THE UNIVERSITY OF CHICAGO

ESTIMATION AND STATISTICAL INFERENCE FOR HIGH DIMENSIONAL MODEL WITH CONSTRAINED PARAMETER SPACE

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE UNIVERSITY OF CHICAGO
BOOTH SCHOOL OF BUSINESS
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

BY
MING YU

CHICAGO, ILLINOIS
MARCH 2020

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank my advisors Mladen Kolar and Varun Gupta for their continuing guidance and support. They are great mentors, and I learned a lot through the collaboration ranging from technical details, academic writing to being a qualified independent researcher. I would also like to thank the rest of my committee members, Tengyuan Liang and Rina Foygel Barber for their valuable discussion.

During the PhD study, I did several wonderful summer internships in industry companies. I would like to thank Aurelie Lozano, Eunho Yang, Karthikeyan Natesan Ramamurthy, and Naoki Abe at IBM Research; Zhaoran Wang and Han Liu at Tencent AI Lab; Qin Gao and Issam Bazzi at Citadel Securities for hosting the internships. I also would like to thank my collaborators Addie Thompson, Zhuoran Yang, Tuo Zhao and Yang Ning for their helpful discussions through my research life. I am also grateful to my friends and schoolmates for the discussion and collaboration on research and coursework. Special thanks to Jingyu He and my four chinchillas for making my PhD life so much fun.

I would like to express my deepest thanks to my parents - my mother, Changping Ji, and my father, Xiaogang Yu. Without the love and support of them, I would never reach this far and fulfill my PhD journey.

Last, I would like to especially thank and acknowledge my wife. Thank you for all your support and encouragement in the past 15 years.

ABSTRACT

This thesis considers estimation and statistical inference for high dimensional model with constrained parameter space. Due to the recent development of data storage and computing technology, it is extremely common for researchers to face a high dimensional problem in practical applications, ranging from health-care, neural imaging, genetic studies, etc. In a high dimensional problem, the number of unknown parameters is usually much larger than the sample size, imposing additional difficulties on accurately estimating the parameters. As a result, it is usually assumed that the parameter satisfies some certain constraints, such as sparsity constraint or low-rank constraint. In this thesis, we develop novel algorithms to obtain accurate parameter estimation and statistical inference for several high dimensional models with constrained parameter space.

Chapter 2 discusses asymptotic inference for high dimensional model under equality constraint. We propose a novel inference method that takes the equality constraint into consideration. The proposed estimator enjoys asymptotically smaller variance than the standard method without constraints, and is semiparametric efficient. Chapter 3 considers high dimensional statistical inference with inequality constraint. We develop tools to test whether the parameters are on the boundary of the constraint or not. The proposed testing procedure has greater power than the standard algorithms where the constraints are ignored. Chapter 4 studies the problem of recovery of matrices that are simultaneously low rank and row and/or column sparse. We propose a GDT (Gradient Descent with hard Thresholding) algorithm that converges linearly to a region within statistical error of an optimal solution. Chapter 5 considers the safe reinforcement learning problem. We construct a sequence of surrogate convex constrained optimization problems by replacing the nonconvex functions locally with convex quadratic functions obtained from policy gradient estimators. We prove that the solutions to these surrogate problems converge to a stationary point of the original nonconvex problem.

CHAPTER 1

INTRODUCTION

In this thesis, we cover a handful of scenarios where we have constraints on the parameters of interest. For the estimation part, we focus on two kinds of constrained problems. The first one is the recovery of matrices that are simultaneously low rank and row and/or column sparse. Such a problem has a wide variety of applications, such as multi-task learning. The second one is the safe reinforcement learning problem. When taking action at state space, we incur a reward value and a cost value. The safety constraint requires that the expected cost value is upper bounded by a predefined constant. The goal of the safe reinforcement learning problem is to estimate the model parameter that maximizes the expected reward while satisfies the safety constraint. For the statistical inference part, most of the existing works perform confidence intervals and hypothesis tests for a low dimensional subset of model parameters under the assumption that the parameters of interest are unconstrained. However, in many applications, there are natural constraints on model parameters. The constraints can be equality constraints, inequality constraints, or the combination of both. For all these scenarios, we propose new estimation and inference methods designed for specific problems.

In Chapter 2, we consider the equality constraints on the model parameters. In many of the applications in high dimensional models, we naturally have such equality constraints, such as linear regression with linear constraint, sparse PCA model with unit norm constraint. We propose a novel method to provide statistical inference in high dimensional models under equality constraints on the parameters. By considering the equality constraints, our proposed estimator enjoys asymptotically smaller variance than the standard method without constraints. Experiments demonstrate the effectiveness of our proposed model, on both asymptotic valid statistical inference and the variance reduction.

In Chapter 3, we consider the inequality constraints on the model parameters. In many problems, there are natural constraints on model parameters and one is interested in whether the parameters are on the boundary of the constraint or not. e.g., non-negativity constraints for transmission rates in network diffusion. We provide algorithms to solve this problem of hypothesis testing in high-dimensional statistical models under constrained parameter space. We show that following our testing procedure we can get asymptotic designed Type I error under the null. Numerical experiments demonstrate that our algorithm has greater power than the standard algorithms where the constraints are ignored. We demonstrate the effectiveness of our algorithms on two real datasets where we have *intrinsic* constraints on the parameters.

In Chapter 4 we study the problem of recovery of matrices that are simultaneously low rank and row and/or column sparse. Such matrices appear in recent applications in cognitive neuroscience, imaging, computer vision, macroeconomics, and genetics. We propose a GDT (Gradient Descent with hard Thresholding) algorithm to efficiently recover matrices with such structure, by minimizing a bi-convex function over a nonconvex set of constraints. We show linear convergence of the iterates obtained by GDT to a region within the statistical error of an optimal solution. As an application of our method, we consider multi-task learning problems and show that the statistical error rate obtained by GDT is near optimal compared to the minimax rate. Experiments demonstrate competitive performance and much faster running speed compared to existing methods, on both simulations and real data sets.

In Chapter 5 we study the safe reinforcement learning problem with nonlinear function approximation, where policy optimization is formulated as a constrained optimization problem with both the objective and the constraint being nonconvex functions. For such a problem, we construct a sequence of surrogate convex constrained optimization problems by replacing the nonconvex functions locally with convex quadratic functions obtained from policy gradient estimators. We prove that the solutions to these surrogate problems converge to a stationary point of the original nonconvex problem. Furthermore, to extend our theoretical results, we apply our algorithm to examples of optimal control and multi-agent reinforcement learning with safety constraints.

CHAPTER 2
ASYMPTOTIC INFERENCE FOR HIGH DIMENSIONAL MODEL UNDER EQUALITY CONSTRAINTS

2.1 Introduction

Quantifying uncertainty of the penalized estimators used for estimation of parameters in high-dimensional models, where sample size is comparable, and often much smaller, to the dimensionality of the parameter vector, is a challenging problem. For example, when the lasso estimator [302] is used to estimate an (approximately) sparse parameter vector, the limiting distribution of the estimator is non-standard even in a low-dimensional setting [179]. It is well understood that inference following model selection is difficult and, in particular, that inference relying on model selection can not be made uniformly valid [196, 197, 198, 249, 250]. Rather than focusing on perfect model selection, recently proposed approaches construct asymptotically linear estimators of low-dimensional components of a high-dimensional parameter vector, which can then be used for statistical inference. Examples include de-biasing [309, 362, 166], decorrelated score method [237, 102], and double selection [29], among others. These approaches do not take into account constraints or restrictions that may be available on the underlying parameter vector beyond assuming that the parameter is approximately sparse.

This chapter studies estimation of a parameter vector in a high-dimensional statistical model that is known to satisfy an equality constraint. Specifically, we focus on developing valid inferential tools for quantifying uncertainty about low-dimensional components of the parameter vector. Suppose we are given n independent and identically distributed multivariate random variables $Z_1, \ldots, Z_n$ generated from a statistical model $\mathcal{P} = \{\mathbb{P}_\theta \mid \theta \in \Omega\}$, where θ is a p dimensional unknown parameter and Ω is the parameter space. The true parameter vector, denoted by θ^*, is the minimizer of

$$\theta^* = \arg\min_{\theta} \mathbb{E}\left[\ell(\theta)\right] \quad \text{subject to} \quad g(\theta) = 0,$$

where $\ell(\theta) = \sum_{i\in[n]} \ell_i(\theta) = \sum_{i\in[n]} \ell(\theta, Z_i)$ is the sample objective function and $g(\theta) = 0$ is the constraint function, $g : \mathbb{R}^p \mapsto \mathbb{R}^q$ with $q \ll p$. We assume that the true parameter θ^* satisfies the constraint, $g(\theta^*) = 0$. The equality constraint can be viewed as a restriction or prior information on the parameters. Such a constraint is common in many applications. For example, in linear regression, we may have prior information that the unknown parameter lies in a linear subspace [9]; in (sparse) principal component analysis, we have a constraint that the leading eigenvector has the unit norm: $\|\theta\|_2 = 1$. In a portfolio selection problem the goal is to allocate the resources to maximize the expected return and minimize the risk. Let $(w_i)_i$ be the portfolio allocation vector denoting the proportion of wealth allocated to each asset. In this problem, the allocation vector satisfies $\sum_i w_i = 1$ and, in addition, we may have a constraint $Aw = a$, which constraints percentage of allocations on each sector or industry [100].

There are two common approaches for solving the equality constrained problem: the reparameterization method and the Lagrangian multiplier method. In the first approach, one expresses a subset of q parameters in terms of the remaining $p - q$ parameters by solving $g(\theta) = 0$, and estimates the $p - q$ unrestricted parameters by minimizing the sample objective function. This approach has major drawbacks: first, it may not be straightforward or even possible to express the q parameters in terms of the remaining $p - q$; second, the choice of which q parameters are reparameterized affects the symmetry of the procedure and may be undesirable in case our interest is in all the parameters. Throughout the chapter we focus on the second approach, the Lagrangian multiplier method, which avoids drawbacks of the reparametrization method. Let $L(\theta, \mu) = \ell(\theta) + \mu^\top g(\theta)$ be the Lagrangian function, where $\mu \in \mathbb{R}^q$ is a vector of Lagrangian multipliers. In the Lagrangian multiplier method, we estimate θ^* by finding a solution of the equations $\nabla_\theta L(\theta, \mu) = \nabla_\mu L(\theta, \mu) = 0$, that is,

$$\nabla_\theta \ell(\theta) + \nabla_\theta g(\theta) \cdot \mu = 0 \tag{2.1}$$

$$g(\theta) = 0. \tag{2.2}$$

In a high-dimensional setting, there may be infinitely many solutions to (2.1) and (2.2). Instead, we find the pair $(\widehat{\theta}, \widehat{\mu})$ by minimizing the following regularized optimization problem

$$\widehat{\theta} = \arg\min_{\theta} \ \ell(\theta) + P_\lambda(\theta) \quad \text{subject to} \quad g(\theta) = 0, \tag{2.3}$$

where $P_\lambda(\theta)$ is a regularization term with a tuning parameter λ. The Lagrange multiplier $\widehat{\mu}$ can be obtained from the Karush–Kuhn–Tucker (KKT) conditions. More generally, we can obtain $(\widehat{\theta}, \widehat{\mu})$ by finding a solution of the equations

$$\nabla_\theta \ell(\widehat{\theta}) + \widehat{\tau} + \nabla_\theta g(\widehat{\theta}) \cdot \widehat{\mu} = 0$$

$$g(\widehat{\theta}) = 0,$$

where $\widehat{\tau} \in \partial P_\lambda(\widehat{\theta})$ is an element of the subgradient at $\widehat{\theta}$. Due to the bias induced by the regularization term, obtaining sampling distribution of $\widehat{\theta}$ is not possible.

Our contributions. In this chapter we perform asymptotic statistical inference for such equality constrained problems in high dimensional models. We make the following three contributions. First, starting from consistent but biased estimator, we develop a novel inferential methodology that takes the equality constraint into consideration. The proposed method is based on debias method. However, different from existing work which approximate the inverse of the Hessian matrix, in our approach, the gradient of the constraint function also plays a role. The proposed estimator is proved to be asymptotic normal under an appropriate asymptotic regime, and from which we could build valid statistical inference on the unknown parameters including confidence interval and hypothesis testing. Second, we show that by considering the equality constraint, our proposed estimator achieves smaller asymptotic variance than the existing methods where we ignore the constraint. This variance reduction helps to obtain narrower confidence interval and improve the power of hypothesis testing. We then prove that our proposed estimator is semiparametriaclly efficient in that it achieves the minimum asymptotic variance for the equality constrained problem. Finally, we show how our approach can be applied to some common equality constrained problems, including linear regression with linear constraint, sparse principal component analysis, and single index model with unit norm constraint.

2.1.1 Related works

Statistical inference in high dimensional models. Our work lies in the literature of high dimensional statistical inference. In high dimensional models, most of the consistent estimators obtained by adding a regularization term to the negative log-likelihood function do not have a tractable asymptotic distribution. For example, [179] shows that the limiting distribution of Lasso estimator may have positive probability mass at 0 when the true value is 0; moreover, the limiting distribution depends on the unknown parameter and is not tractable. More recently, much progress has been made on the statistical inference of low dimensional parameters in high dimensional linear model based on the bias correction approach, also named as debias [166], desparsify [309], or LDPE method [362]. These methods start from the Lasso estimator with KKT condition, and use different approaches to approximate the inverse of the Gram matrix. From then on, many approaches have been proposed for more general models and applications. In [237] the authors propose a general framework for high dimensional statistical inference based on the decorrelation method on penalized M-estimators. In [90, 364] the authors propose simultaneous inference that focuses on the whole high dimensional parameter instead of focusing on a low dimensional subset. Other applications include graphical model [162, 258, 345, 161] which provide inference on the edge between two nodes in a graph; post selection inference [81, 193, 304, 336, 301] which consider the conditional inference given that some covariates are selected; and also semiparametric regression [238, 80], proportional hazards model [102], panel model [30, 180].

Statistical inference with constraint. In the literature, very little work has been done for the statistical inference under equality constraint, especially in high dimensions. An early work [5] gives the maximum likelihood estimation and statistical inference for equality constrained problem in low dimensions. Based on this result, in subsequent work, [283] constructs the Lagrangian multiplier test for the equality hypothesis. [82, 50] consider testing equality hypothesis and show that the Wald, Likelihood ratio, and the Lagrangian multiplier test are equivalent when the log-likelihood is quadratic. [95] considers testing the linear hypothesis in generalized least squares models. [269] considers overparameterized structural model which involves some redundant parameters, and propose minimum discrepancy function (MDF) test statistic that has an asymptotic chi-squared distribution. [271] investigates the asymptotic behavior of the estimator in stochastic programming under equality and inequality constraint, which is not asymptotic normal due to the presence of inequality constraint.

More recently, in high dimensional framework, [277] considers statistical inference for linear regression with linear constraint. [276, 213] performs linear hypothesis testing in the high dimensional generalized linear model. [160] proposes a debiased method to construct statistical inference on the sparse PCA problem. Instead of taking the unit norm constraint $\|\theta\|_2 = 1$ into consideration, the authors “absorb” the leading eigenvalue into the leading eigenvector and solve for an unconstrained problem. Therefore, it is still different from our approach.

Another closely related area is statistical inference with inequality constraint. With an inequality constraint, people usually focus on hypothesis testing on whether the parameter lies on the boundary, or is strictly an interior point. A lot of work has been done in low dimensional problems with inequality constraint, mostly focused on Wald, Score, and Likelihood ratio test

[79, 270, 279, 224, 278]. A recent work [350] generalizes these techniques to high dimensions. The geometries are very different for equality and inequality constrained problems. With an inequality constraint, the limit distribution is given by a mixed χ^2 distribution, whereas for the equality constraint problem, the limit distribution is normal. As another related work, [371] considers hypothesis testing based on constrained maximum likelihood ratio with L_1 constraint, with a focus on high dimensional linear regression and Gaussian graphical model.

2.1.2 Organization of the chapter

The remainder of the chapter is organized as follows. In Section 2.2 we present the setup of the equality constrained problem. We propose our methodology in Section 2.3 and prove theoretical results in Section 2.4. Section 2.5 shows the semiparameteric efficiency of the proposed estimator. In Section 2.6 we show how our proposed method can be applied to some frequently-used models, and in Section 2.7 we demonstrate the effectiveness of our method through extensive experiments. We conclude in Section 2.8.

2.2 Problem setup

In this section we provide the setup of the high dimensional equality constrained problem. Suppose $\theta \in \mathbb{R}^p$ is the model parameter and we have n i.i.d. samples. The loss function (e.g. negative log likelihood) is denoted as $\ell(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(\theta)$ where $\ell_i(\theta)$ denotes the loss function on one sample. In addition, we have an equality constraint on the model parameter given by $g(\theta) = 0$. Throughout the chapter we assume that the constraint is compatible, non-trivial and non-redundant. Also for illustration purpose we assume only one scalar constraint. It is straightforward to generalize to multiple constraints case. The true parameter is denoted as θ^* satisfiying $g(\theta^*) = 0$. The optimization problem (constrained regularized problem) is

$$
\begin{aligned}
\min_{\theta} \quad & \ell(\theta) + P_\lambda(\theta) \\
\text{s.t.} \quad & g(\theta) = 0
\end{aligned}
\tag{2.4}
$$

where $P_\lambda(\theta)$ is some regularization term with tuning parameter λ. Denote μ as the Lagrange multiplier, the Lagrange function of (2.4) is given by

$$L(\theta, \mu) = \ell(\theta) + P_\lambda(\theta) + \mu \cdot g(\theta).$$

Let $(\widehat{\theta}, \widehat{\mu})$ be the solution to (2.4), we have the KKT condition

$$
\begin{aligned}
\nabla \ell(\widehat{\theta}) + \widehat{\tau} + \nabla g(\widehat{\theta}) \cdot \widehat{\mu} &= 0 \\
g(\widehat{\theta}) &= 0
\end{aligned}
$$

where $\widehat{\tau} \in \partial P_\lambda(\widehat{\theta})$ is the subgradient at $\widehat{\theta}$. Next, denote $\ell^*(\theta) = \mathbb{E}\ell_i(\theta)$ as the expected loss function, i.e., the loss function when we have infinite amount of samples. The population version optimization problem is given by

$$
\begin{aligned}
\min_{\theta} \quad & \ell^*(\theta) \\
\text{s.t.} \quad & g(\theta) = 0
\end{aligned}
\tag{2.5}
$$

Here we do not need regularization term since the problem is on population version. The population version Lagrange function of (2.5) is given by

$$L^*(\theta, \mu) = \ell^*(\theta) + \mu \cdot g(\theta).$$

Clearly the solution to (2.5) is given by (θ^*, μ^*) where θ^* is the true parameter and μ^* is the population Lagrange multiplier with

$$\begin{aligned} \nabla \ell^*(\theta^*) + \nabla g(\theta^*) \cdot \mu^* &= 0 \\ g(\theta^*) &= 0 \end{aligned} \tag{2.6}$$

Next, we highlight that there are two kinds of equality constrained problems in which the constraint plays different roles in the optimization problem. We summarize these two cases as below.

Case 1: constraint as additional information. In the first case, the constraint serves as additional information on the unknown parameters. It helps to obtain a more accurate estimation for the model parameter; but even if without the constraint, the problem is still well-defined in that we are still able to obtain a consistent estimator as long as we have enough samples. As an example, consider linear regression where $y = X\theta^* + \epsilon$ with $\|\theta^*\|_0 = s$ and $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. The constraint is given by a linear constraint $A^\top \theta = r$ on the model parameter. The population version optimization problem is

$$\begin{aligned} \min \quad & \mathbb{E} \left\| y - X\theta \right\|_2^2 \\ \text{s.t.} \quad & A^\top \theta - r = 0 \end{aligned}$$

In this case, even if there is no constraint, the solution to the population version problem is still θ^*. Moreover, for the sample version, the unconstrained OLS estimator is still well defined and consistent as long as we have enough samples. We will revisit this problem in details later in Section 2.6.1.

Mathematically, for this kind of problem, we have that $\nabla \ell^*(\theta^*) = 0$ and therefore $\mu^* = 0$ according to (2.6). This indicates that in population version the constraint is *not binding*, and θ^* is the solution to the problem (2.5) *even if without the constraint.*

Case 2: constraint as intrinsic restriction. In the second case, the constraint serves as an intrinsic restriction on the model in that, if we ignore the constraint, then the unconstrained problem will have a completely different solution or is even meaningless. For example, consider the principal component analysis (PCA) where we observe x_i i.i.d. with mean 0 and covariance matrix Σ^*. The goal is to estimate the leading eigenvector θ^* of the true covariance matrix Σ^*. The population version optimization problem is

$$\begin{aligned} \max_{\theta} \quad & \frac{1}{2}\theta^\top \Sigma^* \theta \\ \text{s.t.} \quad & \frac{1}{2}\|\theta\|_2^2 = \frac{1}{2} \end{aligned}$$

Here we have an intrinsic constraint that $\|\theta\|_2 = 1$. If we ignore this constraint, the problem becomes meaningless in that we can choose arbitrary large or small θ to obtain an arbitrarily large or small variance. We will revisit this problem in details later in Section 2.6.2.

Mathematically, for this kind of problem, we have that $\nabla \ell^*(\theta^*) \neq 0$ and therefore $\mu^* \neq 0$ according to (2.6). This indicates that in population version the constraint is *binding*, and θ^* is the solution to the problem (2.5) *only if we have the constraint.*

In practice, the equality constraint can be any one of these two cases, or the combination of both. In the next section, we will propose a unified methodology to construct asymptotic inference for these two cases.

2.3 Methodology

In this section we present our methodology to construct the proposed estimator. We start from some consistent estimators $(\widehat{\theta}, \widehat{\mu})$ where $\widehat{\theta}$ satisfies the constraint $g(\widehat{\theta}) = 0$. The consistent estimator $(\widehat{\theta}, \widehat{\mu})$ may be obtained by solving the constrained regularized problem (2.4), or can be constructed by any problem-specific approach. In Section 2.9.1 we provide an efficient algorithm for solving (2.4) based on projected proximal gradient descent algorithm.

In practice, it may be hard to obtain a consistent estimator $\widehat{\mu}$ for the Lagrangian multiplier other than solving the constrained optimization problem (2.4), and hence most of the time we solve for (2.4) for $(\widehat{\theta}, \widehat{\mu})$. However, there are some cases where we can obtain $\widehat{\mu}$ without solving (2.4), for example for the sparse PCA problem, and therefore we can directly start with the consistent estimator. See Section 2.6.2 for more details.

The estimator $\widehat{\theta}$ is usually biased since we usually have a regularization term when solving for high dimensional models. We then propose the methodology to quantify this bias and to construct asymptotically unbiased estimator. Define $\widehat{\tau}$ as

$$\widehat{\tau} = -\nabla \ell(\widehat{\theta}) - \nabla g(\widehat{\theta}) \cdot \widehat{\mu} \tag{2.7}$$

If $(\widehat{\theta}, \widehat{\mu})$ is the solution to (2.4), then $\widehat{\tau}$ is the subgradient of the problem. Denote $H_\ell(\theta)$ and $H_g(\theta)$ as the Hessian matrices of $\ell(\theta)$ and $g(\theta)$, we do Taylor expansion on (2.7) at θ^* and μ^* and obtain

$$\begin{aligned} 0 &= \nabla \ell(\widehat{\theta}) + \widehat{\tau} + \Big[\nabla g(\widehat{\theta}) - \nabla g(\theta^*)\Big]\widehat{\mu} + \nabla g(\theta^*)(\widehat{\mu} - \mu^*) + \nabla g(\theta^*)\mu^* \\ &= \nabla \ell(\theta^*) + H_\ell(\bar{\theta}_1)(\widehat{\theta} - \theta^*) + \widehat{\tau} + H_g(\bar{\theta}_2)(\widehat{\theta} - \theta^*) \cdot \widehat{\mu} + \nabla g(\theta^*)(\widehat{\mu} - \mu^*) + \nabla g(\theta^*)\mu^* \end{aligned} \tag{2.8}$$

Since $\widehat{\theta}$ satisfies the constraint $g(\widehat{\theta}) = 0$, we have

$$g(\theta^*) + \nabla g(\bar{\theta}_3)^\top \cdot (\widehat{\theta} - \theta^*) = 0 \tag{2.9}$$

where $\bar{\theta}_i = \theta^* + \bar{u}_i(\widehat{\theta} - \theta^*)$ are intermediate values between $\widehat{\theta}$ and θ^* for some $0 \leq \bar{u}_i \leq 1$ with $i = 1, 2, 3$. Using the fact that $g(\theta^*) = 0$ we can combine (2.8) and (2.9) and rewrite them as

$$\begin{bmatrix} H_\ell(\bar{\theta}_1) + \widehat{\mu} H_g(\bar{\theta}_2) & \nabla g(\theta^*) \\ \nabla g(\bar{\theta}_3)^\top & 0 \end{bmatrix} \cdot \begin{bmatrix} \widehat{\theta} - \theta^* \\ \widehat{\mu} - \mu^* \end{bmatrix} = \begin{bmatrix} -\nabla \ell(\theta^*) - \widehat{\tau} - \nabla g(\theta^*)\mu^* \\ 0 \end{bmatrix} \tag{2.10}$$

The population version of the big matrix on the left hand side of (2.10) is given by

$$\begin{bmatrix} H_\ell^*(\theta^*) + \mu^* H_g(\theta^*) & \nabla g(\theta^*) \\ \nabla g(\theta^*)^\top & 0 \end{bmatrix} \tag{2.11}$$

where $H_\ell^*(\theta)$ is the Hessian matrix of $\ell^*(\theta)$. We then wish to find a matrix

$$\begin{bmatrix} P^* & Q^* \\ Q^{*\top} & R^* \end{bmatrix} \tag{2.12}$$

as the inverse of (2.11). Denote $H^* = H_\ell^*(\theta^*) + \mu^* H_g(\theta^*)$ and denote M^* as the exact inverse of H^*, according to the block matrix inversion formula we have

$$\begin{aligned} P^* &= M^* - M^*\nabla g(\theta^*)\Big[\nabla g(\theta^*)^\top M^* \nabla g(\theta^*)\Big]^{-1} \nabla g(\theta^*)^\top M^* \\ Q^* &= M^* \nabla g(\theta^*)\Big[\nabla g(\theta^*)^\top M^* \nabla g(\theta^*)\Big]^{-1} \\ R^* &= -\Big[\nabla g(\theta^*)^\top M^* \nabla g(\theta^*)\Big]^{-1} \end{aligned} \tag{2.13}$$

Note that in Case 1 discussed in Section 2.2 we have $\mu^* = 0$. Therefore, M^* is the exact inverse of $H_\ell^*(\theta^*)$. Also, it is possible that H^* is not invertible, for example for sparse PCA problems to be discussed later in Section 2.6.2. In this situation, we could turn to some substitutes, for example the Moore-Penrose pseudo-inverse, and change the definition of P^*, Q^*, R^* in (2.13) accordingly. We discuss this modification in detail in Section 2.6.2.

Lemma 1. *We have that*

$$\begin{bmatrix} \widehat\theta - \theta^* \\ \widehat\mu - \mu^* \end{bmatrix} = \begin{bmatrix} P^* & Q^* \\ Q^{*\top} & R^* \end{bmatrix} \cdot \begin{bmatrix} -\nabla \ell(\theta^*) - \widehat\tau - \nabla g(\theta^*)\mu^* \\ 0 \end{bmatrix} + \text{error}. \tag{2.14}$$

where the error term is given by

$$\text{error} = \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} P^* & Q^* \\ {Q^*}^\top & R^* \end{bmatrix} \begin{bmatrix} H_\ell(\bar{\theta}_1) + \widehat{\mu} H_g(\bar{\theta}_2) & \nabla g(\theta^*) \\ \nabla g(\bar{\theta}_3)^\top & 0 \end{bmatrix} \right) \cdot \begin{bmatrix} \widehat{\theta} - \theta^* \\ \widehat{\mu} - \mu^* \end{bmatrix} \tag{2.15}$$

Proof. Multiplying the inverse matrix (2.12) to the both sides of (2.10) and rearranging terms we obtain the result. □

Rearranging terms in (2.14) and notice that $P^* \cdot \nabla g(\theta^*) = 0$, ${Q^*}^\top \cdot \nabla g(\theta^*) = I$ according to the definition (2.13), we obtain

$$\begin{aligned} \begin{bmatrix} \widehat{\theta} + P^* \cdot \widehat{\tau} \\ \widehat{\mu} + {Q^*}^\top \cdot \widehat{\tau} \end{bmatrix} &= \begin{bmatrix} \theta^* - P^* \cdot \nabla \ell(\theta^*) - P^* \cdot \nabla g(\theta^*) \mu^* \\ \mu^* - {Q^*}^\top \cdot \nabla \ell(\theta^*) - {Q^*}^\top \cdot \nabla g(\theta^*) \mu^* \end{bmatrix} + \text{error}. \\ &= \begin{bmatrix} \theta^* - P^* \cdot \nabla \ell(\theta^*) \\ -{Q^*}^\top \cdot \nabla \ell(\theta^*) \end{bmatrix} + \text{error}. \end{aligned} \tag{2.16}$$

Motivated by the expression in the first row, our proposed estimator is given by

$$\theta_{\text{est}} = \widehat{\theta} + P \cdot \widehat{\tau} \tag{2.17}$$

where P is an approximation of the true P^* defined as

$$P = M - M \nabla g(\widehat{\theta}) \Big[\nabla g(\widehat{\theta})^\top M \nabla g(\widehat{\theta}) \Big]^{-1} \nabla g(\widehat{\theta})^\top M \tag{2.18}$$

Here M is an approximation of the true M^*, and we will discuss the choice of M later in Section 2.3.1. Under appropriate asymptotic regime such that the error term in (2.16) and the approximation error by using P in place of P^* are negligible, we obtain from (2.16) that

$$\sqrt{n} \big(\theta_{\text{est}} - \theta^* \big) = -P^* \cdot \sqrt{n} \nabla \ell(\theta^*) + o_{\mathbb{P}}(1) \tag{2.19}$$

In most of the applications, the term $\nabla\ell(\theta^*)$ on the right hand side of (2.19) is an average over n i.i.d. samples, and is therefore asymptotically normal by Central Limit Theorem. Moreover, under Case 1 it has mean 0; under Case 2, according to (2.6) and the definition of P^* in (2.13) we have $\mathbb{E}\big[P^*\nabla\ell_i(\theta^*)\big] = P^* \cdot \nabla\ell^*(\theta^*) = -P^* \cdot \nabla g(\theta^*)\mu^* = 0$. We then obtain the asymptotic distribution of our proposed estimator (2.17), and we can construct confidence interval and hypothesis testing based on this.

Asymptotic inference on μ. Another by-product of (2.16) is the asymptotic inference on the Lagrangian multiplier. Although in many applications we care more about the estimator than the Lagrangian multiplier (especially in Case 1 where $\mu^* = 0$), in some cases it is still valuable. For example, later in Section 2.6.2 on the application to sparse PCA, we see that μ^* is the negative leading eigenvalue of the true covariance matrix. This allows us to obtain asymptotic inference for the leading eigenvalue. Back to the second line of (2.16), our proposed estimator on the Lagrangian multiplier is

$$\mu_{\text{est}} = \widehat{\mu} + Q^\top \cdot \widehat{\tau} \tag{2.20}$$

where we estimate Q^* by Q defined as

$$Q = M\nabla g(\widehat{\theta})\Big[\nabla g(\widehat{\theta})^\top M \nabla g(\widehat{\theta})\Big]^{-1}$$

Under appropriate asymptotic regime such that the error term is negligible, we obtain from the second line of (2.16) that

$$\sqrt{n}\big(\mu_{\text{est}} - \mu^*\big) = -\sqrt{n}\Big({Q^*}^\top \cdot \nabla\ell(\theta^*) + \mu^*\Big) + o_{\mathbb{P}}(1) \tag{2.21}$$

According to (2.6) and the definition of Q^* in (2.13) we have

$$\mathbb{E}\Big[{Q^*}^\top \cdot \nabla \ell_i(\theta^*)\Big] = {Q^*}^\top \cdot \nabla \ell^*(\theta^*) = -{Q^*}^\top \cdot \nabla g(\theta^*)\mu^* = -\mu^*$$

We then obtain that the right-hand side of (2.21) is asymptotically normal with mean 0 by Central Limit Theorem. This gives the asymptotic inference for the Lagrangian multiplier.

Remark 2. *Another interesting result is that if $\ell(\theta)$ is a negative log-likelihood function and we are in Case 1 such that $\mu^* = 0$, then θ_{est} and μ_{est} are asymptotically uncorrelated under appropriate conditions. To see this, from* (2.16) *we ignore the error term and calculate the asymptotic covariance between θ_{est} and μ_{est} as*

$$\begin{aligned}
Cov(\theta_{\text{est}}, \mu_{\text{est}}) &= Cov\Big(\theta^* - P^* \cdot \nabla \ell(\theta^*), -{Q^*}^\top \cdot \nabla \ell(\theta^*)\Big) = \mathbb{E}\Big[P^* \cdot \nabla \ell(\theta^*) \nabla \ell(\theta^*)^\top \cdot Q^*\Big] \\
&= P^* \cdot \mathbb{E}\Big[\nabla \ell(\theta^*) \nabla \ell(\theta^*)^\top\Big] \cdot Q^* = P^* H_\ell^* Q^* = P^* H^* Q^* \\
&= P^* H^* \cdot M^* \nabla g(\theta^*) \Big[\nabla g(\theta^*)^\top M^* \nabla g(\theta^*)\Big]^{-1} = P^* \nabla g(\theta^*) \Big[\nabla g(\theta^*)^\top M^* \nabla g(\theta^*)\Big]^{-1} \\
&= 0
\end{aligned}$$

where we use the fact that $P^* \nabla g(\theta^*) = 0$ in the last equality. This result reproduces the asymptotical uncorrelation result for low dimensional models in [5].

2.3.1 Approximate the Hessian inverse

We discuss some practical approaches to approximate the Hessian inverse step in our method. Recall from (2.18) that we could use the following estimator

$$P = M - M \nabla g(\widehat{\theta}) \Big[\nabla g(\widehat{\theta})^\top M \nabla g(\widehat{\theta})\Big]^{-1} \nabla g(\widehat{\theta})^\top M$$

where M is an approximation of the true M^* and M^* is the exact inverse of $H_\ell^*(\theta^*) + \mu^* H_g(\theta^*)$. For notation simplicity we denote $H^* = H_\ell^*(\theta^*) + \mu^* H_g(\theta^*)$ and its sample version $\widehat{H} = H_\ell(\widehat{\theta}) + \widehat{\mu} H_g(\widehat{\theta})$. In practice we could choose M as an approximate inverse of $\widehat{H}$. We then discuss the choices of M under Case 1 and Case 2, respectively.

Case 1. Under Case 1 where $\mu^* = 0$, we have $H^* = H^*_\ell(\theta^*)$ and hence we only need to approximate the inverse of the Hessian of loss function $\ell(\theta)$. Note that this step is the same when we deal with high dimensional unconstrained problems. In low dimensions we can calculate M by taking the matrix inverse on $H_\ell(\widehat{\theta})$ directly and obtain a consistent estimator, as shown in [5]. In high dimensions we have to approximate. There is a vast literature on approximating the inverse of the Hessian matrix [166, 362, 309, 237], and all of them give the same asymptotic approximation error. Here we briefly review two of them: the desparsify method proposed in [309] designed for linear regression (although it is straightforward to generalize), and the decorrelation method proposed in [237] for the general model.

Desparsify method in [309] (nodewise Lasso) Consider the linear regression setting with $y = X\theta^* + \epsilon$. For each $j = 1, ..., p$, denote X_j as the j^{th} column of X and $X_{\backslash j}$ as the design matrix X without the j^{th} column, we solve for the following problem

$$\widehat{\gamma}_j = \arg\min_{\gamma} \frac{1}{2n} \|X_j - X_{\backslash j}\gamma\|_2^2 + \lambda_j \|\gamma\|_1. \tag{2.22}$$

We then construct $C \in \mathbb{R}^{p\times p}$ as $c_{j,j} = 1$ and $c_{j,\backslash j} = -\widehat{\gamma}_j$ where $c_{j,\backslash j}$ denotes the j^{th} row of C without the j^{th} column. Next, we define $T = \text{diag}(\tau_1^2, ..., \tau_p^2)$ with $\tau_j^2 = \|X_j - X_{\backslash j}\gamma_j\|_2^2/n + \lambda_j\|\gamma_j\|_1$. The approximate Hessian inverse M is given by $M = T^{-1}C$.

Decorrelation method in [237] For each $j = 1, ..., p$, denote $H_{j,j}$ as the j^{th} row, j^{th} column of $H(\widehat{\theta})$; denote $H_{j,\backslash j}$ as the j^{th} row of $H(\widehat{\theta})$ without the j^{th} column; denote $H_{\backslash j,\backslash j}$ as $H(\widehat{\theta})$ without the j^{th} row and j^{th} column. We solve for the following problem

$$\widehat{w}_j = \arg\min_{w} \|w\|_1 \quad \text{subject to} \quad \|H_{j,\backslash j} - w^\top H_{\backslash j,\backslash j}\|_\infty \le \lambda_j.$$

Define $\tau_j = H_{j,j} - H_{j,\backslash j}\widehat{w}_j$, The approximate Hessian inverse $M \in \mathbb{R}^{p\times p}$ is constructed as $m_{j,j} = 1/\tau_j$ and $m_{j,\setminus j} = -\widehat{w}_j/\tau_j$.

Case 2. Under Case 2 we aim to approximate the inverse of $\widehat{H} = H_\ell(\widehat{\theta}) + \widehat{\mu} H_g(\widehat{\theta})$. First notice that if the constraint is linear, then $H_g(\widehat{\theta}) = 0$ and it degenerates to Case 1. For non-linear constraint, we see that the Hessian matrix of the constraint $g(\theta)$ also plays a role and reshape the eigenvalue-structure of $H_\ell(\widehat{\theta})$. In general, this step is problem specific and it depends on how the constraint help to reshape the eigenvalue-structure. For example, when the constraint consists of $\|\theta\|_2^2$, then we have $H_g(\widehat{\theta})$ consists of identity matrix. We can then obtain M as the (pseudo-)inverse of $\widehat{H}$. See Section 2.6.2 for an example.

2.4 Theoretical result

In this section we provide theoretical results on the proposed estimator. We start from stating some mild assumptions on the model.

Assumption 3 (Convergence rate for θ)**.** *We have the convergence rate*

$$\lim_{n\to\infty} \mathbb{P}\big(\|\widehat{\theta} - \theta^*\|_1 \le \gamma_1(n)\big) = 1 \tag{2.23}$$

Moreover, for any $\bar{\theta}_i = \theta^* + \bar{u}_i(\widehat{\theta} - \theta^*)$ for some $0 \le \bar{u}_i \le 1$ with $i = 1, 2, 3$, we have

$$\begin{aligned} \lim_{n\to\infty} \mathbb{P}\Big(\sup_{\bar{\theta}_1,\bar{\theta}_2} \big\|I - M^*\big[H_\ell(\bar{\theta}_1) + \widehat{\mu} H_g(\bar{\theta}_2)\big]\big\|_\infty \le \gamma_2(n)\Big) &= 1 \\ \lim_{n\to\infty} \mathbb{P}\Big(\sup_{\bar{\theta}_3} \big\|\nabla g(\theta^*)\big[\nabla g(\theta^*)^\top \nabla g(\theta^*)\big]^{-1}\big[\nabla g(\bar{\theta}_3) - \nabla g(\theta^*)\big]^\top\big\|_\infty \le \gamma_3(n)\Big) &= 1 \\ \lim_{n\to\infty} \mathbb{P}\Big(\big\|(P - P^*)\cdot \widehat{\tau}\big\|_\infty \le \gamma_4(n)\Big) &= 1 \end{aligned} \tag{2.24}$$

Assumption 4 (Convergence rate for μ)**.** *For any $\bar{\theta}_i = \theta^* + \bar{u}_i(\widehat{\theta} - \theta^*)$ for some $0 \le \bar{u}_i \le 1$*

with $i = 1, 2, 3,$ We have the convergence rate

$$\begin{aligned}\lim_{n\to\infty}\mathbb{P}\bigg(\sup_{\bar\theta_1,\bar\theta_2}\Big\|\Big[\nabla g(\theta^*)^\top M^*\nabla g(\theta^*)\Big]^{-1}\nabla g(\theta^*)^\top\Big[I - M^*[H_\ell(\bar\theta_1)+\widehat{\mu}H_g(\bar\theta_2)]\Big]\Big\|_\infty \le \gamma_5(n)\bigg) &= 1\\ \lim_{n\to\infty}\mathbb{P}\bigg(\sup_{\bar\theta_3}\Big\|\Big[\nabla g(\theta^*)^\top M^*\nabla g(\theta^*)\Big]^{-1}\big[\nabla g(\bar\theta_3)-\nabla g(\theta^*)\big]^\top\Big\|_\infty \le \gamma_6(n)\bigg) &= 1\\ \lim_{n\to\infty}\mathbb{P}\Big(\big\|(Q-Q^*)^\top\cdot\widehat{\tau}\big\|_\infty \le \gamma_7(n)\Big) &= 1\end{aligned} \tag{2.25}$$

Assumption 5 (Central limit theorem)**.** *For any matrix V^* such that $\mathbb{E}\big[V^*\nabla\ell_i(\theta^*)\big] = \kappa$, we have*

$$\sqrt{n}\Big(V^*\cdot\nabla\ell(\theta^*)-\kappa\Big)\to\mathcal{N}\Big(0, V^*\mathcal{I}(\theta^*){V^*}^\top\Big)$$

where $\mathcal{I}(\theta^*)$ is the Fisher information matrix if $\ell(\cdot)$ is given by negative log-likelihood function, or some appropriate problem specific covariance matrix otherwise.

Assumption 3 and 4 provide problem specific rates for the estimation and approximation errors. In particular, (2.23) measures the estimation error on $\widehat{\theta}$; the first line in (2.24) and (2.25) measure the approximation on Taylor expansion with intermediate values; the second line in (2.24) and (2.25) assume that the constraint function is well-behaved around θ^*; the third line in (2.24) and (2.25) measure the error on approximating P^*, Q^* with the sample version. For most of the problems, all these terms $\gamma_i(n)$ can be shown to converge to 0, although they may be at different rates which are problem-specific. Assumption 5 is standard central limit theorem result and can be obtained by verifying the Lindeberg's condition. We are then ready for our main theorem.

Theorem 6. *Consider the general constrained optimization problem* (2.4)*. Suppose Assumption 3 and 5 are satisfied and suppose that $n^{1/2}\big(\gamma_1(n)(\gamma_2(n)+\gamma_3(n))+\gamma_4(n)\big) = o(1)$, then the estimator θ_{est} constructed in* (2.17) *satisfies*

$$\sqrt{n}\big(\theta_{\mathrm{est}}-\theta^*\big) = W_\theta + \Delta_\theta,$$

where $\|\Delta_\theta\|_\infty = o_{\mathbb{P}}(1)$ *and*

$$W_\theta = P^* \cdot \sqrt{n}\nabla\ell(\theta^*) \sim \mathcal{N}\Big(0, P^*\mathcal{I}(\theta^*){P^*}^\top\Big).$$

Furthermore, suppose Assumption 4 is satisfied and suppose that $n^{1/2}\big(\gamma_1(n)(\gamma_5(n)+\gamma_6(n))+\gamma_7(n)\big) = o(1)$*, then the estimator* μ_{est} *constructed in* (2.20) *satisfies*

$$\sqrt{n}\big(\mu_{\text{est}} - \mu^*\big) = W_\mu + o_{\mathbb{P}}(1)$$

where

$$W_\mu = -\sqrt{n}\Big({Q^*}^\top \cdot \nabla\ell(\theta^*) + \mu^*\Big) \sim \mathcal{N}\Big(0, {Q^*}^\top\mathcal{I}(\theta^*)Q^*\Big).$$

Remark 7. *We show that by using our method and considering the equality constraint, we can reduce the asymptotic variance of the estimator, and therefore the confidence interval can be narrower and the hypothesis testing will have greater power. In this section since we need to compare with the case where we ignore the constraint, we focus on Case 1 only (otherwise the unconstrained problem would be meaningless). Intuitively, by considering the constraint we utilize additional information and this would help to reduce the uncertainty about our estimation. If we ignore the constraint, we would solve for the following unconstrained problem*

$$\min_{\theta} \; \ell(\theta) + P_\lambda(\theta)$$

and obtain an unconstrained estimator $\widehat{\theta}_{\text{un}}$*. Starting from the unconstrained optimality condition* $\nabla\ell(\widehat{\theta}_{\text{un}}) + \widehat{\tau}_{\text{un}} = 0$ *and follow the similar procedure described in Section 4.2, the proposed estimator would be*

$$\theta_{\text{est,un}} = \widehat{\theta}_{\text{un}} + M^* \cdot \widehat{\tau}_{\text{un}} = \theta^* - M^* \cdot \nabla\ell(\theta^*) + o_{\mathbb{P}}(1/\sqrt{n})$$

and this gives

$$\sqrt{n}\big(\theta_{\text{est,un}} - \theta^*\big) = -M^* \cdot \sqrt{n}\nabla\ell(\theta^*) + o_{\mathbb{P}}(1) \to \mathcal{N}\Big(0, M^*\mathcal{I}(\theta^*){M^*}^\top\Big)$$

We can see that the asymptotic variance is $M^*\mathcal{I}(\theta^*){M^*}^\top$ if we ignore the constraint; while by taking the constraint into consideration, the asymptotic variance of our estimator is $P^*\mathcal{I}(\theta^*){P^*}^\top$. Recall that $P^* = M^* - P_M(\theta^*) \cdot M^*$, we obtain

$$P^*\mathcal{I}(\theta^*){P^*}^\top = \Big[I - P_M(\theta^*)\Big] \cdot M^*\mathcal{I}(\theta^*){M^*}^\top \cdot \Big[I - P_M(\theta^*)\Big]$$

Since $P_M(\theta^*)$ is a projection matrix, we have that $I - P_M(\theta^*)$ is also a projection matrix (project to the orthogonal complement). Therefore, $I - P_M(\theta^*)$ is a non-expansion and hence our estimator has a smaller variance. The amount of variance reduction depends on the size of the projection subspace. Intuitively, if we have more constraints, then the subspace is larger and hence we get much more variance reduction.

2.5 Semiparametric efficiency

In this section, we show that our proposed estimator enjoys the smallest asymptotic variance, and is therefore an efficient estimator. In low dimensional settings, it is well-known that the maximum likelihood estimator is asymptotically efficient under mild conditions. See [37, 310] for the traditional results in low dimensions. When taking the constraint into account in low dimensions, several works have established the lower bound on the variance of the estimator [121, 293, 178]. These results, however, cannot be directly applied to the high dimensional models. For high dimensional models, [309] established the asymptotic optimality in terms of semiparametric efficiency for debiased Lasso estimator. Later on, [163] introduced a framework for semiparametric efficiency for sparse high dimensional models. In this section, to prove the efficiency of our proposed estimator on constrained parameter space, we follow the framework proposed in [163], with a focus on Le Cam's bound.

Throughout this section we only consider our proposed Case 1 where the constraint serves as additional information and we have $\mu^* = 0$. In addition, we require that the loss function $\ell(\theta)$ is given by a negative log-likelihood function, so that the Hessian matrix is identical to the Fisher information matrix: $H^*_\ell(\theta^*) = \mathcal{I}(\theta^*)$. As a result, M^* is the inverse of the Fisher information matrix: $M^* \cdot \mathcal{I}(\theta^*) = I$. The score function is denoted as $s(\theta) = \nabla \ell(\theta)$.

We first follow the framework in [163] to calculate the minimal asymptotic variance of an estimator, when we have equality constraints on θ. Suppose the parameter of interest is $f(\theta)$ where f is a one-dimensional function. Following the setup in [163], we construct a sequence of true parameters $\{\theta^*_n\}$ satisfying the constraint $g(\theta^*_n) = 0$ and also

$$\|\theta^*_n\|_0 \leq c_1 s_n \text{ and } \|\theta^*_n\|_2 \leq c_2$$

for some constant c_1, c_2, and the sparsity level s_n will be specified later. Let $x^1, ..., x^n$ be i.i.d. samples from the model with log-likelihood function $\ell(\theta^*_n)$. For many of the applications, an estimator T_n of $f(\theta^*_n)$ can be written as the following linear form

$$T_n - f(\theta^*_n) = \frac{1}{n}\sum_{i=1}^{n} \varphi_{\theta^*_n}(x^i) + o(n^{-1/2})$$

where $\varphi_{\theta^*_n}$ is an "influence function" satisfying $\mathbb{E}\varphi_{\theta^*_n}(x^i) = 0$ with a finite variance $V_{\theta^*_n} = \mathbb{E}\varphi^2_{\theta^*_n}(x^i) = \mathcal{O}(1)$. With the sequence $\{\theta^*_n\}$, we are interested in deriving the asymptotic variance of a sequence $\widetilde{\theta}_n$ satisfying the constraint $g(\widetilde{\theta}_n) = 0$ and also

$$\|\widetilde{\theta}_n\|_0 \leq c_1 s_n, \ \|\widetilde{\theta}_n\|_2 \leq c_2, \text{ and } \|\theta^*_n - \widetilde{\theta}_n\|_2 \leq c/\sqrt{n} \tag{2.26}$$

In the following, we denote $h = \sqrt{n}(\widetilde{\theta}_n - \theta^*_n)$. To derive the asymptotic variance, we impose the following assumptions adopted from the Condition (D1) - (D3) in [163].

Assumption 8. *The score function $s(\theta)$ is twice differentiable with a finite second derivative.*

The function $f(\theta)$ is differentiable in that

$$f(\widetilde{\theta}_n) - f(\theta_n^*) = \nabla f(\theta_n^*)^\top \cdot (\widetilde{\theta}_n - \theta_n^*) + o(n^{-1/2})$$

The Lindberg's condition is satisfied in that for all $\epsilon > 0$ we have

$$\lim_{n\to\infty} \mathbb{E}_{\theta_n^*}\big[\varphi_{\theta_n^*} + h^\top s_{\theta_n^*}\big]^2 \cdot \mathbf{1}_{|\varphi_{\theta_n^*} + h^\top s_{\theta_n^*}| > \epsilon\sqrt{n}} = 0$$

Furthermore, we have $0 < c_{\min} < \lambda_{\min}\big(\mathcal{I}(\theta_n^*)\big) < \lambda_{\max}\big(\mathcal{I}(\theta_n^*)\big) < c_{\max} < \infty$ and

$$\left\| \frac{1}{n}\sum_{i=1}^n \nabla s_{\theta_n^*} + \mathcal{I}_{\theta_n^*} \right\| = \mathcal{O}_{\mathbb{P}}(\lambda)$$

for some λ. The sparsity level is set as $s_n = o\big(\max\{1/\lambda, n^{1/3}\}\big)$.

The following theorem specifies the lower bound on the asymptotic variance.

Theorem 9. *Let $\widetilde{\theta}_n$ satisfy (2.26). Denote $h = \sqrt{n}(\widetilde{\theta}_n - \theta_n^*)$. Suppose Assumption 8 is satisfied and*

$$\mathbb{E}\big(\varphi_{\theta_n^*} h^\top s_{\theta_n^*}\big) - h^\top \nabla f(\theta_n^*) = o(1) \tag{2.27}$$

Then we have

$$\sqrt{n}\Big(T_n - f\big(\widetilde{\theta}_n\big)\Big) \to \mathcal{N}(0, V_{\theta_n^*}) \tag{2.28}$$

For $\breve{\theta}_n = \widetilde{\theta}_n + \breve{u}(\theta_n^* - \widetilde{\theta}_n)$ with $\breve{u} \in [0,1]$, let

$$\Delta(\breve{\theta}_n; \theta_n^*) = \nabla g(\breve{\theta}_n)\Big(\nabla g(\breve{\theta}_n)^\top \mathcal{I}^{-1} \nabla g(\breve{\theta}_n)\Big)^{-1} \nabla g(\breve{\theta}_n)^\top - \nabla g(\theta_n^*)\Big(\nabla g(\theta_n^*)^\top \mathcal{I}^{-1} \nabla g(\theta_n^*)\Big)^{-1} \nabla g(\theta_n^*)^\top$$

Suppose the constraint function $g(\theta)$ is smooth such that

$$\lim_{n\to\infty} \sup_{\breve{\theta}_n} \nabla f(\theta_n^*)^\top \mathcal{I}^{-1} \cdot \Delta(\breve{\theta}_n; \theta_n^*) \cdot \mathcal{I}^{-1} \nabla f(\theta_n^*) = 0. \tag{2.29}$$

Then the variance $V_{\theta_n^*}$ satisfies

$$V_{\theta_n^*} \geq \nabla f(\theta_n^*)^\top \Big[\mathcal{I}^{-1} - \mathcal{I}^{-1}\nabla g(\theta_n^*)\Big(\nabla g(\theta_n^*)^\top \mathcal{I}^{-1}\nabla g(\theta_n^*)\Big)^{-1}\nabla g(\theta_n^*)^\top \mathcal{I}^{-1}\Big]\nabla f(\theta_n^*) + o(1). \tag{2.30}$$

Condition (2.27) is known to be satisfied for most of the traditional settings. See [163]. Condition (2.29) is a mild condition such that $g(\theta)$ is smooth. The proof of Theorem 9 is relegated to Appendix 2.9.4.

On the other hand, we calculate the asymptotic variance obtained by our approach. According to Theorem 6, our proposed estimator θ_{est} satisfies

$$\sqrt{n}\big(\theta_{\text{est}} - \theta^*\big) \to \mathcal{N}\Big(0, P^*\mathcal{I}(\theta^*){P^*}^\top\Big).$$

Recall that P^* is defined in (2.13), and according to (2.50) we can write $P^* = M^* - P_M(\theta^*)M^*$ with

$$P_M(\theta^*) = M^*\nabla g(\theta^*)\Big[\nabla g(\theta^*)^\top M^*\nabla g(\theta^*)\Big]^{-1}\nabla g(\theta^*)^\top$$

Some simple calculation yields

$$\begin{aligned} P^*\mathcal{I}(\theta^*){P^*}^\top &= \Big(M^* - P_M(\theta^*)M^*\Big)\cdot\mathcal{I}(\theta^*)\cdot\Big(M^* - P_M(\theta^*)M^*\Big)^\top \\ &= M^* - M^*P_M(\theta^*)^\top - P_M(\theta^*)M^* + P_M(\theta^*)M^*P_M(\theta^*)^\top \\ &= M^* - P_M(\theta^*)M^* \end{aligned}$$

Applying the Delta method, we have

$$\sqrt{n}\Big(f(\theta_{\text{est}}) - f(\theta^*)\Big) \to \mathcal{N}\Big(0, \nabla f(\theta^*)^\top\cdot\big[M^* - P_M(\theta^*)M^*\big]\cdot\nabla f(\theta^*)\Big). \tag{2.31}$$

We then obtain the asymptotic variance of our proposed estimator. Since $M^* = \mathcal{I}^{-1}$, it is straightforward to verify that the variance obtained by our proposed method in (2.31) is the same as the minimal population variance in (2.30). This shows that our proposed estimator is asymptotic efficient. The form of the minimal variance is analogous to those in low dimension setting [178].

2.6 Applications to specific models

In this section we present some applications of our proposed methodology and show how to apply them to several commonly used equality constrained problems. Specifically, in Section 2.6.1 we discuss linear regression with linear constraint; Section 2.6.2 is for sparse PCA; Section 2.6.3 for single index model.

2.6.1 Linear regression with a linear constraint

We first consider the simplest case where the model is linear regression and the constraint is linear equality constraint. Specifically, we observe $y = X\theta^* + \epsilon$ where $y \in \mathbb{R}^{n\times 1}$, $X \in \mathbb{R}^{n\times p}$, and $\theta^* \in \mathbb{R}^{p\times 1}$. Here we focus on the random design for X. The true parameter $\|\theta^*\| = s$ is sparse and $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$. Throughout this section we assume σ^2 is known, and in practice it can be estimated by any consistent estimator. The objective function is $\ell(\theta) = \frac{1}{2n}\big\|y - X\theta\big\|_2^2$ and the constraint is $g(\theta) = A^\top\theta - r = 0$ where $r = A^\top\theta^*$ so that the true parameter θ^* satisfies the constraint. This gives $\nabla\ell(\theta) = \frac{1}{n}X^\top(X\theta - y)$, $H_\ell(\theta) = \frac{1}{n}X^\top X$, $\nabla g(\theta) = A$, and $H_g(\theta) = 0$. We see that both the Hessian $H_\ell(\theta)$ and the gradient $\nabla g(\theta)$ do not depend on θ. For notation simplicity we write $H = \frac{1}{n}X^\top X$.

In low dimensions, it is shown in [9] that we can solve the problem in closed form with the constrained least squares (CLS) estimator given by

$$\widehat{\theta}^{CLS} = \widehat{\theta}^{OLS} - \big(X^\top X\big)^{-1}A\Big(A^\top(X^\top X)^{-1}A\Big)^{-1}\big(A^\top\widehat{\theta}^{OLS} - r\big).$$

where $\widehat{\theta}^{OLS} = (X^\top X)^{-1}(X^\top y)$ is the usual OLS estimator. In high dimensions, since the true parameter θ^* is sparse, we consider the L_1 regularized constrained optimization problem

$$\begin{aligned} \min \quad & \frac{1}{2n}\left\|y - X\theta\right\|_2^2 + \lambda\|\theta\|_1 \\ \text{s.t.} \quad & A^\top\theta - r = 0 \end{aligned} \tag{2.32}$$

In practice, we can use the projected proximal gradient descent algorithm in Appendix 2.9.1 to solve for (2.32) to obtain $\widehat{\theta}$ and the subgradient $\widehat{\tau} \in \partial\|\widehat{\theta}\|_1$ satisfying

$$\lambda\tau = \frac{1}{n}X^\top(y - X\widehat{\theta}) - A \cdot \widehat{\mu}$$

Since we can always obtain a consistent estimator from linear regression with enough samples, this problem corresponds to Case 1, and it is straightforward to verify that $\mu^* = 0$.

In the methodology presented for general problem in Section 4.2, we have to define a population version P^* in the theoretical analysis, and plug in an estimator P in practice. This is because in order to apply Central Limit Theorem in (2.19), we must make sure that P^* is independent with the samples. If we instead use P in the theoretical analysis, then it breaks down the independence and hence the Central Limit Theorem is no longer valid. However, in the current setting, we have $\nabla\ell(\theta^*) = \frac{1}{n}X^\top\epsilon$ to be always normally distributed and so is $P \cdot \nabla\ell(\theta^*)$ as long as P is independent with ϵ. Therefore, we can avoid defining P^* and instead we only need to define the sample versions as

$$\begin{aligned} P &= M - MA\left[A^\top MA\right]^{-1}A^\top M \\ Q &= MA\left[A^\top MA\right]^{-1} \\ R &= -\left[A^\top MA\right]^{-1} \end{aligned} \tag{2.33}$$

where M denotes the approximate inverse of the Hessian matrix H, which has been extensively studied in the literature [166, 362, 309, 237]. Here we follow the nodewise Lasso method in [309] as described in Section 2.3. In this way we avoid using P^* and bounding the error terms as in Assumption 3. The proposed estimator is

$$\theta_{\text{est}} = \widehat{\theta} + P \cdot \lambda \widehat{\tau} \tag{2.34}$$

We then obtain the following theorem for linear regression with linear constraint problem with random design. It is asymptotically equivalent with the method in [277]. The sample complexity $s^2 \log^2 p/n = o(1)$ is in line with the literature on the unconstrained case. Note that it is straightforward to generalize our result to a fixed design case. See Theorem 2.1 in [309].

Theorem 10. *Consider the linear model with linear constraint problem where the rows of X are i.i.d. samples from Gaussian distribution $\mathcal{N}(0, \Sigma)$ with $\max_j \Sigma_{j,j} = \mathcal{O}(1)$. Suppose also $\lambda_{\min}(\Sigma) \geq c_{\min}$ for some positive constant $c_{\min} > 0$ where $\lambda_{\min}(\Sigma)$ denotes the smallest eigenvalue of Σ. The regularization parameter is set as $\lambda = \mathcal{O}(\sqrt{\log p/n})$ and $\lambda_j = \mathcal{O}(\sqrt{\log p/n})$ for all j in (2.22). Assume $s^2 \log^2 p/n = o(1)$, we have*

$$\sqrt{n}\big(\theta_{\text{est}} - \theta^*\big) = W + \Delta_\theta,$$

where $\|\Delta_\theta\|_\infty = o_{\mathbb{P}}(1)$ and

$$W = P \cdot \frac{1}{\sqrt{n}} X^\top \epsilon \sim \mathcal{N}\big(0, \sigma^2 P H P^\top\big).$$

2.6.2 Sparse PCA

As another example, we show how our proposed method can be applied to sparse principal component analysis (sparse PCA). In principal component analysis (PCA) [148] researchers aim to estimate the leading eigenvectors of the sample covariance matrix. However, in high dimensions, the classical PCA is shown to be inconsistent in estimating the leading eigenvector [228, 168]. To handle this problem, a common approach is to impose sparsity condition on the leading vectors, termed sparse PCA. Much progress has been made on sparse PCA in both methodology and theoretical perspective [322, 315, 317, 377, 361, 55, 10], with a focus on the estimation of the sparse eigenvectors. See also [378] for a recent overview. Less focus has been made on the statistical inference on the sparse PCA problem. [160] proposes a debiased approach for sparse PCA problems, with a focus on the first *loading vector* instead of our focus which is the leading eigenvector. In that way it does not involve the unit norm constraint and the procedure is different from ours. See later Remark 13 at the end of this section for a detailed comparison.

Problem setup. In sparse PCA we assume that we observe $x_1, \ldots, x_n$ which are p-dimensional random vectors with mean 0 and covariance matrix Σ^* given by

$$\Sigma^* = \sum_{i=1}^{d} \lambda_i u_i u_i^\top,$$

where $\lambda_1 > \lambda_2 \geq \ldots \geq \lambda_p$ are eigenvalues of Σ and u_i are the corresponding eigenvectors. Our interest is on the leading eigenvalue λ_1 and eigenvector u_1. Note that we have an eigen-gap $(\lambda_1 > \lambda_2)$ for identifiability. From now on, we will denote $\lambda_{\max} = \lambda_1$ as the leading eigenvalue and $\theta^* = u_1$ as the leading eigenvector. In addition, we further assume that θ^* is sparse with $\|\theta^*\|_0 = s$. Denote $\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^\top$ as the sample covariance matrix, the objective function we would like to maximize is $\ell(\theta) = \frac{1}{2}\theta^\top \widehat{\Sigma}\theta$ with $\nabla \ell(\theta) = \widehat{\Sigma}\theta$ and $H_\ell(\theta) = \widehat{\Sigma}$. Since we could always scale θ to obtain larger or smaller objective value, as a natural identifiability constraint on PCA, we have the constraint given by $g(\theta) = \frac{1}{2}(\|\theta\|_2^2 - 1)$ with $\nabla g(\theta) = \theta$ and $H_g(\theta) = I$. Compared to the standard PCA setting, we include additional factors 1/2 to both the objective and constraint function so that the gradient and Hessian do not involve any multiplicative constant.

The population version of the problem is given by

$$\begin{aligned} \max_{\theta} \quad & \frac{1}{2}\theta^\top \Sigma^* \theta \\ \text{s.t.} \quad & \frac{1}{2}\|\theta\|_2^2 = \frac{1}{2} \end{aligned}$$

with the Lagrangian function

$$L^*(\theta, \mu) = \frac{1}{2}\theta^\top \Sigma^* \theta + \frac{1}{2}\mu\big(\|\theta\|_2^2 - 1\big)$$

Clearly θ^* is the solution to the expected version problem. Denote μ^* as the corresponding Lagrangian multiplier, we have the KKT condition

$$\begin{aligned} \Sigma^*\theta^* + \mu^*\theta^* &= 0 \\ \frac{1}{2}\|\theta^*\|_2^2 &= \frac{1}{2} \end{aligned}$$

From the fact that $\Sigma^*\theta^* = \lambda_{\max}\theta^*$ we conclude that $\mu^* = -\lambda_{\max}$. Therefore, the analysis on the Lagrangian multiplier would provide asymptotic inference on the leading eigenvalue.

Find consistent $(\widehat{\theta}, \widehat{\mu})$**.** In order to apply our proposed method, we first need to find consistent estimators $(\widehat{\theta}, \widehat{\mu})$ satisfying $\|\widehat{\theta}\|_2 = 1$. There are many practical algorithms that could efficiently find such $\widehat{\theta}$ with the minimax rate $\|\widehat{\theta} - \theta^*\|_2 = \mathcal{O}_{\mathbb{P}}(\sqrt{s \log p/n})$, for example, see [316, 322]. Throughout this section we ignore the fact that $\widehat{\theta}$ is identifiable up to the sign, and assume that $\widehat{\theta}$ are in the same direction of θ^* so that the minimax rate holds. Furthermore, we could perform a hard-thresholding step with sparsity level $\mathcal{O}(s)$ on $\widehat{\theta}$ such that $\|\widehat{\theta}\|_0 = \mathcal{O}(s)$, and then normalize to keep the unit norm. According to Lemma 3.3 in [203], this hard thresholding step only amplifies the error rate by a constant, and hence the minimax rate $\|\widehat{\theta} - \theta^*\|_2 = \mathcal{O}_{\mathbb{P}}(\sqrt{s \log p/n})$ still holds. Another important observation is that

$$\theta^{*\top}(\widehat{\theta} - \theta^*) = \Big[\frac{1}{2}(\widehat{\theta} + \theta^*) - \frac{1}{2}(\widehat{\theta} - \theta^*)\Big]^\top \cdot (\widehat{\theta} - \theta^*) = -\frac{1}{2}\|\widehat{\theta} - \theta^*\|_2^2 \qquad (2.35)$$

is a higher order infinitesimal, where we use the fact that $\|\widehat{\theta}\|_2 = \|\theta^*\|_2 = 1$.

Next, we need to find a $\widehat{\mu}$ that is a consistent estimator of μ^*. Since we know that $\mu^* = -\lambda_{\max}$, we could use the following estimator

$$\widehat{\mu} = -\widehat{\theta}^\top \widehat{\Sigma} \widehat{\theta} \tag{2.36}$$

In order to quantify the error rate of $\widehat{\mu}$, we first quantify the error rate on $\widehat{\Sigma}$. According to Lemma 5.3 in [322], with high probability we have $\|\widehat{\Sigma} - \Sigma^*\|_{2,d} = \mathcal{O}_{\mathbb{P}}(\sqrt{d \log p/n})$ for any $d < p/2$, where we define

$$\|\widehat{\Sigma} - \Sigma^*\|_{2,d} = \sup_{\|v\|_0 \le d, \|v\|_2 = 1} |v^\top (\widehat{\Sigma} - \Sigma^*) v|$$

Therefore, for any u, v with $\|u\|_0 = \mathcal{O}(s), \|v\|_0 = \mathcal{O}(s)$ and $\|u\|_2 = \|v\|_2 = 1$ we have

$$\Big|u^\top (\widehat{\Sigma} - \Sigma^*) v\Big| = \Big|\Big(\frac{u+v}{2}\Big)^\top (\widehat{\Sigma} - \Sigma^*)\Big(\frac{u+v}{2}\Big) - \Big(\frac{u-v}{2}\Big)^\top (\widehat{\Sigma} - \Sigma^*)\Big(\frac{u-v}{2}\Big)\Big| = \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s \log p}{n}}\Big)$$

We then have

$$\begin{aligned}
|\mu^* - \widehat{\mu}| &= |\widehat{\theta}^\top (\widehat{\Sigma} - \Sigma^*) \widehat{\theta} + \widehat{\theta}^\top \Sigma^* \widehat{\theta} - \lambda_{\max}| \\
&\le |\widehat{\theta}^\top (\widehat{\Sigma} - \Sigma^*) \widehat{\theta}| + |(\widehat{\theta} - \theta^*)^\top \Sigma^* (\widehat{\theta} - \theta^*)| + |\theta^{*\top} \Sigma^* \theta^* + 2(\widehat{\theta} - \theta^*)^\top \Sigma^* \theta^* - \lambda_{\max}| \\
&= \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s \log p}{n}}\Big) + \mathcal{O}_{\mathbb{P}}\Big(\frac{s \log p}{n}\Big) + \mathcal{O}_{\mathbb{P}}\Big(\frac{s \log p}{n}\Big) \\
&= \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s \log p}{n}}\Big)
\end{aligned} \tag{2.37}$$

where we use (2.35) for the last term in the second equality. We then find a consistent estimator $\widehat{\mu}$ with the error rate given by (2.37).

Find P^*. The next step is to find the inverse of the Hessian matrix. Denote $H^* = \Sigma^* + \mu^* I$ as the population version Hessian matrix. Since $\mu^* = -\lambda_{\max}$, we can verify that the eigenvalues of H^* are given by $0, \lambda_2 - \lambda_{\max}, ..., \lambda_p - \lambda_{\max}$ with corresponding eigenvectors $u_1, ..., u_p$. Denote $U = [u_1, u_2, ..., u_p]$ with $UU^\top = U^\top U = I$, this gives the matrix form expression

$$H^* = U \cdot \operatorname{diag}\Big(0, (\lambda_2 - \lambda_{\max}), ..., (\lambda_p - \lambda_{\max})\Big) \cdot U^\top$$

We then see that H^* is not invertible and we could not obtain an exact inverse M^*. Fortunately, this problem can be solved in the following way. First, we observe that the choice of M^* (and the subsequent P^*, Q^*, R^*) is to make sure that the error term is negligible, and the error term consists of multiplication with $\widehat{\theta} - \theta^*$ according to (2.15). Next, we see that the Hessian H^* has eigenvalue 0 only in the direction of the first eigenvector $u_1 = \theta^*$. According to (2.35), the inner product of θ^* and $\widehat{\theta} - \theta^*$ is a higher-order infinitesimal. Therefore, although we could not invert H^* in the direction of θ^*, this will not affect the order of the error term since θ^* is almost perpendicular to $\widehat{\theta} - \theta^*$. Motivated by this observation, we could choose

$$M^* = U \cdot \operatorname{diag}\Big(0, (\lambda_2 - \lambda_{\max})^{-1}, ..., (\lambda_p - \lambda_{\max})^{-1}\Big) \cdot U^\top \tag{2.38}$$

In fact, this choice of M^* is the Moore-Penrose pseudo-inverse of H^*. We can verify that

$$I - M^*H^* = I - U \cdot \operatorname{diag}(0, 1, 1, ..., 1) \cdot U^\top = I - (u_2 u_2^\top + ... + u_p u_p^\top) = u_1 u_1^\top \tag{2.39}$$

This gives $(I - M^*H^*)u_1 = u_1$ and $(I - M^*H^*)u_j = 0$ for all $j = 2, ..., p$ and hence the linear combination of them. We then see that this M^* is an appropriate inverse of H^* except for the direction of θ^*. Moreover, we have

$$\begin{aligned} M^*\theta^* &= U \cdot \operatorname{diag}\Big(0, (\lambda_2 - \lambda_{\max})^{-1}, ..., (\lambda_p - \lambda_{\max})^{-1}\Big) \cdot U^\top u_1 \\ &= U \cdot \operatorname{diag}\Big(0, (\lambda_2 - \lambda_{\max})^{-1}, ..., (\lambda_p - \lambda_{\max})^{-1}\Big) \cdot e_1 \\ &= 0 \end{aligned} \tag{2.40}$$

This motivates the choice of P^*, Q^*, and R^* as

$$P^* = M^*, \qquad Q^* = \theta^*, \quad \text{and} \qquad R^* = 0$$

Construct estimator. Finally we construct the proposed estimator. According to (2.7), we define $\widehat{\tau}$ as

$$\widehat{\tau} = -\nabla \ell(\widehat{\theta}) - \nabla g(\widehat{\theta}) \cdot \widehat{\mu} = -\widehat{\Sigma}\widehat{\theta} - \widehat{\theta} \cdot \widehat{\mu} = -(\widehat{\Sigma} + \widehat{\mu} I) \cdot \widehat{\theta}$$

Denote $\widehat{H} = \widehat{\Sigma} + \widehat{\mu} I$ as the sample Hessian matrix. According to (2.17) and (2.20), the proposed estimators are

$$\theta_{\text{est}} = \widehat{\theta} - M \cdot \widehat{\tau} = \widehat{\theta} - M(\widehat{\Sigma} + \widehat{\mu} I)\widehat{\theta} \tag{2.41}$$

and

$$\mu_{\text{est}} = \widehat{\mu} - \widehat{\theta}^\top \cdot \widehat{\tau} = \widehat{\mu} - \widehat{\theta}^\top \cdot (\widehat{\Sigma} + \widehat{\mu} I)\widehat{\theta}. \tag{2.42}$$

In fact, according to the definition of $\widehat{\mu}$ in (2.36) we can verify that $\widehat{\theta}^\top \cdot (\widehat{\Sigma} + \widehat{\mu} I)\widehat{\theta} = \widehat{\theta}^\top \widehat{\Sigma} \widehat{\theta} + \widehat{\mu} \cdot \|\widehat{\theta}\|_2^2 = 0$. This means that we are not doing any modification on $\widehat{\mu}$, and $\widehat{\mu}$ itself is the proposed estimator that is unbiased and asymptotically normal.

To obtain valid statistical inference on eigenvector and eigenvalue, we impose the following assumption on the sparse PCA model.

Assumption 11. *Each x_i is sampled from a sub-Gaussian distribution with zero mean and covariance matrix Σ^*. The leading eigenvector $\theta^* = u_1$ is assumed to be sparse: $\|\theta^*\|_0 = s$. Denote m_j^* as the j^{th} row of M^* defined in* (2.38)*, we assume $\|m_j^*\|_0 = s$, and $\|m_j^*\|_2 = \mathcal{O}(1)$.*

We use the same s for the sparsity level of θ^* and m_j^* for notation simplicity. The sparsity condition on m_j^* is necessary to obtain an accurate estimation of M. This assumption is also assumed in [160].

Spiked covariance model. As an example, consider the spiked covariance model where the covariance matrix Σ^* is given as

$$\Sigma^* = I + w \cdot \theta^* {\theta^*}^\top \tag{2.43}$$

with $\|\theta^*\|_0 = s$ and $\|\theta^*\|_2 = 1$. It is straightforward to calculate that $M^* = -1/w \cdot (I - \theta^* {\theta^*}^\top)$. We can then verify that $\|m_j^*\|_0 \leq s$ and $\|m_j^*\|_2 \leq 1/w$ for each $j = 1, ..., p$. For that reason, we focus on sparse PCA with *a spiked covariance model only* in the main text.

For the spiked covariance model, we can calculate that $M^* = -1/w \cdot (I - \theta^* {\theta^*}^\top)$. Since the maximum eigenvalue is given by $w + 1$ and hence $\mu^* = -w - 1$, we can estimate w by $\widehat{w} = -\widehat{\mu} - 1$ with $\|\widehat{w} - w\|_2 = \mathcal{O}_{\mathbb{P}}(\sqrt{s \log p/n})$. We can then estimate M as

$$M = -\frac{1}{\widehat{w}} \cdot (I - \widehat{\theta}\widehat{\theta}^\top) \tag{2.44}$$

According to Lemma 16, this gives the error rate

$$\|m_j - m_j^*\|_2 = \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s \log p}{n}}\Big)$$

We conclude with the following theorem for the sparse PCA problem with a spiked covariance matrix.

Theorem 12. *Consider the sparse PCA model and suppose the Assumption 11 holds with a spiked covariance matrix. Assume $s^2 \log^2 p/n = o(1)$, the estimator θ_{est} defined in* (2.41) *satisfies*

$$\sqrt{n}\big(\theta_{\text{est}} - \theta^*\big) = W_\theta + \Delta_\theta,$$

where $\|\Delta_\theta\|_\infty = o_{\mathbb{P}}(1)$ *and*

$$W_\theta = n^{-1/2} \sum_{i=1}^{n} M^* x_i x_i^\top \theta^* \sim \mathcal{N}\big(0, \Sigma_\theta^*\big).$$

where Σ_θ^* *is estimated by* $\widehat{\Sigma}_\theta$ *defined in* (2.69). *Furthermore, the estimator* μ_{est} *defined in* (2.42) *satisfies*

$$\sqrt{n}\big(\mu_{\text{est}} - \mu^*\big) = W_\mu + \Delta_\mu$$

with $\mu^* = -\lambda_{\max}$, $\|\Delta_\mu\|_\infty = o_{\mathbb{P}}(1)$ *and*

$$W_\mu = n^{-1/2} \sum_{i=1}^{n} {\theta^*}^\top x_i x_i^\top \theta^* \sim \mathcal{N}\Big(0, {\sigma_\mu^*}^2\Big).$$

where ${\sigma_\mu^*}^2$ *is estimated by* $\widehat{\sigma}_\mu^2$ *defined in* (2.70).

Remark 13. *In [160] the authors propose a debiased procedure and obtain statistical inference for high dimensional sparse PCA problem. Different from our focus, their interest is on the first loading vector of the true covariance matrix* Σ^* *defined as*

$$\beta^* = \arg\min_\beta \frac{1}{4} \|\Sigma^* - \beta\beta^\top\|_F^2$$

Clearly we have $\beta^* = \sqrt{\lambda_{\max}} \cdot \theta^*$ *where* θ^* *is our focus, the leading eigenvector of* Σ^*. *Since we have* $\|\beta^*\|_2^2 = \lambda_{\max}$ *which is unknown in practice, the inference procedure in [160] does not involve norm constraint. Therefore it is different from our proposed method. Moreover, the sample complexity for our method is* $n \gg s^2 \log^2 p$, *which matches the result in [160] for the eigenvector. For the maximum eigenvalue, the result in [160] requires* $n \gg s^3 \log^2 p$, *and our proposed method improves this sample complexity to* $n \gg s^2 \log^2 p$.

2.6.3 Single index model

Single index model ([137]) is a semi-parametric generalization of the linear model where the response variable depends on the covariates through a composition of a univariate function and a linear transformation. Specifically, let $f(\cdot) : \mathbb{R} \to \mathbb{R}$ denote an univariate function and let $\theta^* \in \mathbb{R}^p$ be a vector, a single index model is assumes that the response $Y \in \mathbb{R}$ and the covariate $X \in \mathbb{R}^p$ satisfy

$$Y = f\big(\langle X, \theta^* \rangle\big) + \epsilon \tag{2.45}$$

where $\epsilon \in \mathbb{R}$ is the noise independent of X. Here f and θ^* in (2.45) are both unknown parameters and are called the nonparametric and parametric components, respectively. Since f is unknown, the norm of θ^* can always be absorbed into $f(\cdot)$. For model identifiability, we further assume that θ^* has unit norm, i.e., $\|\theta^*\|_2 = 1$.

The single index model has received great research interest in both theory and applications thanks to its semi-parametric structure. Specifically, the single index model enjoys extra modeling flexibility than parametric models by having the unknown link function f that incorporates possible model misspecification. Moreover, compared with full-fledged nonparametric models, the single-index model does not suffer from the curse of dimensionality by having a linear component $\langle X, \theta^* \rangle$ that reduces the problem of learning f to a univariate nonparametric regression problem once θ^* is known. Thus, the semi-parametric modeling of the single-index model strikes a balance between model flexibility and statistical efficiency.

Furthermore, since estimating f via univariate nonparametric methods hinges on how well we can recover θ^*, it is crucial to access the uncertainty of the estimation of θ^*. In the sequel, we apply the inferential method developed in Section 2.3 to the statistical inference of θ^*. Specifically, based on n i.i.d. observations $\{X_i, Y_i\}_{i=1}^n$ of the model in (2.45), we aim to construct an estimator of θ^* that enjoys asymptotic normality, under the constraint that $\|\theta\|_2 = 1$. Moreover, we consider the high-dimensional setting where θ^* has s nonzero entries with $s \ll n$, and the dimensionality p of θ^* is much larger than n. For ease of presentation, we further assume that the distribution of the covariate X is standard Gaussian $N(0, I)$, and we will introduce how to generalize to non-Gaussian settings in Remark 15.

Single index model is rather well studied in the low dimensional settings. See, e.g, [134, 275, 201, 155, 149]) and the references therein for details. However, much progress has been made for high-dimensional single index model only in recent years. For instance, [235, 247, 246, 135, 340, 118] study the statistical error of Lasso-type estimators and [167, 298, 206] propose estimators based on sliced inverse regression [201]. In addition, for single index model whose link function is similar to the quadratic function, also known as the misspecified phase retrieval model [58, 274], [236, 343] propose efficient methods with near-optimal statistical guarantees based on sparse PCA and Wirtinger flow ([58, 54]), respectively. Moreover, only a few work has focused on statistical inference for high-dimensional single index model, which is more related to our work. Specifically, [126] proposes statistical inference for high dimensional partially linear single index model; [145] proposes an inferential procedure for average partial effect in high dimensional single index model. Both of these two approaches are based on the debias/desparsify method [166, 309, 237], and ignore the unit norm constraint in the inference procedure, which are different from our method.

Most of the aforementioned methods estimate θ^* based on the celebrated Stein's identity, which extracts information of θ^* from the correlation between Y and X in the presence of an unknown function f.

First-order Stein's identity [288]. Let $X \sim \mathcal{N}(0, I)$ be a random vector and $h : \mathbb{R} \to \mathbb{R}$ be a continuous function such that $\mathbb{E}\big[\nabla h(X)\big]$ exists. Then we have $\mathbb{E}\big[h(X) \cdot X\big] = \mathbb{E}\big[\nabla h(X)\big]$.

According to the first order Stein's identity, we have

$$\mathbb{E}[Y \cdot X] = \mathbb{E}[f\big(\langle X, \theta^*\rangle\big) \cdot X] = \theta^* \cdot \mathbb{E}[f'(\langle X, \theta^*\rangle)].$$

Thus, provided that $\mathbb{E}[f'(\langle X, \theta^*\rangle)] \neq 0$, when n is sufficiently large, $n^{-1} \sum_{i=1}^{n} Y_i X_i$ is a reasonable estimator for θ^* up to scaling. This observation motivates the generalized Lasso approach proposed in [247, 246]. However, one can easily see that such a method fails when $\mathbb{E}[f'(\langle X, \theta^* \rangle)] \neq 0$, which is the case when the link function f is symmetric. One such example is the phase retrieval model where we have $f(u) = u^2$. In this case, Y and X are uncorrelated and we need to utilize the correlation between Y and the higher moments of X to extract useful information of θ^*. We need the following second-order Stein's identity.

Second order Stein's identity [164]. Let $X \sim \mathcal{N}(0, I)$ be a random vector and $h : \mathbb{R} \to \mathbb{R}$ be a continuous function such that $\mathbb{E}[\nabla^2 h(X)]$ exists. Let $T(x) = xx^\top - I$ be the second-order score function of the standard Gaussian distribution. Then we have $\mathbb{E}[h(X) \cdot T(X)] = \mathbb{E}[\nabla^2 h(X)]$.

Applying the second-order Stein's identity to the single index model by letting $h(x) = f(\langle x, \theta^* \rangle)$, we have

$$\mathbb{E}[Y \cdot T(X)] = \mathbb{E}[Y \cdot (XX^\top - I)] = C_0 \cdot \theta^* \theta^{*\top} \tag{2.46}$$

where $C_0 = 2\mathbb{E}[f''(\langle X, \theta^* \rangle)]$. Thus, even if $\mathbb{E}[f'(\langle X, \theta^* \rangle)]$ is equal to zero, as long as $C_0 \neq 0$, we can nevertheless extract θ^* from the covariance between Y and $T(X)$. Moreover, from (2.46) we know that θ^* is the only non-trivial eigenvector of $\Sigma^* = \mathbb{E}[Y \cdot T(X)]$, which motivates us to estimate θ^* by solving a sparse PCA problem for the sample version of Σ^*. Thus, we can apply the procedure introduced in Section 2.6.2 on the sample covariance matrix $\widehat{\Sigma} = n^{-1} \sum_{i=1}^n Y_i \cdot T(X_i)$. Suppose the true parameter θ^* is sparse with $\|\theta^*\|_0 = s$, we first apply the algorithm proposed in [236, 342] to obtain a consistent estimator $\widehat{\theta}$ with $\|\widehat{\theta} - \theta^*\|_2 = \mathcal{O}_{\mathbb{P}}(\sqrt{s \log p / n})$ with high probability. Next, we define $\widehat{\mu} = -\widehat{\theta}^\top \widehat{\Sigma} \widehat{\theta}$ as in (2.36) and $M = 1/\widehat{\mu} \cdot (I - \widehat{\theta}\widehat{\theta}^\top)$ as in (2.44). The population version of these two quantities are $\mu^* = -\theta^{*\top} \Sigma^* \theta^* = -C_0$ and $M^* = 1/\mu^* \cdot (I - \theta^* \theta^{*\top})$ Similar as (2.41), the proposed estimator is

$$\theta_{\text{est}} = \widehat{\theta} - M(\widehat{\Sigma} + \widehat{\mu} I)\widehat{\theta} \tag{2.47}$$

We obtain the following inferential results for the single index model.

Corollary 14. *Consider the single index model introduced in* (2.45) *where* $X \sim N(0, I)$ *and the link function* f *satisfies* $\mathbb{E}[f''(Z)] = C_0 > 0$, *where* $Z \sim N(0, 1)$. *We assume that the response* Y *is a sub-exponential random variable. The true parameter* θ^* *is sparse with* $\|\theta^*\|_0 = s$. *Assume* $n > C \log^2 n \cdot (s \log p + \log n)$ *and* $(s \log p + \log n)^2/n = o(1)$, *the estimator* θ_{est} *defined in* (2.47) *satisfies*

$$\sqrt{n}\big(\theta_{\text{est}} - \theta^*\big) = W_\theta + \Delta_\theta,$$

where $\|\Delta_\theta\|_\infty = o_{\mathbb{P}}(1)$ *and*

$$W_\theta = n^{-1/2} \sum_{i=1}^{n} M^* y_i \cdot (x_i x_i^\top - I)\theta^* \sim \mathcal{N}\big(0, \Sigma_\theta^*\big).$$

where Σ_θ^* *is estimated by* $\widehat{\Sigma}_\theta$ *defined in* (2.72).

Remark 15. *We mark that in Corollary 14 we assume* $C_0 > 0$ *without loss of generality. When* $C_0 < 0$ *we can apply the same procedure with* $\Sigma = -\sum_{i=1}^{n} Y_i \cdot T(X_i)$. *Moreover, we note that it is not hard to generalize our approach to non-Gaussian settings where the density* $p_x \colon \mathbb{R}^p \to \mathbb{R}$ *of the covariate* X *is known. In this case, a generalized version of the second-order Stein's identity holds. Specifically, we define the second-order score function as* $T_X(x) = \nabla^2 p_X(x)/p_X(x) \in \mathbb{R}^{p\times p}$. *It can be easily verified that* $T_X(x) = xx^\top - I$ *when* p_X *is the density of the standard Gaussian distribution. Then for any twice-differentiable function* h, *it holds that* $\mathbb{E}[h(X) \cdot T_X(X)] = \mathbb{E}[\nabla^2 h(X)]$. *Thus, for the single index model in* (2.45) *where* X *has density* p_X, *under the assumption that* $\mathbb{E}[f''(\langle X, \theta^*\rangle)] \neq 0$, *we can recover* θ^* *by solving a sparse PCA problem for* $\Sigma^* = \mathbb{E}[Y \cdot T_X(X)]$. *Moreover, since* $T_X(X)$ *can be a heavy-tailed random vector for general density* p_X, *we could apply truncation techniques to each* $T_X(X_i)$ *to obtain near-optimal statistical guarantees in the finite-sample setting. See [338, 339] for more details.*

2.7 Experiments

In this section we demonstrate the effectiveness of our method through experiments. We focus on linear regression with linear constraint and sparse PCA problem.

Linear regression with linear constraint. We first consider linear regression with linear constraint problem as discussed in Section 2.6.1. For the experiment, we set $\sigma = 0.8$ and we assume σ is known. Each row of X is sampled from multivariate normal distribution $X \sim N(0, \Sigma)$, where Σ is a Toeplitz matrix with $\Sigma_{jk} = \rho^{|j-k|}$ and $\rho = 0.5$. The true parameter β^* is set to be sparse with $s = \|\beta^*\|_0 = 10$ where set $\beta_{1:5} = 1$ and $\beta_{6:10} = -1$. The constraint is set as $A^\top \beta = r$ where $r = A^\top \beta^*$ so that the true parameter satisfies the constraint. We have $A \in \mathbb{R}^{p \times q}$ where q denotes the number of constraints and each component of A is i.i.d. drawn from a standard normal distribution. The regularization parameter is set to be $\lambda = \frac{1}{2}\sqrt{\log p/n}$. We initialize with Lasso estimator and follow Algorithm 1 to obtain the constrained estimator $\widehat{\theta}$. Specifically, the proximal step in Algorithm 1 is given by soft-thresholding $\theta_t \leftarrow \mathrm{sign}(\theta_t) \cdot \max\big(\theta_t - \lambda\eta_t, 0\big)$; and the projection step in Algorithm 1 is a projection onto linear space given by $\theta_t \leftarrow \theta_t - A\big(A^\top A\big)^{-1}\big(A^\top \theta_t - r\big)$. We then follow our proposed method to obtain the estimator θ_{est}. For the experiment, we fix $p = 500$ and vary $n \in \{250, 400, 600\}$, $q \in \{5, 15, 40\}$, and the designed coverage rate is 95%. The averaged empirical coverage rate on 1000 replicates with different choices of n and q are shown in Table 2.1 - 2.3. In Table 2.1 - 2.3, column 2 - 6 show empirical coverage rate for some specific components of θ, which include both the support of θ and its complement. Column $\theta_{1:s}$ shows the average empirical coverage rate on the first s components of θ corresponding to the support of θ^*; column $\theta_{s+1:p}$ shows the average empirical coverage rate on the supplement of the support. Moreover, Figure 2.1 shows the histogram for several components of θ with $q = 15$ and $n = 600$. We see that our method works well on this problem.

We then demonstrate that our method achieves smaller asymptotic variance, and hence narrower confidence interval for the parameters, as illustrated in Remark 7. Here we fix

$n = 400, p = 300$ and vary the number of constraints $q \in \{5, 15, 40, 100, 200\}$. Although in practice it is unlikely to have such many constraints, this is merely an illustration of the variance reduction. Following Remark 7, our estimator is given by $\theta_{\text{est}} = \widehat{\theta} + P \cdot \widehat{\tau}$ with variance $\sigma^2 PHP^\top$; while when we ignore the constraint, the estimator is given by $\theta_{\text{est,un}} = \widehat{\theta}_{\text{un}} + M \cdot \widehat{\tau}_{\text{un}}$ with variance $\sigma^2 MHM^\top$. Table 2.4 shows the empirical variance of the estimator of our method and the unconstrained method. As before, column 2 - 6 show the variance of some specific components, while column $\theta_{1:s}$, $\theta_{s+1:p}$ show the average variance on support and its complement of θ^*, respectively. We see that by considering the equality constraint and following our method, we always get a smaller variance. When the number of constraints is small, the projection space is small and hence the amount of variance reduction is not that much. This is not surprising because we cannot expect to reduce the uncertainty on a high dimensional model with only a few constraints.

Table 2.1: Empirical coverage rate on θ for linear regression with linear constraint problem with sample size $n = 250$

	θ_1	θ_5	θ_{10}	θ_{50}	θ_{200}	$\theta_{1:s}$	$\theta_{s+1:p}$
$q = 5$	93.9%	92.9%	93.1%	96.3%	96.7%	91.90%	96.88%
$q = 15$	93.6%	94.0%	94.1%	96.4%	96.6%	93.64%	96.70%
$q = 40$	95.6%	92.8%	95.2%	95.7%	96.1%	94.67%	96.54%

Table 2.2: Empirical coverage rate on θ for linear regression with linear constraint problem with sample size $n = 400$

	θ_1	θ_5	θ_{10}	θ_{50}	θ_{200}	$\theta_{1:s}$	$\theta_{s+1:p}$
$q = 5$	93.8%	91.5%	91.5%	97.3%	97.2%	92.19%	96.26%
$q = 15$	95.1%	91.9%	96.0%	96.6%	96.4%	93.61%	96.12%
$q = 40$	95.6%	94.6%	95.2%	95.6%	96.1%	95.05%	96.05%

Sparse PCA. We then consider sparse PCA problems as discussed in Section 2.6.2. For the experiment, we focus on the spike covariance model (2.43) and set the sparsity level of

Table 2.3: Empirical coverage rate on θ for linear regression with linear constraint problem with sample size $n = 600$

	θ_1	θ_5	θ_{10}	θ_{50}	θ_{200}	$\theta_{1:s}$	$\theta_{s+1:p}$
$q = 5$	95.8%	93.3%	93.5%	96.1%	95.9%	93.39%	95.90%
$q = 15$	94.7%	94.5%	94.5%	96.8%	95.2%	94.22%	95.89%
$q = 40$	96.8%	93.8%	94.9%	96.2%	95.2%	94.97%	95.74%

Table 2.4: Variance reduction by using our method for linear regression with linear constraint problem with different number of constraint q

	method	θ_1	θ_5	θ_{10}	θ_{50}	θ_{200}	$\theta_{1:s}$	$\theta_{s+1:p}$
$q = 5$	our method	1.229	1.431	1.369	1.461	1.350	1.438	1.436
	unconstrained	1.247	1.461	1.388	1.479	1.397	1.442	1.462
$q = 15$	our method	1.159	1.326	1.506	1.460	1.413	1.332	1.395
	unconstrained	1.189	1.393	1.611	1.530	1.447	1.412	1.477
$q = 40$	our method	1.058	1.364	1.365	1.276	1.200	1.239	1.259
	unconstrained	1.184	1.536	1.465	1.487	1.518	1.438	1.474
$q = 100$	our method	0.915	0.936	1.028	0.863	1.015	0.929	0.952
	unconstrained	1.259	1.451	1.413	1.467	1.568	1.444	1.476
$q = 200$	our method	0.371	0.410	0.481	0.419	0.592	0.470	0.463
	unconstrained	1.323	1.523	1.528	1.597	1.604	1.486	1.482

the leading eigenvector θ^* as $s = 10$. Each components of θ^* are set to be $1/\sqrt{s}$ so that $\|\theta^*\|_2 = 1$. The leading eigenvalue of Σ^* is $\lambda_{\max} = w + 1$, and all the other eigenvalues are 1. We consider two choices of $w \in \{2, 5\}$ with $\lambda_{\max} \in \{3, 6\}$ corresponding to large and small eigen-gap. We then get Σ^* according to (2.43) and we sample X from p-dimensional Gaussian distribution $N(0, \Sigma^*)$.

To apply our Algorithm 1, we could initialize by any efficient algorithm for sparse PCA problem. Here we use the inverse power method proposed in [144]. The regularization parameter is set to be $\lambda = \frac{1}{2}\sqrt{\log p/n}$. The proximal step in Algorithm 1 is again a soft-thresholding, and the projection step is to normalize θ_t to unit-norm. For the experiment, we fix $p = 200$ and vary $n \in \{200, 400, 800\}$, $\lambda_{\max} \in \{3, 6\}$, and the designed coverage rate is 95%. The averaged empirical coverage rate on 1000 replicates with different choices of n and $\lambda_{\max}$ are shown in Table 2.5 and 2.6. The last column shows the empirical coverage rate for the leading eigenvalue $\lambda_{\max}$, while the other columns are for several components of leading eigenvector θ^* and their average as before. We see from Table 2.5 that when the eigen-gap is small, we need more samples in order to obtain designed coverage rate; while when the eigen-gap is large, the problem becomes much easier, as shown in Table 2.6. This observation is consistent with the experimental result in [160]. Finally, Figure 2.2 shows the histogram for $\lambda_{\max}$ and several components of θ, for both $\lambda_{\max} = 3$ and $\lambda_{\max} = 6$ with $n = 800$. Once again, we see that our method works well on this sparse PCA problem.

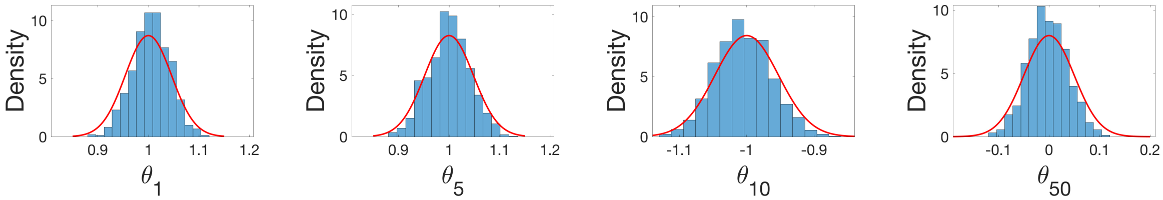


Figure 2.1: Histograms for several components of θ on linear regression with linear constraint problem. The red curve is the population version probability density function.

Table 2.5: Empirical coverage rate on θ and $\lambda_{\max}$ for sparse PCA problem with leading eigenvalue $\lambda_{\max} = 3$ and for different sample sizes

	θ_1	θ_5	θ_{10}	θ_{50}	θ_{200}	$\theta_{1:s}$	$\theta_{s+1:p}$	$\lambda_{\max}$
$n = 200$	80.1%	79.5%	80.1%	94.0%	93.3%	80.68%	94.22%	75.3%
$n = 400$	91.1%	91.5%	91.8%	94.6%	94.8%	90.90%	94.67%	85.6%
$n = 800$	92.1%	94.5%	94.2%	95.0%	94.2%	92.80%	94.76%	91.9%

Table 2.6: Empirical coverage rate on θ and $\lambda_{\max}$ for sparse PCA problem with leading eigenvalue $\lambda_{\max} = 6$ and for different sample sizes

	θ_1	θ_5	θ_{10}	θ_{50}	θ_{200}	$\theta_{1:s}$	$\theta_{s+1:p}$	$\lambda_{\max}$
$n = 200$	93.1%	91.8%	91.9%	94.0%	94.5%	92.45%	94.62%	93.7%
$n = 400$	93.1%	94.0%	94.1%	94.9%	94.8%	93.94%	94.82%	95.1%
$n = 800$	94.6%	95.6%	94.9%	95.4%	94.7%	94.73%	94.90%	94.6%

2.8 Conclusion

Most of the existing works in high dimensional statistical inference literature assume no constraint on the model parameters. In this chapter we consider the statistical inference in high dimensional models with equality constraints on the parameters, where the constraints

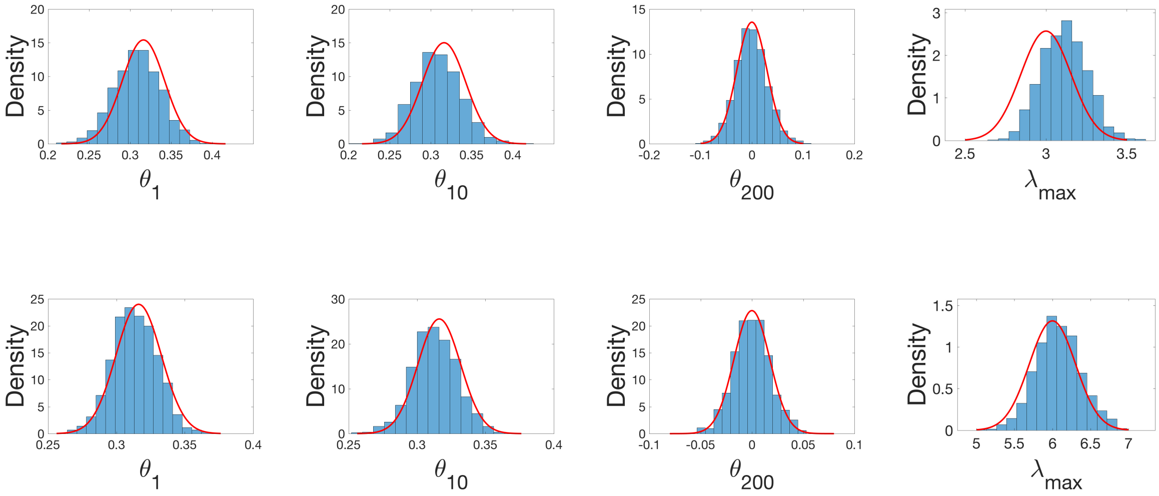


Figure 2.2: Histograms for θ and $\lambda_{\max}$ on sparse PCA problem: the first row is for $\lambda_{\max} = 3$ and the second row is for $\lambda_{\max} = 6$. The red curve is the population version probability density function.

serve as either additional information, or as an intrinsic restriction. For both of the two cases, we show that following our proposed procedure we obtain valid asymptotic confidence intervals on the model parameters with the equality constraints taken into account. The proposed estimator enjoys a smaller variance than the standard estimators, and is semi-parametrically efficient. As demonstrated through numerical experiments, the proposed method has a broad range of applications, including linear regression with linear constraints, sparse PCA and single index model with unit norm constraint, etc. An interesting future extension would be to perform statistical inference with both equality and inequality constraints on the model parameter. Another direction would be to extend our results to non-differentiable constraints.

2.9 Technical details

2.9.1 Efficient algorithm for solving the constrained regularized problem

We provide an efficient algorithm to solve for the constrained regularized problem (2.4). The constrained problem (2.4) is non-convex when the loss function $\ell(\theta)$ is non-convex or the constraint $g(\theta)$ is non-linear. To solve for $(\widehat{\theta}, \widehat{\mu})$, we could initialize with a consistent estimator that may not satisfy the constraint, and apply projected proximal gradient descent algorithm. This algorithm is summarized in Algorithm 1. The choice of initialization is problem specific. For example, for linear regression we can initialize with Lasso estimator. Note that we do not need $\widehat{\theta}$ to be the global minimum of (2.4); all we need is a consistent estimator that satisfies the constraint. For the problems where the projection onto the constraint space is intractable, we could turn to other approaches, for example the interior point method, augmented Lagrangian method, etc. The detailed discussion of the optimization algorithm is beyond the scope of this work.

Algorithm 1 Projected proximal gradient descent algorithm for solving (2.4)

```
  Initialize with a consistent estimator $\theta_0$
  while $tolerance > \epsilon$ do
    $\theta_t \leftarrow \theta_t - \eta_t \nabla \ell(\theta_t)$    // gradient descent
    $\theta_t \leftarrow \text{prox}_{\eta_t P_\lambda(\cdot)}(\theta_t)$    // proximal operator
    $\theta_{t+1} \leftarrow \text{proj}_{g(\theta)=0}(\theta_t)$    // project to constraint space
  end while
```

2.9.2 Degenerate case

In this section we show that our method is asymptotically equivalent to the methods in the current literature in degenerate case. Specifically, we denote $\theta = (\alpha, \beta)$ where α is a scalar and $\beta \in \mathbb{R}^{p-1}$, and the constraint is given by $\alpha = 0$. In this case, we could either view it as a equality constraint problem and follow our proposed method, or we could plug in $\alpha = 0$ and view it as an unconstrained problem on β. With some abuse of notation, we denote the loss function as $\ell(\theta) = \ell(\alpha, \beta)$ and the constraint as $g(\theta) = g(\alpha, \beta) = \alpha$ with $\nabla g(\theta) = e_1$ where e_j is the j^{th} unit column vector. The constrained optimization problem is given by

$$\begin{aligned} \min_{\theta} \quad & \ell(\theta) + P_\lambda(\theta) \\ \text{s.t.} \quad & \alpha = 0 \end{aligned}$$

and the unconstrained problem (after plugging in $\alpha = 0$) is given by

$$\min_{\theta} \quad \ell(0, \beta) + P_\lambda(0, \beta) \tag{2.48}$$

Clearly these two problems have the same solution $\widehat{\theta} = (0, \widehat{\beta})$. To be concise, in this section we focus on asymptotic performance and ignore the error terms. Denote $H^*_{\alpha\alpha}, H^*_{\alpha\beta}, H^*_{\beta\alpha}, H^*_{\beta\beta}$ as the corresponding partitions of H^*_ℓ. We focus on Case 1 only with $\mu^* = 0$, and $\ell(\theta)$ is given by negative log-likelihood so that $\nabla \ell(\theta^*)$ has asymptotic variance H^*. Once again we denote M^* as the inverse of H^*, according to the definition of P^* in (2.13) and after some calculations we have

$$\begin{aligned} P^* &= M^* - M^* e_1 \left[e_1^\top M^* e_1 \right]^{-1} e_1^\top M^* \\ &= \begin{bmatrix} 0 & 0 \\ 0 & \overline{M}^* \end{bmatrix} \end{aligned}$$

with $\overline{M}^* \in \mathbb{R}^{p-1 \times p-1}$ defined as

$$\overline{M}^* = M^*_{\beta\beta} - M^*_{\beta\alpha} \left[M^*_{\alpha\alpha} \right]^{-1} M^*_{\alpha\beta}$$

We then have the proposed estimator $\theta_{\text{est}} = \widehat{\theta} + P^* \cdot \widehat{\tau}_\theta$ and the asymptotic variance

$$P^* H^* P^{*\top} = \begin{bmatrix} 0 & 0 \\ 0 & \overline{M}^* \end{bmatrix} \begin{bmatrix} H^*_{\alpha\alpha} & H^*_{\alpha\beta} \\ H^*_{\beta\alpha} & H^*_{\beta\beta} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & \overline{M}^* \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & \overline{M}^* H^*_{\beta\beta} \overline{M}^{*\top} \end{bmatrix}$$

Specifically, we have $\alpha_{\text{est}} = 0$ with variance 0, which is as expected since $\alpha = 0$ is fixed by the constraint; $\beta_{\text{est}} = \widehat{\beta} + \overline{M}^* \widehat{\tau}_\beta$ with variance $\overline{M}^* H^*_{\beta\beta} \overline{M}^{*\top}$.

On the other hand, after plugging in the constraint $\alpha = 0$, we have the unconstrained problem on β as shown in (2.48), and the Hessian matrix with respect to β is $H^*_{\beta\beta}$. Since M^* is the inverse of H^*, we have

$$\begin{aligned} M^*_{\alpha\alpha} H^*_{\alpha\alpha} + M^*_{\alpha\beta} H^*_{\beta\alpha} &= I \\ M^*_{\alpha\alpha} H^*_{\alpha\beta} + M^*_{\alpha\beta} H^*_{\beta\beta} &= 0 \\ M^*_{\beta\alpha} H^*_{\alpha\alpha} + M^*_{\beta\beta} H^*_{\beta\alpha} &= 0 \\ M^*_{\beta\alpha} H^*_{\alpha\beta} + M^*_{\beta\beta} H^*_{\beta\beta} &= I \end{aligned}$$

It is then straightforward to verify that

$$
\begin{aligned}
\overline{M}^* \cdot H^*_{\beta\beta} &= M^*_{\beta\beta}H^*_{\beta\beta} - M^*_{\beta\alpha}\left[M^*_{\alpha\alpha}\right]^{-1}M^*_{\alpha\beta}H^*_{\beta\beta} \\
&= I - M^*_{\beta\alpha}H^*_{\alpha\beta} + M^*_{\beta\alpha}\left[M^*_{\alpha\alpha}\right]^{-1} \cdot M^*_{\alpha\alpha}H^*_{\alpha\beta} \\
&= I
\end{aligned}
$$

Therefore, after plugging in the constraint $\alpha = 0$, the unconstrained estimator is given by $\beta_{\text{est,plug}} = \widehat{\beta} + \overline{M}^*\widehat{\tau}_\beta$ with asymptotic variance $\overline{M}^* H^*_{\beta\beta}\overline{M}^{*\top}$. We then see that asymptotically this is the same as our proposed method, which verifies that these two methods are asymptotically equivalent.

2.9.3 Proof of Theorem 6

Proof. According to (2.15), the error term is given by

$$
\text{error} = \begin{bmatrix} I - P^*H_\ell(\bar{\theta}_1) - \widehat{\mu}P^*H_g(\bar{\theta}_2) - Q^*\nabla g(\bar{\theta}_3)^\top & -P^*\nabla g(\theta^*) \\ -Q^{*\top}H_\ell(\bar{\theta}_1) - \widehat{\mu}Q^{*\top}H_g(\bar{\theta}_2) - R^*\nabla g(\bar{\theta}_3)^\top & I - Q^{*\top}\nabla g(\theta^*) \end{bmatrix} \cdot \begin{bmatrix} \widehat{\theta} - \theta^* \\ \widehat{\mu} - \mu^* \end{bmatrix} \tag{2.49}
$$

According to the definition of P^*, Q^*, R^* in (2.13), we have $P^*\nabla g(\theta^*) = 0$ and $Q^{*\top}\nabla g(\theta^*) = I$ and hence the error term does not depend on $\widehat{\mu} - \mu^*$. Denote

$$
P_M(\theta^*) = M^*\nabla g(\theta^*)\left[\nabla g(\theta^*)^\top M^*\nabla g(\theta^*)\right]^{-1}\nabla g(\theta^*)^\top \tag{2.50}
$$

as a projection matrix so that $P^* = M^* - P_M(\theta^*)M^*$, we focus on the first line of (2.49) and obtain

$$
\begin{aligned}
& I - P^* H_\ell(\bar\theta_1) - \widehat\mu P^* H_g(\bar\theta_2) - Q^* \nabla g(\bar\theta_3)^\top \\
&= I - \Big[M^* - P_M(\theta^*) M^*\Big]\Big[H_\ell(\bar\theta_1) + \widehat\mu H_g(\bar\theta_2)\Big] \\
&\quad - M^* \nabla g(\theta^*)\Big[\nabla g(\theta^*)^\top M^* \nabla g(\theta^*)\Big]^{-1}\Big[\nabla g(\bar\theta_3) - \nabla g(\theta^*) + \nabla g(\theta^*)\Big]^\top \\
&= I - P_M(\theta^*) - \Big[I - P_M(\theta^*)\Big] M^* \Big[H_\ell(\bar\theta_1) + \widehat\mu H_g(\bar\theta_2)\Big] \\
&\quad - M^* \nabla g(\theta^*)\Big[\nabla g(\theta^*)^\top M^* \nabla g(\theta^*)\Big]^{-1} \nabla g(\theta^*)^\top \nabla g(\theta^*)\Big[\nabla g(\theta^*)^\top \nabla g(\theta^*)\Big]^{-1}\Big[\nabla g(\bar\theta_3) - \nabla g(\theta^*)\Big]^\top \\
&= \Big[I - P_M(\theta^*)\Big]\Big[I - M^*\big[H_\ell(\bar\theta_1) + \widehat\mu H_g(\bar\theta_2)\big]\Big] \\
&\qquad\quad + P_M(\theta^*) \nabla g(\theta^*)\Big[\nabla g(\theta^*)^\top \nabla g(\theta^*)\Big]^{-1}\Big[\nabla g(\bar\theta_3) - \nabla g(\theta^*)\Big]^\top
\end{aligned}
\tag{2.51}
$$

Back to the equation (2.14), we focus on the first line and obtain

$$
\widehat\theta - \theta^* = -P^*\Big(\nabla \ell(\theta^*) + \widehat\tau\Big) + \Big[I - P^* H_\ell(\bar\theta_1) - \widehat\mu P^* H_g(\bar\theta_2) - Q^* \nabla g(\bar\theta_3)^\top\Big](\widehat\theta - \theta^*)
$$

and hence

$$
\begin{aligned}
\theta_{\text{est}} = \widehat\theta + P \cdot \widehat\tau = \theta^* - P^* \cdot \nabla \ell(\theta^*) + (P - P^*) \cdot \widehat\tau \\
+ \Big[I - P^* H_\ell(\bar\theta_1) - \widehat\mu P^* H_g(\bar\theta_2) - Q^* \nabla g(\bar\theta_3)^\top\Big](\widehat\theta - \theta^*)
\end{aligned}
$$

According to Assumption 3, (2.51), and the fact that both $P_M(\theta^*)$ and $I - P_M(\theta^*)$ are projection matrices, we have

$$
\sqrt{n}(\theta_{\text{est}} - \theta^*) = -P^* \cdot \sqrt{n} \nabla \ell(\theta^*) + o_{\mathbb{P}}(1)
$$

Our claim then follows from Assumption 5. Similarly, we focus on the second line of (2.49) and obtain

$$
\begin{aligned}
&- Q^{*\top} H_\ell(\bar{\theta}_1) - \widehat{\mu} Q^{*\top} H_g(\bar{\theta}_2) - R^* \nabla g(\bar{\theta}_3)^\top \\
&= -\Big[\nabla g(\theta^*)^\top M^* \nabla g(\theta^*)\Big]^{-1} \nabla g(\theta^*)^\top M^* [H_\ell(\bar{\theta}_1) + \widehat{\mu} H_g(\bar{\theta}_2)] \\
&\qquad + \Big[\nabla g(\theta^*)^\top M^* \nabla g(\theta^*)\Big]^{-1} [\nabla g(\theta^*) + \nabla g(\bar{\theta}_3) - \nabla g(\theta^*)]^\top \\
&= \Big[\nabla g(\theta^*)^\top M^* \nabla g(\theta^*)\Big]^{-1} \nabla g(\theta^*)^\top \Big[I - M^* [H_\ell(\bar{\theta}_1) + \widehat{\mu} H_g(\bar{\theta}_2)]\Big] \\
&\qquad + \Big[\nabla g(\theta^*)^\top M^* \nabla g(\theta^*)\Big]^{-1} [\nabla g(\bar{\theta}_3) - \nabla g(\theta^*)]^\top
\end{aligned}
$$

Similar as the analysis for θ, we obtain

$$
\sqrt{n}\big(\mu_{\text{est}} - \mu^*\big) = -\sqrt{n}\Big(Q^{*\top} \cdot \nabla \ell(\theta^*) + \mu^*\Big) + o_{\mathbb{P}}(1)
$$

and our claim again follows from Assumption 5.

□

2.9.4 Proof of Theorem 9

Proof. With Assumption 8 and (2.27), Theorem 9 in [163] gives

$$
\sqrt{n}\Big(T_n - f\big(\widetilde{\theta}_n\big)\Big) \to \mathcal{N}(0, V_{\theta_n^*})
$$

This proves (2.28). We then calculate the minimal asymptotic variance. According to (2.27), the Cauchy-Schwarz inequality gives

$$
V_{\theta_n^*} \cdot h^\top \mathcal{I}(\theta_n^*) h \geq \big(h^\top \nabla f(\theta_n^*)\big)^2 - o\big(V_{\theta_n^*} \cdot h^\top \mathcal{I}(\theta_n^*) h\big)^{1/2} \tag{2.52}
$$

We can then choose h to obtain a tight lower bound on asymptotic variance, where we need to consider the constraint given as

$$
g(\widetilde{\theta}_n) = g\Big(\theta_n^* + \frac{h}{\sqrt{n}}\Big) = 0
$$

Taking the Taylor expansion and using the fact that $g(\theta_n^*) = 0$, we obtain

$$\nabla g(\breve{\theta}_n)^\top \cdot h = 0$$

for some intermediate value $\breve{\theta}_n$. With this constraint, the minimal variance can be calculated through solving the following optimization problem over h:

$$\begin{aligned} \max_h \quad & \frac{\big(h^\top \nabla f(\theta_n^*)\big)^2}{h^\top \mathcal{I}(\theta_n^*) h} \\ \text{s.t.} \quad & \nabla g(\breve{\theta}_n)^\top \cdot h = 0 \end{aligned} \tag{2.53}$$

Note that h is identifiable up to a scalar. To solve for h, we can write down the Lagrangian function and the KKT condition gives

$$\begin{aligned} h^\top \nabla f \cdot \Big(\nabla f \cdot h^\top \mathcal{I} h - h^\top \nabla f \cdot \mathcal{I} h\Big) + \lambda_h \cdot \nabla g \cdot (h^\top \mathcal{I} h)^2 = 0 \\ \nabla g^\top \cdot h = 0 \end{aligned}$$

where we omit θ for notation simplicity. Here λ_h denotes the Lagrangian multiplier of (2.53). Similar as (2.10), we can rewrite as a matrix form

$$\begin{bmatrix} -(h^\top \nabla f)^2 \cdot \mathcal{I} & \nabla g \\ \nabla g^\top & 0 \end{bmatrix} \cdot \begin{bmatrix} h \\ \lambda_h \cdot (h^\top \mathcal{I} h)^2 \end{bmatrix} = \begin{bmatrix} -(h^\top \nabla f) \cdot (h^\top \mathcal{I} h) \cdot \nabla f \\ 0 \end{bmatrix} \tag{2.54}$$

Since h is identifiable up to a scalar, the scalars in (2.54) do not play a role and therefore we can view them as some constant. According to the block matrix inversion formula, we solve for (2.54) and obtain the optimal $\widehat{h}$ as

$$\widehat{h} = c \cdot \Big[\mathcal{I}^{-1} - \mathcal{I}^{-1} \nabla g \Big(\nabla g^\top \mathcal{I}^{-1} \nabla g\Big)^{-1} \nabla g^\top \mathcal{I}^{-1}\Big] \cdot \nabla f$$

as long as $\widehat{h}$ is feasible in that $\theta_n^* + \widehat{h}/\sqrt{n}$ satisfies (2.26). Plug back into (2.52), the (asymptotic) minimum variance is given as

$$V_{\theta_n^*} \geq \nabla f(\theta_n^*)^\top \Big[\mathcal{I}^{-1} - \mathcal{I}^{-1}\nabla g(\breve{\theta}_n)\Big(\nabla g(\breve{\theta}_n)^\top \mathcal{I}^{-1}\nabla g(\breve{\theta}_n)\Big)^{-1}\nabla g(\breve{\theta}_n)^\top \mathcal{I}^{-1}\Big]\nabla f(\theta_n^*) \quad (2.55)$$

We then obtain the lower bound on the asymptotic variance. Equation (2.29) then ensures that the difference between the minimum variance in (2.55) and its population version as in (2.30) is $o(1)$.

□

2.9.5 Proof of Theorem 10

Proof. According to (2.10) we have

$$\begin{bmatrix} \frac{1}{n}X^\top X & A \\ A^\top & 0 \end{bmatrix} \cdot \begin{bmatrix} \widehat{\theta} - \theta^* \\ \widehat{\mu} \end{bmatrix} = \begin{bmatrix} \frac{1}{n}X^\top \epsilon - \lambda\widehat{\tau} \\ 0 \end{bmatrix} \quad (2.56)$$

Recall that M denotes the approximate inverse of the Hessian matrix $H = 1/nX^\top X$. We define the projection matrix $P_M = MA\big[A^\top MA\big]^{-1}A^\top$. According to the definition of P, Q, and R in (2.33), we obtain from (2.56) that

$$\begin{bmatrix} \widehat{\theta} - \theta^* \\ \widehat{\mu} \end{bmatrix} = \begin{bmatrix} P & Q \\ Q^\top & R \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{n}X^\top \epsilon - \widehat{\tau} \\ 0 \end{bmatrix} + \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} P & Q \\ Q^\top & R \end{bmatrix} \begin{bmatrix} H & A \\ A^\top & 0 \end{bmatrix} \right) \cdot \begin{bmatrix} \widehat{\theta} - \theta^* \\ \widehat{\mu} \end{bmatrix} \quad (2.57)$$

The error term in (2.57) is given by

$$\text{error} = \begin{bmatrix} I - PH - QA^\top & -PA \\ -Q^\top H - RA^\top & I - Q^\top A \end{bmatrix} \cdot \begin{bmatrix} \widehat{\theta} - \theta^* \\ \widehat{\mu} \end{bmatrix} \quad (2.58)$$

According to the definition (2.33), it is straightforward to obtain $PA = 0$ and $Q^\top A = I$, and hence $\widehat{\mu}$ does not play a role in the error term as usual. Furthermore, we have

$$
\begin{aligned}
I - PH - QA^\top &= I - MH + MA\Big[A^\top MA\Big]^{-1}A^\top MH - MA\Big[A^\top MA\Big]^{-1}A^\top \\
&= \Big(I - MA\Big[A^\top MA\Big]^{-1}A^\top\Big)(I - MH) \qquad (2.59)\\
&= (I - P_M)(I - MH)
\end{aligned}
$$

Combine (2.57), (2.58), and (2.59) we obtain

$$
\widehat{\theta} - \theta^* = P\Big(\frac{1}{n}X^\top\epsilon - \lambda\widehat{\tau}\Big) + (I - P_M)(I - MH)(\widehat{\theta} - \theta^*)
$$

and hence the estimator in (2.34) satisfies

$$
\theta_{\text{est}} = \widehat{\theta} + P \cdot \lambda\widehat{\tau} = \theta^* + P \cdot \frac{1}{n}X^\top\epsilon + (I - P_M)(I - MH)(\widehat{\theta} - \theta^*) \qquad (2.60)
$$

According to Theorem 2.2 in [309], we have that $\big\|\sqrt{n}(I - MH)(\widehat{\theta} - \theta^*)\big\|_\infty = o_{\mathbb{P}}(1)$. Since P_M is a projection matrix, we have that $I - P_M$ is also a projection matrix (project to the orthogonal complement). It is therefore a non-expansion and hence the error term in (2.60) is $o_{\mathbb{P}}(1/\sqrt{n})$. This completes the proof.

□

2.9.6 Proof of Theorem 12

Proof. We start from the following lemma that quantifies the error in M for spiked covariance model.

Lemma 16. *Consider the sparse PCA model and suppose the Assumption 11 holds with a spiked covariance model* (2.43). *Assume* $s^2 \log^2 p/n = o(1)$, *the estimator* M *in* (2.44) *satisfies*

$$
\|m_j - m_j^*\|_2 = \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s\log p}{n}}\Big)
$$

Proof. We have

$$\begin{aligned}
\|m_j - m_j^*\|_2 &= \Big\|\frac{1}{\widehat{w}}(e_j - \widehat{\theta}_j \cdot \widehat{\theta}) - \frac{1}{w}(e_j - \theta_j^* \cdot \theta^*)\Big\|_2 \\
&\le \Big\|\Big(\frac{1}{\widehat{w}} - \frac{1}{w}\Big)e_j\Big\|_2 + \Big\|\frac{1}{\widehat{w}}\widehat{\theta}_j \cdot \widehat{\theta} - \frac{1}{w}\theta_j^* \cdot \theta^*\Big\|_2 \\
&\le \Big\|\Big(\frac{1}{\widehat{w}} - \frac{1}{w}\Big)e_j\Big\|_2 + \Big\|\Big(\frac{1}{\widehat{w}} - \frac{1}{w}\Big)\widehat{\theta}_j \cdot \widehat{\theta}\Big\|_2 + \Big\|\frac{1}{w}(\widehat{\theta}_j \cdot \widehat{\theta} - \theta_j^* \cdot \theta^*)\Big\|_2 \\
&\le \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s\log p}{n}}\Big) + \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s\log p}{n}}\Big) + \Big\|\frac{1}{w}\widehat{\theta}_j \cdot (\widehat{\theta} - \theta^*)\Big\|_2 + \Big\|\frac{1}{w}(\widehat{\theta}_j - \theta_j^*) \cdot \theta^*)\Big\|_2 \\
&= \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s\log p}{n}}\Big)
\end{aligned}$$

□

With Assumption 11 and Lemma 17, we can derive the asymptotic results from (2.41) and (2.42). We first focus on θ_{est} where we have

$$\begin{aligned}
\theta_{\text{est}} &= \widehat{\theta} - M^*(\widehat{\Sigma} + \widehat{\mu}I)\widehat{\theta} + (M^* - M)(\widehat{\Sigma} + \widehat{\mu}I)\widehat{\theta} \\
&= \theta^* + \widehat{\theta} - \theta^* - M^*(\widehat{\Sigma} + \widehat{\mu}I)\theta^* - M^*(\widehat{\Sigma} + \widehat{\mu}I)(\widehat{\theta} - \theta^*) + (M^* - M)(\widehat{\Sigma} + \widehat{\mu}I)\widehat{\theta} \\
&= \theta^* - M^*\widehat{\Sigma}\theta^* - \underbrace{\Big(M^*(\widehat{\Sigma} + \widehat{\mu}I) - I\Big)(\widehat{\theta} - \theta^*)}_{R_1} + \underbrace{(M^* - M)(\widehat{\Sigma} + \widehat{\mu}I)\widehat{\theta}}_{R_2}
\end{aligned} \qquad (2.61)$$

We see that in order to obtain the desired asymptotic results on θ_{est}, we need to show that both R_1 and R_2 are small. For R_1, according to (2.39) we have

$$\begin{aligned}
R_1 &= \Big(M^*(\widehat{\Sigma} + \widehat{\mu}I) - I\Big)(\widehat{\theta} - \theta^*) \\
&= \Big(M^*(\widehat{\Sigma} + \widehat{\mu}I - \Sigma^* - \mu^*I) - \theta^*{\theta^*}^\top\Big)(\widehat{\theta} - \theta^*) \\
&= M^*(\widehat{\Sigma} - \Sigma^*)(\widehat{\theta} - \theta^*) + (\widehat{\mu} - \mu^*) \cdot M^*(\widehat{\theta} - \theta^*) - \theta^*{\theta^*}^\top(\widehat{\theta} - \theta^*)
\end{aligned}$$

Focusing on the j^{th} row, we have

$$\Big|[R_1]_j\Big| \le |m_j^*(\widehat{\Sigma} - \Sigma^*)(\widehat{\theta} - \theta^*)| + |\widehat{\mu} - \mu^*| \cdot |m_j^*(\widehat{\theta} - \theta^*)| + |\theta_j^*| \cdot {\theta^*}^\top(\widehat{\theta} - \theta^*) = \mathcal{O}_{\mathbb{P}}\Big(\frac{s\log p}{n}\Big) \qquad (2.62)$$

For R_2, focusing on the j^{th} row, we have

$$\begin{aligned}
\Big|[R_2]_j\Big| &= |(m_j^* - m_j)^\top \cdot (\widehat{\Sigma} + \widehat{\mu} I)\widehat{\theta}| \\
&= \Big|(m_j^* - m_j)^\top \Big[(\Sigma^* + \mu^* I)\widehat{\theta} + (\widehat{\Sigma} - \Sigma^*)\widehat{\theta} + (\widehat{\mu} - \mu^*)\widehat{\theta}\Big]\Big| \\
&= \Big|(m_j^* - m_j)^\top \Big[(\Sigma^* + \mu^* I)(\widehat{\theta} - \theta^*) + (\widehat{\Sigma} - \Sigma^*)\widehat{\theta} + (\widehat{\mu} - \mu^*)\widehat{\theta}\Big]\Big| \\
&\leq |(m_j^* - m_j)^\top (\Sigma^* + \mu^* I)(\widehat{\theta} - \theta^*)| + |(m_j^* - m_j)^\top (\widehat{\Sigma} - \Sigma^*)\widehat{\theta}| + |(m_j^* - m_j)^\top (\widehat{\mu} - \mu^*)\widehat{\theta}| \\
&= \mathcal{O}_{\mathbb{P}}\Big(\frac{s \log p}{n}\Big) + \mathcal{O}_{\mathbb{P}}\Big(\frac{s \log p}{n}\Big) + \mathcal{O}_{\mathbb{P}}\Big(\frac{s \log p}{n}\Big) \\
&= \mathcal{O}_{\mathbb{P}}\Big(\frac{s \log p}{n}\Big)
\end{aligned} \tag{2.63}$$

Plug in (2.62) and (2.63) into (2.61) we obtain

$$\theta_{\text{est}} = \theta^* - M^* \widehat{\Sigma} \theta^* + \mathcal{O}_{\mathbb{P}}\Big(\frac{s \log p}{n}\Big) \tag{2.64}$$

We then focus on μ_{est} where we have

$$\begin{aligned}
\mu_{\text{est}} &= \widehat{\mu} - \widehat{\theta}^\top \cdot (\widehat{\Sigma} + \widehat{\mu} I)\widehat{\theta} \\
&= \widehat{\mu} - \Big[(\widehat{\theta} - \theta^*)^\top (\widehat{\Sigma} + \widehat{\mu} I)(\widehat{\theta} - \theta^*) + \theta^{*\top} (\widehat{\Sigma} + \widehat{\mu} I)\theta^* + 2(\widehat{\theta} - \theta^*)^\top (\widehat{\Sigma} + \widehat{\mu} I)\theta^*\Big] \\
&= -\theta^{*\top} \widehat{\Sigma} \theta^* - \underbrace{(\widehat{\theta} - \theta^*)^\top (\widehat{\Sigma} + \widehat{\mu} I)(\widehat{\theta} - \theta^*)}_{T_1} - \underbrace{2(\widehat{\theta} - \theta^*)^\top (\widehat{\Sigma} + \widehat{\mu} I)\theta^*}_{T_2}
\end{aligned} \tag{2.65}$$

For T_1, we have

$$\begin{aligned}
|T_1| &\leq |(\widehat{\theta} - \theta^*)^\top (\Sigma^* + \mu^* I)(\widehat{\theta} - \theta^*)| + |(\widehat{\theta} - \theta^*)^\top (\widehat{\Sigma} - \Sigma^*)(\widehat{\theta} - \theta^*)| + |\widehat{\mu} - \mu^*| \cdot \|(\widehat{\theta} - \theta^*)\|_2^2 \\
&= \mathcal{O}_{\mathbb{P}}\Big(\frac{s \log p}{n}\Big)
\end{aligned} \tag{2.66}$$

For T_2, we have

$$\begin{aligned}|T_2| &\leq |2(\widehat{\theta}-\theta^*)^\top(\Sigma^*+\mu^* I)\theta^*| + |2(\widehat{\theta}-\theta^*)^\top(\widehat{\Sigma}-\Sigma^*)\theta^*| + |\widehat{\mu}-\mu^*|\cdot|2(\widehat{\theta}-\theta^*)^\top\theta^*| \\ &= \mathcal{O}_{\mathbb{P}}\Big(\frac{s\log p}{n}\Big)\end{aligned} \tag{2.67}$$

Plug in (2.66) and (2.67) into (2.65) we obtain

$$\mu_{\text{est}} = -\theta^{*\top}\widehat{\Sigma}\theta^* + \mathcal{O}_{\mathbb{P}}\Big(\frac{s\log p}{n}\Big) \tag{2.68}$$

Since $\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^n x_i x_i^\top$ with $\mathbb{E}[\widehat{\Sigma}] = \Sigma^*$, we can then verify the terms in (2.64) and (2.68) that

$$\begin{aligned}&\mathbb{E}[M^*\cdot\widehat{\Sigma}\theta^*] = M^*\Sigma^*\theta^* = \lambda_{\max}M^*\theta^* = 0 \\ &\mathbb{E}[\theta^{*\top}\cdot\widehat{\Sigma}\theta^*] = \theta^{*\top}\Sigma^*\theta^* = \lambda_{\max}\theta^{*\top}\theta^* = \lambda_{\max} = -\mu^*\end{aligned}$$

According to (2.64) and (2.68), in order to make sure that the error terms are negligible, the sample complexity is given by $n \gg s^2\log^2 p$, which matches the result in [160] for the leading eigenvector. For the maximum eigenvalue, the result in [160] requires $n \gg s^3\log^2 p$, and our proposed method improves this sample complexity to $n \gg s^2\log^2 p$.

In sparse PCA, the objective function is not a negative log-likelihood function, so we need to calculate the asymptotic variance explicitly. As an illustrative example, we focus on the j^{th} component of θ and denote $m_j^{*\top}$ as the j^{th} row of M^* and obtain (asymptotically, ignore the error term)

$$\sqrt{n}(\theta_{\text{est}}-\theta^*)_{[j]} \approx \sqrt{n}\cdot m_j^{*\top}\widehat{\Sigma}\theta^* = \sqrt{n}\cdot m_j^{*\top}\Big[\frac{1}{n}\sum_{i=1}^n x_i x_i^\top\Big]\theta^* = n^{-1/2}\sum_{i=1}^n m_j^{*\top}x_i x_i^\top\theta^*$$

Since we have already verified that it has mean 0, the variance is given by $\mathbb{E}\big[m_j^{*\top}x_i x_i^\top\theta^*\big]^2$, and can be estimated as $\frac{1}{n}\sum_{i=1}^n\big(\widehat{m}_j^\top x_i x_i^\top\widehat{\theta}\big)^2$ in practice where $\widehat{m}_j^\top$ is the j^{th} row of M. We follow this procedure for all the components of θ, and for μ where the only difference is that its mean is not 0 and we estimate by sample variance instead of sample second moment.

More specifically, the sample covariance matrix of $\sqrt{n}(\theta_{\text{est}} - \theta^*)$ can be estimated in practice by $\widehat{\Sigma}_\theta$ with

$$\big[\widehat{\Sigma}_\theta\big]_{j,k} = \frac{1}{n}\sum_{i=1}^{n}\Big(\widehat{m}_j^\top x_i x_i^\top \widehat{\theta}\Big) \cdot \Big(\widehat{m}_k^\top x_i x_i^\top \widehat{\theta}\Big) \tag{2.69}$$

and the variance of $\sqrt{n}(\mu_{\text{est}} - \mu^*)$ is estimated by the sample variance of $\big(x_i^\top \widehat{\theta}\big)^2$:

$$\widehat{\sigma}_\mu^2 = \text{Var}_i\big(x_i^\top \widehat{\theta}\big)^2 \tag{2.70}$$

□

2.9.7 Proof of Corollary 14

Proof. The proof is similar to that of sparse PCA, except that the concentration rate of $\widehat{\Sigma} - \Sigma^*$ is different since we no longer have sub-Gaussianity. According to Lemma A.2 and Lemma A.3 in [342], by taking union bound on all the support S with $|\text{supp}(S)| = s$, with probability at least $1 - 1/(10n)$ we have

$$\|\widehat{\Sigma} - \Sigma^*\|_{2,s} = \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s\log p + \log n}{n}}\Big)$$

The rest of the proof follows from the proof of Theorem 12 where we end up with

$$\theta_{\text{est}} = \theta^* - M^*\widehat{\Sigma}\theta^* + \mathcal{O}_{\mathbb{P}}\Big(\frac{s\log p + \log n}{n}\Big) \tag{2.71}$$

The error term in (2.71) is $o_{\mathbb{P}}(1/\sqrt{n})$ by assumption. We then have

$$\sqrt{n}(\theta_{\text{est}} - \theta^*)_{[j]} \approx \sqrt{n} \cdot m_j^{*\top}\widehat{\Sigma}\theta^* = n^{-1/2}\sum_{i=1}^{n} m_j^{*\top} y_i \cdot (x_i x_i^\top - I)\theta^*$$

The sample covariance matrix of $\sqrt{n}(\theta_{\text{est}} - \theta^*)$ can be estimated in practice by $\widehat{\Sigma}_\theta$ with

$$\big[\widehat{\Sigma}_\theta\big]_{j,k} = \frac{1}{n}\sum_{i=1}^{n}\Big(\widehat{m}_j^\top y_i \cdot (x_i x_i^\top - I)\widehat{\theta}\Big) \cdot \Big(\widehat{m}_k^\top y_i \cdot (x_i x_i^\top - I)\widehat{\theta}\Big) \tag{2.72}$$

□

2.9.8 Sparse PCA with a general covariance matrix

In this section we provide the application of our method to sparse PCA problem with a general (non-spiked) covariance matrix. It turns out the only difference is the estimation of M, where we can use the following Dantzig selector to obtain the j^{th} row $m_j^\top$, and stack them to obtain M as an approximation of M^*.

$$\begin{aligned} \underset{m_j}{\text{minimize}} \quad & \|m_j\|_1 \\ \text{subject to} \quad & \Big\|e_j - m_j^\top \cdot \widehat{H} - [\widehat{\theta}\widehat{\theta}^\top]_j\Big\|_{2,2s} \le \lambda' \\ & \big|m_j^\top \cdot \widehat{\theta}\big| \le \lambda'' \end{aligned} \tag{2.73}$$

where the sparse 2-norm of a vector u with sparsity level d is defined as

$$\|u\|_{2,d} = \sup_{\|v\|_0 = d, \|v\|_2 = 1} \big|u^\top v\big|$$

The two constraints are the analogue of the properties of M^*: $I - M^* H^* = u_1 u_1^\top = \theta^* \theta^{*\top}$ as in (2.39) and $M^*\theta^* = 0$ as in (2.40). The following lemma quantifies the estimation error of this choice of M.

Lemma 17. *Suppose we obtain m_j by (2.73) with $\lambda' \asymp \lambda'' \asymp \sqrt{s \log p / n}$. We then have*

$$\|m_j - m_j^*\|_2 = \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s \log p}{n}}\Big)$$

Proof. We first show that m_j^* satisfies the constraints in (2.73). For the first constraint, we have

$$I - M^*\widehat{H} - \widehat{\theta}\widehat{\theta}^\top = M^*(H^* - \widehat{H}) + \theta^*{\theta^*}^\top - \widehat{\theta}\widehat{\theta}^\top = M^*(\Sigma^* - \widehat{\Sigma} + (\mu^* - \widehat{\mu})I) + \theta^*{\theta^*}^\top - \widehat{\theta}\widehat{\theta}^\top$$

Taking the j^{th} row we obtain

$$\begin{aligned}\Big\|e_j - {m_j^*}^\top \cdot \widehat{H} - [\widehat{\theta}\widehat{\theta}^\top]_j\Big\|_{2,2s} &\le \|{m_j^*}^\top(\Sigma^* - \widehat{\Sigma})\|_{2,2s} + |\mu^* - \widehat{\mu}| \cdot \|m_j^*\|_2 + \|\theta_j^* \cdot \theta^* - \widehat{\theta}_j \cdot \widehat{\theta}\|_2 \\ &\le \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s\log p}{n}}\Big) + \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s\log p}{n}}\Big) + \|\theta_j^*(\theta^* - \widehat{\theta})\|_2 + \|(\theta_j^* - \widehat{\theta}_j) \cdot \widehat{\theta}\|_2 \\ &= \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s\log p}{n}}\Big)\end{aligned}$$

This shows that m_j^* satisfies the first constraint. For the second constraint, we have

$$\big|{m_j^*}^\top \cdot \widehat{\theta}\big| = \big|{m_j^*}^\top \cdot (\widehat{\theta} - \theta^*)\big| = \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s\log p}{n}}\Big)$$

This shows that m_j^* satisfies the second constraint. From now on we omit the subscript j for notation simplicity. According to the optimality of m, we have $\|m\|_1 \le \|m^*\|_1$. Denote $\delta = m - m^*$ and $S = \operatorname{supp}(m^*)$ as the support of m^*, this further gives

$$\|m_S\|_1 + \|m_{S^c}\|_1 \le \|m_S^*\|_1 = \|m_S - \delta_S\|_1 \le \|m_S\|_1 + \|\delta_S\|_1$$

This gives $\|m_{S^c}\|_1 \le \|\delta_S\|_1$, which further implies $\|\delta_{S^c}\|_1 \le \|\delta_S\|_1$ since $m_{S^c}^* = 0$. This shows that δ is approximately sparse and we have $\|\delta\|_1 = \|\delta_S\|_1 + \|\delta_{S^c}\|_1 \le 2\|\delta_S\|_1$.

Next, since both m and m^* satisfy the first constraint, we have

$$\|\delta^\top \cdot \widehat{H}\|_{2,2s} \le \Big\|e_j - m_j^\top \cdot \widehat{H} - [\widehat{\theta}\widehat{\theta}^\top]_j\Big\|_{2,2s} + \Big\|e_j - {m_j^*}^\top \cdot \widehat{H} - [\widehat{\theta}\widehat{\theta}^\top]_j\Big\|_{2,2s} \le 2\lambda'$$

According to the proof of Lemma 4.9 in [27] we have

$$|\delta^\top \widehat{H}\delta| \le (\|\delta\|_2 + \|\delta\|_1/\sqrt{2s}) \cdot \|\delta^\top \cdot \widehat{H}\|_{2,2s} \le 3\|\delta\|_2 \cdot \|\delta^\top \cdot \widehat{H}\|_{2,2s} \le 6\lambda'\|\delta\|_2 \tag{2.74}$$

where we use that

$$\|\delta\|_1 \le 2\|\delta_S\|_1 \le 2\sqrt{s}\|\delta_S\|_2 \le 2\sqrt{s}\|\delta\|_2$$

On the other hand, we have

$$-\delta^\top \widehat{H}\delta = -\delta^\top H^*\delta - \delta^\top(\widehat{H} - H^*)\delta \tag{2.75}$$

For H^*, we know that the eigenvalues of H^* are given by $0, \lambda_2 - \lambda_{\max}, ..., \lambda_p - \lambda_{\max}$ with eigenvectors $u_1, ..., u_p$ where $u_1 = \theta^*$. Therefore, to lower bound $-\delta^\top H^*\delta$, we hope that δ does not span too much on the direction of θ^*. This is exactly the motivation of the second constraint. Denote $\delta = \delta^{||} + \delta^\perp$ where $\delta^{||}$ is the projection of δ onto the direction of θ^*, and $\delta^\perp$ is the projection onto the complement of θ^*. Since H^* does not span onto θ^* by definition, we have $H^*\theta^* = 0$ and hence $H^*\delta^{||} = 0$. This gives

$$-\delta^\top H^*\delta = -(\delta^{||} + \delta^\perp)^\top H^*(\delta^{||} + \delta^\perp) = -(\delta^\perp)^\top H^*\delta^\perp \ge \kappa \cdot \|\delta^\perp\|_2^2 = \kappa \cdot (\|\delta\|_2^2 - \|\delta^{||}\|^2) \tag{2.76}$$

where the inequality comes from the fact that $-H^*$ is κ-positive definite in the direction perpendicular to θ^* with minimum eigenvalue $\kappa = \lambda_{\max} - \lambda_2$ (the eigengap).

Next, we upper bound the term $\|\delta^{||}\|^2$ as

$$\begin{aligned}
\|\delta^{||}\|_2 &= \left|\delta^\top \cdot \theta^*\right| = \left|\delta^\top \cdot \widehat{\theta} + \delta^\top \cdot (\theta^* - \widehat{\theta})\right| = \left|m^\top \widehat{\theta} - m^{*\top}\widehat{\theta} + \delta^\top \cdot (\theta^* - \widehat{\theta})\right| \\
&\le \left|m^\top \widehat{\theta}\right| + \left|m^{*\top}\widehat{\theta}\right| + \|\delta\|_2 \cdot \|\theta^* - \widehat{\theta}\|_2 \\
&\le \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s\log p}{n}}\Big) + \|\delta\|_2 \cdot \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s\log p}{n}}\Big)
\end{aligned}$$

This gives

$$\|\delta^{\|}\|_2^2 \leq 2C\frac{s\log p}{n} + \|\delta\|_2^2 \cdot 2C\frac{s\log p}{n} \tag{2.77}$$

Moreover, for the term $\delta^\top(\widehat{H} - H^*)\delta$, according to Lemma 3.2.3 in [315], with high probability we have

$$\sup_{\|b\|_2=1} |b^\top(\widehat{\Sigma} - \Sigma^*)b| \leq C\max\left\{\|b\|_1\sqrt{\frac{\log p}{n}}, \|b\|_1^2\frac{\log p}{n}\right\}$$

Taking $b = \delta/\|\delta\|_2$, we have

$$|\delta^\top(\widehat{\Sigma} - \Sigma^*)\delta| \leq \|\delta\|_2^2 \cdot C\max\left\{\frac{\|\delta\|_1}{\|\delta\|_2}\sqrt{\frac{\log p}{n}}, \left(\frac{\|\delta\|_1}{\|\delta\|_2}\right)^2\frac{\log p}{n}\right\} \tag{2.78}$$

Recall that $\|\delta\|_1 \leq 2\|\delta_S\|_1$, we have

$$\|\delta\|_1 \leq 2\|\delta_S\|_1 \leq 2\sqrt{s}\|\delta_S\|_2 \leq 2\sqrt{s}\|\delta\|_2$$

Plug into (2.78) we obtain

$$|\delta^\top(\widehat{\Sigma} - \Sigma^*)\delta| \leq \|\delta\|_2^2 \cdot C\max\left\{\sqrt{\frac{s\log p}{n}}, \frac{s\log p}{n}\right\} \tag{2.79}$$

Finally, we have

$$|\delta^\top(\widehat{\mu} - \mu^*) \cdot I \cdot \delta| = \|\delta\|_2^2 \cdot |\widehat{\mu} - \mu^*| = \|\delta\|_2^2 \cdot \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{s\log p}{n}}\Big) \tag{2.80}$$

Together with (2.77), with the asymptotic regime $n \gg s^2\log^2 p$, we can guarantee that $\kappa - C\sqrt{s\log p/n} - Cs\log p/n \geq \kappa/2$. Plug in (2.76), (2.77), (2.79) and (2.80) into (2.75) we have

$$| - \delta^\top\widehat{H}\delta| \geq \frac{\kappa}{2} \cdot \|\delta\|_2^2 - C \cdot \frac{s\log p}{n}$$

Combining with (2.74) we have

$$\frac{\kappa}{2} \cdot \|\delta\|_2^2 - C \cdot \frac{s \log p}{n} \leq 6\lambda' \|\delta\|_2$$

Solving this quadratic inequality on $\|\delta\|_2$ we obtain

$$\|\delta\|_2 \leq \frac{1}{\kappa}\left(6\lambda' + \sqrt{(6\lambda')^2 + 4 \cdot \frac{\kappa}{2} \cdot C \frac{s \log p}{n}}\right) = \mathcal{O}_{\mathbb{P}}\left(\sqrt{\frac{s \log p}{n}}\right)$$

Recall that $\|\delta\|_1 \leq 2\|\delta_S\|_1$, using the norm inequality we obtain

$$\|\delta\|_1 \leq 2\|\delta_S\|_1 \leq 2\sqrt{s}\|\delta_S\|_2 \leq 2\sqrt{s}\|\delta\|_2 = \mathcal{O}_{\mathbb{P}}\left(s\sqrt{\frac{\log p}{n}}\right)$$

□

Remark 18. *The estimation problem* (2.73) *involves sparse 2-norm, and is therefore computationally intractable. In practice, we can instead use infinity norm in the first constraint in* (2.73)*.*

CHAPTER 3

CONSTRAINED HIGH DIMENSIONAL STATISTICAL INFERENCE

3.1 Introduction

Statistical estimation of high dimensional problems has been attracting more and more attention due to the abundance of such data in many emerging fields such as genetic studies, social network analysis, etc. High dimensional geometry is inherently different from low-dimensional geometry. As an example, for linear regression, in low dimensions the Ordinary Least Square (OLS) estimator allows for constructing confidence intervals and hypothesis tests for the true coefficient. In high dimensional models OLS is ill-conditioned so instead we have to solve for penalized estimators like LASSO. In low dimensions we can test for hypothesis such as $H_0 : \alpha^* = 0$ by partial likelihood function while in high dimensions this also fails, due to the large amount of nuisance parameters.

In this chapter we consider a hypothesis testing problem in a high dimensional model under constrained parameter space. For many problems, before analyzing data and fitting models we might already know some constraints on the parameters. This can also be viewed as prior information on the parameters. For example in isotonic regression [34, 335, 86] we have a constraint that the variables are non-decreasing; in non-negative least square problem [284] we have a constraint that the coefficients are non-negative. in real-world reinforcement learning applications, we need to take into consideration the safety of the agent [31, 326, 355]. Also, in Gaussian process, it is sometimes assumed that the parameters satisfy some linear inequality constraints [211].

With this additional information the statistical inference and hypothesis testing for the parameters may be different. For example, consider a simple model: $X \sim N(\mu, 1)$. In general if we want to test whether μ is 0 or not, i.e. test for $H_0 : \mu = 0$ versus $H_A : \mu \neq 0$, we will reject H_0 if the absolute value of the mean $|\bar{x}|$ is relatively large. However, if we have the constraint that $\mu \geq 0$, then we are testing $H_0 : \mu = 0$ versus $H_A : \mu > 0$, and we reject H_0 only when $\bar{x}$ is relatively large.

When we have constraints on parameters, a natural question we want to answer is whether the parameter lies on the boundary or is away from the boundary, since these two cases are usually very different. For example for nonnegativity constraint, we want to know whether the parameter is exactly zero or strictly positive; for monotonic constraint we want to know whether the two variables are equal or one is strictly greater than the other.

In this chapter we perform statistical inference (hypothesis testing) for low dimensional parameters in a high dimensional model under cone constraint. Denote the parameter $\boldsymbol{\beta} = (\boldsymbol{\alpha}, \boldsymbol{\theta})$, where $\boldsymbol{\alpha}$ is the low dimensional parameter of interest and $\boldsymbol{\theta}$ denotes nuisance parameters. Denote the constraint set C as a closed and convex cone, and let $M \in C$ be a linear space in C. In most of the cases C is a polyhedron and the linear space M denotes the (subset of) the boundary set of C. In this chapter we want to test

$$H_0 : \boldsymbol{\alpha} \in M \text{ versus } H_A : \boldsymbol{\alpha} \in C \backslash M,$$

where we have the constraint $\boldsymbol{\alpha} \in C$. We develop an algorithm for this constrained testing problem in high dimensional models. Following our procedure we show that the hypothesis test method we propose has asymptotic designed Type I error, and it has much greater power than when the constraints are ignored.

3.1.1 Related Work

High-dimensional inference without constraint. There is a vast literature on performing statistical inference for high dimensional models and here we provide a brief overview. Early work [179] shows that the limiting distribution of LASSO estimator is not normal even in low dimensions. More recently, several approaches have been proposed to obtain asymptotic distribution on low dimensional parameters in high dimensional linear model, mostly by approximating the inverse of the Gram matrix. [362] gives confidence intervals for low dimensional parameters in a high dimensional linear model using low dimensional projection estimator (LDPE). [166] provides asymptotic confidence interval of LASSO estimator for high dimensional linear regression by introducing the debiasing method. [309] further extends their work to a more general setting, including Generalized Linear Model and other nonlinear models. [237] deals with general model on Hessian matrix with Dantzig type estimator. Related works also include [90, 364] for simultaneous inference, [29] for double selection method, [258, 345, 351] for graphical model, [299, 336, 193] for post selective inference, [200, 63, 65] for for synthetic control, [286] for noisy labels, etc.

Low-dimensional constrained inference. The literature on constrained testing dates back to [79], where the authors prove the asymptotic distribution of the likelihood ratio (LR) test statistic for constrained testing to be weighted Chi-square. [243] further considers testing with unknown covariance matrix, and gives sharp upper and lower bounds for the weights. [122] introduces the test statistics for likelihood ratio test, Wald test, and Kuhn-Tucker test with inequality constraint in linear model, and proves the equivalence of these three tests. [176] proposes one-sided t-test when the coefficients' signs are known. [262] introduces a modified Lagrange multiplier test for testing one-sided problem. [181] proposes Wald test for jointly testing equality and inequality constraints on the parameters. [328] develops asymptotically equivalent tests under linear inequality restrictions for linear models. [177] introduces a locally most mean powerful (LMMP) test. [17] introduces directed tests, which is optimal in terms of power. [40] introduces multiple-endpoint testing in clinical trials. [133] provides Order-Restricted Score Tests for generalized linear and nonlinear mixed models. [18] proposes test when nuisance parameters appear under the alternative hypothesis, but not under the null. [244] gives improved LRT and UIT test. More recently, [215] has discussed halfline test for inequality constraints. [292] gives conservative likelihood ratio test using data-dependent degree of freedom. [370] gives Wald test under inequality constraint in linear model with spherically symmetric disturbance. [216] proposes an extended MaxT test and gets the power improvement. However, all these existing results are for low dimensional models.

In terms of statistical inference, our work is most related to [237], where the authors establish inference for high dimensional models using decorrelation method. We will review this method in Section 3.2. For constrained testing, our work is most related to [270] where the authors introduce and discuss Chi-bar-squared statistic, and [279] and [224] which form the one sided test to test whether a parameter is zero or strictly positive. Recent works [165, 324] consider hypothesis testing on whether the parameters lie in some convex cone. This is still different from our setting where we know the parameters lie in the convex cone and the goal is to test whether they lie on the boundary of the cone.

3.1.2 Organization of the chapter

In Section 3.2 we give the detailed procedure for our algorithm. Section 3.3 gives assumptions under which our method is valid, and states our main theorem. Sections 3.4 and 3.5 present experimental results on synthetic datasets and real world datasets, respectively. We conclude in Section 3.6.

3.2 Algorithm

In this section we describe our main algorithm. Consider a high dimensional statistical model with parameters $\boldsymbol{\beta} \in \mathbb{R}^p$ and the partition $\boldsymbol{\beta} = (\boldsymbol{\alpha}, \boldsymbol{\theta})$, where $\boldsymbol{\alpha}$ is d dimensional parameter of interest, and $\boldsymbol{\theta}$ is a $p - d$ dimensional nuisance parameter with $d \ll p$. We write $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_d)$, and the true parameter as $\boldsymbol{\beta}^* = (\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*)$ with $\|\boldsymbol{\beta}^*\|_0 = s$. Moreover, we have the constraint $\boldsymbol{\alpha}^* \in C$ where C is a closed and convex cone. Let $M \in C$ be a linear space in C. In most of the cases C is an polyhedron and the linear space M denotes the (subset of) the boundary set of C. The hypothesis we want to test is

$$H_0 : \boldsymbol{\alpha}^* \in M \text{ versus } H_A : \boldsymbol{\alpha}^* \in C \backslash M, \tag{3.1}$$

i.e. we want to test whether $\boldsymbol{\alpha}^*$ lies on the boundary of C, or is a strict interior point of C in at least one direction. For example, with nonnegativity constraint we have $C = \mathbb{R}_+^d = \{\boldsymbol{\alpha} : \boldsymbol{\alpha} \geq \mathbf{0}\}$ and $M = \{\boldsymbol{\alpha} : \boldsymbol{\alpha} = \mathbf{0}\}$. The hypothesis we want to test is

$$H_0 : \boldsymbol{\alpha}^* = \mathbf{0} \quad \text{versus} \quad H_A : \exists j \in \{1, ..., d\} \text{ s.t. } \alpha_j^* > 0.$$

Another example is monotonic constraint where we have $C = \{\boldsymbol{\alpha} : \alpha_1 \leq \alpha_2 \leq ... \leq \alpha_d\}$ and $M = \{\boldsymbol{\alpha} : \alpha_1 = \alpha_2 = ... = \alpha_d\}$. The hypothesis we want to test is

$$H_0 : \alpha_1^* = \alpha_2^* = ... = \alpha_d^* \quad \text{versus} \quad H_A : \exists j \in \{1, ..., d-1\} \text{ s.t. } \alpha_j^* < \alpha_{j+1}^*.$$

Suppose we have n independent trials where we allow for $n < p$. Denote the sample negative log likelihood function as

$$\ell(\boldsymbol{\beta}) = -\frac{1}{n} \sum_{i=1}^{n} \log \mathcal{L}_i(\boldsymbol{\beta}), \tag{3.2}$$

where $\mathcal{L}_i(\boldsymbol{\beta})$ is the likelihood function for one trial i. In low dimensions we can estimate the parameter $\boldsymbol{\beta}$ by maximum likelihood estimation (MLE). However in high dimensions, MLE may not work. Instead we use the penalized estimator

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \Big\{ \ell(\boldsymbol{\beta}) + P_\lambda(\boldsymbol{\beta}) \Big\}, \tag{3.3}$$

where P_λ is some penalty function with tuning parameter λ. Note that this estimation can be performed with or without the cone constraint $\boldsymbol{\alpha} \in C$. In Section 3.3 we will see that all we need is the consistency of this estimator.

Let $\nabla \ell(\boldsymbol{\beta}) = \nabla \ell(\boldsymbol{\alpha}, \boldsymbol{\theta})$ be the gradient of the negative log likelihood function and $\nabla \ell_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}, \boldsymbol{\theta})$, $\nabla \ell_{\boldsymbol{\theta}}(\boldsymbol{\alpha}, \boldsymbol{\theta})$ be the corresponding partitions. Similarly let $\nabla^2 \ell(\boldsymbol{\beta})$ be the sample Hessian matrix, and let $\nabla^2_{\boldsymbol{\alpha\alpha}} \ell(\boldsymbol{\beta})$, $\nabla^2_{\boldsymbol{\alpha\theta}} \ell(\boldsymbol{\beta})$, $\nabla^2_{\boldsymbol{\theta\alpha}} \ell(\boldsymbol{\beta})$ and $\nabla^2_{\boldsymbol{\theta\theta}} \ell(\boldsymbol{\beta})$ be the corresponding partitions. Let $H(\boldsymbol{\beta}) = \mathbb{E}(\nabla^2 \ell(\boldsymbol{\beta}))$ be the population Fisher information matrix. Denote $H^* = H(\boldsymbol{\beta}^*)$ and $H^*_{\boldsymbol{\alpha\alpha}}$, $H^*_{\boldsymbol{\alpha\theta}}$, $H^*_{\boldsymbol{\theta\alpha}}$, $H^*_{\boldsymbol{\theta\theta}}$ as the corresponding partitions for H^*.

The difficulty of the problem comes from two aspects: the problem is high dimensional, and that we have the constraint on $\boldsymbol{\alpha}$. We first deal with the difficulty from high dimensions. It is well known that in low dimensions we can test for $H_0 : \boldsymbol{\alpha}^* = 0$ based on the partial score function

$$S(\boldsymbol{\alpha}) = \nabla_{\boldsymbol{\alpha}}(\boldsymbol{\alpha}, \widehat{\boldsymbol{\theta}}(\boldsymbol{\alpha})),$$

where $\widehat{\boldsymbol{\theta}}(\boldsymbol{\alpha}) = \text{argmin}_{\boldsymbol{\theta}} \ell(\boldsymbol{\alpha}, \boldsymbol{\theta})$ is the partial maximum likelihood estimator. Under the null hypothesis we have

$$\sqrt{n} S(\mathbf{0}) \xrightarrow{d} N(\mathbf{0}, H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}),$$

where $H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}} = H^*_{\boldsymbol{\alpha\alpha}} - H^*_{\boldsymbol{\alpha\theta}} {H^*_{\boldsymbol{\theta\theta}}}^{-1} H^*_{\boldsymbol{\theta\alpha}}$ is the partial information matrix. We then reject the null when $S(\mathbf{0})$ is relatively large. However, in high dimensions this method does not work. To overcome this issue, we follow the decorrelation procedure introduced in [102, 237] as described in Step 1 in Algorithm 2.

Remark 19. *In Step 1.2, we want to get a linear combination of $\nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\alpha^*}, \boldsymbol{\theta}^*)$ to best approximate $\nabla_{\alpha} \ell(\boldsymbol{\alpha^*}, \boldsymbol{\theta}^*)$. The population version of this vector should be*

$$\boldsymbol{W}^* = \underset{\boldsymbol{W}}{\mathit{argmin}}\, \mathbb{E}\Big\{\nabla_{\alpha} \ell(\boldsymbol{\alpha^*}, \boldsymbol{\theta}^*) - \boldsymbol{W}^\top \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\alpha^*}, \boldsymbol{\theta}^*)\Big\}^2 = H^{*-1}_{\boldsymbol{\theta\theta}} H^*_{\boldsymbol{\theta\alpha}}.$$

However, in high dimensions, we cannot directly estimate $\boldsymbol{W}^*$ by the corresponding sample version since the problem is ill-conditioned. So instead we estimate $\boldsymbol{W}^*$ by the Dantzig selector $\widehat{\boldsymbol{W}}$.

Remark 20. *In Step 1.3 we get decorrelated score function which is approximately orthogonal*

Algorithm 2 Two-step procedure for statistical inference with cone constraint

Step 1

1.1 Get penalized estimator $\widehat{\boldsymbol{\beta}} = (\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\theta}})$ using (3.3) for some tuning parameter λ.

1.2 For each $j = 1, ..., d$, estimate $\widehat{\boldsymbol{w}}_j$ by the following Dantzig selector

$$\widehat{\boldsymbol{w}}_j = \text{argmin}_{\boldsymbol{w}} \|\boldsymbol{w}\|_1 \ \text{ s.t. } \ \big\|\nabla^2_{\alpha_j \boldsymbol{\theta}} \ell(\widehat{\boldsymbol{\beta}}) - \boldsymbol{w}^\top \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \ell(\widehat{\boldsymbol{\beta}})\big\|_\infty \leq \lambda',$$

where λ' is a hyper-parameter which we describe how to choose later. Combine them to get matrix $\widehat{\boldsymbol{W}}$, i.e., $\widehat{\boldsymbol{W}} = (\widehat{\boldsymbol{w}}_1, ..., \widehat{\boldsymbol{w}}_d)$.

1.3 Define the decorrelated score function:

$$\widehat{\boldsymbol{U}}(\boldsymbol{\alpha}) = \nabla_{\boldsymbol{\alpha}} \ell(\boldsymbol{\alpha}, \widehat{\boldsymbol{\theta}}) - \widehat{\boldsymbol{W}}^\top \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\alpha}, \widehat{\boldsymbol{\theta}}).$$

1.4 Define the decorrelated estimator:

$$\widetilde{\boldsymbol{\alpha}} = \widehat{\boldsymbol{\alpha}} - \Big(\frac{\partial \widehat{\boldsymbol{U}}(\widehat{\boldsymbol{\alpha}})}{\partial \boldsymbol{\alpha}}\Big)^{-1} \cdot \widehat{\boldsymbol{U}}(\widehat{\boldsymbol{\alpha}}). \tag{3.4}$$

1.5 Define the decorrelated likelihood function:

$$\ell_{\text{de}}(\boldsymbol{\alpha}) = \ell\big(\boldsymbol{\alpha}, \widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{W}}(\boldsymbol{\alpha} - \widehat{\boldsymbol{\alpha}})\big).$$

Step 2

2.1 Get one-sided Wald test statistic

$$T_w = \inf_{\boldsymbol{b} \in M} \Big\{(\widetilde{\boldsymbol{\alpha}} - \boldsymbol{b})^\top \widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}} (\widetilde{\boldsymbol{\alpha}} - \boldsymbol{b})\Big\} - \inf_{\boldsymbol{b} \in C} \Big\{(\widetilde{\boldsymbol{\alpha}} - \boldsymbol{b})^\top \widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}} (\widetilde{\boldsymbol{\alpha}} - \boldsymbol{b})\Big\}.$$

2.2 Get one-sided Likelihood ratio test statistic

$$T_L = 2n\Big(\inf_{\boldsymbol{b} \in M} \ell_{\text{de}}(\boldsymbol{b}) - \inf_{\boldsymbol{b} \in C} \ell_{\text{de}}(\boldsymbol{b})\Big).$$

2.3 Get one-sided Score test statistic

$$T_s = \Big(\widehat{\boldsymbol{U}}(\boldsymbol{b}_M) - \widehat{\boldsymbol{U}}(\boldsymbol{b}_C)\Big)^\top \widehat{H}^{-1}_{\boldsymbol{\alpha}|\boldsymbol{\theta}} \Big(\widehat{\boldsymbol{U}}(\boldsymbol{b}_M) - \widehat{\boldsymbol{U}}(\boldsymbol{b}_C)\Big),$$

where

$$\boldsymbol{b}_M = \arg \inf_{\boldsymbol{b} \in M} \ell_{\text{de}}(\boldsymbol{b}), \ \text{ and } \ \boldsymbol{b}_C = \arg \inf_{\boldsymbol{b} \in C} \ell_{\text{de}}(\boldsymbol{b}).$$

to any component of the nuisance score function $\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\alpha}^*, \boldsymbol{\theta}^*)$. This is approximately an unbiased estimating equation for $\boldsymbol{\alpha}$ so the root of this equation should give us an approximately unbiased estimator for $\boldsymbol{\alpha}^*$. Since searching for the root may be computational intensive, we use one Newton step, as stated in (3.4).

With the decorrelated score function, the decorrelated estimator, and the decorrelated likelihood function, under mild conditions we will specify in Section 3.3, we have the following asymptotic distributions [237]:

$$\sqrt{n}\widehat{\boldsymbol{U}}(\boldsymbol{\alpha}^*)\to N(\mathbf{0}, H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}),$$
$$\sqrt{n}(\widetilde{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*)\to N(\mathbf{0}, H^{*-1}_{\boldsymbol{\alpha}|\boldsymbol{\theta}}),$$
$$2n\Big(\ell_{\text{de}}(\boldsymbol{\alpha}^*) - \ell_{\text{de}}(\widetilde{\boldsymbol{\alpha}})\Big)\to\chi^2_d,$$

where $H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}} = H^*_{\boldsymbol{\alpha}\boldsymbol{\alpha}} - H^*_{\boldsymbol{\alpha}\boldsymbol{\theta}}H^{*-1}_{\boldsymbol{\theta}\boldsymbol{\theta}}H^*_{\boldsymbol{\theta}\boldsymbol{\alpha}}$, and in practice it can be estimated by

$$\widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}} = \nabla^2_{\boldsymbol{\alpha}\boldsymbol{\alpha}}\ell(\widehat{\boldsymbol{\beta}}) - \widehat{\boldsymbol{W}}^\top\nabla^2_{\boldsymbol{\theta}\boldsymbol{\alpha}}\ell(\widehat{\boldsymbol{\beta}}).$$

We then deal with the second difficulty: cone constraint. Since we already get asymptotic normality, we follow the procedure in [270] to construct the Score, Wald and likelihood ratio test statistics, as described in Step 2 in Algorithm 2.

This two-step procedure gives us the final test statistics T_s, T_w and T_L. In the next section we will show that under null hypothesis, all of them converge weakly to the weighted Chi-square distribution, and from which we can construct valid hypothesis test with asymptotic designed Type I error.

3.3 Theoretical result

In this section, we outline the main theoretical properties of our method. We start by providing high-level conditions in Section 3.3.1, and state our main theorem in Section 3.3.2

that the null distribution is a weighted Chi-square distribution. In Section 3.3.3 we describe the way to calculate the weights. We analyze the power of our method in Section 3.3.4 and the proof of the main theorem is given in Section 3.3.5.

3.3.1 Assumptions

In this section we provide high-level assumptions that allow us to establish properties of each step in our procedure.

Sparsity Condition: Both $\boldsymbol{\beta}^*$ and $\boldsymbol{w}^*$ are sparse: $\|\boldsymbol{\beta}^*\|_0 = \|\boldsymbol{w}^*\|_0 = s$. (We use a single s for notational simplicity, but this is not required for our method to work).

Score Condition: The expected value of the score function at true $\boldsymbol{\beta}^*$ is 0:

$$\mathbb{E}\Big(\nabla \ell(\boldsymbol{\beta}^*)\Big) = 0.$$

Sparse Eigenvalue Condition: We have $\boldsymbol{v}^\top H^* \boldsymbol{v} \geq c_{\min}\|\boldsymbol{v}\|_2^2$ and $\boldsymbol{v}^\top \nabla^2 \ell(\widehat{\boldsymbol{\beta}})\boldsymbol{v} \geq c_{\min}\|\boldsymbol{v}\|_2^2$ for any $\boldsymbol{v}$ with $\|\boldsymbol{v}\|_0 = \mathcal{O}(s)$. Also both $\nabla \ell(\boldsymbol{\beta}^*)$, $\nabla^2 \ell(\boldsymbol{\beta}^*)$, and H^* are bounded element-wise, i.e., the maximum element is $\mathcal{O}(1)$ and each element has absolute value bounded by some constant a.

Denote $\|A\|_\infty$ as the maximum absolute value of elements in A, i.e., $\|A\|_\infty = \max_{j,k} |A_{jk}|$. By saying the maximum element of H^* is $\mathcal{O}(1)$, we are assuming $\|H^*\|_\infty = \mathcal{O}(1)$.

Estimation Accuracy Condition: The penalized estimator $\widehat{\boldsymbol{\beta}}$ in (3.3) is a consistent estimator for the true $\boldsymbol{\beta}^*$:

$$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_1 = \mathcal{O}(\lambda s) \text{ and } \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 = \mathcal{O}(\lambda\sqrt{s}),$$

where λ is the hyper-parameter in the penalty P_λ.

Smooth Hessian Condition: The Hessian matrix $\nabla^2 \ell(\boldsymbol{\beta})$ is Lipschitz continuous:

$$\|\nabla^2 \ell(\boldsymbol{\beta}_1) - \nabla^2 \ell(\boldsymbol{\beta}_2)\|_\infty \leq L \cdot \|\boldsymbol{\beta}_1 - \boldsymbol{\beta}_2\|_1,$$

for some constant L.

Remark 21. *The score condition holds for most of the log likelihood functions. In fact, let f be the likelihood function and θ be the parameter, then under certain regularity conditions [253], we have*

$$\mathbb{E}\frac{d}{d\theta}\log f = \mathbb{E}\frac{df}{d\theta} \cdot \frac{1}{f} = \int \frac{df}{d\theta} \cdot \frac{1}{f} \cdot f\, dx = \int \frac{df}{d\theta}\, dx = \frac{d}{d\theta}\int f\, dx = \frac{d}{d\theta}1 = 0.$$

Remark 22. *The sparse eigenvalue (SE) condition can be replaced by restricted eigenvalue (RE) condition: let $\mathcal{S} = supp(\boldsymbol{\beta^*}) \cup supp(\boldsymbol{w^*})$, RE condition requires $\boldsymbol{v}^\top H^* \boldsymbol{v} \geq c_{\min}\|\boldsymbol{v}\|_2^2$ and $\boldsymbol{v}^\top \nabla^2 \ell(\widehat{\boldsymbol{\beta}})\boldsymbol{v} \geq c_{\min}\|\boldsymbol{v}\|_2^2$ for any $\boldsymbol{v}$ in the cone $\mathcal{C}(\mathcal{S}) = \{\boldsymbol{v} : \|\boldsymbol{v}_{\mathcal{S}^c}\| \leq c_0\|\boldsymbol{v}_{\mathcal{S}}\|\}$ for some $c_{\min}, c_0 > 0$. Both sparse eigenvalue condition and restricted eigenvalue condition are common in high dimensional statistical estimation literature, and are known to hold for a large number of models. See Remark 37 in the supplementary material for the proof.*

Remark 23. *The estimation condition is also common for penalized estimators. For example, [232] shows that, if the sample loss function $\mathcal{L}$ (e.g. negative log likelihood function $\ell(\boldsymbol{\beta})$ here) is convex, differentiable, and satisfies Restricted Strong Convexity:*

$$\mathcal{L}(\boldsymbol{\beta}^* + \Delta) - \mathcal{L}(\boldsymbol{\beta}^*) - \langle \nabla\mathcal{L}(\boldsymbol{\beta}^*), \Delta\rangle \geq \kappa\|\Delta\|^2$$

for certain Δ, then for P_λ being L_1 penalty, with $\lambda \geq 2\|\nabla\mathcal{L}(\boldsymbol{\beta}^*)\|_\infty$ we have

$$\|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_1 = \mathcal{O}(\lambda s) \text{ and } \|\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 = \mathcal{O}(\lambda\sqrt{s}).$$

Remark 24. *The smooth Hessian condition is to make sure the Hessian matrix is well-*

behaved locally, hence to make sure the Dantzig selector $\widehat{\boldsymbol{w}}$ is consistent. This condition is also known to hold for general models.

3.3.2 Main theorem

Before we proceed with our main theorem, we first introduce the following Lemma 25 which shows the asymptotic distribution of the decorrelated score function and decorrelated estimator constructed in Step 1 of Algorithm 2. It is in the same spirit as and corresponds to Theorem 4.4 and 4.7 in [102]. All the other related lemmas and proofs are provided in the supplementary material. For ease of presentation, in the following Lemma 25 we focus on the case where α is a scalar. It is straightforward to generalize to the vector case.

Lemma 25. *Suppose all the conditions in Section 3.3.1 are satisfied. Let $\lambda = \mathcal{O}(\sqrt{\log p/n})$ in Step 1.1, $\lambda' = \mathcal{O}(s^2\sqrt{\log p/n})$ in Step 1.2, and $s^6 \log^2 p/n = o(1)$, we have*

$$\sqrt{n}\widehat{U}(\alpha^*) \rightarrow N(0, H^*_{\alpha|\boldsymbol{\theta}}), \tag{3.5}$$

$$\sqrt{n}(\widetilde{\alpha} - \alpha^*) \rightarrow N(0, H^{*-1}_{\alpha|\boldsymbol{\theta}}), \tag{3.6}$$

$$\left| H^*_{\alpha|\boldsymbol{\theta}} - \widehat{H}_{\alpha|\boldsymbol{\theta}} \right| = o_{\mathbb{P}}(1) \tag{3.7}$$

where $H^*_{\alpha|\boldsymbol{\theta}} = H^*_{\alpha\alpha} - H^*_{\alpha\boldsymbol{\theta}} H^{*-1}_{\boldsymbol{\theta}\boldsymbol{\theta}} H^*_{\boldsymbol{\theta}\alpha}$ and is estimated by the sample version

$$\widehat{H}_{\alpha|\boldsymbol{\theta}} = \nabla^2_{\alpha\alpha}\ell(\widehat{\boldsymbol{\beta}}) - \widehat{\boldsymbol{w}}^\top \nabla^2_{\boldsymbol{\theta}\alpha}\ell(\widehat{\boldsymbol{\beta}}). \tag{3.8}$$

Proof. The outline of the proof follows from [102]. We start from the proof of (3.5) for $\widehat{U}(\alpha^*)$

where by mean value theorem we have:

$$
\begin{aligned}
\widehat{U}(\alpha^*) &= \nabla_\alpha \ell(\alpha^*, \widehat{\boldsymbol{\theta}}) - \widehat{\boldsymbol{w}}^\top \nabla_{\boldsymbol{\theta}} \ell(\alpha^*, \widehat{\boldsymbol{\theta}}) \\
&= \nabla_\alpha \ell(\alpha^*, \boldsymbol{\theta}^*) + \nabla^2_{\alpha\boldsymbol{\theta}} \ell(\alpha^*, \bar{\boldsymbol{\theta}})(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*) - [\widehat{\boldsymbol{w}}^\top \nabla_{\boldsymbol{\theta}} \ell(\alpha^*, \boldsymbol{\theta}^*) + \widehat{\boldsymbol{w}}^\top \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \ell(\alpha^*, \widetilde{\boldsymbol{\theta}})(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)] \\
&= \underbrace{\left[\nabla_\alpha \ell(\alpha^*, \boldsymbol{\theta}^*) - \boldsymbol{w}^{*T} \nabla_{\boldsymbol{\theta}} \ell(\alpha^*, \boldsymbol{\theta}^*)\right]}_{E_1} + \underbrace{\left[(\boldsymbol{w}^* - \widehat{\boldsymbol{w}})^\top \nabla_{\boldsymbol{\theta}} \ell(\alpha^*, \boldsymbol{\theta}^*)\right]}_{E_2} \\
&\qquad\qquad + \underbrace{\left[\nabla^2_{\alpha\boldsymbol{\theta}} \ell(\alpha^*, \widetilde{\boldsymbol{\theta}}) - \widehat{\boldsymbol{w}}^\top \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \ell(\alpha^*, \bar{\boldsymbol{\theta}})\right](\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)}_{E_3} \\
&= E_1 + E_2 + E_3,
\end{aligned}
$$

where $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}^* + \bar{u}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*), \widetilde{\boldsymbol{\theta}} = \boldsymbol{\theta}^* + \widetilde{u}(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)$ for some $\bar{u}, \widetilde{u} \in [0, 1]$. We consider the three terms separately. For E_1, by taking $\boldsymbol{v} = (1; -\boldsymbol{w}^*)$ in Lemma 31, under the null hypothesis we have

$$\sqrt{n} E_1 \to N(0, H^*_{\alpha|\boldsymbol{\theta}}).$$

For E_2, according to Hölder's inequality, Lemma 32, and Lemma 36 we have

$$|E_2| \le \|\widehat{\boldsymbol{w}} - \boldsymbol{w}^*\|_1 \cdot \|\nabla_{\boldsymbol{\theta}} \ell(0, \boldsymbol{\theta}^*)\|_\infty = \mathcal{O}_{\mathbb{P}}(\lambda' s \sqrt{\log p/n}) = \mathcal{O}_{\mathbb{P}}(s^3 \log p/n).$$

For E_3 we have

$$
\begin{aligned}
|E_3| &\le \underbrace{\left|(\widehat{\boldsymbol{w}} - \boldsymbol{w}^*)^\top \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \ell(\alpha^*, \bar{\boldsymbol{\theta}})(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)\right|}_{R_1} + \left|[\nabla^2_{\alpha\boldsymbol{\theta}} \ell(\alpha^*, \widetilde{\boldsymbol{\theta}}) - {\boldsymbol{w}^*}^\top \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \ell(\alpha^*, \bar{\boldsymbol{\theta}})](\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)\right| \\
&\le R_1 + \underbrace{\left|[\nabla^2_{\alpha\boldsymbol{\theta}} \ell(\alpha^*, \widetilde{\boldsymbol{\theta}}) - H^*_{\alpha\boldsymbol{\theta}}](\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)\right|}_{R_2} + \left|[H^*_{\alpha\boldsymbol{\theta}} - {\boldsymbol{w}^*}^\top \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \ell(\alpha^*, \bar{\boldsymbol{\theta}})](\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)\right| \\
&\le R_1 + R_2 + \underbrace{\left|{\boldsymbol{w}^*}^\top [H^*_{\boldsymbol{\theta}\boldsymbol{\theta}} - \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \ell(\alpha^*, \bar{\boldsymbol{\theta}})](\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*)\right|}_{R_3} \\
&\le R_1 + R_2 + R_3.
\end{aligned}
$$

Considering the three terms R_1, R_2 and R_3 separately, according to Lemma 34 and Lemma

36 we have

$$
\begin{aligned}
R_1 \leq \|\widehat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2 \|\nabla_{\boldsymbol{\theta}\boldsymbol{\theta}} \ell(\alpha^*, \bar{\boldsymbol{\theta}})\|_2 \|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_2 \leq C\sqrt{s}\lambda' c\sqrt{s \log p/n} = \mathcal{O}_{\mathbb{P}}(s^3 \log p/n), \\
R_2 \leq \|\nabla^2_{\alpha\boldsymbol{\theta}} \ell(\alpha^*, \widetilde{\boldsymbol{\theta}}) - H^*_{\alpha\boldsymbol{\theta}}\|_\infty \cdot \|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_1 = \mathcal{O}_{\mathbb{P}}(s^2 \log p/n), \\
R_3 \leq \|\boldsymbol{w}^*\|_1 \|H^*_{\boldsymbol{\theta}\boldsymbol{\theta}} - \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \ell(\alpha^*, \bar{\boldsymbol{\theta}})\|_\infty \|\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}^*\|_1 = \mathcal{O}_{\mathbb{P}}(s^3 \log p/n).
\end{aligned}
$$

Combining all these terms we show that (3.5) holds. We then turn to the proof of (3.7) that $\widehat{H}_{\alpha|\boldsymbol{\theta}}$ is an consistent estimator. By definition we have

$$
\begin{aligned}
|\widehat{H}_{\alpha|\boldsymbol{\theta}} - H^*_{\alpha|\boldsymbol{\theta}}| &\leq \underbrace{|H^*_{\alpha\alpha} - \nabla^2_{\alpha\alpha} \ell(\widehat{\alpha}, \widehat{\boldsymbol{\theta}})|}_{T_1} + |H^*_{\alpha\boldsymbol{\theta}} H^{*-1}_{\boldsymbol{\theta}\boldsymbol{\theta}} H^*_{\boldsymbol{\theta}\alpha} - \widehat{\boldsymbol{w}}^\top \nabla^2_{\boldsymbol{\theta}\alpha} \ell(\widehat{\alpha}, \widehat{\boldsymbol{\theta}})| \\
&\leq T_1 + \underbrace{|(\boldsymbol{w}^* - \widehat{\boldsymbol{w}})^\top H^*_{\boldsymbol{\theta}\alpha}|}_{T_2} + \underbrace{|\widehat{\boldsymbol{w}}^\top (H^*_{\boldsymbol{\theta}\alpha} - \nabla^2_{\boldsymbol{\theta}\alpha} \ell(\widehat{\alpha}, \widehat{\boldsymbol{\theta}}))|}_{T_3} \\
&\leq T_1 + T_2 + T_3.
\end{aligned}
$$

Considering the terms T_1, T_2 and T_3 separately, according to Lemma 34 and Lemma 36 we have

$$
\begin{aligned}
T_1 = \mathcal{O}_{\mathbb{P}}(s\sqrt{\log p/n}), \\
T_2 \leq \|\boldsymbol{w}^* - \widehat{\boldsymbol{w}}\|_1 \cdot \|H^*_{\boldsymbol{\theta}\alpha}\|_\infty = \mathcal{O}_{\mathbb{P}}(s^3\sqrt{\log p/n}), \\
T_3 \leq \|\widehat{\boldsymbol{w}}\|_1 \cdot \|H^*_{\boldsymbol{\theta}\alpha} - \nabla^2_{\boldsymbol{\theta}\alpha} \ell(\widehat{\alpha}, \widehat{\theta})\|_\infty = \mathcal{O}_{\mathbb{P}}(s^2\sqrt{\log p/n}).
\end{aligned}
$$

Combining the three terms we show that (3.7) holds. Finally we prove the result (3.6) for

$\widetilde{\alpha}$. By construction we have

$$\begin{aligned}\widetilde{\alpha} &= \widehat{\alpha} - \Big(\frac{\partial \widehat{U}(\widehat{\alpha})}{\partial \alpha}\Big)^{-1} \cdot \widehat{U}(\widehat{\alpha}) = \widehat{\alpha} - H^{*-1}_{\alpha|\boldsymbol{\theta}} \widehat{U}(\widehat{\alpha}) + \underbrace{\widehat{U}(\widehat{\alpha})\Big[H^{*-1}_{\alpha|\boldsymbol{\theta}} - \Big(\frac{\partial \widehat{U}(\widehat{\alpha})}{\partial \alpha}\Big)^{-1}\Big]}_{S_1} \\ &= \widehat{\alpha} - H^{*-1}_{\alpha|\boldsymbol{\theta}}\Big[\widehat{U}(\alpha^*) + (\widehat{\alpha} - \alpha^*) \cdot \frac{\partial \widehat{U}(\breve{\alpha})}{\partial \alpha}\Big] + S_1 \\ &= \widehat{\alpha} - H^{*-1}_{\alpha|\boldsymbol{\theta}} \widehat{U}(\alpha^*) - (\widehat{\alpha} - \alpha^*) H^{*-1}_{\alpha|\boldsymbol{\theta}} \cdot H^{*}_{\alpha|\boldsymbol{\theta}} + \underbrace{(\widehat{\alpha} - \alpha^*) H^{*-1}_{\alpha|\boldsymbol{\theta}}\Big[H^{*}_{\alpha|\boldsymbol{\theta}} - \Big(\frac{\partial \widehat{U}(\breve{\alpha})}{\partial \alpha}\Big)\Big]}_{S_2} + S_1 \\ &= \alpha^* - H^{*-1}_{\alpha|\boldsymbol{\theta}} \widehat{U}(\alpha^*) + S_1 + S_2,\end{aligned} \tag{3.9}$$

where $\breve{\alpha} = \alpha^* + \breve{u}(\widehat{\alpha} - \alpha^*)$ for some $\breve{u} \in [0, 1]$. We consider the terms S_1 and S_2 separately. For S_1 we have

$$|\widehat{U}(\alpha^*) - \widehat{U}(\widehat{\alpha})| \leq |\alpha^* - \widehat{\alpha}| \cdot \Big|\frac{\partial \widehat{U}(\breve{\alpha})}{\partial \alpha}\Big| = \mathcal{O}_{\mathbb{P}}(\lambda).$$

Moreover, from the analysis of $\widehat{U}(\alpha^*)$ above we have that $|\widehat{U}(\alpha^*)| = \mathcal{O}_{\mathbb{P}}(n^{-1/2})$. We then obtain

$$|S_1| \leq \Big(|\widehat{U}(\alpha^*) - \widehat{U}(\widehat{\alpha})| + |\widehat{U}(\alpha^*)|\Big) \cdot \Big[H^{*-1}_{\alpha|\boldsymbol{\theta}} - \Big(\frac{\partial \widehat{U}(\widehat{\alpha})}{\partial \alpha}\Big)^{-1}\Big] \leq \mathcal{O}_{\mathbb{P}}(s^3 \log p/n). \tag{3.10}$$

For S_2 we have that

$$|S_2| \leq |\widehat{\alpha} - \alpha^*| \cdot H^{*-1}_{\alpha|\boldsymbol{\theta}} \cdot \Big|H^{*}_{\alpha|\boldsymbol{\theta}} - \Big(\frac{\partial \widehat{U}(\breve{\alpha})}{\partial \alpha}\Big)\Big| \leq \mathcal{O}_{\mathbb{P}}(s^3 \log p/n). \tag{3.11}$$

Plugging in (3.10) and (3.11) into (3.9) we obtain

$$\sqrt{n}(\widetilde{\alpha} - \alpha^*) = -\sqrt{n} H^{*-1}_{\alpha|\boldsymbol{\theta}} \widehat{U}(\alpha^*) + o_{\mathbb{P}}(1).$$

According to (3.5), this gives

$$\sqrt{n}(\widetilde{\alpha} - \alpha^*) \to N(0, H^{*-1}_{\alpha|\boldsymbol{\theta}}),$$

and our claim (3.6) holds. □

Remark 26. *The stated sample complexity $s^6 \log^2 p/n = o(1)$ is for a general model. For specific models we may be able to get sharper results. For example for linear model and generalized linear model $s^2 \log^2 p/n = o(1)$ suffices [237].*

In Lemma 25 we focus on the case where α is a scalar. It is straightforward to generalize to the vector case. We are now almost ready for our main theorem. For any positive definite matrix V, denote $\langle x, y\rangle_V = x^\top V y$ and $\|x\|_V = (x^\top V x)^{\frac{1}{2}}$ as the inner product and the norm, respectively. For the linear space M, the usual orthogonal complement of M associated with V is defined as

$$M_V^\perp = \Big\{ y : \langle x, y\rangle_V = 0 \text{ for all } x \in M \Big\}.$$

For any positive definite matrix $V \in \mathbb{R}^{m\times m}$ and convex cone $C \subseteq \mathbb{R}^m$, let $y \sim N(0, V)$ and consider

$$T_0 = y^\top V^{-1} y - \min_{\eta\in C} (y-\eta)^\top V^{-1} (y-\eta). \tag{3.12}$$

It can be shown [270] that T_0 is distributed as a weighted mixture of Chi-squared distribution associated with V and C denoted as $T_0 \sim \bar{\chi}^2(V, C)$. That is

$$\Pr\big\{T_0 \geq c\big\} = \Pr\big\{\bar{\chi}^2(V, C) \geq c\big\} = \sum_{i=0}^{m} w_i(m, V, C) \cdot \Pr\big\{\chi_i^2 \geq c\big\}, \tag{3.13}$$

where χ_i^2 is a Chi-squared random variable with i degrees of freedom and χ_0^2 is the point mass at 0. Here $w_i(m, V, C)$ are non-negative weights satisfying $\sum_{i=1}^m w_i(m, V, C) = 1$. See Section 3.3.3 for details. We then have the following main theorem:

Theorem 27. *Suppose the hypothesis we would like to test is $H_0 : \alpha^* \in M$ versus $H_A : \alpha^* \in C\backslash M$ where we have the constraint $\alpha^* \in C$, and suppose all the conditions in Section 3.3.1 are satisfied. Then under the null hypothesis, the test statistics T_s, T_w and T_L constructed in*

Step 2 satisfy

$$T_s, T_w, T_L \rightarrow \bar{\chi}^2(H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}, C^*), \tag{3.14}$$

where $C^* = C \cap M^{\perp}_{H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}}$.

The proof of Theorem 27 is postponed to Section 3.3.5.

Remark 28. *Our method is also valid for cones not centered at the origin, for example* $C = \{\boldsymbol{\alpha} : R\boldsymbol{\alpha} \geq r\}$ *and* $M = \{\boldsymbol{\alpha} : R\boldsymbol{\alpha} = r\}$*. The two-step procedure is exactly the same as before. Under the null hypothesis* $R\boldsymbol{\alpha}^* = r$*, by removing* $\boldsymbol{\alpha}^*$ *from both* $\widetilde{\boldsymbol{\alpha}}$ *and* $\boldsymbol{b}$*, we see that* T_w *has the same distribution with the case* $C = \{\boldsymbol{\alpha} : R\boldsymbol{\alpha} \geq 0\}$ *and* $M = \{\boldsymbol{\alpha} : R\boldsymbol{\alpha} = 0\}$*. This is also validated experimentally by the sum constraint in Section 3.4.*

Remark 29. *In this work we focus on hypothesis on a low dimension parameter* $\boldsymbol{\alpha}$ *only. It is in fact straightforward to extend Theorem 27 to the whole parameter* $\boldsymbol{\beta}$*. However, as we will see in Section 3.3.3, the weights of the null distribution* (3.14) *usually lack closed form expression and can only be calculated using numerical methods in practice. When dimension of parameter of interest is large, this could be computationally intractable.*

With this weighted Chi-square distribution under the null, we can build hypothesis test for $\boldsymbol{\alpha}^*$ with any designed Type I error. It remains to calculate the weights w_i and the critical value c in (3.13). We describe the calculation of the weights in the next section. The critical value can be calculated numerically as follow.

Critical value. The final step is to calculate the critical value. Specifically, we want to find critical value c such that

$$\Pr\left\{\bar{\chi}^2\big(\widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}}, C^*\big) \geq c\right\} = \sum_{i=0}^{m} w_i\big(m, \widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}}, C^*\big) \cdot \Pr\left\{\chi_i^2 \geq c\right\} = \gamma,$$

where γ is the designed Type I error. This can be solved numerically by binary search on c.

Combining all these result we are able to build valid testing procedure for the original hypothesis (3.1) with asymptotic designed Type I error γ by calculating the $1-\gamma$ quantile of the weighted Chi-squared distribution, and reject H_0 when T_s, T_w or T_L is greater than this quantile.

3.3.3 Weights Calculation

According to Lemma 25, the covariance matrix $H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}$ can be consistently estimated by sample version (3.8). The cone C^* depends on the constraint space C, M and $H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}$. For example for non-negative constraint, we have $M = \{\boldsymbol{\alpha} : \boldsymbol{\alpha} = \mathbf{0}\}$ and hence $M^{\perp}_{H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}} = \mathbb{R}^d$ and $C^* = C \cap M^{\perp}_{H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}} = C$; for monotonic constraint, we have $M = \{\boldsymbol{\alpha} : \alpha_1 = \alpha_2 = ... = \alpha_d\}$ and hence $M^{\perp}_{H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}} = \{\boldsymbol{\alpha} : \mathbf{1}^\top H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}} \cdot \boldsymbol{\alpha} = 0\}$. Since $C = \{\boldsymbol{\alpha} : \alpha_1 \leq \alpha_2 \leq ... \leq \alpha_d\}$, we have

$$C^* = C \cap M^{\perp}_{H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}} = \{\boldsymbol{\alpha} : \alpha_1 \leq \alpha_2 \leq ... \leq \alpha_d, \mathbf{1}^\top H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}} \cdot \boldsymbol{\alpha} = 0\}.$$

The weights $w_i(d, H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}, C^*)$ depend on $H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}$ and C^* and can be complicated and without closed form expression. Here we briefly review the expression of general weights $w_i(m, V, C)$ for some general dimension m, covariance matrix V, and cone C obtained in [188]. We refer to [270] for more detailed formulas. We start from the simplest case where $C = \mathbb{R}^m_+$ and $V = I$. From (3.12) we have

$$T_0 = \sum_{i=1}^{m} \max(y_i, 0)^2 \sim \bar{\chi}^2(I, \mathbb{R}^m_+).$$

We can see that the weight $w_i(m, I, \mathbb{R}^m_+)$ depends on the number of positive components of y: if i of them are positive then the distribution would be χ^2_i. There are in total 2^m choices of signs on each component of y and therefore

$$w_i(m, I, \mathbb{R}^m_+) = 2^{-m}\binom{m}{i}.$$

We then consider $C = \mathbb{R}^m_+$ with general V where the weights are given by

$$w_i(m, V, \mathbb{R}^m_+) := \sum_{|\mathcal{A}|=i,\, \mathcal{A}\subseteq[m]} p\Big\{(V_{\mathcal{A}^c})^{-1}\Big\} \cdot p\Big\{V_{\mathcal{A};\mathcal{A}^c}\Big\}, \tag{3.15}$$

where the summation runs over all subsets $\mathcal{A}$ of $\{1, ..., m\}$ having i elements. $\mathcal{A}^c$ is the complement of $\mathcal{A}$ and $V_{\mathcal{A}}$ is the submatrix of V corresponding to those y_i where $i \in \mathcal{A}$. $V_{\mathcal{A};\mathcal{A}^c}$ is the covariance matrix under the condition $y_j = 0$ where $j \in \mathcal{A}^c$. Finally $p(\Lambda)$ denotes the probability that $z \geq 0$ for a Gaussian random variable $z \sim N(0, \Lambda)$.

The weight (3.15) can be approximated using Monte Carlo simulation when m is relatively small. For large m this could be computational intensive, but since we are interested in $\boldsymbol{\alpha} \in \mathbb{R}^d$ with $d \ll p$ we expect d to be relatively small.

We then consider more general cones C and show how they can be reduced to the above case $C = \mathbb{R}^m_+$ as proposed in [270]. First suppose C is defined by linear inequality constraints

$$C_R = \{\boldsymbol{\alpha} : R\boldsymbol{\alpha} \geq 0\},$$

where $R \in \mathbb{R}^{m\times m}$ is nonsingular. In this case by linear transformation we have

$$w_i(m, V, C_R) = w_i\big(m, RVR^\top, \mathbb{R}^m_+\big).$$

More generally suppose $R \in \mathbb{R}^{k\times m}$ with rank k, we have

$$w_{m-k+j}(m, V, C_R) = w_j\big(k, RVR^\top, \mathbb{R}^m_+\big),$$

and the remaining weights vanish.

Finally consider the standard linear constraint with $C = \{\boldsymbol{\alpha} : R\boldsymbol{\alpha} \geq 0\}$ and $M = \{\boldsymbol{\alpha} : R\boldsymbol{\alpha} = 0\}$ where $R \in \mathbb{R}^{k\times m}$ has full row rank. In this case we can calculate the final weights associated with C^* directly. We first find $A \in \mathbb{R}^{(m-k)\times m}$ as the null space of RV satisfying

$RVA^\top = 0$. Then the cone C^* is given by

$$C^* = \{\boldsymbol{\alpha} : R\boldsymbol{\alpha} \geq 0, A\boldsymbol{\alpha} = 0\},$$

and the final weights associated with C^* are given by

$$w_j(m, V, C^*) = w_j(k, RVR^\top, \mathbb{R}_+^m).$$

3.3.4 Power analysis

In this section we analyze the power of our proposed method and compare with the standard method where the constraints are ignored. Since it is unclear how to define the margin and alternative hypothesis for general cone constraint, in this section we focus on the nonnegativity constraint. The idea can be generalized to general cone straightforwardly.

We start from the scalar case where $d = 1$ and α is a scalar. In this case we want to test for

$$H_0 : \alpha^* = 0 \quad \text{versus} \quad H_A : \alpha^* > 0.$$

To ease calculation we assume we have $n = 1$ sample and the variance is known as $\sigma^2 = H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}} = 1$. Since the three tests are asymptotically equivalent, we focus on Wald test only. According to Step 2 we have $T_w = \max(\widetilde{\alpha}, 0)^2$ where $\widetilde{\alpha} \rightarrow N(\alpha^*, 1)$ by Lemma 25. Under the null hypothesis $\alpha^* = 0$, the asymptotic null distribution of T_w is given by

$$T_w \rightarrow \frac{1}{2}\chi_0^2 + \frac{1}{2}\chi_1^2.$$

Based on this asymptotic null distribution we reject the null hypothesis when T_w is large. For standard Wald test where the nonnegativity constraint is ignored, the asymptotic null distribution is $\widetilde{\alpha} \rightarrow N(0, 1)$ and we reject the null hypothesis when $\widetilde{\alpha}$ is large. Denote $\Phi(\cdot)$ as the cumulative distribution function of standard normal variable. Given the designed Type I error γ, the critical value for standard method is given by $\tau_1 = \Phi^{-1}(1-\gamma/2)$ and the critical value for our method is given by $\tau_2 = \Phi^{-1}(1-\gamma)$. Under the alternative hypothesis that $\alpha^* > 0$, the power of the standard method is given by

$$\mathbb{P}_{\text{standard}}(\text{reject} \,|\, \alpha^*) = 1 - \Phi(\tau_1 - \alpha^*) + \Phi(-\tau_1 - \alpha^*),$$

while the power of our method is given by

$$\mathbb{P}_{\text{our}}(\text{reject} \,|\, \alpha^*) = 1 - \Phi(\tau_2 - \alpha^*).$$

It is straightforward to calculate that

$$f(\alpha^*) \coloneqq \mathbb{P}_{\text{our}}(\text{reject} \,|\, \alpha^*) - \mathbb{P}_{\text{standard}}(\text{reject} \,|\, \alpha^*) = \big[\Phi(\tau_1 - \alpha^*) - \Phi(\tau_2 - \alpha^*)\big] - \Phi(-\tau_1 - \alpha^*).$$

For small α^* ($0 < \alpha^* \leq \tau_1 + \tau_2$), we have

$$\Phi(\tau_1 - \alpha^*) - \Phi(\tau_2 - \alpha^*) \geq \frac{\gamma}{2} > \Phi(-\tau_1 - \alpha^*),$$

and hence $f(\alpha^*) > 0$. For large α^* ($\alpha^* > \tau_1 + \tau_2$), we write $\alpha^* = \tau_1 + \tau_2 + \epsilon$ with some $\epsilon > 0$ and rewrite $f(\alpha^*)$ as

$$g(\epsilon) \coloneqq f(\alpha^*) = \Phi(-\tau_2 - \epsilon) - \Phi(-\tau_1 - \epsilon) - \Phi(-2\tau_1 - \tau_2 - \epsilon).$$

Clearly we have $g(0) > 0$ and $g(+\infty) = 0$ and also

$$\begin{aligned} g'(\epsilon) &= -\phi(-\tau_2 - \epsilon) + \phi(-\tau_1 - \epsilon) + \phi(-2\tau_1 - \tau_2 - \epsilon) \\ &= -\phi(\tau_2 + \epsilon) + \phi(\tau_1 + \epsilon) + \phi(2\tau_1 + \tau_2 + \epsilon), \end{aligned}$$

where $\phi(\cdot)$ denotes the probability density function of standard normal variable. Since $\phi(\cdot)$

decays exponentially, some simple calculation shows that $g'(\epsilon) < 0$ for any $\epsilon > 0$. Together with the fact that $g(0) > 0$ and $g(+\infty) = 0$, we know $g(\epsilon) > 0$ for any $\epsilon > 0$, which indicates that $f(\alpha^*) > 0$ for $\alpha^* > \tau_1 + \tau_2$. Therefore, for any $\alpha^* > 0$, we have $f(\alpha^*) > 0$ which shows that our method has greater power than the standard method.

Figure 3.1 shows the powers obtained by our method and standard method for $\gamma = 0.05$. We can see that when $\alpha^* = 0$ (i.e. under the null) both methods have Type I error 0.05. As α^* increases and the null is violated, our method has much larger power compared to the standard method. Finally when α^* is sufficiently large, both methods has power close to 1.

We then turn to the vector case where $\boldsymbol{\alpha} \in \mathbb{R}^d$. Again to ease calculation we assume we have $n = 1$ sample and the variance is known as $\Sigma = H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}} = I_d$. In this case we have $\widetilde{\boldsymbol{\alpha}} \rightarrow N(\boldsymbol{\alpha}^*, I_d)$ by Lemma 25, and the asymptotic null distribution of T_w is given by

$$T_w \rightarrow \sum_{i=0}^{d} 2^{-d} \binom{d}{i} \chi_i^2.$$

To violate the null hypothesis, we increase the value α_1^* and $\alpha_2^*, ..., \alpha_d^*$ remain to be 0. Figure 3.2 shows the comparison result for $d = 4$ and $\gamma = 0.05$. The pattern is similar to Figure 3.1.

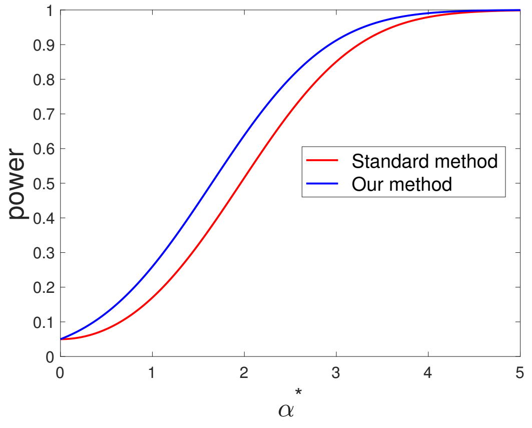


Figure 3.1: Comparison of power with $d = 1$

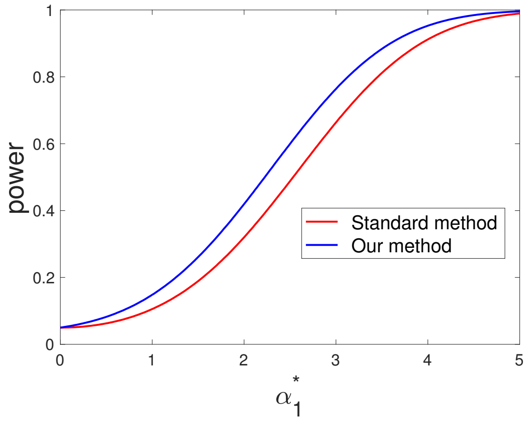


Figure 3.2: Comparison of power with $d = 4$

3.3.5 Proof of Theorem 27

Before we proceed with the main proof, we first introduce the following lemma in [270].

Lemma 30. *For any positive definite matrix $V \in \mathbb{R}^{m\times m}$, convex cone $C \subseteq \mathbb{R}^m$ and linear space $M \subseteq C$, let $y \sim N(\mu, V)$ with $\mu \in M$. Then the statistic*

$$T = \min_{\eta\in M}(y-\eta)^\top V^{-1}(y-\eta) - \min_{\eta\in C}(y-\eta)^\top V^{-1}(y-\eta) \tag{3.16}$$

has the distribution $\bar{\chi}^2(V, C^*)$ where $C^* = C \cap M^\perp_{V^{-1}}$.

Proof. Denote $P(\cdot, C)$ as the orthogonal projection onto C according to norm $\|\cdot\|_{V^{-1}}$. We have

$$\big\|y - P(y,C)\big\|^2_{V^{-1}} = \min_{\eta\in C}(y-\eta)^\top V^{-1}(y-\eta).$$

Since M is linear space, we have the Pythagoras' theorem

$$\big\|y-P(y,M)\big\|^2_{V^{-1}} = \big\|y-P(y,C)\big\|^2_{V^{-1}} + \big\|P(y,C^*)\big\|^2_{V^{-1}}.$$

Then (3.16) follows directly from (3.12). □

We then proceed with the proof of Theorem 27. According to Lemma 25 we have

$$\sqrt{n}(\widetilde{\boldsymbol{\alpha}} - \boldsymbol{\alpha}^*) \rightarrow N(\mathbf{0}, H^{*-1}_{\boldsymbol{\alpha}|\boldsymbol{\theta}}).$$

Under the null $\boldsymbol{\alpha}^* \in M$, Lemma 30 immediately indicates that $T_w \rightarrow \bar{\chi}^2(H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}, C^*)$, where $C^* = C \cap M^\perp_{H^*_{\boldsymbol{\alpha}|\boldsymbol{\theta}}}$. We then show that T_L and T_s are asymptotically equivalent to T_w. For T_L, following Proposition 4.2.2 in [278], we have the local quadratic approximation

$$\begin{aligned}\ell_{\text{de}}(\boldsymbol{b}) &= \ell_{\text{de}}(\widetilde{\boldsymbol{\alpha}}) + \frac{\partial \ell_{\text{de}}(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}\bigg|^\top_{\boldsymbol{\alpha}=\widetilde{\boldsymbol{\alpha}}}(\boldsymbol{b}-\widetilde{\boldsymbol{\alpha}}) + \frac{1}{2}(\boldsymbol{b}-\widetilde{\boldsymbol{\alpha}})^\top \frac{\partial^2 \ell_{\text{de}}(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}^2}\bigg|_{\boldsymbol{\alpha}=\widetilde{\boldsymbol{\alpha}}}(\boldsymbol{b}-\widetilde{\boldsymbol{\alpha}}) + o_P(1) \\ &= \ell_{\text{de}}(\widetilde{\boldsymbol{\alpha}}) + \frac{1}{2}(\boldsymbol{b}-\widetilde{\boldsymbol{\alpha}})^\top \widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}}(\boldsymbol{b}-\widetilde{\boldsymbol{\alpha}}) + o_P(1),\end{aligned}$$

where the second term follows from the definition of $\widetilde{\boldsymbol{\alpha}}$ and the following Taylor expansion

$$\frac{\partial \ell_{\text{de}}(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}}\bigg|_{\boldsymbol{\alpha}=\widetilde{\boldsymbol{\alpha}}} = \widehat{\boldsymbol{U}}(\widetilde{\boldsymbol{\alpha}}) = \widehat{\boldsymbol{U}}(\widehat{\boldsymbol{\alpha}}) + \Big(\frac{\partial \widehat{\boldsymbol{U}}(\widehat{\boldsymbol{\alpha}})}{\partial \boldsymbol{\alpha}}\Big) \cdot (\widetilde{\boldsymbol{\alpha}} - \widehat{\boldsymbol{\alpha}}) + o_p(1) = o_p(1).$$

The first term $\ell_{\text{de}}(\widetilde{\boldsymbol{\alpha}})$ is a constant over $\boldsymbol{b}$, therefore we have

$$\begin{aligned} T_L &= 2\Big(\inf_{\boldsymbol{b}\in M} \ell_{\text{de}}(\boldsymbol{b}) - \inf_{\boldsymbol{b}\in C} \ell_{\text{de}}(\boldsymbol{b})\Big) \\ &= \inf_{\boldsymbol{b}\in M} \Big\{(\widetilde{\boldsymbol{\alpha}} - \boldsymbol{b})^\top \widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}} (\widetilde{\boldsymbol{\alpha}} - \boldsymbol{b}) + o_p(1)\Big\} - \inf_{\boldsymbol{b}\in C} \Big\{(\widetilde{\boldsymbol{\alpha}} - \boldsymbol{b})^\top \widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}} (\widetilde{\boldsymbol{\alpha}} - \boldsymbol{b}) + o_p(1)\Big\} \\ &= T_w + o_p(1). \end{aligned}$$

This shows that T_L has the same asymptotic distribution as T_w. Similarly, for T_s we have the local approximation

$$\widehat{\boldsymbol{U}}(b) = \widehat{\boldsymbol{U}}(\widetilde{\boldsymbol{\alpha}}) + \widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}} \cdot (\boldsymbol{b} - \widetilde{\boldsymbol{\alpha}}) + o_p(1).$$

Plugging in $\boldsymbol{b}_M$ and $\boldsymbol{b}_C$ we obtain

$$\begin{aligned} T_s &= \Big(\widehat{\boldsymbol{U}}(\boldsymbol{b}_M) - \widehat{\boldsymbol{U}}(\boldsymbol{b}_C)\Big)^\top \widehat{H}^{-1}_{\boldsymbol{\alpha}|\boldsymbol{\theta}} \Big(\widehat{\boldsymbol{U}}(\boldsymbol{b}_M) - \widehat{\boldsymbol{U}}(\boldsymbol{b}_C)\Big) \\ &= \big(\boldsymbol{b}_M - \boldsymbol{b}_C\big)^\top \widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}} \big(\boldsymbol{b}_M - \boldsymbol{b}_C\big) + o_p(1) \\ &= \big(\widetilde{\boldsymbol{\alpha}} - \boldsymbol{b}_M\big)^\top \widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}} \big(\widetilde{\boldsymbol{\alpha}} - \boldsymbol{b}_M\big) - \big(\widetilde{\boldsymbol{\alpha}} - \boldsymbol{b}_C\big)^\top \widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}} \big(\widetilde{\boldsymbol{\alpha}} - \boldsymbol{b}_C\big) + o_p(1) \\ &= T_w + o_p(1), \end{aligned}$$

where the third equality comes from the Pythagoras' theorem

$$\big\|\widetilde{\boldsymbol{\alpha}} - P(\widetilde{\boldsymbol{\alpha}}, C)\big\|^2 = \big\|\widetilde{\boldsymbol{\alpha}} - P(\widetilde{\boldsymbol{\alpha}}, M)\big\|^2 + \big\|P(\widetilde{\boldsymbol{\alpha}}, M) - P(\widetilde{\boldsymbol{\alpha}}, C)\big\|^2,$$

and the fact that

$$\begin{aligned}\boldsymbol{b}_M &= \arg\inf_{\boldsymbol{b}\in M} \ell_{\mathrm{de}}(\boldsymbol{b}) = \arg\inf_{\boldsymbol{b}\in M} \left\{\ell_{\mathrm{de}}(\widetilde{\boldsymbol{\alpha}}) + \frac{1}{2}(\boldsymbol{b}-\widetilde{\boldsymbol{\alpha}})^\top \widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}}(\boldsymbol{b}-\widetilde{\boldsymbol{\alpha}}) + o_p(1)\right\} \\ &= \arg\inf_{\boldsymbol{b}\in M} \left\{(\boldsymbol{b}-\widetilde{\boldsymbol{\alpha}})^\top \widehat{H}_{\boldsymbol{\alpha}|\boldsymbol{\theta}}(\boldsymbol{b}-\widetilde{\boldsymbol{\alpha}})\right\} + o_p(1).\end{aligned}$$

This shows that T_s has the same asymptotic distribution as T_w, which completes the proof.

3.4 Synthetic Data

In this section we apply our method on synthetic datasets. We consider linear model $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta}^* + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim N_n(\mathbf{0}, \sigma^2\boldsymbol{I}_n)$, and impose different kinds of constraints on the first two variables $\boldsymbol{\alpha}^* = (\alpha_1^*, \alpha_2^*) = (\beta_1^*, \beta_2^*)$. Specifically, we consider the following three constraints.

1. Monotonicity constraint. We have the monotonic constraint $\alpha_1 \leq \alpha_2$, and the hypothesis we would like to test is

   $$H_0 : \alpha_1^* = \alpha_2^* \quad \text{versus} \quad H_A : \alpha_1^* < \alpha_2^*.$$

   For the experiment we set $\alpha_1^* = \alpha_2^* = -1$ and $\alpha_i^* = 0$ elsewhere.

2. Non-negativity constraint. We have the non-negative constraint $\alpha_1, \alpha_2 \geq 0$, and the hypothesis we would like to test is

   $$H_0 : \alpha_1^* = \alpha_2^* = 0 \quad \text{versus} \quad H_A : \alpha_1^* > 0 \text{ or } \alpha_2^* > 0.$$

   For the experiment we set $\alpha_1^* = \alpha_2^* = 0$, $\alpha_p^* = \alpha_{p-1}^* = 1$ where p is the dimension of $\boldsymbol{\beta}$, and $\alpha_i^* = 0$ elsewhere.

3. Sum constraint. We have the sum constraint $\alpha_1 + \alpha_2 \leq -2$, and the hypothesis we

would like to test is

$$H_0 : \alpha_1^* + \alpha_2^* = -2 \quad \text{versus} \quad H_A : \alpha_1^* + \alpha_2^* < -2.$$

For the experiment we set $\alpha_1^* = \alpha_2^* = -1$ and $\alpha_i^* = 0$ elsewhere.

In low dimensions we have the Least Square estimator $\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{Y}$ with $\widehat{\boldsymbol{\beta}} \sim N\Big(\boldsymbol{\beta}^*, \sigma^2 (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\Big)$, from which we can construct confidence interval and hypothesis testing for $\boldsymbol{\beta}^*$. In high dimension Least Square estimator is ill-conditioned so we instead calculate penalized estimator $\widehat{\boldsymbol{\beta}}$ according to (3.3). For example letting P_λ be L_1 penalty we get the LASSO estimator. Alternatively we can get the estimator $\widehat{\boldsymbol{\beta}}$ under constraint. For example for non-negativity constraint, we can get nonnegative sparse estimator directly [284].

In [237] the authors show that our conditions in Section 3.3.1 are satisfied for linear regression so we then follow our procedure to calculate the test statistics T_s, T_w and T_L. We set $\sigma = 1$ and we assume σ is known. Each row of $\boldsymbol{X}$ is sampled from multivariate normal distribution $\boldsymbol{X} \sim N_p(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a Toeplitz matrix with $\Sigma_{jk} = \rho^{|j-k|}$. The tuning parameter is set to be $\lambda = \sqrt{\log p/n}$ and $\lambda' = \frac{1}{2}\sqrt{\log p/n}$. We vary $\rho \in \{0.2, 0.4, 0.6, 0.8\}$, $p \in \{100, 300, 500\}$ and for each setting we generate $n = 200$ samples. The averaged empirical Type I error on 500 replicates under the three different constraints are shown in Table 3.1 - 3.3. The designed Type I error is 5%.

From the three tables we see that our algorithm works well for all these three constraints. We then check the power of our algorithm. For each constraint, we introduce a variable *margin* that measures how much we violate the null hypothesis (i.e. how far we are away from the boundary). Specifically, for monotonic constraint, we set $\alpha_1^* = -1, \alpha_2^* = -1 + margin$; for non-negative constraint we set $\alpha_1^* = \alpha_2^* = margin/2$; for sum constraint, we set $\alpha_1^* = -1, \alpha_2^* = -1 - margin$. Intuitively, as the margin increases, the power of the test will increase. For all the three constraints, we compare the power of our testing procedures to the standard Wald/Score/Likelihood ratio tests where we ignore the constraint. For example for monotonic

Table 3.1: Empirical Type I error for monotonic constraint

Method	ρ p	0.2	0.4	0.6	0.8
Score	100	6.4%	5.6%	5.4%	6.8%
	300	6.6%	5.4%	6.4%	5.6%
	500	6.8%	5.4%	6.6%	6.6%
Wald	100	5.4%	4.4%	4.8%	7.0%
	300	5.0%	3.6%	5.2%	6.2%
	500	3.8%	3.2%	3.8%	5.2%
LR	100	6.2%	4.8%	5.4%	6.4%
	300	5.2%	5.2%	5.8%	6.4%
	500	4.8%	4.0%	6.0%	5.6%

Table 3.2: Empirical Type I error for non-negative constraint

Method	ρ p	0.2	0.4	0.6	0.8
Score	100	5.6%	6.2%	6.2%	4.8%
	300	4.6%	5.2%	5.2%	6.2%
	500	5.4%	5.6%	5.0%	5.2%
Wald	100	5.6%	4.8%	5.0%	4.6%
	300	5.0%	3.6%	4.0%	3.6%
	500	3.2%	4.2%	3.4%	3.2%
LR	100	6.0%	5.0%	5.8%	4.6%
	300	3.6%	4.2%	3.6%	4.8%
	500	3.6%	3.4%	4.8%	4.4%

constraint our method tests for

$$H_0 : \alpha_1^* = \alpha_2^* \quad \text{versus} \quad H_A : \alpha_1^* < \alpha_2^*,$$

while the standard method tests for

$$H_0 : \alpha_1^* = \alpha_2^* \quad \text{versus} \quad H_A : \alpha_1^* \neq \alpha_2^*.$$

We vary *margin* $\in \{0, 0.05, 0.1, 0.2, 0.3, 0.5, 1\}$ where *margin* $= 0$ corresponds to null

Table 3.3: Empirical Type I error for sum constraint

Method	ρ p	0.2	0.4	0.6	0.8
Score	100	4.8%	6.2%	5.4%	5.2%
	300	4.8%	4.4%	5.8%	5.4%
	500	4.4%	3.8%	3.6%	4.0%
Wald	100	3.6%	5.4%	4.2%	4.0%
	300	3.8%	4.4%	3.6%	4.2%
	500	4.0%	3.6%	4.2%	4.0%
LR	100	4.2%	5.6%	4.4%	5.2%
	300	3.8%	4.4%	4.8%	5.0%
	500	3.4%	4.0%	5.2%	4.6%

hypothesis and others corresponds to alternative hypothesis. Under the alternative hypothesis, for both our method and standard method, Wald/Score/Likelihood ratio tests gives nearly identical power. Therefore we only report the mean of them. The comparison results on 500 replicates are shown in Table 3.4, and we can see that by considering the known constraint, our tests have much stronger power.

Table 3.4: Power of the tests

Constraint	margin Method	0	0.05	0.1	0.2	0.3	0.5	1
Monotonic	Our method	0.045	0.061	0.113	0.211	0.331	0.597	0.988
	Standard method	0.047	0.044	0.068	0.138	0.235	0.488	0.978
Non-negative	Our method	0.036	0.069	0.112	0.278	0.504	0.879	1.000
	Standard method	0.039	0.032	0.047	0.134	0.323	0.788	1.000
Sum	Our method	0.060	0.169	0.266	0.478	0.712	0.950	1.000
	Standard method	0.041	0.097	0.156	0.340	0.596	0.922	1.000

3.5 Real Data

In this section we apply our method to two real datasets on ARCH model and information diffusion model. For both the models, we have the *intrinsic* non-negative constraint on the parameters. Therefore, to provide statistical inference on the parameters, we should use constrained testing method.

3.5.1 ARCH Model

As a first example, we consider the application of our method in financial economics, where most of the existing works focus on estimations and predictions [171, 107, 214, 106, 25]. However, people are usually more interested in testing whether a specific factor affects the prediction results, with a focus on testing inequality constraints [328]. The model we consider is the autoregressive conditional heteroscedasticity (ARCH) model introduced in [94]. ARCH model is very popular in modeling financial economic time series like exchange rates, commodity prices. The main feature is that ARCH model attempts to model the variance as well. More formally, ARCH models assume the variance of the current error term to be a function of the actual sizes of the previous time periods' error terms. To introduce the model, let $\mathcal{F}_t$ be the information up to time t, y_t be the dependent variable and $\boldsymbol{x}_t$ be exogeneous variables included in $\mathcal{F}_{t-1}$ ($\boldsymbol{x}_t$ may contain lagged dependent variables like y_{t-1} and y_{t-2}). An ARCH model with lag length q can be written as

$$\begin{aligned} y_t|\mathcal{F}_{t-1} &\sim N(\boldsymbol{x}_t^\top \boldsymbol{\beta}, h_t), \\ h_t &= \alpha_0 + \alpha_1 \epsilon_{t-1}^2 + ... + \alpha_q \epsilon_{t-q}^2, \\ \epsilon_t &= y_t - \boldsymbol{x}_t^\top \boldsymbol{\beta}. \end{aligned} \tag{3.17}$$

From the definition of the model we can see that, if some α_i in (3.17) is negative, then a large value for ϵ_{t-i} would lead to negative variance for y_t. Hence the admissible range for $\alpha_1, ..., \alpha_q$ should be $\left\{\alpha_1 \geq 0, ..., \alpha_q \geq 0\right\}$. Therefore, the testing problem should be

$$H_0 : \alpha_i = 0 \text{ versus } H_A : \alpha_i > 0,$$

instead of

$$H_0 : \alpha_i = 0 \text{ versus } H_A : \alpha_i \neq 0.$$

In this section we focus on α_1 and test for $H_0 : \alpha_1 = 0$ versus $H_A : \alpha_1 > 0$.

The data we use are the All Ordinaries Index (Australia) from January 5, 1984 to November 29, 1985, denoted as I_t. This index is a weighted average of the prices of selected shares in Australia which corresponds to the Dow-Jones Index in the United States. The data are from the *Australian Financial Review*. We have a total of 484 observations. The return variable y_i is defined as $\log(I_t/I_{t-1})$, and $\boldsymbol{x}_t$ are the lagged dependent variables.

We estimate $\boldsymbol{\alpha}$ by first estimating the best fitting autoregressive model AR(q):

$$y_t = a_0 + a_1 y_{t-1} + ... + a_q y_{t-q} + \epsilon_t.$$

We then obtain the squares of the error $\widehat{\epsilon}^2$ and regress them on a constant and q lagged values:

$$\widehat{\epsilon}_t^2 = \widehat{\alpha}_0 + \widehat{\alpha}_1 \widehat{\epsilon}_{t-1}^2 + ... + \widehat{\alpha}_q \widehat{\epsilon}_{t-q}^2.$$

The estimation is based on LASSO estimator with L_1 penalty. We then follow our procedure to give the p-value. We choose $q = 30$ here and it turns out that the result is not sensitive to the choice of q. All the three tests give p-value 0.41, indicating that we should not reject the null. This result is consistent with the claim in [279].

3.5.2 Information Diffusion

The second model we consider is the network model. Network and graphical models have been widely used in fields including neuroscience, social sciences, and statistics [117, 376, 111, 366, 226, 92, 109, 110, 38, 174, 112]. We consider the time diffusion model where a diffusion matrix A quantifies the structure between nodes. If $a_{ij} \neq 0$ then there is a link

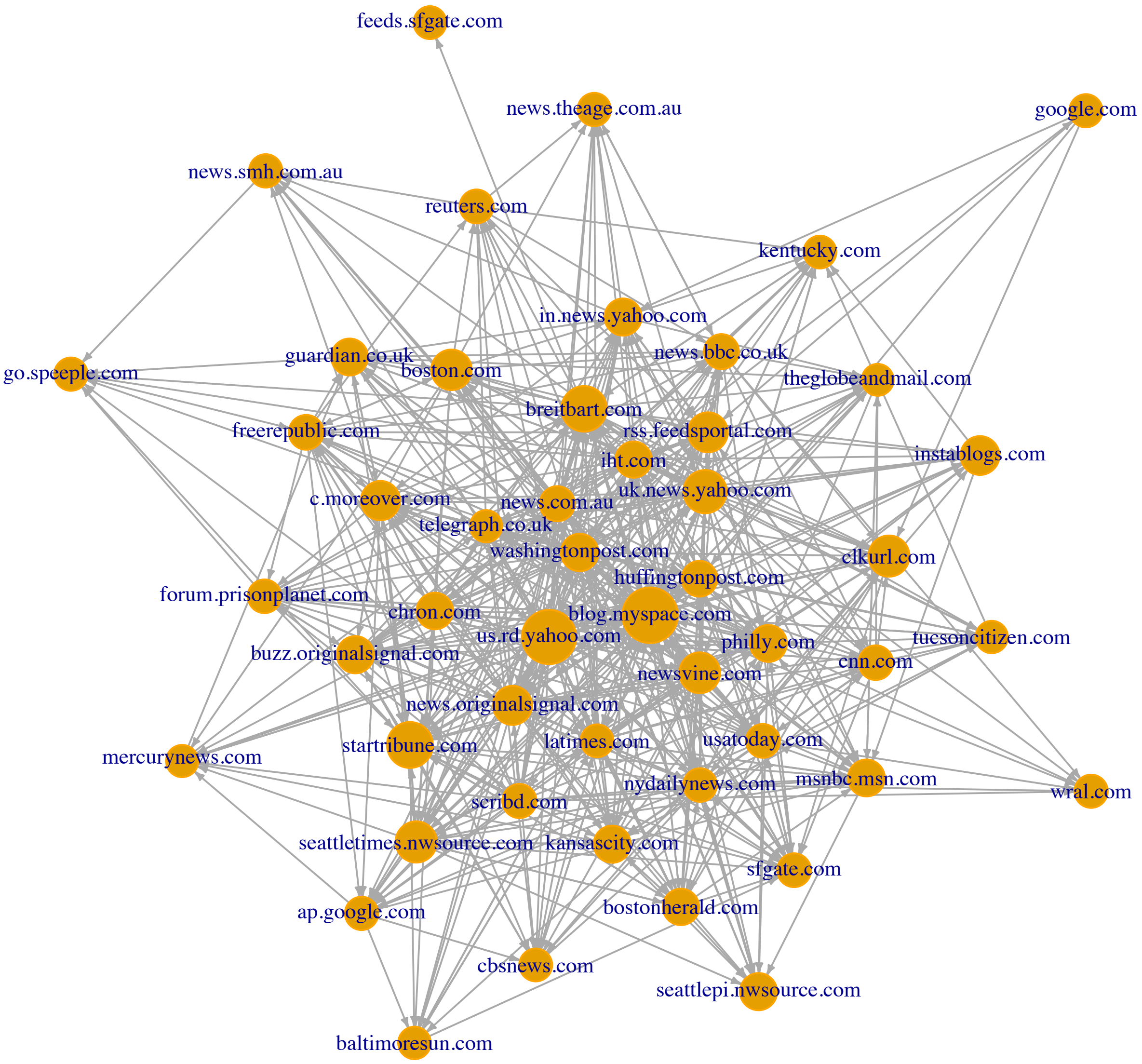


Figure 3.3: Network structure by MLE

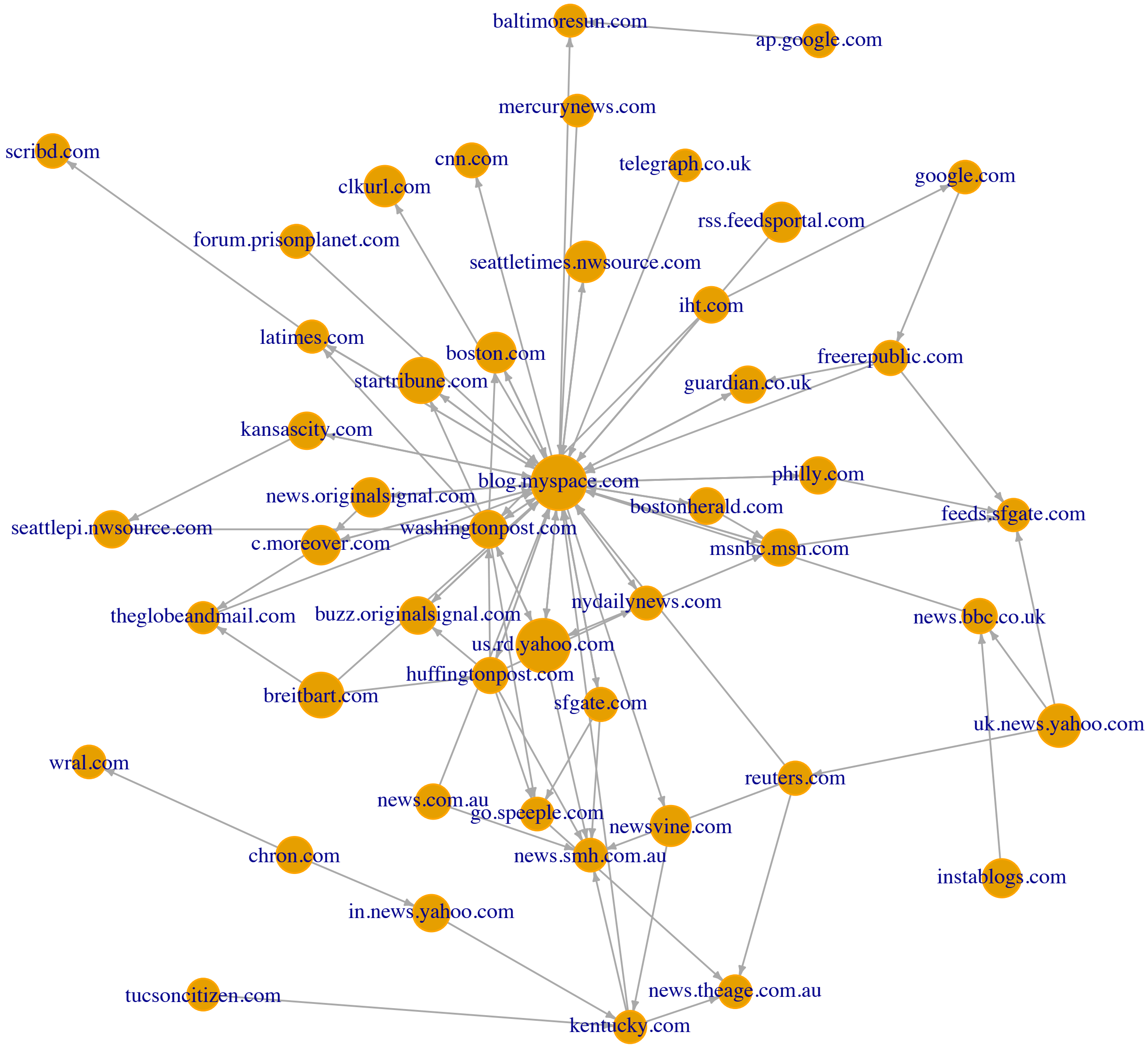


Figure 3.4: Network structure by our algorithm with fixed critical p-value 0.05

$i \to j$ and information from node i may propagate to j. The parameter a_{ij} measures how strong the relation is. Clearly only nonnegative a_{ij} is meaningful in this model. Therefore if we want to know whether there exists an edge from i to j (i.e. whether $a_{ij} \neq 0$), this is a constrained testing problem with nonnegativity constraint (i.e. we should test for whether $a_{ij} > 0$). For this network diffusion problem, many existing methods [261, 119, 349, 347] have been proposed to recover the diffusion matrix A. However, all of them focus on point estimation with no statistical inference.

The specific diffusion model we use is the discretized CICE model introduced in [251]. We use the Memetracker dataset [199][1] which contains more than 172 million news articles and blog posts from 1 million online sources. This dataset contains many textual phrases (like 'lipstick on a pig') extracted from websites, and the time each website mentioned it. We cluster the phrases to aggregate different textual variants of the same phrase. After aggregating different textual variants of the same phrase, we consider each phrase cluster as a separate cascade c. Since all documents are time stamped, a cascade c is simply a set of time-stamps when websites first mentioned a phrase in the phrase cluster c. We can observe the times when websites mention a particular phrase but we don't know where they copied that phrase from.

For the experiments we extract top 50 sites with about 2000 cascades among it. We first use penalized Maximum Likelihood Estimation for discrete CICE model in [251] with appropriate penalty parameter to estimate the network diffusion matrix: this network structure is shown in Figure 3.3. It is very dense and has many false positive edges. We then apply our algorithm to check the significance of each discovered edge. We fix the critical p-value to be 0.05 and keep the edges with p-value less than or equal to 0.05. After applying our algorithm the estimated network structure is shown in Figure 3.4. This network structure is much sparser and clearer. Note that this is different from using larger penalty on MLE which also gives a more sparse network structure but without statistical significance. In contrast, our procedure

[1] Data available at `http://memetracker.org`

is able to test the significance of each edge. Also note that this 95% confidence is for each edge individually, not the whole graph. If we want to recover the whole graph, that is a multiple testing problem for which we can apply multiple testing techniques on the p-values given by our algorithm, for example the Holm-Bonferroni method [147].

3.6 Conclusion

In this chapter we consider the hypothesis testing problem on low dimensional parameters in high dimensional models with cone constraint on the parameters. We provide modified Wald/Score/Likelihood ratio procedures to test whether the low dimensional parameters are on the boundary of the cone constraint or not. We prove that following our procedure we can get an asymptotic designed Type I error under the null. Our algorithm has stronger power compared to the standard methods where we ignore the constraint.

For future work, it is of interest to consider more general constraint $C = \{\boldsymbol{\alpha} : f(\boldsymbol{\alpha}) \geq 0\}$ and possibly nonlinear boundary set M. Another future extension is to develop algorithms for models where some of our assumptions are violated. For example, for continuous time diffusion model, our Score Condition is violated [119]. Extending our algorithm to incorporate this model is work in progress.

3.7 Technical proofs

We provide the proofs of Lemmas used in the previous sections. Some of the proofs are motivated by [102].

Lemma 31. *Suppose all the conditions in Section 3.3.1 are satisfied, for any vector $\boldsymbol{v} \in \mathbb{R}^p$ with $\|\boldsymbol{v}\|_0 \leq s$, we have*

$$\frac{\sqrt{n}\boldsymbol{v}^\top \nabla \ell(\boldsymbol{\beta}^*)}{\sqrt{\boldsymbol{v}^\top\ H^* \boldsymbol{v}}} \xrightarrow{d} N(0,1). \tag{3.18}$$

Proof. We define $\xi_i(\boldsymbol{\beta}^*) = -\nabla \log \mathcal{L}_i(\boldsymbol{\beta}^*)$, where $\mathcal{L}_i$ is the likelihood function for one trial i. According to (3.2), we have $\nabla \ell(\boldsymbol{\beta}^*) = -\frac{1}{n}\sum_i \nabla \log \mathcal{L}_i(\boldsymbol{\beta}^*) = \frac{1}{n}\sum_i \xi_i(\boldsymbol{\beta}^*)$ and from now we write ξ_i for simplicity. From the definition of H^* we have

$$H^* = n\text{Var}\Big(\ell(\boldsymbol{\beta}^*)\Big) = \text{Var}(\xi_i),$$

and hence

$$\text{Var}(\boldsymbol{v}^\top \xi_i) = \boldsymbol{v}^\top H^* \boldsymbol{v}.$$

From the score condition we have $\mathbb{E}[\xi_i] = 0$ and hence

$$\mathbb{E}[\boldsymbol{v}^\top \xi_i] = 0.$$

We then know that $\frac{\boldsymbol{v}^\top \xi_i}{\sqrt{\boldsymbol{v}^\top \boldsymbol{H}^* \boldsymbol{v}}}$ has mean 0 and variance 1. Therefore the LHS of (3.18) is sum of n independent random variables. We then verify the Lyapunov condition [154]:

$$\begin{aligned}
&\lim_{n\to\infty} n^{-\frac{3}{2}} \sum_i \mathbb{E}\left|\frac{\boldsymbol{v}^\top \xi_i}{\sqrt{\boldsymbol{v}^\top H^* \boldsymbol{v}}}\right|^3 \\
&\le \lim_{n\to\infty} n^{-\frac{3}{2}} \sum_i \mathbb{E}\left|\frac{\boldsymbol{v}^\top \xi_i}{\sqrt{c_{\min}}\|\boldsymbol{v}\|_2}\right|^3 \\
&\le \lim_{n\to\infty} n^{-\frac{3}{2}} \sum_i \mathbb{E}\left|\frac{\boldsymbol{v}^\top \xi_i}{\sqrt{\frac{c_{\min}}{s}}\|\boldsymbol{v}\|_1}\right|^3 \\
&= \lim_{n\to\infty} n^{-\frac{3}{2}} \Big(\frac{s}{c_{\min}}\Big)^{\frac{3}{2}} \sum_i \mathbb{E}\left|\frac{\boldsymbol{v}^\top \xi_i}{\|\boldsymbol{v}\|_1}\right|^3 \\
&\le \lim_{n\to\infty} n^{-\frac{1}{2}} \Big(\frac{s}{c_{\min}}\Big)^{\frac{3}{2}} \max(\xi_i) \\
&= 0,
\end{aligned}$$

where the first inequality comes from the sparse eigenvalue condition on H^* with sparse $\boldsymbol{v}$; the second inequality comes from Cauchy-Schwarz inequality and $\|\boldsymbol{v}\|_0 \le s$.

Now since the Lyapunov condition is satisfied, we can apply the central limit theorem and obtain

$$\frac{1}{\sqrt{n}} \frac{\sum_i \boldsymbol{v}^\top \xi_i}{\sqrt{\boldsymbol{v}^\top H^* \boldsymbol{v}}} \xrightarrow{d} N(0,1),$$

which is just

$$\frac{\sqrt{n}\boldsymbol{v}^\top \nabla \ell(\beta^*)}{\sqrt{\boldsymbol{v}^\top H^* \boldsymbol{v}}} \xrightarrow{d} N(0,1).$$

□

Lemma 32. *Suppose all the conditions in Section 3.3.1 are satisfied, we have*

$$\|\nabla \ell(\boldsymbol{\beta}^*)\|_\infty = \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{\log p}{n}}\Big). \tag{3.19}$$

Proof. Each element $\big[\nabla \ell(\boldsymbol{\beta}^*)\big]_j$ is the average over n terms with absolute value bounded by a. According to Hoeffding's inequality [146] we have

$$P\Big(\big|\big[\nabla \ell(\boldsymbol{\beta}^*)\big]_j\big| \geq t\Big) \leq 2e^{-\frac{nt^2}{2a^2}}.$$

Apply union bound and let $t = C\sqrt{\frac{\log p}{n}}$ we have

$$P\Bigg(\|\nabla \ell(\boldsymbol{\beta}^*)\|_\infty > C\sqrt{\frac{\log p}{n}}\Bigg) \leq p \cdot P\Bigg(\big|\big[\ell(\boldsymbol{\beta}^*)\big]_j\big| \geq C\sqrt{\frac{\log p}{n}}\Bigg) \leq p \cdot 2e^{-\frac{C^2 \log p}{2a^2}} \leq 2p^{1-\frac{C^2}{2a^2}}.$$

We can take large enough C so that (3.19) holds with high probability. □

Lemma 33. *Suppose all the conditions in Section 3.3.1 are satisfied, we have*

$$\Big\|\nabla^2 \ell(\boldsymbol{\beta}^*) - H^*\Big\|_\infty = \mathcal{O}_{\mathbb{P}}\Big(\sqrt{\frac{\log p}{n}}\Big). \tag{3.20}$$

Proof. By Hoeffding's inequality again we have

$$P\Bigg(\big|\nabla^2_{jk} \ell(\boldsymbol{\beta}^*) - H^*_{jk}\big| \geq C\sqrt{\frac{\log p}{n}}\Bigg) \leq 2\exp\Big\{-\frac{2n^2C^2\frac{\log p}{n}}{4na^2}\Big\} \leq p^{-\frac{C^2}{2a^2}}.$$

Apply union bound we have

$$P\Big(\Big\|\nabla^2\ell(\boldsymbol{\beta}^*) - H^*\Big\|_\infty \geq C\sqrt{\frac{\log p}{n}}\Big) \leq \sum_{j,k=1\dots p} P\Big(\Big|\nabla^2_{jk}\ell(\boldsymbol{\beta}^*) - H^*_{jk}\Big| \geq C\sqrt{\frac{\log p}{n}}\Big) \leq 2p^{2-\frac{C^2}{2a^2}}.$$

We can take large enough C so that (3.20) holds with high probability.

□

Lemma 34. *Suppose all the conditions in Section 3.3.1 are satisfied, for any* $\widetilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^* + u(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)$ *with* $u \in [0,1]$ *we have*

$$\|\nabla^2\ell(\widetilde{\boldsymbol{\beta}})\|_\infty = \mathcal{O}_{\mathbb{P}}(1),$$
$$\|\nabla^2\ell(\widetilde{\boldsymbol{\beta}}) - H^*\|_\infty = \mathcal{O}_{\mathbb{P}}\Big(s\sqrt{\frac{\log p}{n}}\Big).$$

Proof. From the definition we know $\widetilde{\boldsymbol{\beta}}$ is of the same order with $\boldsymbol{\beta}^*$ and $\widehat{\boldsymbol{\beta}}$. The first claim comes from the second claim and the condition $\|H^*\|_\infty = \mathcal{O}(1)$. For the second claim, we have,

$$\|\nabla^2\ell(\widetilde{\boldsymbol{\beta}}) - H^*\|_\infty \leq \|\nabla^2\ell(\widetilde{\boldsymbol{\beta}}) - \nabla^2\ell(\boldsymbol{\beta}^*)\|_\infty + \|\nabla^2\ell(\boldsymbol{\beta}^*) - H^*\|_\infty. \tag{3.21}$$

For the first term in (3.21), according to Smooth Hessian Condition and Estimation Accuracy Condition we have

$$\|\nabla^2\ell(\widetilde{\boldsymbol{\beta}}) - \nabla^2\ell(\boldsymbol{\beta}^*)\|_\infty \leq L \cdot \|\widetilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_1 = \mathcal{O}\Big(s\sqrt{\frac{\log p}{n}}\Big).$$

For the second term in (3.21), by Lemma 33 it is $\mathcal{O}_{\mathbb{P}}\big(\sqrt{\frac{\log p}{n}}\big)$. Combining this two terms we get our desired result.

□

Lemma 35. *Suppose all the conditions in Section 3.3.1 are satisfied, we have*

$$\|\nabla^2_{\alpha\boldsymbol{\theta}}\ell(\widehat{\boldsymbol{\beta}}) - \boldsymbol{w}^{*T}\nabla^2_{\boldsymbol{\theta\theta}}\ell(\widehat{\boldsymbol{\beta}})\|_\infty = \mathcal{O}_{\mathbb{P}}\Big(s^2\sqrt{\frac{\log p}{n}}\Big). \tag{3.22}$$

Proof. By triangle inequality we have

$$\begin{aligned}\|\nabla^2_{\alpha\boldsymbol{\theta}}\ell(\widehat{\boldsymbol{\beta}}) - \boldsymbol{w}^{*T}\nabla^2_{\boldsymbol{\theta\theta}}\ell(\widehat{\boldsymbol{\beta}})\|_\infty \le \|H^*_{\alpha\boldsymbol{\theta}} - \boldsymbol{w}^{*T}H^*_{\boldsymbol{\theta\theta}}\|_\infty + \|\nabla^2_{\alpha\boldsymbol{\theta}}\ell(\widehat{\boldsymbol{\beta}}) - H^*_{\alpha\boldsymbol{\theta}}\|_\infty \\ + \|\boldsymbol{w}^{*T}\{H^*_{\boldsymbol{\theta\theta}} - \nabla^2_{\boldsymbol{\theta\theta}}\ell(\widehat{\boldsymbol{\beta}})\}\|_\infty.\end{aligned}$$

The first term is 0 by definition. The second term is $\mathcal{O}_{\mathbb{P}}\big(s\sqrt{\frac{\log p}{n}}\big)$ according to Lemma 34. The third term is $\mathcal{O}_{\mathbb{P}}\Big(s^2\sqrt{\frac{\log p}{n}}\Big)$ according to Lemma 34 and the sparse condition $\|\boldsymbol{w}^*\|_1 = s$. Combining these three terms we get our desired result.

□

Lemma 36. *Suppose all the conditions in Section 3.3.1 are satisfied, we have*

$$\|\widehat{\boldsymbol{w}} - \boldsymbol{w}^*\|_1 = \mathcal{O}_{\mathbb{P}}(\lambda' s) \ \textit{and} \ \|\widehat{\boldsymbol{w}} - \boldsymbol{w}^*\|_2 = \mathcal{O}_{\mathbb{P}}(\lambda'\sqrt{s}).$$

Proof. By definition we know $\widehat{\boldsymbol{w}}$ satisfies

$$\|\nabla^2_{\alpha\boldsymbol{\theta}}\ell(\widehat{\boldsymbol{\beta}}) - \widehat{\boldsymbol{w}}^T\nabla^2_{\boldsymbol{\theta\theta}}\ell(\widehat{\boldsymbol{\beta}})\|_\infty \le \lambda'.$$

Define $\boldsymbol{\delta} = \widehat{\boldsymbol{w}} - \boldsymbol{w}^*$, according to (3.22) we have

$$\|\nabla^2_{\boldsymbol{\theta\theta}}\ell(\widehat{\boldsymbol{\beta}})\cdot\boldsymbol{\delta}\|_\infty \le \|\nabla^2_{\alpha\boldsymbol{\theta}}\ell(\widehat{\boldsymbol{\beta}}) - \widehat{\boldsymbol{w}}^T\nabla^2_{\boldsymbol{\theta\theta}}\ell(\widehat{\boldsymbol{\beta}})\|_\infty + \|\nabla^2_{\alpha\boldsymbol{\theta}}\ell(\widehat{\boldsymbol{\beta}}) - \boldsymbol{w}^{*T}\nabla^2_{\boldsymbol{\theta\theta}}\ell(\widehat{\boldsymbol{\beta}})\|_\infty \le C\lambda',$$

for some constant C. Therefore we have

$$\boldsymbol{\delta}^\top\cdot\nabla^2_{\boldsymbol{\theta\theta}}\ell(\widehat{\boldsymbol{\beta}})\cdot\boldsymbol{\delta} \le \|\boldsymbol{\delta}\|_1\cdot\|\nabla^2_{\boldsymbol{\theta\theta}}\ell(\widehat{\boldsymbol{\beta}})\cdot\boldsymbol{\delta}\|_\infty \le C\lambda'\|\boldsymbol{\delta}\|_1. \tag{3.23}$$

Following Lemma 3 in [345] we know $\|\widehat{\boldsymbol{w}}\|_0 = cs$ for some constant c. By Sparse Eigenvalue Condition, we have

$$\boldsymbol{\delta}^\top \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \ell(\widehat{\boldsymbol{\beta}}) \boldsymbol{\delta} \geq c_{\min} \|\boldsymbol{\delta}\|_2^2. \tag{3.24}$$

Plug into (3.23) we obtain

$$C\lambda' \|\boldsymbol{\delta}\|_1 \geq c_{\min} \|\boldsymbol{\delta}\|_2^2 \geq c_{\min} \|\boldsymbol{\delta}\|_1^2 \cdot \frac{1}{s},$$

which gives

$$\|\boldsymbol{\delta}\|_1 \leq \frac{C\lambda' s}{c_{\min}} = \mathcal{O}_{\mathbb{P}}(\lambda' s),$$

and also

$$\|\boldsymbol{\delta}\|_2 = \mathcal{O}_{\mathbb{P}}(\lambda' \sqrt{s}),$$

□

Remark 37. *We show that Restricted Eigenvalue Condition also works here, as discussed in Remark 22. According to the optimality condition of Dantzig selector we have* $\|\widehat{\boldsymbol{w}}\|_1 \leq \|\boldsymbol{w}^*\|_1$. *Also note that since* $\|\boldsymbol{w}^*_{\mathcal{S}^c}\|_1 = 0$ *we have*

$$\|\widehat{\boldsymbol{w}}_{\mathcal{S}}\|_1 + \|\widehat{\boldsymbol{w}}_{\mathcal{S}^c}\|_1 \leq \|\boldsymbol{w}^*_{\mathcal{S}}\|_1.$$

By triangle inequality we have

$$\|\boldsymbol{w}^*_{\mathcal{S}}\|_1 \leq \|\widehat{\boldsymbol{w}}_{\mathcal{S}}\|_1 + \|\boldsymbol{\delta}_{\mathcal{S}}\|_1.$$

Summing up these two inequalities we obtain

$$\|\boldsymbol{\delta}_{\mathcal{S}^c}\|_1 \leq \|\boldsymbol{\delta}_{\mathcal{S}}\|_1,$$

which means $\boldsymbol{\delta} \in \mathcal{C}(\mathcal{S})$. *Therefore with Restricted Eigenvalue Condition we can still get* (3.24) *and everything follows.*

Moreover, in the proof of Lemma 25, since we take $\boldsymbol{v} = (1; -\boldsymbol{w}^*)$*, clearly we have* $\boldsymbol{v} \in \mathcal{C}(\mathcal{S})$*. Therefore the proof of Lemma 31 also hold under Restricted Eigenvalue Condition. Combining these two results we see that Restricted Eigenvalue Condition also suffices for our algorithm to be valid.*

CHAPTER 4

RECOVERY OF SIMULTANEOUS LOW RANK AND TWO-WAY SPARSE COEFFICIENT MATRICES, A NONCONVEX APPROACH

4.1 Introduction

Many problems in machine learning, statistics and signal processing can be formulated as optimization problems with a smooth objective and nonconvex constraints. The objective usually measures the fit of a model, parameter, or signal to the data, while the constraints encode structural requirements on the model. Examples of nonconvex constraints include sparsity where the parameter is assumed to have only a few non-zero coordinates [151, 272, 307, 372], group sparsity where the parameter is comprised of several groups only few of which are non-zero [212, 175, 153, 70], and low-rankness where the parameter is believed to be a linear combination of few factors [12, 71, 78, 124, 159]. Common approach to dealing with nonconvex constraints is via convex relaxations, which allow for application of simple optimization algorithms and easy theoretical analysis [4, 62, 104, 61, 184]. From a practical point of view, it has been observed that directly working with a nonconvex optimization problem can lead to both faster and more accurate algorithms [290, 367, 348, 322]. As a result, a body of literature has recently emerged that tries to characterize good performance of these algorithms [26, 365, 127].

In this work, we focus on the following optimization problem

$$\widehat{\Theta} \in \arg\min_{\Theta \in \Xi} f(\Theta) \tag{4.1}$$

where $\Xi \subset \mathbb{R}^{m_1 \times m_2}$ is a nonconvex set comprising of low rank matrices that are also row and/or column sparse,

$$\Xi = \Xi(r, s_1, s_2) = \{\Theta \in \mathbb{R}^{m_1 \times m_2} \mid \operatorname{rank}(\Theta) \leq r, \|\Theta\|_{2,0} \leq s_1, \|\Theta^\top\|_{2,0} \leq s_2\},$$

where $\|\Theta\|_{2,0} = |\{i \in [m_1] \mid \sum_{j \in [m_2]} \Theta_{ij}^2 \neq 0\}|$ is the number of non-zero rows of Θ. Such an optimization problem arises in a number of applications including sparse singular value decomposition and principal component analysis [322, 217, 141], sparse reduced-rank regression [47, 218, 71, 72, 313], and reinforcement learning [57, 294, 191, 327, 285]. Rather than considering convex relaxations of the optimization problem (4.1), we directly work with a nonconvex formulation. Under an appropriate statistical model, the global minimizer $\widehat{\Theta}$ approximates the "true" parameter Θ^* with an error level ϵ. Since the optimization problem (4.1) is highly nonconvex, our aim is to develop an iterative algorithm that, with appropriate initialization, converges linearly to a stationary point $\breve{\Theta}$ that is within $c \cdot \epsilon$ distance of $\widehat{\Theta}$. In order to develop a computationally efficient algorithm, we reparametrize the $m_1 \times m_2$ matrix variable Θ as $UV^\top$ with $U \in \mathbb{R}^{m_1 \times r}$ and $V \in \mathbb{R}^{m_2 \times r}$, and optimize over U and V. That is, we consider (with some abuse of notation) the following optimization problem

$$(\widehat{U}, \widehat{V}) \in \arg \min_{U \in \mathcal{U}, V \in \mathcal{V}} f(U, V), \tag{4.2}$$

where

$$\mathcal{U} = \mathcal{U}(s_1) = \left\{U \in \mathbb{R}^{m_1 \times r} \mid \|U\|_{2,0} \leq s_1\right\}$$

and

$$\mathcal{V} = \mathcal{V}(s_2) = \left\{V \in \mathbb{R}^{m_2 \times r} \mid \|V\|_{2,0} \leq s_2\right\}.$$

Such a reparametrization automatically enforces the low rank structure and will allow us to develop an algorithm with low computational cost per iteration. Note that even though $\widehat{U}$ and $\widehat{V}$ are only unique up to scaling and a rotation by an orthogonal matrix, $\widehat{\Theta} = \widehat{U}\widehat{V}^\top$ is usually unique.

We make several contributions in this work. First, we develop an efficient algorithm for minimizing (4.2), which uses projected gradient descent on a nonconvex set in each iteration. Under conditions on the function $f(\Theta)$ that are common in the high-dimensional literature, we establish linear convergence of the iterates to a statistically relevant solution. In particular, we require that the function $f(\Theta)$ satisfies restricted strong convexity (RSC) and restricted strong smoothness (RSS), conditions that are given in Condition (**RSC/RSS**) below. Compared to the existing work for optimization over low rank matrices with (alternating) gradient descent, we need to study a projection onto a nonconvex set in each iteration, which in our case is a hard-thresholding operation, that requires delicate analysis and novel theory. Our second contribution, is in the domain of multi-task learning. Multi-task learning is a widely used learning framework where similar tasks are considered jointly for the purpose of improving performance compared to learning the tasks separately [66]. We study the setting where the number of input variables and the number of tasks can be much larger than the sample size (see [218] and references there in). Our focus is on simultaneous variable selection and dimensionality reduction. We want to identify which variables are relevant predictor variables for different tasks and at the same time we want to combine the relevant predictor variables into fewer features that can be explained as latent factors that drive the variation in the multiple responses. We provide a new algorithm for this problem and improve the theoretical results established in [218]. In particular, our algorithm does not require a new independent sample in each iteration and allows for non-Gaussian errors, while at the same time achieves nearly optimal error rate compared to the information theoretic minimax lower bound for the problem. Moreover, our prediction error is much better than the error bound proposed in [47], and matches the error bound in [272]. However, all of the existing algorithms are slow and cannot scale to high dimensions. Finally, our third contribution is in the area of reinforcement learning. We study the Multi-task Reinforcement Learning (MTRL) problem via value function approximation. In MTRL the decision maker needs to solve a sequence of Markov Decision Processes (MDPs). A common approach to Reinforcement Learning when the state space is large is to approximate the value function of linear basis functions (linear in some appropriate feature representation of the states) with sparse support. Thus, it is natural to assume the resulting coefficient matrix is low rank and row sparse. Our proposed algorithm can be applied to the regression step of any MTRL algorithm (we chose Fitted Q-iteration (FQI) for presentation purposes) to solve for the optimal policies for MDPs. Compared to [57] which uses convex relaxation, our algorithm is much more efficient in high dimensions.

4.1.1 Related Work

Our work contributes to several different areas, and thus is naturally related to many existing works. We provide a brief overview of the related literature and describe how it is related to our contributions. For the sake of brevity, we do not provide an extensive review of the existing literature.

Low-rank Matrix Recovery. A large body of literature exists on recovery of low-rank matrices as they arise in a wide variety of applications throughout science and engineering, ranging from quantum tomography to signal processing and machine learning [1, 210, 287, 88]. Recovery of a low-rank matrix can be formulated as the following optimization problem

$$\widehat{\Theta} \in \arg \min_{\Theta \in \mathbb{R}^{m_1 \times m_2}} f(\Theta) \quad \text{subject to } \operatorname{rank}(\Theta) \leq r, \tag{4.3}$$

where the objective function $f : \mathbb{R}^{m_1 \times m_2} \mapsto \mathbb{R}$ is convex and smooth. The problem (4.3) is highly nonconvex and NP-hard in general [104, 103]. A lot of the progress in the literature has focused on convex relaxations where one replaces the rank constraint using the nuclear norm. See, for example, [61, 62, 60, 256, 52, 254, 124, 67, 151, 263, 184, 136, 230, 70, 329, 231, 4, 257, 75, 76, 77, 141, 56, 333, 372, 323, 346, 358, 352] and references therein. However, developing efficient algorithms for solving these convex relaxations is challenging in regimes with large m_1 and m_2 [150]. A practical approach, widely used in large scale applications such as recommendation systems or collaborative filtering [297, 186, 115, 375] relies on solving a nonconvex optimization problem where the decision variable Θ is factored as $UV^\top$, usually referred to as the Burer-Monteiro type decomposition [48, 49]. A stationary point of this nonconvex problem is usually found via a block coordinate descent-type algorithm, such as alternating minimization or (alternating) gradient descent. Unlike for the convex relaxation approaches, the theoretical understanding of these nonconvex optimization procedures has been developed only recently [172, 173, 159, 138, 140, 139, 290, 367, 74, 226, 101, 368, 36, 35, 305, 78, 373, 69, 374, 114, 209, 202, 221, 128]. Compared to the classical nonconvex optimization theory, which only shows a sublinear convergence to a local optima, the focus of the recent literature is on establishing linear rates of convergence or characterizing that the objective does not have spurious local minima. In addition to the methods that work on the factorized form, [157, 194, 158, 26, 91] consider projected gradient-type methods which optimize over the matrix variable $\Theta \in \mathbb{R}^{m_1 \times m_2}$. These methods involve calculating the top r singular vectors of an $m_1 \times m_2$ matrix at each iteration. When r is much smaller than m_1 and m_2, they incur much higher computational cost per iteration than the methods that optimize over $U \in \mathbb{R}^{m_1 \times r}$ and $V \in \mathbb{R}^{m_2 \times r}$.

Our work contributes to this body of literature by studying gradient descent with a projection step on a non-convex set, which requires hard-thresholding. Hard-thresholding in this context has not been considered before. Theoretically we need a new argument to establish linear convergence to a statistically relevant point. [78] considered projected gradient descent in a symmetric and positive semidefinite setting with a projection on a convex set. Our work is most closely related to [367], which used the notion of inexact first order oracle to establish their results, but did not consider the hard-thresholding step.

Structured Low-rank Matrices. Low-rank matrices with additional structure also commonly arise in different problems ranging from sparse principal component analysis (PCA) and sparse singular value decomposition to multi-task learning. In a high-dimensional setting, the classical PCA is inconsistent [168] and recent work has focused on PCA with additional sparse structure on the eigenvectors [11, 32, 39, 55, 317, 219, 361]. Similar sparse structure in singular vectors arises in sparse SVD and biclustering [195, 71, 217, 308, 334, 169, 22, 182, 23]. While the above papers use the sparsity structure of the eigenvectors and singular vectors, it is also possible to have simultaneous low rank and sparse structure directly on the matrix Θ. Such a structure arises in multi-task learning, covariance estimation, graph denoising and link prediction [220, 260]. Additional structure on the sparsity pattern was imposed in the context of sparse rank-reduced regression, which is an instance of multi-task learning [72, 47, 218, 20, 272]. Our algorithm described in Section 4.2 can be applied to the above mentioned problems. In Section 4.4, we theoretically study multi-task learning in the setting of [218]. We relax conditions imposed in [218], specifically allowing for non-Gaussian errors and not requiring independent samples at each step of the algorithm, while still achieving the near minimax rate of convergence. We provide additional discussion in Section 4.4 after formally providing results for the multi-task learning setting. In Section 4.5, we further corroborate our theoretical results in extensive simulations and show that our algorithm outperforms existing methods in multi-task learning.

Low-rank Plus Sparse Matrix Recovery. At this point, it is worth mentioning another commonly encountered structure on the decision variable Θ that we do not study in the current work. In various applications it is common to model Θ as a sum of two matrices, one of which is low-rank and the other one sparse. Applications include robust PCA, latent Gaussian graphical models, factor analysis and multi-task learning [59, 151, 67, 77, 4, 125, 365, 330, 127, 131, 64]. While Burer-Monteiro factorization has been considered for the low-rank component in this context (see, for example, [365] and references therein), the low-rank component is dense as it needs to be incoherent. The incoherence assumption guarantees that the low-rank component is not too spiky and can be identified [61]. An alternative approach was taken in [127] where alternating minimization over the low-rank and sparse component with a projection on a nonconvex set was investigated.

4.1.2 Organization of the chapter

In Section 4.2 we provide details for our proposed algorithm. Section 4.3 states our assumptions and the theoretical result with a proof sketch. Section 4.4 shows applications to multi-task learning, while Section 4.5 presents experimental results. Section 4.6 provides detailed technical proofs. Conclusion is given in Section 4.7.

4.2 Gradient Descent With Hard Thresholding

In this section, we detail our proposed algorithm, which is based on gradient descent with hard thresholding (GDT). Our focus is on developing an efficient algorithm for minimizing $f(\Theta)$ with $\Theta \in \Xi$. In statistical estimation and machine learning a common goal is to find Θ^*, which is an (approximate) minimizer of $\mathbb{E}[f(\Theta)]$ where the expectation is with respect to randomness in data. In many settings, the global minimizer of (4.1) can be shown to approximate Θ^* up to statistical error, which is problem specific. In Section 4.3, we will show that iterates of our algorithm converge linearly to Θ^* up to a statistical error. It is worth noting that an argument similar to that in the proof of Theorem 38 can be used to establish linear convergence to the global minimizer $\widehat{\Theta}$ in a deterministic setting. That is, suppose $(\widehat{U}, \widehat{V})$ is a global minimizer of the problem (4.2) and $\widehat{\Theta} = \widehat{U}\widehat{V}^\top$. Then as long as the conditions in Section 4.3 hold for $\widehat{U}, \widehat{V}$ in place of U^*, V^*, we can show linear convergence to $\widehat{\Theta}$ up to an error level defined by the gradient of the objective function at $\widehat{\Theta}$. See the discussion after Theorem 38.

Our algorithm, GDT, uses a Burer-Monteiro factorization to write $\Theta = UV^\top$, where $U \in \mathbb{R}^{m_1 \times r}$ and $V \in \mathbb{R}^{m_2 \times r}$, and minimizes

$$(\widehat{U}, \widehat{V}) \in \arg \min_{U \in \mathcal{U}, V \in \mathcal{V}} f(U, V) + g(U, V), \tag{4.4}$$

where $g(U, V)$ is the penalty function defined as

$$g(U, V) = \frac{1}{4}\|U^\top U - V^\top V\|_F^2.$$

The role of the penalty is to find a balanced decomposition of $\widehat{\Theta}$, one for which $\sigma_i(\widehat{U}) = \sigma_i(\widehat{V})$, $i = 1, \dots, r$ [374, 365]. Note the value of the penalty is equal to 0 for a balanced solution, so we can think of the penalized objective as looking through minimizer of (4.2) for a one that satisfies $\widehat{U}^\top\widehat{U} - \widehat{V}^\top\widehat{V} = 0$. In particular, adding the penalty function g does not change the minimizer of f over Ξ. The convergence rate of GDT depends on the condition number of (U^*, V^*), the point algorithm converges to. The penalty ensures that the iterates U, V are not ill-conditioned. Gradient descent with hard-thresholding on U and V is used to minimize (4.4). Details of GDT are given in Algorithm 3. The algorithm takes as input parameters η, the step size; s_1, s_2, the sparsity level; T, the number of iterations; and a starting point Θ^0.

The choice of starting point Θ^0 is very important as the algorithm performs a local search in its neighborhood. In Section 4.3 we will formalize how close Θ^0 needs to be to Θ^*, while in Section 4.4 we provide a concrete way to initialize under a multi-task learning model. In general, we envisage finding Θ^0 by solving the following optimization problem

$$\Theta^0 = \arg \min_{\Theta \in \mathbb{R}^{m_1 \times m_2}} f(\Theta) + \text{pen}(\Theta), \tag{4.5}$$

where $\text{pen}(\Theta)$ is a (simple) convex penalty term making the objective (4.5) a convex optimization problem. For example, we could use the vector ℓ_1 norm, $\text{pen}(\Theta) = \|\Theta\|_1$. The choice of penalty $\text{pen}(\Theta)$ should be such that solving the optimization problem in (4.5) can be done efficiently in a high dimensional setting. In practice, if solving the convex relaxation is slow, we can start from the all zero matrix and perform several (proximal) gradient steps to get an appropriate initialization. See for example [365]. Once an initial estimate Θ^0 is obtained, we find the best rank r approximation $\widetilde{\Theta} = \widetilde{U}\widetilde{\Sigma}\widetilde{V}^\top$ to Θ^0 and use it to obtain the initial iterates U^0 and V^0. In each step, GDT updates U and V by taking a gradient step and hard-thresholding the result. The operation $\text{Hard}(U, s)$ keeps s rows of U with the largest ℓ_2 row-norm, while setting to zero other rows.

Suppose that the target statistical parameter Θ^* is in $\Xi(r^*, s_1^*, s_2^*)$. The sparsity level s_1^* and s_2^* as well as the rank r^* are not known in practice, but are needed in Algorithm 3. For the convergence proof we require that the input parameters to the algorithm are set as $s_1 = c \cdot s_1^*$ and $s_2 = c \cdot s_2^*$ for some $c > 1$. From simulations, we observe that the estimation accuracy is not very sensitive to the choice of s_1 and s_2 as long as they are chosen greater than the true values s_1^* and s_2^*. This suggests that in practice, we could set s_1 and s_2 to be reasonably large values whenever a reasonable guess of the sparsity level is available, as incorrectly omitting nonzero value (false negative) is more troublesome than including one zero value (false positive). Alternatively, as we do in simulations, we can use a validation set or an information criteria to select these tuning parameters. However, it is noted in [273] that conventional cross validation may select an inconsistent model, especially when using a non-convex penalty. As an improvement, we can adopt the techniques in [272], which develops the scale-free predictive information criterion to select the best sparsity parameters. Also, [273] proposes structural cross validation method that can achieve the minimax optimal error rate.

Following the same guideline as in the literature, in our analysis we assume that we are using the true rank $r = r^*$. In practice, the rank r can be estimated as in [46], which guarantees consistent rank estimation with high probability. Although [46] considers low-rank structure without sparsity, in practice, it still provides a reasonable rank estimator. The usage of [46] in a low-rank and sparse model is also suggested in [218]. The performance of the GDT algorithm is robust to the choice of rank r, as we will demonstrate through extensive experiments in Section 4.5. Finally, we remark that a joint tuning scheme for the rank and sparsity parameters can also be considered.

To the best of our knowledge, GDT is the first gradient based algorithm to deal with a nonconvex optimization problem over a parameter space that is simultaneously low rank

Algorithm 3 Gradient Descent with Hard Thresholding (GDT)

```
Input: Initial estimate $\Theta^0$
Parameters: Step size $\eta$, Rank $r$, Sparsity level $s_1, s_2$, Total number of iterations $T$
$(\widetilde{U}, \widetilde{\Sigma}, \widetilde{V}) =$ rank $r$ SVD of $\Theta^0$
$U^0 = \text{Hard}(\widetilde{U}(\widetilde{\Sigma})^{\frac{1}{2}}, s_1), V^0 = \text{Hard}(\widetilde{V}(\widetilde{\Sigma})^{\frac{1}{2}}, s_2)$
for $t = 1$ to $T$ do
    $V^{t+0.5} = V^t - \eta\nabla_V f(U^t, V^t) - \eta\nabla_V g(U^t, V^t)$,
    $V^{t+1} = \text{Hard}(V^{t+0.5}, s_2)$
    $U^{t+0.5} = U^t - \eta\nabla_U f(U^t, V^t) - \eta\nabla_U g(U^t, V^t)$,
    $U^{t+1} = \text{Hard}(U^{t+0.5}, s_1)$
end for
Output: $\Theta^T = U^T(V^T)^\top$
```

and row and column sparse. In the following section we will provide conditions on the objective function f and the starting point Θ^0 which guarantee linear convergence to Θ^* up to a statistical error. As an application, we consider the multi-task learning problem in Section 4.4. We show that the statistical error nearly matches the optimal minimax rate, while the algorithm achieves the best performance in terms of estimation and prediction error in simulations.

4.3 Theoretical Result

In this section, we formalize the conditions and state the main result on the linear convergence of our algorithm. We begin in Section 4.3.1 by stating the conditions on the objective function f and initialization that are needed for our analysis. In Section 4.3.2, we state Theorem 38 that guarantees linear convergence under the conditions to a statistically useful point. The proof outline is given in Section 4.3.3. In Section 4.4 to follow, we derive results for multi-task learning as corollaries of our main result.

4.3.1 Regularity Conditions

We start by stating mild conditions on the objective function f, which have been used in the literature on high-dimensional estimation and nonconvex optimization, and they hold with high-probability for a number of statistical models of interest [367, 365, 127]. Note that all the conditions depend on the choice of s_1 and s_2 (or equivalently, on c).

For $\Theta^* \in \Xi(r^*, s_1^*, s_2^*)$, let $\Theta^* = U_{\Theta^*}\Sigma_{\Theta^*}V_{\Theta^*}^\top$ be its singular value decomposition. Let $U^* = U_{\Theta^*}\Sigma_{\Theta^*}^{1/2}$ and $V^* = V_{\Theta^*}\Sigma_{\Theta^*}^{1/2}$ be the balanced decomposition of $\Theta^* = U^*V^{*\top}$. Note that the decomposition is not unique as $\Theta^* = (U^*O)(V^*O)^\top$ for any orthogonal matrix $O \in \mathcal{O}(r)$. Let $\sigma_1(\Theta^*) = \sigma_{\max}(\Theta^*)$ and $\sigma_r(\Theta^*) = \sigma_{\min}(\Theta^*)$ denote the maximum and minimum nonzero singular values of Θ^* with $r = r^*$. The first condition is Restricted Strong Convexity and Smoothness on f.

Restricted Strong Convexity and Smoothness (RSC/RSS). There exist universal constants μ and L such that

$$\frac{\mu}{2}\|\Theta_2 - \Theta_1\|_F^2 \le f(\Theta_2) - f(\Theta_1) - \langle \nabla f(\Theta_1), \Theta_2 - \Theta_1\rangle \le \frac{L}{2}\|\Theta_2 - \Theta_1\|_F^2$$

for all $\Theta_1, \Theta_2 \in \Xi(2r, \widetilde{s}_1, \widetilde{s}_2)$ where $\widetilde{s}_1 = (2c+1)s_1^*$ and $\widetilde{s}_2 = (2c+1)s_2^*$.

The next condition is on the initial estimate Θ^0. It quantifies how close the initial estimator needs to be to Θ^* so that iterates of GDT converge to statistically useful solution.

Initialization (I). Define $\mu_{\min} = \frac{1}{8}\min\{1, \frac{\mu L}{\mu+L}\}$ and

$$I_0 = \frac{4}{5}\mu_{\min}\sigma_r(\Theta^*) \cdot \min\left\{\frac{1}{\mu+L}, 2\right\}. \tag{4.6}$$

We require

$$\|\Theta^0 - \Theta^*\|_F \le \frac{1}{5}\min\left\{\sigma_r(\Theta^*), \frac{I_0}{\xi}\sqrt{\sigma_r(\Theta^*)}\right\}, \tag{4.7}$$

where $\xi^2 = 1 + \frac{2}{\sqrt{c-1}}$.

We note that, in general, (4.7) defines a ball of constant radius around Θ^* in which the initial estimator needs to fall into. In particular, when considering statistical learning problems, the initial estimator can be inconsistent as the sample size increases.

Next, we define the notion of the statistical error,

$$e_{\text{stat}} = \sup_{\substack{\Delta \in \Xi(2r, \widetilde{s}_1, \widetilde{s}_2) \\ \|\Delta\|_F \leq 1}} \langle \nabla f(\Theta^*), \Delta \rangle. \tag{4.8}$$

Note that the statistical error quantifies how large the gradient of the objective evaluated at the true parameter Θ^* can be in the directions of simultaneously low-rank and sparse matrices. It implicitly depends on the choice of c and as we will see later there is a trade-off in balancing the statistical error and convergence rate of GDT. As c increases, statistical error gets larger, but requires us to choose a smaller step size in order to guarantee convergence.

With these two conditions, we are ready to the choice of the step size in Algorithm 3.

Step Size Selection. Let $Z^0 = \begin{bmatrix} U^0 \\ V^0 \end{bmatrix}$. We choose the step size η to satisfy

$$\eta \leq \frac{1}{16\|Z^0\|_2^2} \cdot \min\left\{\frac{1}{2(\mu+L)}, 1\right\}, \tag{4.9}$$

Furthermore, we require η and c to satisfy

$$\beta = \xi^2 \left(1 - \eta \cdot \frac{2}{5}\mu_{\min}\sigma_r(\Theta^*)\right) < 1, \tag{4.10}$$

and

$$e_{\text{stat}}^2 \leq \frac{1-\beta}{\xi^2 \eta} \cdot \frac{L\mu}{L+\mu} \cdot I_0^2. \tag{4.11}$$

The condition that the step size η satisfies (4.9) is typical in the literature on convex optimization of strongly convex and smooth functions. Under (4.10) we will be able to show contraction after one iteration and progress towards Θ^*. The second term in (4.10) is always smaller than 1, while the first term ξ^2 is slightly larger than 1 and is the price we pay for the hard thresholding step. In order to show linear convergence we need to balance the choice of η and ξ^2 to ensure that $\beta < 1$. From (4.10), we see that if we select a small step size η, then we need to have a small ξ^2, which means a large c. Intuitively, if η is too small, it may be impossible to change row and column support in each iteration. In this case we have to keep many active rows and columns to make sure we do not miss the true signal. This leads to large s_1 and s_2, or equivalently to a large c. However, the statistical error (4.8) will increase with increase of c and these are the trade-off on the selection of η and c.

Finally, (4.11) guarantees that the iterates do not run outside of the initial ball given in (4.7). In case (4.11) is violated, then the initialization point Θ^0 is already a good enough estimate of Θ^*. Therefore, this requirement is not restrictive. In practice, we found that the selection of η and c is not restrictive and the convergence is guaranteed for a wide range of values of their values.

In order to satisfy these regularity conditions, we may need to choose a relatively large c. However, the condition on c is purely a technical conditions. To the best of our knowledge, all the literature on iterative hard thresholding requires some restrictive conditions on c. Without the hard thresholding step, we can guarantee contraction $\beta < 1$ after one step of the gradient descent. However, the hard thresholding step amplifies the estimation error and, therefore, we need a relatively large c to guarantee contraction. In theory, we require an upper bound on c that does not scale with n, p, or K. In practice, we do not know the true sparsity level s^* and choose s directly based on prior knowledge or select it via cross validation. Moreover, the step size η could be selected in a heuristic way when implementing the algorithm for specific applications. While the techniques needed to establish better theoretical control of the parameter c still require improvement, in practice, even with small values of c the method performs well. Experiments in Section 4.5 show that the tuning parameters can be chosen in way to yield good finite sample performance. In practice, selecting inappropriate model parameters or initialization may worsen the performance of the algorithm, resulting in possibly sublinear convergence rate.

4.3.2 Main Result

Our main result establishes linear convergence of GDT iterates to Θ^* up to statistical error. Since the factorization of Θ^* is not unique, we turn to measure the subspace distance of the iterates (U^t, V^t) to the balanced decomposition $U^*(V^*)^\top = \Theta^*$.

Subspace distance. Let $Z^* = \begin{bmatrix} U^* \\ V^* \end{bmatrix}$ where $\Theta^* = U^*V^{*\top}$ and $\sigma_i(U^*) = \sigma_i(V^*)$ for each $i = 1, ..., r$. Define the subspace distance between $Z = \begin{bmatrix} U \\ V \end{bmatrix}$ and $Z^* = \begin{bmatrix} U^* \\ V^* \end{bmatrix}$ as

$$d^2(Z, Z^*) = \min_{O \in \mathcal{O}(r)} \left\{ \|U - U^*O\|_F^2 + \|V - V^*O\|_F^2 \right\}.$$

With this, we are ready to state our main result.

Theorem 38. *Suppose the conditions* **(RSC/RSS)**, **(I)** *are satisfied and the step size* η *satisfies* (4.9) - (4.11)*. Then after* T *iterations of GDT (Algorithm 3), we have*

$$d^2(Z^T, Z^*) \le \beta^T \cdot d^2(Z^0, Z^*) + \frac{\xi^2\eta}{1-\beta} \cdot \frac{L+\mu}{L \cdot \mu} \cdot e_{\text{stat}}^2. \tag{4.12}$$

Furthermore, for $\Theta^T = U^T(V^T)^\top$ *we have*

$$\|\Theta^T - \Theta^*\|_F^2 \le 4\sigma_1(\Theta^*) \cdot \Big[\beta^T \cdot d^2(Z^0, Z^*) + \frac{\xi^2\eta}{1-\beta} \cdot \frac{L+\mu}{L \cdot \mu} \cdot e_{\text{stat}}^2\Big]. \tag{4.13}$$

The proof sketch of Theorem 38 is given in the following section. Conceptually, Theorem 38 provides a minimal set of conditions for convergence of GDT. The first term in equations (4.12) and (4.13) correspond to the optimization error, whereas the second term corresponds to the statistical error. These bounds show that the distance between the iterates and Θ^* drop exponentially up to the statistical limit e_{stat}, which is problem specific. In statistical learning problem, it commonly depends on the sample size and the signal-to-noise ratio of the problem.

Theorem 38 provides convergence in a statistical setting to the "true" parameter Θ^*. However, as mentioned in Section 4.2, Algorithm 3 and Theorem 38 can also be used to establish linear convergence to a global minimizer in a deterministic setting. Suppose $(\widehat{U}, \widehat{V}) \in \arg\min_{U \in \mathcal{U}, V \in \mathcal{V}} \{f(U, V)\}$ is a global minimizer and $\widehat{\Theta} = \widehat{U}\widehat{V}^\top$. Furthermore, assume that the conditions in Section 4.3.1 are satisfied with $\widehat{\Theta}$ in place of Θ^*. Then we have that the iterates $\{\Theta^t\}$ obtained by GDT converge linearly to a global minimum $\widehat{\Theta}$ up to the error $\widehat{e}_{\text{stat}}$ defined similar to (4.8) with $\widehat{\Theta}$ in place of Θ^*. This error comes from sparsity and hard thresholding. In particular, suppose there are no row or column sparsity constraints in the optimization problem (4.2), so that we do not have hard-thresholding steps in Algorithm 3. Then we have $\widehat{e}_{\text{stat}} = 0$, so that iterates $\{\Theta^t\}$ converge linearly to $\widehat{\Theta}$, recovering the result of [367].

4.3.3 Proof Sketch of Theorem 38

In this section we sketch the proof of our main result. The proof combines three lemmas. We first one quantify the accuracy of the initialization step. The following one quantifies the improvement in the accuracy by one step of GDT. The third lemma shows that the step size assumed in Theorem 38 satisfies conditions of the second lemma. Detailed proofs of these lemmas are relegated to Section 4.6.

Our first lemma quantifies the accuracy of the initialization step.

Lemma 39. *Suppose that the input to GDT, Θ^0, satisfies initialization condition* (4.7). *Then the initial iterates U^0 and V^0 obtained in lines* 3 *and* 4 *of Algorithm 3 satisfy*

$$d(Z^0, Z^*) \leq I_0, \tag{4.14}$$

where $Z^0 = \begin{bmatrix} U^0 \\ V^0 \end{bmatrix}$ and I_0 is defined in (4.6).

The proof of Lemma 39 is given in Section 4.6.1.

Lemma 40. *Suppose the conditions* **(RSC/RSS)***,* **(I)** *are satisfied. Assume that the point* $Z = \begin{bmatrix} U \\ V \end{bmatrix}$ *satisfies* $d(Z, Z^*) \leq I_0$*. Let* (U^+, V^+) *denote the next iterate obtained with Algorithm 3 with the step size* η *satisfying*

$$\eta \leq \frac{1}{8\|Z\|_2^2} \cdot \min\Big\{\frac{1}{2(\mu + L)}, 1\Big\}. \tag{4.15}$$

Then we have

$$d^2(Z^+, Z^*) \leq \xi^2\bigg[\Big(1 - \eta \cdot \frac{2}{5}\mu_{\min}\sigma_r(\Theta^*)\Big) \cdot d^2(Z, Z^*) + \eta \cdot \frac{L + \mu}{L \cdot \mu} \cdot e_{\text{stat}}^2\bigg], \tag{4.16}$$

where $\xi^2 = 1 + \frac{2}{\sqrt{c-1}}$*.*

The proof of Lemma 40 is given in Section 4.6.2.

Lemma 41. *Suppose* $Z = \begin{bmatrix} U \\ V \end{bmatrix}$ *satisfies* $d(Z, Z^*) \leq I_0$*. We have that the choice of step size* (4.9) *in Theorem 38 satisfies the condition* (4.15) *in Lemma 40.*

The proof of Lemma 41 is given in Section 4.6.3.

Combining the three results above, we can complete the proof of Theorem 38. Starting from initialization Θ^0 satisfying the initialization condition (4.7), Lemma 39 ensures that (4.14) is satisfied for Z^0 and Lemma 41 ensures that the choice of step size (4.9) satisfies the step size condition (4.15) in Lemma 40. We can then apply Lemma 40 and get the next iterate $Z^1 = Z^+$, which satisfies (4.16). Using the condition on statistical error (4.11), initialization (4.7), and a simple calculation, we can verify that Z^1 satisfies $d(Z^1, Z^*) \leq I_0$. Therefore we can apply Lemma 39, Lemma 40, and Lemma 41 repeatedly to obtain

$$d^2(Z^{t+1}, Z^*) \leq \beta \cdot d^2(Z^t, Z^*) + \xi^2\eta \cdot \frac{L + \mu}{L \cdot \mu} \cdot e_{\text{stat}}^2,$$

for each $t = 0, 1, ..., T$. We then have

$$d^2(Z^T, Z^*) \le \beta^T \cdot d^2(Z^0, Z^*) + \frac{\xi^2 \eta}{1-\beta} \cdot \frac{L+\mu}{L \cdot \mu} \cdot e_{\text{stat}}^2.$$

Finally, for $\Theta^T = U^T (V^T)^\top$, let $O^T \in \mathcal{O}(r)$ be such that

$$d^2(Z^T, Z^*) = \|U^T - U^* O^T\|_F^2 + \|V^T - V^* O^T\|_F^2.$$

We have

$$\begin{aligned}
\|\Theta^T - \Theta^*\|_F^2 &= \|U^T (V^T)^\top - U^* O^T (V^* O^T)^\top\|_F^2 \\
&\le \Big[\|U^T\|_2 \|V^T - V^* O^T\|_F + \|V^*\|_2 \|U^T - U^* O^T\|_F\Big]^2 \\
&\le \|U^T\|_2^2 \|V^T - V^* O^T\|_F^2 + \|V^*\|_2^2 \|U^T - U^* O^T\|_F^2 \\
&\le 2\|Z^*\|_2^2 \cdot d^2(Z^T, Z^*) \\
&\le 4\sigma_1(\Theta^*) \cdot \Big[\beta^T \cdot d^2(Z^0, Z^*) + \frac{\xi^2 \eta}{1-\beta} \cdot \frac{L+\mu}{L \cdot \mu} \cdot e_{\text{stat}}^2\Big],
\end{aligned}$$

which shows linear convergence up to the statistical error.

4.4 Application to Multi-task Learning

In this section, we apply the theory developed in Section 4.3 on two specific problems. First, in Section 4.4.1, we apply GDT algorithm to a multi-task learning problem. We show that under commonly used statistical conditions the conditions on the objective function stated in Section 4.3.1 are satisfied with high-probability. Next, in Section 4.4.2 we discuss an application to multi-task reinforcement learning problem.

4.4.1 GDT for Multi-task Learning

We apply GDT algorithm to the problem of multi-task learning, which has been successfully applied in a wide range of application areas, ranging from neuroscience [313], natural language understanding [85], speech recognition [268], computer vision [272], and genetics [354, 353] to remote sensing [332], image classification [190], spam filtering [325], web search [68], disease prediction [369], and eQTL mapping [175]. By transferring information between related tasks it is hoped that samples will be better utilized, leading to improved generalization performance.

We consider the following linear multi-task learning problem

$$Y = X\Theta^* + E, \tag{4.17}$$

where $Y \in \mathbb{R}^{n\times k}$ is the response matrix, $X \in \mathbb{R}^{n\times p}$ is the matrix of predictors, $\Theta^* \in \mathbb{R}^{p\times k}$ is an unknown matrix of coefficients, and $E \in \mathbb{R}^{n\times k}$ is an unobserved noise matrix with i.i.d. mean zero and variance σ^2 entries. Here n denotes the sample size, k is the number of responses, and p is the number of predictors. In general multi-task learning problems, the design matrix X may be different for different tasks. Throughout the chapter we assume a common design matrix X for simplicity, and it is straightforward to generalize the result to problem with different X for different tasks.

There are a number of ways to capture relationships between different tasks and success of different methods relies on this relationship. [98] studied a setting where linear predictors are close to each other. In a high-dimensional setting, with large number of variables, it is common to assume that there are a few variables predictive of all tasks, while others are not predictive [307, 239, 212, 183, 320]. Another popular condition is to assume that the predictors lie in a shared lower dimensional subspace [16, 12, 360, 19, 319]. In contemporary applications, however, it is increasingly common that both the number of predictors and the number of tasks is large compared to the sample size. For example, in a study of regulatory relationships between genome-wide measurements, where micro-RNA measurements are used to explain the gene expression levels, it is commonly assumed that a small number of micro-RNAs regulate genes participating in few regulatory pathways [217]. In such a setting, it is reasonable to assume that the coefficients are both sparse and low rank. That is, one believes that the predictors can be combined into fewer latent features that drive the variation in the multiple response variables and are composed only of relevant predictor variables. Compared to a setting where either variables are selected or latent features are learned, there is much less work on simultaneous variable selection and rank reduction [46, 71, 72, 272]. In addition, when both p and k are large, it is also needed to assume the column sparsity on the matrix Θ^* to make estimation feasible [218], a model that has been referred to as the two-way sparse reduced-rank regression model. We focus on this model here.

Multi-task Model (MTM) In the model (4.17), we assume that the true coefficient matrix $\Theta^* \in \Xi(r, s_1^*, s_2^*)$. The noise matrix E has i.i.d. sub-Gaussian elements with variance proxy σ^2, which requires that each element e_{ij} satisfies $\mathbb{E}(e_{ij}) = 0$ and its moment generating function satisfies $\mathbb{E}[\exp(te_{ij})] \leq \exp(\sigma^2 t^2/2)$. The design matrix X is considered fixed with columns normalized to have mean 0 and standard deviation 1. Moreover, we assume X satisfies the following Restricted Eigenvalue (RE) condition [233] for some constant $\underline{\kappa}(s_1)$ and $\bar{\kappa}(s_1)$.

$$\underline{\kappa}(s_1) \cdot \|\theta\|_2^2 \leq \frac{1}{n}\|X\theta\|_2^2 \leq \bar{\kappa}(s_1) \cdot \|\theta\|_2^2 \quad \text{for all } \|\theta\|_0 \leq s_1.$$

We will show that under the condition **(MTM)**, GDT converges linearly to the optimal coefficient Θ^* up to a region of statistical error. Compared to the previous methods for estimating jointly sparse and low rank coefficients [46, 71, 72, 218], GDT is more scalable and improves estimation accuracy as illustrated in the simulation Section 5.6.

In the context of the multi-task learning with the model in (4.17), we are going to use the least squares loss. The objective function in is $f(\Theta) = \frac{1}{2n}\|Y - X\Theta\|_F^2$ and we write $\Theta = UV^\top$ with $U \in \mathbb{R}^{p\times r}$ and $V \in \mathbb{R}^{k\times r}$. The constraint set is set as before as $U \in \mathcal{U}(s_1)$ and $V \in \mathcal{U}(s_2)$ with $s_1 = c \cdot s_1^*, s_2 = c \cdot s_2^*$ for some $c > 1$. The rank r and the sparsity levels

s_1, s_2 are tuning parameters, which can be selected using the information criterion as in [272].

In order to apply the results of Theorem 38, we first verify the conditions in Section 4.3.1. The condition **(RSC/RSS)** in is equivalent to

$$\mu \big\| \Theta_2 - \Theta_1 \big\|_F^2 \leq \Big\langle \frac{1}{n} X^\top X (\Theta_2 - \Theta_1), \Theta_2 - \Theta_1 \Big\rangle \leq L \big\| \Theta_2 - \Theta_1 \big\|_F^2,$$

and it holds with $\mu = \underline{\kappa}(s_1)$ and $L = \bar{\kappa}(s_1)$.

Next, we discuss how to initialize GDT in the context of multi-task learning. Under the structural conditions on Θ^* in the condition **(MTM)** there are a number of way to obtain an initial estimator Θ^0. For example, we can use row and column screening [99], group lasso [359], and lasso [303] among other procedures. Here and in simulations we use the lasso estimator, which takes the form

$$\Theta^0 = \arg \min_{\Theta \in \mathbb{R}^{p \times k}} \frac{1}{2n} \| Y - X\Theta \|_F^2 + \lambda \| \Theta \|_1.$$

The benefit of this approach is that it is scalable to the high-dimensional setting and trivially parallelizable, since each column of Θ^0 can be estimated separately. The requirement of the initialization condition **(I)** is effectively a requirement on the sample size. Under the condition **(MTM)**, a result of [233] shows that these conditions are satisfied with $n \geq s_1^* s_2^* \log p \log k$.

We then characterize the statistical error e_{stat} under the condition **(MTM)**.

Lemma 42. *Under the condition* **(MTM)**, *with probability at least* $1 - (p \vee k)^{-1}$ *we have*

$$e_{stat} \leq C\sigma \sqrt{\frac{(s_1^* + s_2^*)\big(r + \log(p \vee k)\big)}{n}}$$

for some constant C.

The proof of Lemma 42 is given in Section 4.6.4.

With these conditions, we have the following result on GDT when applied to the multi-task learning model in (4.17).

Corollary 43. *Suppose that the condition* **(MTM)** *is satisfied and the step size η satisfies* (4.9) - (4.11). *Then for all*

$$T \geq C \log \left[\frac{n}{(s_1^* + s_2^*)\big(r + \log(p \vee k)\big)} \right],$$

with probability at least $1 - (p \vee k)^{-1}$, we have

$$\|\Theta^T - \Theta^*\|_F \leq C\sigma \sqrt{\frac{(s_1^* + s_2^*)\big(r + \log(p \vee k)\big)}{n}}$$

for some constant C.

Each iteration of the algorithm requires computing the gradient step with time complexity $r(n+r)(p+k)$. Note that if there is no error term E in the model (4.17), then Algorithm 3 converges linearly to the true coefficient matrix Θ^*, since $e_{\text{stat}} = 0$ in that case. The error rate in Corollary 43 matches the error rate of the algorithm proposed in [218]. However, our algorithm does not require a new independent sample in each iteration and allows for non-Gaussian errors. Compared to the minimax rate

$$\sigma \sqrt{\frac{1}{n} \left[(s_1^* + s_2^*)r + s_1^* \log \frac{ep}{s_1^*} + s_2^* \log \frac{ek}{s_2^*} \right]} \tag{4.18}$$

established in [218], both our algorithm and that of [218] match the rate up to a multiplicative log factor. To the best of our knowledge, achieving the minimax rate (4.18) with a computationally scalable procedure is still an open problem. Note, however, that when r is comparable to $\log(p \vee k)$ the rates match up to a constant multiplier. Therefore for large enough T, GDT algorithm attains near optimal rate.

In case we do not consider column sparsity, that is, when $s_2^* = k$, Corollary 43 gives error rate

$$\|\Theta^T - \Theta^*\|_F \leq C\sigma\sqrt{\frac{kr + s_1^*\big(r + \log p\big)}{n}} \tag{4.19}$$

and prediction error

$$\|X\Theta^T - X\Theta^*\|_F^2 \leq C\sigma^2\Big(kr + s_1^*\big(r + \log p\big)\Big).$$

Compared to the prediction error bound $kr + s_1^* r \log \frac{p}{s}$ proved in [47], we see that GDT error is much smaller with $r + \log p$ in place of $r \log p$. Moreover, GDT error matches the prediction error $(k + s_1^* - r)r + s_1^* \log p$ established in [272], as long as $k \geq Cr$ which is typically satisfied.

As mentioned before, in practice we use the criterion in [46] to select the rank r. In order to obtain a consistent rank estimator with high probability, the procedure in [46] requires that $\sigma_{r^*}(X\Theta^*)$ is lower bounded. We emphasize that although this condition is not required to obtain a near minimax optimal statistical error, it does affect the convergence rate of the GDT algorithm.

4.4.2 Application to Multi-task Reinforcement Learning

Reinforcement learning (RL) and approximate dynamic programming (ADP) are popular algorithms that help decision makers find optimal policies for decision making problems under uncertainty that can be cast in the framework of Markov Decision Processes (MDP) [33, 294]. Similar to many other approaches, when the sample size is small these algorithms may have poor performance. A possible workaround then is to simultaneously solve multiple related tasks and take advantage of their similarity and shared structure. This approach is called multi-task reinforcement learning (MTRL) and has been studied extensively [191, 327, 285]. In this section we show how GDT algorithm can be applied to the MTRL problem.

A Markov decision process (MDP) is represented by a 5-tuple $\mathcal{M} = (S, A, P, R, \gamma)$ where S represents the state space (which we assume to be finite for simplicity); A is a finite set of actions; $P_a(s, s') = \Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$ is the Markovian transition kernel that measures the probability that action a in state s at time t will lead to state s' at time $t+1$ (we assume P_a to be time homogeneous); $R(s,a)$ is the state-action reward function measuring the instantaneous reward received when taking action a in state s; and γ is the discount factor. The core problem of MDP is to find a deterministic policy $\pi : S \to A$ that specifies the action to take when decision maker is in some state s. Define the Bellman operator

$$\mathcal{T}Q(s,a) = R(s,a) + \gamma \sum_{s'} P_a(s,s') \max_{a'} Q(s',a'),$$

where $Q : S \times A \to \mathbb{R}$ is the state-action value function. The MDP can then be solved by calculating the optimal state-action value function Q^* which gives the total discounted reward obtained starting in state s and taking action a, and then following the optimal policy in subsequent time steps. Given Q^*, the optimal policy is recovered by the greedy policy: $\pi^*(s) = \arg\max_{a \in A} Q^*(s,a)$.

In MTRL the objective is to solve k related tasks simultaneously where each task $k_0 \in \{1, \ldots, k\}$ corresponds to an MDP: $\mathcal{M}_{k_0} = (S, A, P_{k_0}, R_{k_0}, \gamma_{k_0})$. Thus, these k tasks share the same state and action space but each task has a different transition dynamics P_{k_0}, state-action reward function R_{k_0}, and discount factor γ_{k_0}. The decision maker's goal is to find an optimal policy for each MDP. If these MDPs do not share any information or structure, then it is straightforward to solve each of them separately. Here we assume the MDPs do share some structure so that the k tasks can be learned together with smaller sample complexity than learning them separately.

We follow the structure in [57] and solve this MTRL problem by the fitted-Q iteration (FQI) algorithm [96], one of the most popular method for ADP. In contrast to exact value iteration ($Q^t = \mathcal{T}Q^{t-1}$), in FQI this iteration is approximated by solving a regression problem by representing $Q(s,a)$ as a linear function in some features representing the state-action pairs. To be more specific, we denote $\varphi(s) = [\varphi_1(s), \varphi_2(s), ..., \varphi_{p_s}(s)]$ as the feature mapping for state s where $\varphi_i : S \to \mathbb{R}$ denotes the ith feature. We then extend the state-feature vector

φ to a feature vector mapping state s and action a as:

$$\phi(s,a) = [\underbrace{0,0,...,0}_{(a-1)\times p_s \text{ times}}, \varphi_1(s), \varphi_2(s), ..., \varphi_{p_s}(s), \underbrace{0,0,...,0}_{(|A|-a)\times p_s \text{ times}}] \in \mathbb{R}^p,$$

where $p = |A| \times p_s$. Finally, for MDP k_0, we represent the state-action value function $Q_{k_0}(\cdot,\cdot)$ as an $|S| \times |A|$ dimensional column vector with:

$$Q_{k_0}(s,a) = \phi(s,a)^\top \cdot \Theta_{k_0}$$

where Θ_{k_0} is a $p \times 1$ dimensional column vector. If $\Theta \in \mathbb{R}^{p\times k}$ represents the matrix with columns Θ_{k_0}, $k \in \{1,\dots,k\}$, then we see that given the $Q_{k_0}(s,a)$ state-action value functions, estimating the Θ matrix is just a Multi-Task Learning problem of the form (4.17) with the response matrix $Y \doteq Q \in \mathbb{R}^{n\times k}$ where $n = |S| \times |A|$ denotes the "sample size" with rows indexed by pairs $(s,a) \in S \times A$, $X \doteq \Phi \in \mathbb{R}^{n\times p}$ represents the matrix of predictors (features) with $(s,a)^{th}$ row as $\phi(s,a)$, and Θ^* is the unknown matrix of ADP coefficients. Consistent with the GDT algorithm, to exploit shared sparsity and structure across the k MDP tasks, we will subsequently assume that the coefficient matrix Θ^* is row sparse and low rank.

Algorithm 4 provides details of MTRL with GDT. We assume we have access to the generative model of the k MDPs so that we can sample reward r and state s' from $R(s,a)$ and $P_a(s,s')$. With "design states" $S_k \subseteq S$, $n_s \doteq |S_k|$ given as input, for each action a and each state $s \in S_k$, FQI first generates samples (reward r and transition state s') from the generative model of each MDP. These samples form a new dataset according to

$$y^t_{i,a,k_0} = r^t_{i,a,k_0} + \gamma \max_{a'} \widehat{Q}^{t-1}_{k_0}(s'^t_{i,a,k_0}, a').$$

Here $\widehat{Q}^{t-1}_{k_0}$ is calculated using the coefficient matrix from previous iteration:

$$\widehat{Q}^{t-1}_{k_0}(s'^t_{i,a,k_0}, a') = \phi(s'^t_{i,a,k_0}, a')^\top \cdot \Theta^{t-1}_{k_0}$$

We then build dataset $\mathcal{D}_{k_0}^t = \big\{(s_i, a), y_{i,a,k_0}^t\big\}_{s_i \in S_k, a \in A}$ with s as predictor and y as response, and apply GDT algorithm on the dataset $\{D_{k_0}^t\}_{k_0=1}^k$ to get estimator Θ^t. This completes an iteration t and we repeat this process until convergence. Finally the optimal policy $\pi_{k_0}^t$ is given by greedy policy: $\pi_{k_0}^t(s) = \arg\max_{a \in A} \widehat{Q}_{k_0}^t(s, a)$ at each iteration t.

To derive theoretical result analogous to [57], we further assume $R(s, a) \in [0, 1]$ and hence the maximum cumulative discounted reward $Q_{\max} = 1/(1-\gamma)$. Since each task is a meaningful MDP, we do not assume sparsity on columns. Suppose $\sup_s \|\varphi(s)\|_2 \le L$, we have the following theoretical result:

Theorem 44. *Suppose the linear model holds and suppose the conditions in Section 5.4 are satisfied for each Θ_a^* with rank r and row sparsity s_1^*, then after T iterations, with probability at least $\big(1 - (p \wedge k)^{-1}\big)^T$ we have*

$$\frac{1}{k}\sum_{k_0=1}^{k} \Big\| Q_{k_0}^* - Q_{k_0}^{\pi_{k_0}^T} \Big\|_2^2 \le \frac{C}{(1-\gamma)^4}\left[\frac{1}{n} Q_{\max}^2 L^4 \Big(r + \frac{s_1^*}{k}(r + \log p)\Big)\right] + \frac{4Q_{\max}^2}{(1-\gamma)^4}\Big[C\beta^T + \gamma^T\Big]^2$$

for some constant C.

Proof. We start from the intermediate result in [225]:

$$\Big| Q_{k_0}^* - Q_{k_0}^{\pi_{k_0}^T} \Big| \le \frac{2\gamma(1-\gamma^{T+1})}{(1-\gamma)^2}\Big[\sum_{t=0}^{T-1} \alpha_t |\epsilon_{k_0}^t| + \alpha_T \big| Q_t^* - Q_t^0 \big|\Big],$$

where

$$\alpha_t = \frac{(1-\gamma)\gamma^{T-t-1}}{1-\gamma^{T+1}}, \text{ for } t < T, \text{ and } \alpha_T = \frac{(1-\gamma)\gamma^T}{1-\gamma^{T+1}}.$$

The error term $\epsilon_{k_0}^t(s', b)$ measures the approximation error in state $s' \in S$ and action

Algorithm 4 Multi-Task Reinforcement Learning with GDT

```
Input: States $S_k = \{s_i\}_{i=1}^{n_s} \subseteq S$.
Initialize $\Theta^0 = 0$
for $t = 1$ to $T$ do
  for $a = 1$ to $|A|$ do
    for $k_0 = 1$ to $k$, $i = 1$ to $n_s$ do
      Generate samples $r_{i,a,k_0}^t = R_{k_0}(s_i, a)$ and ${s'}_{i,a,k_0}^t \sim P_{a,k_0}(s_i, s')$
      Calculate $y_{i,a,k_0}^t = r_{i,a,k_0}^t + \gamma \max_{a'} \widehat{Q}_{k_0}^{t-1}({s'}_{i,a,k_0}^t, a')$
    end for
  end for
  Estimate $\Theta^t$ using GDT algorithm with $X = \Big\{X((s_i, a), \cdot) = \phi(s_i, a)^\top\Big\}_{s_i \in S_k, a \in A}$ and $Y = \Big\{Y((s_i, a), k_0) = y_{i,a,k_0}^t\Big\}_{s_i \in S, a \in A, k_0 \in [k]}$.
end for
Output: $\Theta^T$
```

$b \in A$. It can be bounded by

$$\begin{aligned}
\big|\epsilon_{k_0}^t(s', b)\big| = \big|\varphi(s')^\top \Theta_{k_0,b}^t - \varphi(s')^\top \Theta_{k_0,b}^*\big| &\le \big\|\varphi(s')\big\|_2 \big\|\Theta_{k_0,b}^t - \Theta_{k_0,b}^*\big\|_2 \\
&\le L\big\|\Theta_{k_0,b}^t - \Theta_{k_0,b}^*\big\|_2.
\end{aligned}$$

We then have

$$\Big|Q_{k_0}^* - Q_{k_0}^{\pi_{k_0}^T}\Big| \le \frac{2\gamma(1-\gamma^{T+1})}{(1-\gamma)^2}\Bigg[\sum_{t=0}^{T-1} \alpha_t L \max_b \big\|\Theta_{k_0,b}^t - \Theta_{k_0,b}^*\big\|_2 + 2\alpha_T Q_{\max}\Bigg].$$

Taking average, and plugging in the main result (4.13) and the statistical error (4.19) we obtain our desired result. □

4.5 Experiment

In this section we demonstrate the effectiveness of the GDT algorithm by extensive experiments[1]. Section 4.5.1 shows results on synthetic datasets while Section 4.5.2 and 4.5.3 show results on two real datasets.

[1] The codes are available at `https://github.com/ming93/GDT_nonconvex`

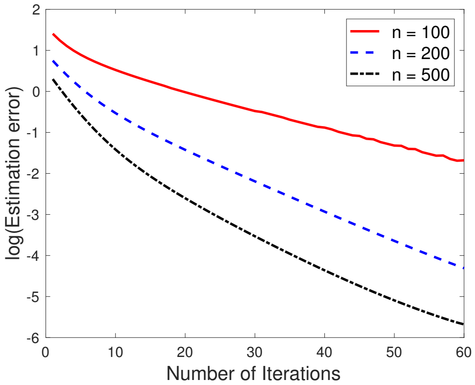


Figure 4.1: No error case

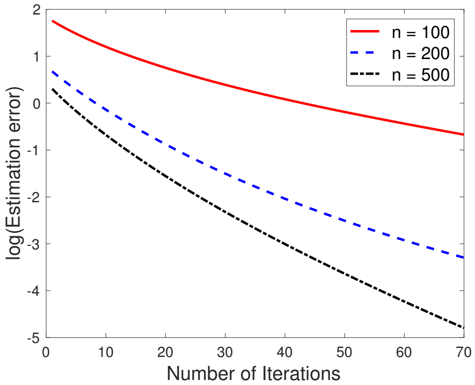


Figure 4.2: No sparsity case

4.5.1 Synthetic Datasets

We present numerical experiments on MTL problem to support our theoretical analysis. Throughout this section, we generate the instances by sampling all entries of design matrix X, all nonzero entries of the true signal U^* and V^*, and all entries of the noise matrix E as i.i.d. standard normal.

Linear convergence. We first demonstrate our linear convergence result. Because it is hard to quantify linear convergence with statistical error, we turn to show the linear convergence in some special cases. Firstly, as we discussed after Corollary 43, suppose there is no error term E in the model (4.17), then Algorithm 3 converges linearly to the true coefficient matrix Θ^*. In this case we choose $p = 100, k = 50, r = 8, s_1^* = s_2^* = 10$, and the estimation error is shown in Figure 4.1. Secondly, as we discussed at the end of Section 4.3.2, suppose there are no row or column sparsity constraints on Θ^*, then Algorithm 3 converges linearly to global minimum $\widehat{\Theta}$. In this case it is more likely that we are in low dimensions, therefore we choose $p = 50$. The estimation error is shown in Figure 4.2. We see that in both cases GDT has linear convergence rate.

Estimation accuracy. We compare our algorithm with the Double Projected Penalization (DPP) method in [218], the thresholding SVD method (TSVD) method in [217], the exclusive extraction algorithm (EEA) in [71], the two methods (denoted by RCGL and JRRS) in [47], and the standard Multitask learning method (MTL, with $L_{2,1}$ penalty). Here we set $n = 50, p = 100, k = 50, r = 8, s_1^* = s_2^* = 10$. The reason why we choose a relatively small scale is that many other methods do not scale to high dimensions, as will shown in Table 4.5. We will show the effectiveness of our method in high dimensions later. Except for standard MTL, all the other methods need an estimate of the rank to proceed for which we apply the rank estimator in [46]. For the methods that rely on tuning parameters, we generate an independent validation set to select the tuning parameters.

We consider two coefficient matrix settings, one is only row sparse and the other one is both row sparse and column sparse. We also consider strong signal and weak signal settings. The strong signal setting is described above and for the weak signal setting, we divide the true Θ^* by 5, resulting in a signal for which recovering true non-zero variables becomes much more difficult. Table 4.1 (strong signal, row sparse), Table 4.2 (strong signal, row *and* column sparse), Table 4.3 (weak signal, row sparse) and Table 4.4 (weak signal, row *and* column sparse) report the mean and the standard deviation of prediction errors, estimation errors and size of selected models based on 50 replications in each setting. We can see that in all the cases GDT has the lowest estimation error and prediction error. When the signal is weak, GDT may underselect the number of nonzero rows/columns, but it still has the best performance.

Running time. We then compare the running time of all these methods. We fix a baseline model size $n = 50, p = 80, k = 50, r = 4, s_1^* = s_2^* = 10$, and set a free parameter ζ. For $\zeta = \{1, 5, 10, 20, 50, 100\}$, each time we increase n, p, s_1^*, s_2^* by a factor of ζ and increase k, r by a factor of $\lfloor\sqrt{\zeta}\rfloor$ and record the running time (in seconds) of each method for a fixed tolerance level, whenever possible. We run each algorithm with a fixed reasonable tuning parameter without any validation step. If for some ζ the algorithm does not converge in 2 hours then we simply record ">2h" and no longer increase ζ for that method. Table 4.5 summarizes the results. We can see that GDT is fast even in very high dimension, while all of the other methods are computationally expensive. We note that even though GDT

Table 4.1: Strong signal, Row sparse

	Estimation error	Prediction error	|Row support|
GDT	0.0452 ± 0.0110	1.1060 ± 0.0248	10.16 ± 0.51
DPP	0.0584 ± 0.0113	1.1290 ± 0.0357	52.64 ± 15.2
TSVD	0.3169 ± 0.1351	2.4158 ± 0.9899	25.62 ± 8.03
EEA	0.3053 ± 0.0998	1.2349 ± 0.0362	84.28 ± 6.70
RCGL	0.0591 ± 0.0148	1.1101 ± 0.0168	49.60 ± 10.6
JRRS	0.0877 ± 0.0227	1.1857 ± 0.0214	12.26 ± 2.02
MTL	0.0904 ± 0.0243	1.1753 ± 0.0204	73.40 ± 2.67

Table 4.2: Strong signal, Row sparse and column sparse

	Estimation error	Prediction error	|Row support|	|Column support|
GDT	0.0624 ± 0.0121	1.0353 ± 0.0167	10.24 ± 0.65	10.24 ± 0.68
DPP	0.0921 ± 0.0251	1.0790 ± 0.0295	54.10 ± 18.25	10.38 ± 0.60
TSVD	0.3354 ± 0.1053	1.7600 ± 0.3415	28.66 ± 7.27	30.88 ± 8.46
EEA	0.2604 ± 0.1159	1.1023 ± 0.0220	64.44 ± 9.88	12.10 ± 2.69
RCGL	0.1217 ± 0.0325	1.1075 ± 0.0174	42.06 ± 7.93	50 ± 0
JRRS	0.1682 ± 0.0410	1.1612 ± 0.0174	13.96 ± 4.69	50 ± 0
MTL	0.1837 ± 0.0499	1.1652 ± 0.0160	73.50 ± 3.17	50 ± 0

uses the lasso estimator in the initialization step, all the variables are used in the subsequent iterations and not only the ones selected by the lasso. In particular, the speed of the method does not come from the initialization step. Table 4.6 summarizes the averaged number of iterations with different choices of ζ with a tolerance value 10^{-3} on the objective function. We see that the number of iteration T scales reasonably with the size of the problem.

Effectiveness in high dimension. Next, we demonstrate the effectiveness of GDT algorithm in high dimensions. Table 4.1 and Table 4.2 are both in low dimensions because we want to compare with other algorithms and they are slow in high dimensions, as shown in Table 4.5. Now we run our algorithm only and we choose $p = 5000, k = 3000, r = 50, s_1^* = s_2^* = 100$. The estimation error and objective value are shown in Figure 4.3 and Figure 4.4, respectively. In each figure, iteration 0 is for initialization we obtained by Lasso.

We can see that both estimation error and objective value continue to decrease, which

Table 4.3: Weak signal, Row sparse

	Estimation error	Prediction error	|Row support|
GDT	0.2328 ± 0.0474	1.1282 ± 0.0231	10.08 ± 0.56
DPP	0.2954 ± 0.0640	1.1624 ± 0.0315	47.26 ± 11.7
TSVD	0.5842 ± 0.1020	1.4271 ± 0.0903	30.81 ± 4.72
EEA	0.3802 ± 0.0787	1.1647 ± 0.0206	46.16 ± 8.97
RCGL	0.2775 ± 0.0605	1.1493 ± 0.0291	37.92 ± 14.4
JRRS	0.3600 ± 0.0752	1.1975 ± 0.0392	11.74 ± 1.35
MTL	0.3577 ± 0.0721	1.2140 ± 0.0418	69.92 ± 12.8

Table 4.4: Weak signal, Row sparse and column sparse

	Estimation error	Prediction error	|Row support|	|Column support|
GDT	0.3173 ± 0.0949	1.0380 ± 0.0218	9.56 ± 1.56	10.06 ± 1.21
DPP	0.3899 ± 0.0737	1.0580 ± 0.0216	50.66 ± 12.86	13.52 ± 5.02
TSVD	0.6310 ± 0.1074	1.1372 ± 0.0246	49.94 ± 5.53	43.38 ± 2.55
EEA	0.6016 ± 0.0965	1.0874 ± 0.0197	30.64 ± 8.65	30.64 ± 8.65
RCGL	0.4601 ± 0.0819	1.1017 ± 0.0262	28.9 ± 12.36	50 ± 0
JRRS	0.5535 ± 0.0866	1.1164 ± 0.0262	12.42 ± 6.02	50 ± 0
MTL	0.5776 ± 0.0873	1.1286 ± 0.0296	53.0 ± 18.41	50 ± 0

Table 4.5: Running time comparison (in seconds)

	$\zeta = 1$	$\zeta = 5$	$\zeta = 10$	$\zeta = 20$	$\zeta = 50$	$\zeta = 100$
GDT	0.11	0.20	0.51	2.14	29.3	235.8
DPP	0.19	0.61	3.18	17.22	315.4	2489
TSVD	0.07	1.09	6.32	37.8	543	6075
EEA	0.50	35.6	256	>2h	>2h	>2h
RCGL	0.18	1.02	7.15	36.4	657.4	>2h
JRRS	0.19	0.82	6.36	30.0	610.2	>2h
MTL	0.18	3.12	30.92	184.3	>2h	>2h

Table 4.6: Number of iterations in GDT algorithm with different choices of ζ

$\zeta = 1$	$\zeta = 5$	$\zeta = 10$	$\zeta = 20$	$\zeta = 50$	$\zeta = 100$
31.9	41.6	33.6	53.0	72.4	153.8

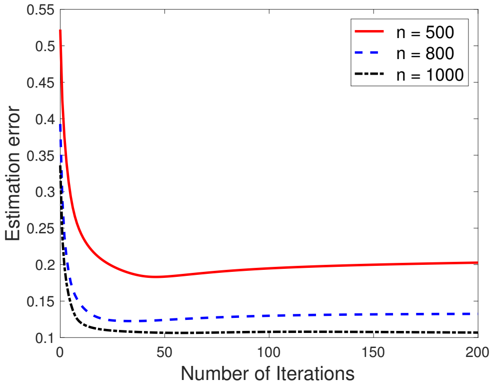


Figure 4.3: Estimation error

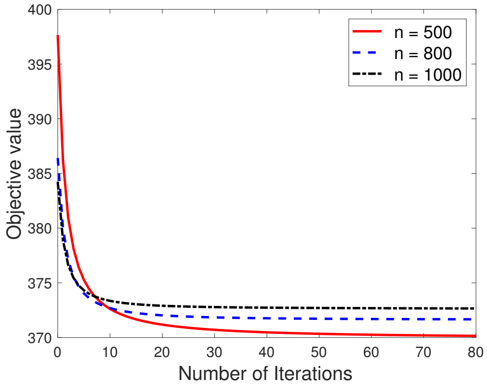


Figure 4.4: Objective value

demonstrates the effectiveness and necessity of GDT algorithm. From Figure 4.3 we also find that early stopping can help to avoid overfitting (although not too much), especially when n is small.

Effect of sparsity and rank. We finally check the effect of the choices of sparsity level s and rank r on the performance of the algorithm. Here we set $n = 100, p = 500, k = 300, r = 15, s_1^* = s_2^* = 25$. We again consider strong signal and weak signal setting, where we divide the true Θ^* by 5 for weak signal setting. Table 4.7 (strong signal, estimation error), Table 4.8 (strong signal, prediction error), Table 4.9 (strong signal, estimation error), and Table 4.10 (strong signal, prediction error) report the average performance (estimation error and prediction error) of GDT algorithm with different choices of sparsity level s and rank r, based on 50 replicates in each setting. The row and column with true sparsity level and rank are highlighted as bold. In the last column, we select the rank based on the rank estimator in [46].

From the tables we see that the performance of the algorithm is poor when we underselect a sparsity level or rank. This is more significant when the signal is strong, since we are missing too many large nonzero values in the estimator. This demonstrates the necessity to be conservative when selecting sparsity level and rank. When both the sparsity level and rank are selected as greater than the true value, the algorithm performs well in a relatively large range of sparsity level and rank, especially for prediction error. As a baseline, if we estimate

Θ with Lasso estimator on each column where the regularization parameter is selected by validation set, the averaged estimation and prediction error is 0.2453 and 1.0902 for strong signal case, and 0.4716 and 1.0636 for weak signal case. Moreover, we see that overselecting a rank does not harm the performance too much, compared to overselecting a sparsity level. The rank selected by [46] performs well, and sometimes it even performs better than any of the fixed ranks.

Table 4.7: Estimation error of different choices of sparsity level and rank, with strong signal

	$r = 10$	$r = 12$	$\boldsymbol{r = 15}$	$r = 20$	$r = 30$	$r = 50$	$r = 80$	r selected by [46]
$s = 15$	0.6367	0.6193	0.6080	0.6228	0.6109	0.6241	0.6259	0.6164
$s = 20$	0.4752	0.4644	0.4562	0.4715	0.4532	0.4536	0.4717	0.4693
$\boldsymbol{s = 25}$	0.2467	0.1668	0.0238	0.0251	0.0256	0.0261	0.0261	0.0240
$s = 30$	0.2567	0.1687	0.0288	0.0318	0.0316	0.0310	0.0314	0.0286
$s = 40$	0.2460	0.1715	0.0411	0.0424	0.0425	0.0413	0.0428	0.0437
$s = 50$	0.2495	0.1588	0.0559	0.0500	0.0518	0.0523	0.0532	0.0585
$s = 80$	0.2468	0.1725	0.1166	0.1084	0.1077	0.1095	0.1084	0.1273

Table 4.8: Prediction error of different choices of sparsity level and rank, with strong signal

	$r = 10$	$r = 12$	$\boldsymbol{r = 15}$	$r = 20$	$r = 30$	$r = 50$	$r = 80$	r selected by [46]
$s = 15$	3.6055	3.6932	3.6038	3.6659	3.6220	3.6203	3.5750	3.5658
$s = 20$	2.5595	2.5062	2.4969	2.5078	2.4935	2.4514	2.5026	2.4780
$\boldsymbol{s = 25}$	1.3722	1.1432	1.0096	1.0097	1.0102	1.0103	1.0111	1.0101
$s = 30$	1.3808	1.1422	1.0129	1.0136	1.0149	1.0152	1.0146	1.0128
$s = 40$	1.3831	1.1519	1.0176	1.0208	1.0246	1.0262	1.0263	1.0196
$s = 50$	1.3966	1.1427	1.0252	1.0289	1.0371	1.0410	1.0402	1.0253
$s = 80$	1.3935	1.1650	1.0415	1.0527	1.0754	1.0969	1.1013	1.0419

4.5.2 Norwegian Paper Quality Dataset

In this section we apply GDT to Norwegian paper quality dataset. This data was obtained from a controlled experiment that was carried out at a paper factory in Norway to uncover the

Table 4.9: Estimation error of different choices of sparsity level and rank, with weak signal

	$r = 10$	$r = 12$	$\boldsymbol{r = 15}$	$r = 20$	$r = 30$	$r = 50$	$r = 80$	r selected by [46]
$s = 15$	0.6292	0.6340	0.6361	0.6313	0.6169	0.6430	0.6276	0.6203
$s = 20$	0.4733	0.4732	0.4905	0.4714	0.4747	0.4792	0.4734	0.4715
$\boldsymbol{s = 25}$	0.2525	0.1877	0.1307	0.1296	0.1349	0.1365	0.1474	0.1484
$s = 30$	0.2704	0.1981	0.1474	0.1396	0.1441	0.1443	0.1364	0.1561
$s = 40$	0.2677	0.2033	0.1702	0.1772	0.1830	0.1824	0.1820	0.1810
$s = 50$	0.2695	0.2211	0.2075	0.2156	0.2281	0.2258	0.2254	0.2137
$s = 80$	0.3114	0.3008	0.2836	0.3060	0.3239	0.3474	0.3461	0.2889

Table 4.10: Prediction error of different choices of sparsity level and rank, with weak signal

	$r = 10$	$r = 12$	$\boldsymbol{r = 15}$	$r = 20$	$r = 30$	$r = 50$	$r = 80$	r selected by [46]
$s = 15$	1.2218	1.2173	1.2203	1.2221	1.2304	1.2265	1.2191	1.2256
$s = 20$	1.1107	1.1055	1.1076	1.1057	1.1059	1.1054	1.1050	1.1051
$\boldsymbol{s = 25}$	1.0251	1.0150	1.0100	1.0096	1.0103	1.0098	1.0112	1.0110
$s = 30$	1.0291	1.0166	1.0133	1.0131	1.0142	1.0135	1.0136	1.0140
$s = 40$	1.0334	1.0235	1.0188	1.0217	1.0255	1.0267	1.0257	1.0202
$s = 50$	1.0352	1.0264	1.0247	1.0308	1.0390	1.0405	1.0376	1.0250
$s = 80$	1.0466	1.0396	1.0412	1.0547	1.0732	1.0919	1.0878	1.0390

effect of three control variables X_1, X_2, X_3 on the quality of the paper which was measured by 13 response variables. Each of the control variables X_i takes values in $\{-1, 0, 1\}$. To account for possible interactions and nonlinear effects, second order terms were added to the set of predictors, yielding $X_1, X_2, X_3, X_1^2, X_2^2, X_3^2, X_1 \cdot X_2, X_1 \cdot X_3, X_2 \cdot X_3$.

The data set can be downloaded from the website of [156] and its structure clearly indicates that dimension reduction is possible, making it a typical application for reduced rank regression methods [156, 7, 47, 272]. Based on the analysis of [46] and [7] we select the rank $\widehat{r} = 3$; also suggested by [46] we take $s_1 = 6$ and $s_2 = k = 13$ which means we have row sparsity only. GDT selects 6 of the original 9 predictors, with $X_1^2, X_1 \cdot X_2$ and $X_2 \cdot X_3$ discarded, which is consistent with the result in [46].

To compare prediction errors, we split the whole dataset at random, with 70% for training and 30% for test, and repeat the process 50 times to compare the performance of the above methods. All tuning parameters are selected by cross validation and we always center the responses in the training data (and transform the test data accordingly). The average RMSE on test set is shown in Table 4.11. We can see that GDT is competitive with the best method, demonstrating its effectiveness on real datasets.

Table 4.11: RMSE on paper quality dataset

GDT	DPP	TSVD	EEA	RCGL	JRRS	MTL
1.002	1.012	1.094	1.161	1.001	1.013	1.014

4.5.3 Calcium Imaging Data

As a microscopy technique in neuroscience, calcium imaging is gaining more and more attentions [129]. It records fluorescent images from neurons and allows us to identify the spiking activity of the neurons. To achieve this goal, [248] introduces a spatiotemporal model and we briefly introduce this model here. More detailed description can be found in [248] and [218]. Denote $k = \ell_1 \times \ell_2$ as the pixels we observe, and denote K as the total number of neurons. The observation time step is $t = 1, ..., T$. Let $S \in \mathbb{R}^{T \times K}$ be the number of spikes at each time step and for each neuron; $A \in \mathbb{R}^{K \times k}$ be the nonnegative spatial footprint for each neuron at each pixel; $Y \in \mathbb{R}^{T \times k}$ be the observation at each time step and at each pixel; and $E \in \mathbb{R}^{T \times k}$ be the observation error. Ignore the baseline vector for all the pixels, the model in [248] is given by

$$Y = G^{-1} S A + E = X \Theta^* + E$$

where $\Theta^* = SA$ is the coefficient matrix and $X = G^{-1}$ is observed with

$$G = \begin{pmatrix} 1 & 0 & \dots & 0 \\ -\gamma & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & -\gamma & 1 \end{pmatrix}.$$

Here γ is set to be $\gamma = 1 - 1/(\text{frame rate})$ as suggested by [312]. From the settings we see that each row of S is the activation for all the neurons, and therefore it is natural to have S to be row sparse since usually we would not observe too many activations in a fixed time period; also, each column of A is the footprint for all the neurons at each pixel, and therefore it is natural to have A to be column sparse since we expect to see only a few neurons in a fixed area. Therefore our coefficient matrix $\Theta^* = SA$ would be both row sparse and column sparse. It is also low rank since it is the product of two "tall" matrices because the number of neurons K are usually small. Now we see this is a multi-task learning problem with simultaneous row-sparse, column-sparse and low rank coefficient matrix where $n = p = T$ and $k = \ell_1 \times \ell_2$.

We consider the calcium imaging data in [6] which is a movie with 559 frames (acquired at approximately 8.64 frames/sec), where each frame is 135 × 131 pixels. This dataset is also analyzed in [218] and [129]. For this dataset, we have $n = p = 559$ and $k = 135 \times 131 = 17,685$. We use $r = 50$, more conservative than the estimator given by [46] and we set $s_1 = 100$ row sparsity and $s_2 = 3000$ column sparsity. Figure 4.5 shows five most significant manually

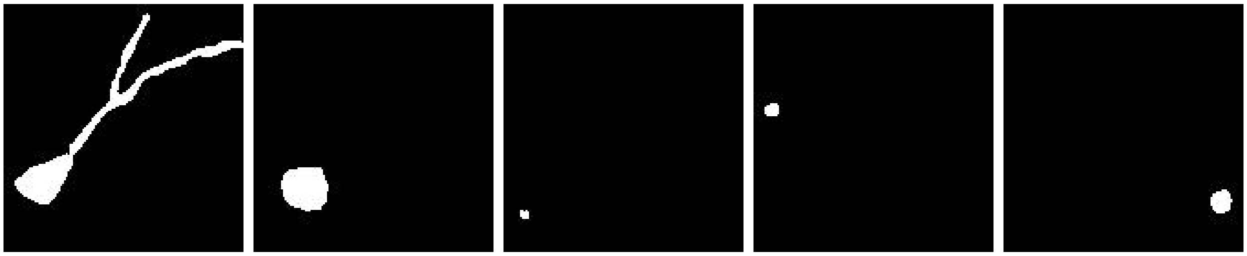


Figure 4.5: Manually selected top 5 labeled regions

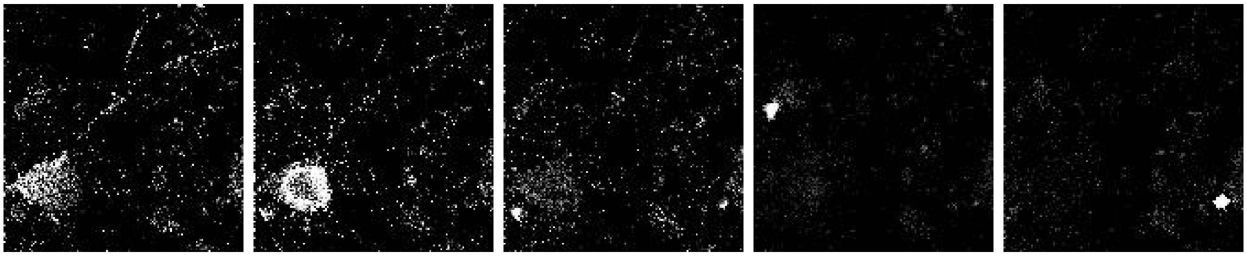


Figure 4.6: Corresponding signals estimated by our GDT algorithm

labeled regions; Figure 4.6 are the corresponding signals estimated by our GDT algorithm. We can see that they match very well, which demonstrates the effectiveness of our method.

4.6 Technical Proofs

This section collects technical proofs.

4.6.1 Proof of Lemma 39

Let $[\widetilde{U}, \widetilde{\Sigma}, \widetilde{V}] = \mathrm{rSVD}(\Theta^0)$ be the rank r SVD of the matrix Θ^0 and let

$$\widetilde{\Theta} = \widetilde{U}\widetilde{\Sigma}(\widetilde{V})^\top = \arg\min_{\mathrm{rank}(\Theta)\leq r} \|\Theta - \Theta^0\|_F.$$

Since $\widetilde{\Theta}$ is the best rank r approximation to Θ^0, we have

$$\|\widetilde{\Theta} - \Theta^0\|_F \leq \|\Theta^0 - \Theta^*\|_F.$$

The triangle inequality gives us

$$\|\widetilde{\Theta} - \Theta^*\|_F \leq \|\Theta^0 - \Theta^*\|_F + \|\Theta^0 - \widetilde{\Theta}\|_F \leq 2\|\Theta^0 - \Theta^*\|_F.$$

Now that both $\widetilde{\Theta}$ and Θ^* are rank r matrices, and according to (4.7) we have

$$\|\widetilde{\Theta} - \Theta^*\|_F \leq 2\|\Theta^0 - \Theta^*\|_F \leq \frac{1}{2}\sigma_r(\Theta^*).$$

Then, Lemma 5.14 in [305] gives us

$$\begin{aligned} d^2\Big(\begin{bmatrix} \widetilde{U}\widetilde{\Sigma}^{\frac{1}{2}} \\ \widetilde{V}\widetilde{\Sigma}^{\frac{1}{2}} \end{bmatrix}, \begin{bmatrix} U^* \\ V^* \end{bmatrix}\Big) &\leq \frac{2}{\sqrt{2}-1} \cdot \frac{\|\widetilde{\Theta} - \Theta^*\|_F^2}{\sigma_r(\Theta^*)} \\ &\leq \frac{2}{\sqrt{2}-1} \cdot \frac{4}{\sigma_r(\Theta^*)} \cdot \frac{I_0^2}{25\xi^2} \cdot \sigma_r(\Theta^*) \\ &\leq \frac{I_0^2}{\xi^2} \end{aligned}$$

where the second inequality comes from the initialization condition (4.7). Finally, Lemma 3.3 in [204] gives

$$d^2\Big(\begin{bmatrix} U^0 \\ V^0 \end{bmatrix}, \begin{bmatrix} U^* \\ V^* \end{bmatrix}\Big) \leq \xi^2 d^2\Big(\begin{bmatrix} \widetilde{U}\widetilde{\Sigma}^{\frac{1}{2}} \\ \widetilde{V}\widetilde{\Sigma}^{\frac{1}{2}} \end{bmatrix}, \begin{bmatrix} U^* \\ V^* \end{bmatrix}\Big) \leq I_0^2.$$

4.6.2 Proof of Lemma 40

For notation simplicity, let $Z = \begin{bmatrix} U \\ V \end{bmatrix}$ denote the current iterate and let $Z^+ = \begin{bmatrix} U^+ \\ V^+ \end{bmatrix}$ denote the next iterate. Let $S_U = \mathcal{S}(U) \cup \mathcal{S}(U^+) \cup \mathcal{S}(U^*)$ and $S_V = \mathcal{S}(V) \cup \mathcal{S}(V^+) \cup \mathcal{S}(V^*)$. With some abuse of notation, we define the index set $S_Z = S_U \cup S_V$ to represent coordinates of Z corresponding to U_{S_U} and V_{S_V}. For an index set S, let $\mathcal{P}(U, S) = \begin{bmatrix} U_S \\ 0_{S^C} \end{bmatrix}$. Let

$G(U, V) = f(U, V) + g(U, V)$. Finally, let $\Delta_U = U - U^* \widehat{O}$, $\Delta_V = V - V^* \widehat{O}$ and $\Delta_Z = Z - Z^* \widehat{O}$. With these notations, we can write

$$U^+ = \mathrm{Hard}(U - \eta \cdot \nabla G_U(U, V), s_1) = \mathrm{Hard}\left(U - \eta \cdot \mathcal{P}\left(\nabla G_U(U, V), S_U\right), s_1\right)$$

and

$$V^+ = \mathrm{Hard}(V - \eta \cdot \nabla G_V(U, V), s_2) = \mathrm{Hard}\left(V - \eta \cdot \mathcal{P}\left(\nabla G_V(U, V), S_V\right), s_2\right).$$

Let $\widehat{O} \in \mathcal{O}(r)$ be such that

$$d^2(Z, Z^*) = \|U - U^* \widehat{O}\|_F^2 + \|V - V^* \widehat{O}\|_F^2.$$

We have that

$$\begin{aligned} d^2(Z^+, Z^*) &= \min_{O \in \mathcal{O}(r)} \left\| \begin{bmatrix} U^+ \\ V^+ \end{bmatrix} - \begin{bmatrix} U^* O \\ V^* O \end{bmatrix} \right\|_F^2 \\ &\leq \left\| \begin{bmatrix} \mathrm{Hard}\left(U - \eta \cdot \mathcal{P}\left(\nabla G_U(U, V), S_U\right), s_1\right) \\ \mathrm{Hard}\left(V - \eta \cdot \mathcal{P}\left(\nabla G_V(U, V), S_V\right), s_2\right) \end{bmatrix} - \begin{bmatrix} U^* \widehat{O} \\ V^* \widehat{O} \end{bmatrix} \right\|_F^2 \\ &\leq \left(1 + \frac{2}{\sqrt{c-1}}\right) \left\| Z - \eta \cdot \mathcal{P}\left(\nabla G_Z(Z), S_Z\right) - Z^* \widehat{O} \right\|_F^2, \end{aligned}$$

where the last inequality follows from Lemma 3.3 of [204]. Therefore,

$$d^2(Z^+, Z^*) \leq \left(1 + \frac{2}{\sqrt{c-1}}\right) \left[d^2(Z, Z^*) - 2\eta \cdot (T_1 + R_1) + 2\eta^2 \cdot (T_2 + R_2)\right] \tag{4.20}$$

where $T_1 = \langle \mathcal{P}\left(\nabla f_Z(Z), S_Z\right), \Delta_Z \rangle$, $T_2 = \left\| \left[\nabla f_Z(Z)\right]_{S_Z} \right\|_F^2$, $R_2 = \left\| \left[\nabla g_Z(Z)\right]_{S_Z} \right\|_F^2$, and $R_1 = \langle \mathcal{P}\left(\nabla g_Z(Z), S_Z\right), \Delta_Z \rangle$.

For the term T_1, we have

$$\begin{aligned} T_1 &= \left\langle \mathcal{P}\left(\nabla f(UV^\top)V, S_U\right), \Delta_U \right\rangle + \left\langle \mathcal{P}\left(\nabla f(UV^\top)^\top U, S_V\right), \Delta_V \right\rangle \\ &= \underbrace{\left\langle \left[\nabla f(UV^\top) - \nabla f(U^*V^{*\top})\right]_{S_U,S_V}, \left[UV^\top - U^*V^{*\top}\right]_{S_U,S_V} \right\rangle}_{T_{11}} \\ &\quad + \underbrace{\left\langle \left[\nabla f(U^*V^{*\top})\right]_{S_U,S_V}, \left[UV^\top - U^*V^{*\top}\right]_{S_U,S_V} \right\rangle}_{T_{12}} \\ &\quad + \underbrace{\left\langle \left[\nabla f(UV^\top)\right]_{S_U,S_V}, \left[\Delta_U \Delta_V^\top\right]_{S_U,S_V} \right\rangle}_{T_{13}}. \end{aligned}$$

By restricting all the variables to the low-rank and sparse space, Theorem 2.1.11 of [234] gives

$$T_{11} \geq \frac{L \cdot \mu}{L+\mu} \cdot \left\| UV^\top - U^*V^{*\top} \right\|_F^2 + \frac{1}{L+\mu} \cdot \left\| \left[\nabla f(UV^\top) - \nabla f(U^*V^{*\top})\right]_{S_U,S_V} \right\|_F^2$$

Next, we have

$$\begin{aligned} T_{12} &\geq - \left| \left\langle \left[\nabla f(U^*V^{*\top})\right]_{S_U,S_V}, \left[UV^\top - U^*V^{*\top}\right]_{S_U,S_V} \right\rangle \right| \\ &\overset{(i)}{\geq} -e_{\text{stat}} \cdot \left\| UV^\top - U^*V^{*\top} \right\|_F \\ &\overset{(ii)}{\geq} -\frac{1}{2}\frac{L+\mu}{L\cdot\mu} e_{\text{stat}}^2 - \frac{1}{2}\frac{L\cdot\mu}{L+\mu} \cdot \left\| UV^\top - U^*V^{*\top} \right\|_F^2 \end{aligned}$$

where in (i) follows from the definition of statistical error and in (ii) we used the Young's inequality $ab \leq \frac{a^2}{2\epsilon} + \frac{\epsilon b^2}{2}$, for $a, b, \epsilon > 0$. Therefore,

$$\begin{aligned} T_{11} + T_{12} \geq \frac{1}{2}\frac{L\cdot\mu}{L+\mu} \cdot \left\| UV^\top - U^*V^{*\top} \right\|_F^2 - \frac{1}{2}\frac{L+\mu}{L\cdot\mu} \cdot e_{\text{stat}}^2 \\ + \frac{1}{L+\mu} \cdot \left\| \left[\nabla f(UV^\top) - \nabla f(U^*V^{*\top})\right]_{S_U,S_V} \right\|_F^2. \end{aligned} \tag{4.21}$$

Finally, for the term T_{13}, we have

$$\begin{aligned}
T_{13} &\geq -\left|\left\langle \Big[\nabla f(UV^\top)\Big]_{S_U,S_V}, \Big[\Delta_U\Delta_V^\top\Big]_{S_U,S_V}\right\rangle\right| \\
&\geq -\left|\left\langle \Big[\nabla f(U^*V^{*\top})\Big]_{S_U,S_V}, \Big[\Delta_U\Delta_V^\top\Big]_{S_U,S_V}\right\rangle\right| \\
&\qquad\qquad -\left|\left\langle \Big[\nabla f(UV^\top) - \nabla f(U^*V^{*\top})\Big]_{S_U,S_V}, \Big[\Delta_U\Delta_V^\top\Big]_{S_U,S_V}\right\rangle\right| \\
&\geq -\left(e_{\text{stat}} + \Big\| \Big[\nabla f(UV^\top) - \nabla f(U^*V^{*\top})\Big]_{S_U,S_V} \Big\|_F\right) \cdot d^2(Z,Z^*),
\end{aligned}$$

where the last inequality follows from the definition of statistical error and the observation $\|\Delta_U\Delta_V^\top\|_F \leq \|\Delta_V\|_F \cdot \|\Delta_U\|_F \leq d^2(Z,Z^*)$. Under the assumptions,

$$d^2(Z,Z^*) \leq \frac{4\mu_{\min}\sigma_r(\Theta^*)}{5(\mu+L)}$$

and therefore

$$\begin{aligned}
T_{13} &\geq -\left(e_{\text{stat}} + \Big\| \Big[\nabla f(UV^\top) - \nabla f(U^*V^{*\top})\Big]_{S_U,S_V} \Big\|_F\right) \cdot \sqrt{\frac{4\mu_{\min}\sigma_r(\Theta^*)}{5(\mu+L)}} \cdot d(Z,Z^*) \\
&\geq -\frac{1}{2(\mu+L)} \cdot \left(e_{\text{stat}}^2 + \Big\| \Big[\nabla f(UV^\top) - \nabla f(U^*V^{*\top})\Big]_{S_U,S_V} \Big\|_F^2\right) \\
&\qquad\qquad - \frac{4}{5}\mu_{\min}\sigma_r(\Theta^*) \cdot d^2(Z,Z^*).
\end{aligned} \tag{4.22}$$

Combining (4.21) and (4.22) we have

$$\begin{aligned}
T_1 \geq\ & \underbrace{\frac{1}{2}\frac{L\cdot\mu}{L+\mu} \cdot \Big\|UV^\top - U^*V^{*\top}\Big\|_F^2}_{T_{1a}} - \frac{4}{5}\mu_{\min}\sigma_r(\Theta^*) \cdot d^2(Z,Z^*) \\
& - \frac{1}{2}\left(\frac{L+\mu}{L\cdot\mu} + \frac{1}{L+\mu}\right) \cdot e_{\text{stat}}^2 \\
& + \frac{1}{2(L+\mu)} \cdot \Big\| \Big[\nabla f(UV^\top) - \nabla f(U^*V^{*\top})\Big]_{S_U,S_V} \Big\|_F^2.
\end{aligned} \tag{4.23}$$

For the term T_2, we have

$$\begin{aligned}\Big\|[\nabla f(U^*{V^*}^\top)V]_{S_U}\Big\|_F &= \sup_{\|U_{S_U}\|_F=1} \operatorname{tr}\left(\nabla f(U^*{V^*}^\top)VU_{S_U}^\top\right)\\ &= \sup_{\|U_{S_U}\|_F=1} \langle \nabla f(U^*{V^*}^\top), U_{S_U}V^\top\rangle\\ &\leq e_{\text{stat}}\cdot\|V\|_2.\end{aligned}$$

We then have

$$\begin{aligned}&\Big\|[\nabla f(UV^\top)V]_{S_U}\Big\|_F^2\\ &= \Big\|[\nabla f(UV^\top)V - \nabla f(U^*{V^*}^\top)V + \nabla f(U^*{V^*}^\top)V]_{S_U}\Big\|_F^2\\ &\leq 2\Big\|[\nabla f(UV^\top)V - \nabla f(U^*{V^*}^\top)V]_{S_U}\Big\|_F^2 + 2\Big\|[\nabla f(U^*{V^*}^\top)V]_{S_U}\Big\|_F^2\\ &\leq 2\Big\|\left[\nabla f(UV^\top) - \nabla f(U^*{V^*}^\top)\right]_{S_U,S_V}\Big\|_F^2\cdot\|V\|_2^2 + 2e_{\text{stat}}^2\cdot\|V\|_2^2\\ &\leq 2\bigg(\Big\|\left[\nabla f(UV^\top) - \nabla f(U^*{V^*}^\top)\right]_{S_U,S_V}\Big\|_F^2 + e_{\text{stat}}^2\bigg)\cdot\|Z\|_2^2,\end{aligned}$$

where the first inequality follows since $\|A+B\|_F^2 \leq 2\|A\|_F^2 + 2\|B\|_F^2$, and the last inequality follows since $\max(\|U\|_2, \|V\|_2) \leq \|Z\|_2$. Combining the results, we have

$$\begin{aligned}T_2 &= \Big\|[\nabla f(UV^\top)V]_{S_U}\Big\|_F^2 + \Big\|[\nabla f(UV^\top)^\top U]_{S_V}\Big\|_F^2\\ &\leq 4\cdot\bigg(\Big\|\left[\nabla f(UV^\top) - \nabla f(U^*{V^*}^\top)\right]_{S_U,S_V}\Big\|_F^2 + e_{\text{stat}}^2\bigg)\cdot\|Z\|_2^2.\end{aligned}\tag{4.24}$$

For R_1, Lemma B.1 of [240] gives

$$R_1 \geq \underbrace{\frac{1}{8}\Big[\|UU^\top - U^*{U^*}^\top\|_F^2 + \|VV^\top - V^*{V^*}^\top\|_F^2 - 2\|UV^\top - U^*{V^*}^\top\|_F^2\Big]}_{R_{12}} + \underbrace{\frac{1}{2}\|\nabla g\|_F^2}_{R_{11}} - \underbrace{\frac{1}{2}\|\nabla g\|_2\cdot\|\Delta Z\|_F^2}_{R_{13}}.\tag{4.25}$$

For R_{12}, we have that

$$
\begin{aligned}
R_{12} + T_{1a} &= R_{12} + \frac{1}{8}\frac{L\cdot\mu}{L+\mu}\cdot 4\Big\|UV^\top - U^*V^{*\top}\Big\|_F^2 \\
&\geq \mu_{\min}\Big[\big\|UU^\top - U^*U^{*\top}\big\|_F^2 + \big\|VV^\top - V^*V^{*\top}\big\|_F^2 + 2\big\|UV^\top - U^*V^{*\top}\big\|_F^2\Big] \\
&= \mu_{\min}\big\|ZZ^\top - Z^*Z^{*\top}\big\|_F^2 \\
&\geq \frac{4}{5}\mu_{\min}\sigma_r^2(Z^*)\cdot d^2(Z,Z^*) \\
&= \frac{8}{5}\mu_{\min}\sigma_r(\Theta^*)\cdot d^2(Z,Z^*),
\end{aligned} \tag{4.26}
$$

where the first inequality follows from the definition of $\mu_{\min}$, the second inequality follows from Lemma 5.4 of [305], and the last equality follows from $\sigma_r(Z^*) = \sqrt{2\sigma_r(\Theta^*)}$.

For R_{13}, recall that ΔZ satisfies (4.14), we have that

$$
\begin{aligned}
R_{13} &\leq \frac{1}{2}\|\nabla g\|_2\cdot\|\Delta Z\|_F\cdot\sqrt{\frac{8}{5}\mu_{\min}\sigma_r(\Theta^*)} \\
&\leq \frac{2}{5}\mu_{\min}\sigma_r(\Theta^*)\cdot d^2(Z,Z^*) + \frac{1}{4}\|\nabla g\|_F^2.
\end{aligned} \tag{4.27}
$$

Combining (4.23), (4.25), (4.26), and (4.27), we obtain

$$
\begin{aligned}
T_1 + R_1 \geq \frac{2}{5}\mu_{\min}\sigma_r(\Theta^*)\cdot d^2(Z,Z^*) + \frac{1}{4}\|\nabla g\|_F^2 - \frac{1}{2}\left(\frac{L+\mu}{L\cdot\mu} + \frac{1}{L+\mu}\right)\cdot e_{\text{stat}}^2 \\
+ \frac{1}{2(L+\mu)}\cdot\Big\|\Big[\nabla f(UV^\top) - \nabla f(U^*V^{*\top})\Big]_{S_U,S_V}\Big\|_F^2.
\end{aligned} \tag{4.28}
$$

For R_2, we have

$$
R_2 = \|U\nabla g\|_F^2 + \|V\nabla g\|_F^2 \leq 2\|Z\|_2^2\cdot\|\nabla g\|_F^2. \tag{4.29}
$$

Combining (4.24), (4.28), and (4.29), we have

$$\begin{aligned}
&d^2(Z, Z^*) - 2\eta \cdot (T_1 + R_1) + 2\eta^2 \cdot (T_2 + R_2) \\
&\quad \le \left(1 - \eta \cdot \frac{2}{5}\mu_{\min}\sigma_r(\Theta^*)\right) \cdot d^2(Z, Z^*) \\
&\qquad + \eta \left(4\eta \cdot \|Z\|_2^2 - \frac{1}{2(L+\mu)}\right) \cdot \left\| \left[\nabla f(UV^\top) - \nabla f(U^*V^{*\top})\right]_{S_U, S_V} \right\|_F^2 \qquad (4.30) \\
&\qquad + \eta\left(2\eta \cdot \|Z\|_2^2 - \frac{1}{4}\right)\|\nabla g\|_F^2 \\
&\qquad + \eta \left(\frac{L+\mu}{2\mu L} + \frac{1}{2(L+\mu)} + 4\eta \cdot \|Z\|_2^2\right) \cdot e_{\text{stat}}^2.
\end{aligned}$$

Under the choice of the step size,

$$\eta \le \frac{1}{8\|Z\|_2^2} \cdot \min\left\{\frac{1}{2(\mu + L)}, 1\right\},$$

the second term and third term in (4.30) are non-positive and we drop them to get

$$\begin{aligned}
d^2(Z, Z^*) - &2\eta \cdot (T_1 + R_1) + 2\eta^2 \cdot (T_2 + R_2) \\
&\le \left(1 - \eta \cdot \frac{2}{5}\mu_{\min}\sigma_r(\Theta^*)\right) \cdot d^2(Z, Z^*) + \eta \cdot \frac{L+\mu}{L \cdot \mu} \cdot e_{\text{stat}}^2.
\end{aligned} \qquad (4.31)$$

Plugging (4.31) into (4.20) we finish the proof.

4.6.3 Proof of Lemma 41

Comparing (4.9) and (4.15) we see that we only need to show $\|Z\|_2^2 \le 2\|Z_0\|_2^2$. Let $O \in \mathcal{O}(r)$ be such that

$$d^2(Z, Z^*) = \|U - U^*O\|_F^2 + \|V - V^*O\|_F^2.$$

By triangular inequality we have

$$\begin{aligned}\|Z\|_2 &\le \|Z^*O\|_2 + \|Z - Z^*O\|_2 \\ &\le \|Z^*\|_2 + \sqrt{\frac{4}{5}\mu_{\min}\sigma_r(\Theta^*)\cdot\frac{1}{\mu+L}} \\ &\le \|Z^*\|_2 + \sqrt{\frac{4}{5}\cdot\frac{1}{8}\frac{\mu L}{\mu+L}\cdot\frac{1}{2}\sigma_r^2(Z^*)\cdot\frac{1}{\mu+L}} \\ &\le \|Z^*\|_2 + \sqrt{\frac{1}{80}\sigma_r^2(Z^*)} \\ &\le \frac{9}{8}\|Z^*\|_2,\end{aligned} \tag{4.32}$$

where the third inequality follows from the definition of $\mu_{\min}$ and $\sigma_r^2(Z^*) = 2\sigma_r(\Theta^*)$, and the fourth inequality follows from $\frac{ab}{(a+b)^2} \le \frac{1}{4}$. Similarly, we have

$$\begin{aligned}\|Z_0\|_2 &\ge \|Z^*O\|_2 - \|Z_0 - Z^*O\|_2 \\ &\ge \|Z^*\|_2 - \sqrt{\frac{1}{80}\sigma_r^2(Z^*)} \\ &\ge \frac{7}{8}\|Z^*\|_2.\end{aligned} \tag{4.33}$$

Combining (4.32) and (4.33) we have

$$\|Z\|_2 \le \frac{9}{8}\cdot\frac{8}{7}\|Z_0\|_2 \le \sqrt{2}\|Z_0\|_2,$$

which completes the proof.

4.6.4 Proof of Lemma 42

Let $\Omega(s,m)$ denote a collection of subsets of $\{1,\ldots,m\}$ of size s. Let $S_U \in \Omega(s_1,p)$ and $S_V \in \Omega(s_2,k)$ be fixed. With some abuse of notation, let $\mathcal{W}(S_U) = \{U \in \mathbb{R}^{p\times 2r} \mid \|U_{S_U^c}\| = 0, \|U_{S_U}\|_2 = 1\}$ and $\mathcal{W}(S_V) = \{V \in \mathbb{R}^{k\times 2r} \mid \|V_{S_V^c}\| = 0, \|V_{S_V}\|_F = 1\}$. Let $\mathcal{N}_U(\epsilon)$ and $\mathcal{N}_V(\epsilon)$ be the epsilon net of $\mathcal{W}_U$ and $\mathcal{W}_V$, respectively. Using Lemma 10 and Lemma 11 of

[314], we know that $|\mathcal{N}_U(\epsilon)| \leq (3\epsilon^{-1})^{2r\cdot s_1}$, $|\mathcal{N}_V(\epsilon)| \leq (3\epsilon^{-1})^{2r\cdot s_2}$, and

$$\sup_{\substack{U\in\mathcal{W}(S_U)\\ V\in\mathcal{W}(S_V)}} \frac{1}{n}\operatorname{tr}\left(E^\top XUV^\top\right) \leq (1-\epsilon)^{-2} \max_{\substack{U\in\mathcal{N}_U(\epsilon)\\ V\in\mathcal{N}_V(\epsilon)}} \frac{1}{n}\operatorname{tr}\left(E^\top XUV^\top\right).$$

For fixed U and V, the random variable $\operatorname{tr}\left(E^\top XUV^\top\right)$ is a sub-Gaussian with variance proxy $\sigma^2\|X_{S_U}U_{S_U}V_{S_V}^\top\|_F^2$. This variance proxy can be bounded as

$$\sigma^2\|X_{S_U}U_{S_U}V_{S_V}^\top\|_F^2 \leq \sigma^2 \cdot \max_{S_U\in\Omega(s_1,p)} \|(X^\top X)_{S_U S_U}\|_2 = n\sigma^2\bar{\kappa}(s_1).$$

Using a tail bound for sub-Gaussian random variables, we get

$$\frac{1}{n}\operatorname{tr}\left(E^\top XU_{S_U}V_{S_V}^\top\right) \leq 2\sigma\sqrt{\frac{\bar{\kappa}(s_1)\log\frac{1}{\delta}}{n}}$$

with probability at least $1-\delta$. To obtain an upper bound on e_{stat}, we will apply the union bound $\Omega(s_1,p)$, $\Omega(s_2,k)$, $\mathcal{N}_U(\epsilon)$ and $\mathcal{N}_V(\epsilon)$. We set $\epsilon = \frac{1}{2}$ and obtain

$$e_{\text{stat}} \leq 8\sigma\sqrt{\frac{\bar{\kappa}(s_1)}{n}\left(s_1\log p + s_2\log k + 2r(s_1+s_2)\log 6 + \log\frac{1}{\delta}\right)}$$

with probability at least $1-\delta$. Taking $\delta = (p\vee k)^{-1}$ completes the proof.

4.7 Conclusion

We proposed a new GDT algorithm to efficiently solve for optimization problem with simultaneous low rank and row and/or column sparsity structure on the coefficient matrix. We show the linear convergence of GDT algorithm up to statistical error. As an application, for multi-task learning problem we show that the statistical error is near optimal compared to the minimax rate. Experiments on multi-task learning demonstrate competitive performance and much faster running speed compared to existing methods. For future extensions, it would be of interest to extend GDT algorithm to non-linear models. Another potential direction would be to adaptively select the sparsity level s_1 and s_2 in hard thresholding step.

CHAPTER 5

CONVERGENT POLICY OPTIMIZATION FOR SAFE REINFORCEMENT LEARNING

5.1 Introduction

Reinforcement learning [295] has achieved tremendous success in video games [223, 242, 289, 192, 331] and board games, such as chess and Go [280, 282, 281], in part due to powerful simulators [28, 311]. In contrast, due to physical limitations, real-world applications of reinforcement learning methods often need to take into consideration the safety of the agent [14, 113]. For instance, in expensive robotic and autonomous driving platforms, it is pivotal to avoid damages and collisions [108, 31]. In medical applications, we need to consider the switching cost [21].

A popular model of safe reinforcement learning is the constrained Markov decision process (CMDP), which generalizes the Markov decision process by allowing for inclusion of constraints that model the concept of safety [8]. In a CMDP, the cost is associated with each state and action experienced by the agent, and safety is ensured only if the expected cumulative cost is below a certain threshold. Intuitively, if the agent takes an unsafe action at some state, it will receive a huge cost that punishes risky attempts. Moreover, by considering the cumulative cost, the notion of safety is defined for the whole trajectory enabling us to examine the long-term safety of the agent, instead of focusing on individual state-action pairs. For a CMDP, the goal is to take sequential decisions to achieve the expected cumulative reward under the safety constraint.

Solving a CMDP can be written as a linear program [8], with the number of variables being the same as the size of the state and action spaces. Therefore, such an approach is only feasible for the tabular setting, where we can enumerate all the state-action pairs. For large-scale reinforcement learning problems, where function approximation is applied, both the objective and constraint of the CMDP are nonconvex functions of the policy parameter. One common method for solving CMDP is to formulate an unconstrained saddle-point optimization problem via Lagrangian multipliers and solve it using policy optimization algorithms [83, 300]. Such an approach suffers the following two drawbacks:

First, for each fixed Lagrangian multiplier, the inner minimization problem itself can be viewed as solving a new reinforcement learning problem. From the computational point of view, solving the saddle-point optimization problem requires solving a sequence of MDPs with different reward functions. For a large scale problem, even solving a single MDP requires huge computational resources, making such an approach computationally infeasible.

Second, from a theoretical perspective, the performance of the saddle-point approach hinges on solving the inner problem optimally. Existing theory only provides convergence to a stationary point where the gradient with respect to the policy parameter is zero [123, 205]. Moreover, the objective, as a bivariate function of the Lagrangian multiplier and the policy parameter, is not convex-concave and, therefore, first-order iterative algorithms can be unstable [120].

In contrast, we tackle the nonconvex constrained optimization problem of the CMDP directly. We propose a novel policy optimization algorithm, inspired by [207]. Specifically, in each iteration, we replace both the objective and constraint by quadratic surrogate functions and update the policy parameter by solving the new constrained optimization problem. The two surrogate functions can be viewed as first-order Taylor-expansions of the expected reward and cost functions where the gradients are estimated using policy gradient methods [296]. Additionally, they can be viewed as convex relaxations of the original nonconvex reward and cost functions. In Section 5.4 we show that, as the algorithm proceeds, we obtain a sequence of convex relaxations that gradually converge to a smooth function. More importantly, the sequence of policy parameters converges almost surely to a stationary point of the nonconvex constrained optimization problem.

Related work. Our work is pertinent to the line of research on CMDP [8]. For CMDPs with large state and action spaces, [84] proposed an iterative algorithm based on a novel construction of Lyapunov functions. However, their theory only holds for the tabular setting. Using Lagrangian multipliers, [252, 83, 2, 300] proposed policy gradient [296], actor-critic [185], or trust region policy optimization [266] methods for CMDP or constrained risk-sensitive reinforcement learning [113]. These algorithms either do not have convergence guarantees or are shown to converge to saddle-points of the Lagrangian using two-time-scale stochastic approximations [41]. However, due to the projection on the Lagrangian multiplier, the saddle-point achieved by these approaches might not be the stationary point of the original CMDP problem. In addition, [326] proposed a cross-entropy-based stochastic optimization algorithm, and proved the asymptotic behavior using ordinary differential equations. In contrast, our algorithm and the theoretical analysis focus on the discrete time CMDP. Outside of the CMDP setting, [152, 189] studied safe reinforcement learning with demonstration data, [306] studied the safe exploration problem with different safety constraints, and [13] studied multi-task safe reinforcement learning.

Our contribution. Our contribution is three-fold. First, for the CMDP policy optimization problem where both the objective and constraint function are nonconvex, we propose to optimize a sequence of convex relaxation problems using convex quadratic functions. Solving these surrogate problems yields a sequence of policy parameters that converge almost surely to a stationary point of the original policy optimization problem. Second, to reduce the variance in the gradient estimator that is used to construct the surrogate functions, we propose an online actor-critic algorithm. Finally, as concrete applications, our algorithms are also applied to optimal control (Section 5.5.1) and parallel and multi-agent reinforcement learning problems with safety constraints (Section 5.5.2 and 5.5.3).

5.2 Background

A Markov decision process is denoted by $(\mathcal{S}, \mathcal{A}, P, \gamma, r, \mu)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is the transition probability distribution, $\gamma \in (0, 1)$ is the discount factor, $r \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\mu \in \mathcal{P}(\mathcal{S})$ is the distribution of the initial state $s_0 \in \mathcal{S}$, where we denote $\mathcal{P}(\mathcal{X})$ as the set of probability distributions over $\mathcal{X}$ for any $\mathcal{X}$. A policy is a mapping $\pi : \mathcal{S} \to \mathcal{P}(\mathcal{A})$ that specifies the action that an agent will take when it is at state s.

Policy gradient method. Let $\{\pi_\theta \colon \mathcal{S} \to \mathcal{P}(\mathcal{A})\}$ be a parameterized policy class, where $\theta \in \Theta$ is the parameter defined on a compact set Θ. This parameterization transfers the original infinite dimensional policy class to a finite dimensional vector space and enables gradient based methods to be used to maximize (5.1). For example, the most popular Gaussian policy can be written as $\pi(\cdot|s, \theta) = \mathcal{N}\big(\mu(s, \theta), \sigma(s, \theta)\big)$, where the state dependent mean $\mu(s, \theta)$ and standard deviation $\sigma(s, \theta)$ can be further parameterized as $\mu(s, \theta) = \theta_\mu^\top \cdot x(s)$ and $\sigma(s, \theta) = \exp\big(\theta_\sigma^\top \cdot x(s)\big)$ with $x(s)$ being a state feature vector. The goal of an agent is to maximize the expected cumulative reward

$$R(\theta) = \mathbb{E}_\pi \biggl[\sum_{t \geq 0} \gamma^t \cdot r(s_t, a_t)\biggr], \tag{5.1}$$

where $s_0 \sim \mu$, and for all $t \geq 0$, we have $s_{t+1} \sim P(\cdot \,|\, s_t, a_t)$ and $a_t \sim \pi(\cdot \,|\, s_t)$. Given a policy $\pi(\theta)$, we define the state- and action-value functions of π_θ, respectively, as

$$V^\theta(s) = \mathbb{E}_{\pi_\theta} \biggl[\sum_{t \geq 0} \gamma^t r(s_t, a_t) \,\bigg|\, s_0 = s\biggr], \tag{5.2}$$

and

$$Q^\theta(s, a) = \mathbb{E}_{\pi_\theta} \biggl[\sum_{t \geq 0} \gamma^t r(s_t, a_t) \,\bigg|\, s_0 = s, a_0 = a\biggr].$$

The policy gradient method updates the parameter θ through gradient ascent

$$\theta_{k+1} = \theta_k + \eta \cdot \widehat{\nabla}_\theta R(\theta_k),$$

where $\widehat{\nabla}_\theta R(\theta_k)$ is a stochastic estimate of the gradient $\nabla_\theta R(\theta_k)$ at k-th iteration. Policy gradient method, as well as its variants (e.g. policy gradient with baseline [295], neural policy gradient [321, 208, 53]) is widely used in reinforcement learning. The gradient $\nabla_\theta R(\theta)$ can be estimated according to the policy gradient theorem [296],

$$\nabla_\theta R(\theta) = \mathbb{E}\Big[\nabla_\theta \log \pi_\theta(s, a) \cdot Q^\theta(s, a)\Big]. \tag{5.3}$$

Actor-critic method. To further reduce the variance of the policy gradient method, we could estimate both the policy parameter and value function simultaneously. This kind of method is called actor-critic algorithm [185], which is widely used in reinforcement learning. Specifically, in the value function evaluation (*critic*) step we estimate the action-value function $Q^\theta(s, a)$ using, for example, the temporal difference method TD(0) [87]. The policy parameter update (*actor*) step is implemented as before by the Monte-Carlo method according to the policy gradient theorem (5.3) with the action-value $Q^\theta(s, a)$ replaced by the estimated value in the policy evaluation step.

Constrained MDP. In this work, we consider an MDP problem with an additional constraint on the model parameter θ. Specifically, when taking action at some state we incur some cost value. The constraint is such that the expected cumulative cost cannot exceed some pre-defined constant. A constrained Markov decision process (CMDP) is denoted by $(\mathcal{S}, \mathcal{A}, P, \gamma, r, d, \mu)$, where $d \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the cost function and the other parameters are as before. The goal of an agent in CMDP is to solve the following constrained problem

$$\begin{aligned} \underset{\theta\in\Theta}{\text{minimize}} \quad & J(\theta) = \mathbb{E}_{\pi_\theta}\Big[-\sum_{t\geq 0}\gamma^t \cdot r(s_t, a_t)\Big], \\ \text{subject to} \quad & D(\theta) = \mathbb{E}_{\pi_\theta}\Big[\sum_{t\geq 0}\gamma^t \cdot d(s_t, a_t)\Big] \leq D_0, \end{aligned} \tag{5.4}$$

where D_0 is a fixed constant. We consider only one constraint $D(\theta) \leq D_0$, noting that it is straightforward to generalize to multiple constraints. Throughout this chapter, we assume that both the reward and cost value functions are bounded: $\big|r(s_t, a_t)\big| \leq r_{\max}$ and $\big|d(s_t, a_t)\big| \leq d_{\max}$. Also, the parameter space Θ is assumed to be compact.

5.3 Algorithm

In this section, we develop an algorithm to solve the optimization problem (5.4). Note that both the objective function and the constraint in (5.4) are nonconvex and involve expectation without closed-form expression. As a constrained problem, a straightforward approach to solve (5.4) is to define the following Lagrangian function

$$L(\theta, \lambda) = J(\theta) + \lambda \cdot \big[D(\theta) - D_0\big],$$

and solve the dual problem

$$\inf_{\lambda\geq 0} \sup_{\theta} L(\theta, \lambda).$$

However, this problem is a nonconvex minimax problem and, therefore, is hard to solve and establish theoretical guarantees for solutions [3]. Another approach to solve (5.4) is to replace $J(\theta)$ and $D(\theta)$ by surrogate functions with nice properties. For example, one can iteratively construct local quadratic approximations that are strongly convex [267], or are an upper bound for the original function [291]. However, an immediate problem of this naive approach is that, even if the original problem (5.4) is feasible, the convex relaxation problem need not be. Also, these methods only deal with deterministic and/or convex constraints.

In this work, we propose an iterative algorithm that approximately solves (5.4) by constructing a sequence of convex relaxations, inspired by [207]. Our method is able to handle the possible infeasible situation due to the convex relaxation as mentioned above, and handle stochastic and nonconvex constraint. Since we do not have access to $J(\theta)$ or $D(\theta)$, we first define the sample negative cumulative reward and cost functions as

$$J^*(\theta) = -\sum_{t\geq 0} \gamma^t \cdot r(s_t, a_t) \qquad \text{and} \qquad D^*(\theta) = \sum_{t\geq 0} \gamma^t \cdot d(s_t, a_t).$$

Given θ, $J^*(\theta)$ and $D^*(\theta)$ are the sample negative cumulative reward and cost value of a realization (i.e., a trajectory) following policy π_θ. Note that both $J^*(\theta)$ and $D^*(\theta)$ are stochastic due to the randomness in the policy, state transition distribution, etc. With some abuse of notation, we use $J^*(\theta)$ and $D^*(\theta)$ to denote both a function of θ and a value obtained by the realization of a trajectory. Clearly we have $J(\theta) = \mathbb{E}\big[J^*(\theta)\big]$ and $D(\theta) = \mathbb{E}\big[D^*(\theta)\big]$.

We start from some (possibly infeasible) θ_0. Let θ_k denote the estimate of the policy parameter in the k-th iteration. As mentioned above, we do not have access to the expected cumulative reward $J(\theta)$. Instead we sample a trajectory following the current policy π_{θ_k} and obtain a realization of the negative cumulative reward value and the gradient of it as $J^*(\theta_k)$ and $\nabla_\theta J^*(\theta_k)$, respectively. The cumulative reward value is obtained by Monte-Carlo estimation, and the gradient is also obtained by Monte-Carlo estimation according to the policy gradient theorem in (5.3). We provide more details on the realization step later in this section. Similarly, we use the same procedure for the cost function and obtain realizations $D^*(\theta_k)$ and $\nabla_\theta D^*(\theta_k)$.

We approximate $J(\theta)$ and $D(\theta)$ at θ_k by the quadratic surrogate functions

$$\widetilde{J}(\theta, \theta_k, \tau) = J^*(\theta_k) + \langle \nabla_\theta J^*(\theta_k), \theta - \theta_k \rangle + \tau \|\theta - \theta_k\|_2^2, \tag{5.5}$$

$$\widetilde{D}(\theta, \theta_k, \tau) = D^*(\theta_k) + \langle \nabla_\theta D^*(\theta_k), \theta - \theta_k \rangle + \tau \|\theta - \theta_k\|_2^2, \tag{5.6}$$

where $\tau > 0$ is any fixed constant. In each iteration, we solve the optimization problem

$$\overline{\theta}_k = \operatorname*{argmin}_{\theta} \overline{J}^{(k)}(\theta) \qquad \text{subject to} \qquad \overline{D}^{(k)}(\theta) \leq D_0, \tag{5.7}$$

where we define

$$\begin{aligned} \overline{J}^{(k)}(\theta) &= (1 - \rho_k) \cdot \overline{J}^{(k-1)}(\theta) + \rho_k \cdot \widetilde{J}(\theta, \theta_k, \tau), \\ \overline{D}^{(k)}(\theta) &= (1 - \rho_k) \cdot \overline{D}^{(k-1)}(\theta) + \rho_k \cdot \widetilde{D}(\theta, \theta_k, \tau), \end{aligned} \tag{5.8}$$

with the initial value $\overline{J}^{(0)}(\theta) = \overline{D}^{(0)}(\theta) = 0$. Here ρ_k is the weight parameter to be specified later. According to the definition (5.5) and (5.6), problem (5.7) is a convex quadratically constrained quadratic program (QCQP). Therefore, it can be efficiently solved by, for example, the interior point method. However, as mentioned before, even if the original problem (5.4) is feasible, the convex relaxation problem (5.7) could be infeasible. In this case, we instead solve the following feasibility problem

$$\overline{\theta}_k = \operatorname*{argmin}_{\theta, \alpha} \; \alpha \qquad \text{subject to} \qquad \overline{D}^{(k)}(\theta) \leq D_0 + \alpha. \tag{5.9}$$

In particular, we relax the infeasible constraint and find $\overline{\theta}_k$ as the solution that gives the minimum relaxation. Due to the specific form in (5.6), $\overline{D}^{(k)}(\theta)$ is decomposable into quadratic forms of each component of θ, with no terms involving $\theta_i \cdot \theta_j$. Therefore, the solution to problem (5.9) can be written in a closed form. Given $\overline{\theta}_k$ from either (5.7) or (5.9), we update

Algorithm 5 Successive convex relaxation algorithm for constrained MDP

```
1: Input: Initial value $\theta_0$, $\tau$, $\{\rho_k\}, \{\eta_k\}$.
2: for $k = 1, 2, 3, \ldots$ do
3:     Obtain a sample $J^*(\theta_k)$ and $D^*(\theta_k)$ by Monte-Carlo sampling.
4:     Obtain a sample $\nabla_\theta J^*(\theta_k)$ and $\nabla_\theta D^*(\theta_k)$ by policy gradient theorem.
5:     if problem (5.7) is feasible then
6:         Obtain $\overline{\theta}_k$ by solving (5.7).
7:     else
8:         Obtain $\overline{\theta}_k$ by solving (5.9).
9:     end if
10:    Update $\theta_{k+1}$ by (5.10).
11: end for
```

θ_k by

$$\theta_{k+1} = (1 - \eta_k) \cdot \theta_k + \eta_k \cdot \overline{\theta}_k, \tag{5.10}$$

where η_k is the learning rate to be specified later. Note that although we consider only one constraint in the algorithm, both the algorithm and the theoretical result in Section 5.4 can be directly generalized to multiple constraints setting. The whole procedure is summarized in Algorithm 5.

Obtaining realizations $J^*(\theta_k)$ and $\nabla_\theta J^*(\theta_k)$. We detail how to obtain realizations $J^*(\theta_k)$ and $\nabla_\theta J^*(\theta_k)$ corresponding to the lines 3 and 4 in Algorithm 5. The realizations of $D^*(\theta_k)$ and $\nabla_\theta D^*(\theta_k)$ can be obtained similarly.

First, we discuss finite horizon setting, where we can sample the full trajectory according to the policy π_θ. In particular, for any θ_k, we use the policy π_{θ_k} to sample a trajectory and obtain $J^*(\theta_k)$ by Monte-Carlo method. The gradient $\nabla_\theta J(\theta)$ can be estimated by the policy gradient theorem [296],

$$\nabla_\theta J(\theta) = -\mathbb{E}_{\pi_\theta}\Big[\nabla_\theta \log \pi_\theta(s, a) \cdot Q^\theta(s, a)\Big]. \tag{5.11}$$

Again we can sample a trajectory and obtain the policy gradient realization $\nabla_\theta J^*(\theta_k)$ by

Monte-Carlo method.

In infinite horizon setting, we cannot sample the infinite length trajectory. In this case, we utilize the truncation method introduced in [259], which truncates the trajectory at some stage T and scales the undiscounted cumulative reward to obtain an unbiased estimation. Intuitively, if the discount factor γ is close to 0, then the future reward would be discounted heavily and, therefore, we can obtain an accurate estimate with a relatively small number of stages. On the other hand, if γ is close to 1, then the future reward is more important compared to the small γ case and we have to sample a long trajectory. Taking this intuition into consideration, we define T to be a geometric random variable with parameter $1-\gamma$: $\Pr(T=t)=(1-\gamma)\gamma^t$. Then, we simulate the trajectory until stage T and use the estimator $J_{\text{truncate}}(\theta) = -(1-\gamma)\cdot\sum_{t=0}^{T} r(s_t,a_t)$, which is an unbiased estimator of the expected negative cumulative reward $J(\theta)$, as proved in proposition 5 in [241]. We can apply the same truncation procedure to estimate the policy gradient $\nabla_\theta J(\theta)$.

Variance reduction. Using the naive sampling method described above, we may suffer from high variance problem. To reduce the variance, we can modify the above procedure in the following ways. First, instead of sampling only one trajectory in each iteration, a more practical and stable way is to sample several trajectories and take average to obtain the realizations. As another approach, we can subtract a baseline function from the action-value function $Q^\theta(s,a)$ in the policy gradient estimation step (5.11) to reduce the variance without changing the expectation. A popular choice of the baseline function is the state-value function $V^\theta(s)$ as defined in (5.2). In this way, we can replace $Q^\theta(s,a)$ in (5.11) by the advantage function $A^\theta(s,a)$ defined as

$$A^\theta(s,a) = Q^\theta(s,a) - V^\theta(s).$$

This modification corresponds to the standard REINFORCE with Baseline algorithm [295] and can significantly reduce the variance of policy gradient.

Actor-critic method. Finally, we can use an actor-critic update to improve the performance further. In this case, since we need unbiased estimators for both the reward value and its gradient in (5.5) and (5.6) in online fashion, we modify our original problem (5.4) to average reward setting as

$$\begin{aligned}
\underset{\theta\in\Theta}{\text{minimize}} \quad J(\theta) &= \lim_{T\to\infty} \mathbb{E}_{\pi_\theta}\left[-\frac{1}{T}\sum_{t=0}^{T} r(s_t, a_t)\right], \\
\text{subject to} \quad D(\theta) &= \lim_{T\to\infty} \mathbb{E}_{\pi_\theta}\left[\frac{1}{T}\sum_{t=0}^{T} d(s_t, a_t)\right] \le D_0.
\end{aligned}$$

Let $V_\theta^J(s)$ and $V_\theta^D(s)$ denote the value and cost functions corresponding to (5.2). We use possibly nonlinear approximation with parameter w for the value function: $V_w^J(s)$ and v for the cost function: $V_v^D(s)$. In the critic step, we update w and v by TD(0) with step size β_w and β_v; in the actor step, we solve our proposed convex relaxation problem to update θ. The actor-critic procedure is summarized in Algorithm 6. Here J and D are estimators of $J(\theta_k)$ and $D(\theta_k)$. Both of J and D, and the TD error δ^J, δ^D can be initialized as 0.

The usage of the actor-critic method helps reduce variance by using a value function instead of Monte-Carlo sampling. Specifically, in Algorithm 5 we need to obtain a sample trajectory and calculate $J^*(\theta)$ and $\nabla_\theta J^*(\theta)$ by Monte-Carlo sampling. This step has a high variance since we need to sample a potentially long trajectory and sum up a lot of random rewards. In contrast, in Algorithm 6, this step is replaced by a value function $V_w^J(s)$, which reduces the variance.

5.4 Theoretical Result

In this section, we show almost sure convergence of the iterates obtained by our algorithm to a stationary point. We start by stating some mild assumptions on the original problem (5.4) and the choice of some parameters in Algorithm 5.

Assumption 45. *The choice of* $\{\eta_k\}$ *and* $\{\rho_k\}$ *satisfy* $\lim_{k\to\infty}\sum_k \eta_k = \infty$, $\lim_{k\to\infty}\sum_k \rho_k =$

Algorithm 6 Actor-Critic update for constrained MDP

```
for $k = 1, 2, 3, \ldots$ do
    Take action $a$, observe reward $r$, cost $d$, and new state $s'$.
    Critic step:
        $w \leftarrow w + \beta_w \cdot \delta^J \nabla_w V_w^J(s), \quad J \leftarrow J + \beta_w \cdot \big(r - J\big)$.
        $v \leftarrow v + \beta_v \cdot \delta^D \nabla_v V_v^J(s), \quad D \leftarrow D + \beta_v \cdot \big(d - D\big)$.
    Calculate TD error:
        $\delta^J = r - J + V_w^J(s') - V_w^J(s)$.
        $\delta^D = d - D + V_v^D(s') - V_v^D(s)$.
    Actor step:
        Solve $\overline{\theta}_k$ by (5.7) or (5.9) with
          $J^*(\theta_k)$, $\nabla_\theta J^*(\theta_k)$ in (5.5) replaced by $J$ and $\delta^J \cdot \nabla_\theta \log \pi_\theta(s, a)$;
          $D^*(\theta_k)$, $\nabla_\theta D^*(\theta_k)$ in (5.6) replaced by $D$ and $\delta^D \cdot \nabla_\theta \log \pi_\theta(s, a)$.
    $s \leftarrow s'$.
end for
```

∞ and $\lim_{k\to\infty} \sum_k \eta_k^2 + \rho_k^2 < \infty$. Furthermore, we have $\lim_{k\to\infty} \eta_k/\rho_k = 0$ and η_k is decreasing.

Assumption 46. *For any realization, $J^*(\theta)$ and $D^*(\theta)$ are continuously differentiable as functions of θ. Moreover, $J^*(\theta)$, $D^*(\theta)$, and their derivatives are uniformly Lipschitz continuous.*

Assumption 45 allows us to specify the learning rates. A practical choice would be $\eta_k = k^{-c_1}$ and $\rho_k = k^{-c_2}$ with $0.5 < c_2 < c_1 < 1$. This assumption is standard for gradient-based algorithms. Assumption 46 is also standard and is known to hold for a number of models. It ensures that the reward and cost functions are sufficiently regular. In fact, it can be relaxed such that each realization is Lipschitz (not uniformly), and the event that we keep generating realizations with monotonically increasing Lipschitz constant is an event with probability 0. See condition iv) in [337] and the discussion thereafter. Also, see [245] for sufficient conditions such that both the expected cumulative reward function and the gradient of it are Lipschitz.

The following Assumption 47 is useful only when we initialize with an infeasible point. We first state it here and we will discuss this assumption after the statement of the main theorem.

Assumption 47. *Suppose (θ_S, α_S) is a stationary point of the optimization problem*

$$\underset{\theta,\alpha}{\textit{minimize}} \quad \alpha \qquad \textit{subject to} \qquad D(\theta) \leq D_0 + \alpha. \tag{5.12}$$

We have that θ_S is a feasible point of the original problem (5.4), *i.e. $D(\theta_S) \leq D_0$.*

We are now ready to state the main theorem.

Theorem 48. *Suppose the Assumptions 45 and 46 are satisfied with small enough initial step size η_0. Suppose also that, either θ_0 is a feasible point, or Assumption 47 is satisfied. If there is a subsequence $\{\theta_{k_j}\}$ of $\{\theta_k\}$ that converges to some $\widetilde{\theta}$, then there exist uniformly continuous functions $\widehat{J}(\theta)$ and $\widehat{D}(\theta)$ satisfying*

$$\lim_{j\to\infty} \overline{J}^{(k_j)}(\theta) = \widehat{J}(\theta) \qquad \textit{and} \qquad \lim_{j\to\infty} \overline{D}^{(k_j)}(\theta) = \widehat{D}(\theta).$$

Furthermore, suppose there exists θ such that $\widehat{D}(\theta) < D_0$ (i.e. the Slater's condition holds), then $\widetilde{\theta}$ is a stationary point of the original problem (5.4) *almost surely.*

Proof. According to the choice of the surrogate function (5.5) and Assumption 46, it is straightforward to verify that the function $\overline{J}^{(k)}(\theta)$ defined in (5.8) is uniformly strongly convex in θ for each iteration t. Moreover, both $\overline{J}^{(k)}(\theta)$ and $\nabla_\theta \overline{J}^{(k)}(\theta)$ are Lipschitz continuous functions.

From Lemma 1 in [264] we have

$$\lim_{t\to\infty} \left| \overline{J}^{(k)}(\theta) - \mathbb{E}\big[\widetilde{J}(\theta, \theta_k, \tau)\big] \right| = 0.$$

Since the function $\mathbb{E}\big[\widetilde{J}(\theta, \theta_k, \tau)\big]$ is Lipschitz continuous in θ_k, we obtain that

$$\left| \overline{J}^{(k_1)}(\theta) - \overline{J}^{(k_2)}(\theta) \right| \leq L_0 \cdot \|\theta_{k_1} - \theta_{k_2}\| + \epsilon,$$

for some constant L_0 and the error term ϵ that goes to 0 as k_1, k_2 go to infinity. This shows that the function sequence $\overline{J}^{(k_j)}(\theta)$ is equicontinuous. Since Θ is compact and the discounted cumulative reward function is bounded by $r_{\max}/(1-\gamma)$, we can apply Arzela-Ascoli theorem [93, 170] to prove existence of $\widehat{J}(\theta)$ that converges uniformly. Moreover, since we apply the same operations on the constraint function $D(\theta)$ as to the reward function $J(\theta)$ in Algorithm 5, the above properties also hold for $D(\theta)$.

The rest of the proof follows in a similar way as the proof of Theorem 1 in [207]. Under Assumptions 45 - 47, the technical conditions in [207] are satisfied by the choice of the surrogate functions (5.5) and (5.6). According to Lemma 2 in [207], with probability one we have

$$\limsup_{k\to\infty} D(\theta_k) \leq D_0.$$

This shows that, although in some of the iterations the convex relaxation problem (5.7) is infeasible, and we have to solve the alternative problem (5.9), the iterates $\{\theta_k\}$ converge to the feasible region of the original problem (5.4) with probability one. Furthermore, with probability one, the convergent point $\widetilde{\theta}$ is the optimal solution to the following problem

$$\underset{\theta\in\Theta}{\text{minimize}} \quad \widehat{J}(\theta) \qquad \text{subject to} \qquad \widehat{D}(\theta) \leq D_0. \tag{5.13}$$

The KKT conditions for (5.13) together with the Slater condition show that the KKT conditions of the original problem (5.4) are also satisfied at $\widetilde{\theta}$. This shows that $\widetilde{\theta}$ is a stationary point of the original problem almost surely.

□

Note that Assumption 47 is not necessary if we start from a feasible point, or we reach a feasible point in the iterates, which could be viewed as an initializer. Assumption 47 makes sure that the iterates in Algorithm 5 keep making progress without getting stuck at any infeasible stationary point. A similar condition is assumed in [207] for an infeasible initializer. If it turns out that θ_0 is infeasible and Assumption 47 is violated, then the convergent point may be an infeasible stationary point of (5.12). In practice, if we can find a feasible point of the original problem, then we proceed with that point. Alternatively, we could generate multiple initializers and obtain iterates for all of them. As long as there is a feasible point in one of the iterates, we can view this feasible point as the initializer and Theorem 48 follows without Assumption 47. In our later experiments, for every single replicate, we could reach a feasible point, and therefore Assumption 47 is not necessary.

Our algorithm does not guarantee safe exploration during the training phase. Ensuring safety during learning is a more challenging problem. Sometimes even finding a feasible point is not straightforward, otherwise Assumption 47 is not necessary.

Our proposed algorithm is inspired by [207]. Compared to [207] which deals with an optimization problem, solving the safe reinforcement learning problem is more challenging. We need to verify that the Lipschitz condition is satisfied, and also the policy gradient has to be estimated (instead of directly evaluated as in a standard optimization problem). The usage of the Actor-Critic algorithm reduces the variance of the sampling, which is unique to reinforcement learning.

5.5 Applications

In this section, we apply our algorithm to some specific MDP models and show how it works for these models in details.

5.5.1 Constrained Linear-Quadratic Regulator

We apply our algorithm to the linear-quadratic regulator (LQR), which is one of the most fundamental problems in control theory. In the LQR setting, the state dynamic equation is linear, the cost function is quadratic, and the optimal control theory tells us that the optimal control for LQR is a linear function of the state [97, 15]. LQR can be viewed as an MDP problem and it has attracted a lot of attention in the reinforcement learning literature [43, 44, 89, 255].

We consider the infinite-horizon, discrete-time LQR problem. Denote x_t as the state variable and u_t as the control variable. The state transition and the control sequence are given by

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + v_t, \\ u_t &= -Fx_t + w_t, \end{aligned} \tag{5.14}$$

where v_t and w_t represent possible Gaussian white noise, and the initial state is given by x_0. The goal is to find the control parameter matrix F such that the expected total cost is minimized. The usual cost function of LQR corresponds to the negative reward in our setting and we impose an additional quadratic constraint on the system. The overall optimization problem is given by

$$\begin{aligned} \text{minimize} \quad & J(F) = \mathbb{E}\Bigg[\sum_{t\geq 0} x_t^\top Q_1 x_t + u_t^\top R_1 u_t\Bigg], \\ \text{subject to} \quad & D(F) = \mathbb{E}\Bigg[\sum_{t\geq 0} x_t^\top Q_2 x_t + u_t^\top R_2 u_t\Bigg] \leq D_0, \end{aligned}$$

where Q_1, Q_2, R_1, and R_2 are positive definite matrices. Note that even thought the matrices are positive definite, both the objective function J and the constraint D are nonconvex with respect to the parameter F. Furthermore, with the additional constraint, the optimal control sequence may no longer be linear in the state x_t. Nevertheless, in this work, we still consider linear control given by (5.14) and the goal is to find the best linear control for this constrained LQR problem. We assume that the choice of A, B are such that the optimal cost is finite.

Random initial state. We first consider the setting where the initial state $x_0 \sim \mathcal{D}$ follows a random distribution $\mathcal{D}$, while both the state transition and the control sequence (5.14) are deterministic (i.e. $v_t = w_t = 0$). In this random initial state setting, [105] showed that without the constraint, the policy gradient method converges efficiently to the global optima in polynomial time. In the constrained case, we can explicitly write down the objective and constraint function, since the only randomness comes from the initial state. Therefore, we have the state dynamic $x_{t+1} = (A - BF)x_t$ and the objective function has the following expression ([105], Lemma 1)

$$J(F) = \mathbb{E}_{x_0 \sim \mathcal{D}}\big[x_0^\top P_F x_0\big], \tag{5.15}$$

where P_F is the solution to the following equation

$$P_F = Q_1 + F^\top R_1 F + (A - BF)^\top P_F (A - BF). \tag{5.16}$$

The gradient is given by

$$\nabla_F J(F) = 2\Big(\big(R_1 + B^\top P_F B\big)F - B^\top P_F A\Big) \cdot \Bigg[\mathbb{E}_{x_0 \sim \mathcal{D}} \sum_{t=0}^{\infty} x_t x_t^\top\Bigg]. \tag{5.17}$$

Let $S_F = \sum_{t=0}^{\infty} x_t x_t^\top$ and observe that

$$S_F = x_0 x_0^\top + (A - BF) S_F (A - BF)^\top. \tag{5.18}$$

We start from some F_0 and apply our Algorithm 5 to solve the constrained LQR problem. In iteration k, with the current estimator denoted by F_k, we first obtain an estimator of P_{F_k} by starting from Q_1 and iteratively applying the recursion $P_{F_k} \leftarrow Q_1 + F_k^\top R_1 F_k + (A - BF_k)^\top P_{F_k}(A - BF_k)$ until convergence. Next, we sample an x_0^* from the distribution $\mathcal{D}$ and follow a similar recursion given by (5.18) to obtain an estimate of S_{F_k}. Plugging the sample x_0^* and the estimates of P_{F_k} and S_{F_k} into (5.15) and (5.17), we obtain the sample reward value $J^*(F_k)$ and $\nabla_F J^*(F_k)$, respectively. With these two values, we follow (5.5) and (5.8) and obtain $\overline{J}^{(k)}(F)$. We apply the same procedure to the cost function $D(F)$ with Q_1, R_1 replaced by Q_2, R_2 to obtain $\overline{D}^{(k)}(F)$. Finally we solve the optimization problem (5.7) (or (5.9) if (5.7) is infeasible) and obtain F_{k+1} by (5.10).

Random state transition and control. We then consider the setting where both v_t and w_t are independent standard Gaussian white noise. In this case, the state dynamic can be written as $x_{t+1} = (A - BF)x_t + \epsilon_t$ where $\epsilon_t \sim \mathcal{N}(0, I + BB^\top)$. Let P_F be defined as in (5.16) and S_F be the solution to the following Lyapunov equation

$$S_F = I + BB^\top + (A - BF)S_F(A - BF)^\top.$$

The objective function has the following expression ([341], Proposition 3.1)

$$J(F) = \mathbb{E}_{x \sim \mathcal{N}(0,S_F)}\Big[x^\top(Q_1 + F^\top R_1 F)x\Big] + \operatorname{tr}(R_1), \tag{5.19}$$

and the gradient is given by

$$\nabla_F J(F) = 2\Big(\big(R_1 + B^\top P_F B\big)F - B^\top P_F A\Big) \cdot \mathbb{E}_{x \sim \mathcal{N}(0,S_F)}\Big[xx^\top\Big]. \tag{5.20}$$

Although in this setting it is straightforward to calculate the expectation in a closed form, we keep the current expectation form to be in line with our algorithm. Moreover, when the error distribution is more complicated or unknown, we can no longer calculate the closed form expression and have to sample in each iteration. With the formulas given by (5.19) and (5.20), we again apply our Algorithm 5. We sample $x \sim \mathcal{N}(0, S_F)$ in each iteration and solve the optimization problem (5.7) or (5.9). The whole procedure is similar to the random initial state case described above.

5.5.2 Constrained Parallel Markov Decision Process

We consider the parallel MDP problem [187, 229, 73] where we have a single-agent MDP task and N workers, where each worker acts as an individual agent and aims to solve the *same* MDP problem. In the parallel MDP setting, each agent is characterized by a tuple $(\mathcal{S}, \mathcal{A}, P, \gamma, r^i, d^i, \mu^i)$, where each agent has the same but individual state space, action space, transition probability distribution, and the discount factor. However, the reward function, cost function, and the distribution of the initial state $s_0 \in \mathcal{S}$ could be different for each agent, but satisfy $\mathbb{E}[r^i(s,a)] = r(s,a)$, $\mathbb{E}[d^i(s,a)] = d(s,a)$, and $\mathbb{E}[\mu^i(s,a)] = \mu(s,a)$. Each agent i generates its own trajectory $\{s_0^i, a_0^i, s_1^i, a_1^i, \dots\}$ and collects its own reward/cost value $\{r_0^i, d_0^i, r_1^i, d_1^i, \dots\}$.

The hope is that by solving the single-agent problem using N agents in parallel, the algorithm could be more stable and converge much faster [222]. Intuitively, each agent i may have a different initial state and will explore different parts of the state space due to the randomness in the state transition distribution and the policy. It also helps to reduce the correlation between agents' behaviors. As a result, by running multiple agents in parallel, we are more likely to visit different parts of the environment and get the experience of the reward/cost function values more efficiently. This mimics the strategy used in tree-based supervised learning algorithms [45, 143, 142].

Following the settings in [73], we have N agents (i.e., N workers) and one central controller in the system. The global parameter is denoted by θ, and we consider the constrained parallel MDP problem where the goal is to solve the following optimization problem:

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & J(\theta) = \sum_{i=1}^{N} \mathbb{E}_{\pi_\theta}\Big[-\sum_{t\geq 0} \gamma^t \cdot r^i(s_t^i, a_t^i)\Big], \\ \text{subject to} \quad & D(\theta) = \mathbb{E}_{\pi_\theta}\Big[\sum_{t\geq 0} \gamma^t \cdot d^i(s_t^i, a_t^i)\Big] \leq D_0, \quad i \in \mathcal{N}. \end{aligned}$$

During the estimation step, the controller broadcasts the current parameter θ_k to each agent and each agent samples its own trajectory and obtains estimators for function value/gradient of the reward/cost function. Next, each agent uploads its estimators to the central controller and the central controller takes the average over these estimators, and then follow our proposed algorithm to solve for the QCQP problem and update the parameter to θ_{k+1}. This process continues until convergence.

5.5.3 Constrained Multi-agent Markov Decision Process

A natural extension of the (single-agent) MDP is to consider a model with N agents termed multi-agent Markov decision process (MMDP). Recently this kind of problem has been attracting more and more attention. See [51] for a comprehensive survey. Most of the work on multi-agent MDP problems consider the setting where the agents share the same global state space, but each with their own collection of actions and rewards [42, 318, 363]. In each stage of the system, each agent observes the global state and chooses its own action individually. As a result, each agent receives its reward and the state evolves according to the joint transition distribution. An MMDP problem can be fully collaborative where all the agents have the same goal, or fully competitive where the problem consists of two agents with an opposite goal, or the mix of the two.

Here we consider a slightly different setting where each agent has its own state space. The only connection between the agents is that the global reward is a function of the overall states and actions. Furthermore, each agent has its own constraint which depends on its own state and action only. This problem is known as Transition-Independent Multi-agent MDP and is considered in [265]. Specifically, each agent's task is characterized by a tuple $(\mathcal{S}^i, \mathcal{A}^i, P^i, \gamma, d^i, \mu^i)$ with each component defined as usual. Note that $P^i \colon \mathcal{S}^i \times \mathcal{A}^i \to \mathcal{P}(\mathcal{S}^i)$ and $d^i \colon \mathcal{S}^i \times \mathcal{A}^i \to \mathbb{R}$ are functions of each agent's state and action only and do not depend on other agents. Denote $\mathcal{S} = \Pi_{i \in \mathcal{N}} \mathcal{S}^i$ and $\mathcal{A} = \Pi_{i \in \mathcal{N}} \mathcal{A}^i$ as the joint state space and action space. The global reward function is given by $r \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ that depends on the joint state and action. Here we consider the fully collaborative setting where all the agents have the same goal. Under this setting, the policy set of each agent is parameterized as $\{\pi^i_{\theta^i} \colon \mathcal{S}^i \to \mathcal{P}(\mathcal{A}^i)\}$ and we denote $\theta = [\theta^1, \ldots, \theta^N]$ as the overall parameters and π_θ as the overall policy. In the following, we use $\mathcal{N} = \{1, 2, \ldots, N\}$ to denote the N agents. Denote a^i_t as the action chosen by agent i at stage t and $a_t = \Pi_{i \in \mathcal{N}}\ a^i_t$ as the joint action chosen by all the agents. The goal of this constrained MMDP is to solve the following problem

$$\begin{aligned} \underset{\theta}{\text{minimize}} \quad & J(\theta) = \mathbb{E}_{\pi_\theta}\Big[-\sum_{t\geq 0}\gamma^t \cdot r(s_t, a_t)\Big], \\ \text{subject to} \quad & D^i(\theta^i) = \mathbb{E}_{\pi_{\theta^i}}\Big[\sum_{t\geq 0}\gamma^t \cdot d^i(s_t^i, a_t^i)\Big] \leq D_0^i, \quad i \in \mathcal{N}. \end{aligned} \tag{5.21}$$

Inspired by the parallel implementation ([207], Section V), our algorithm applies naturally to constrained MMDP problem with some modifications. This modified procedure can also be viewed as a distributed version of the original algorithm. The overall problem (5.21) can be viewed as a large "single-agent" problem where the constraints are decomposable into N parts. In this case, instead of solving a large QCQP problem in each iteration, each agent could solve its own QCQP problem in a distributed manner which is much more efficient. As before, we denote the sample negative reward and cost function as

$$J^*(\theta) = -\sum_{t\geq 0}\gamma^t \cdot r(s_t, a_t) \qquad \text{and} \qquad D^{i,*}(\theta^i) = \sum_{t\geq 0}\gamma^t \cdot d^i(s_t^i, a_t^i).$$

In each iteration with $\theta_k = [\theta_k^1, ..., \theta_k^N]$, we approximate $J(\theta)$ and $D(\theta)$ as

$$\begin{aligned} \widetilde{J}^i(\theta^i, \theta_k, \tau) &= \frac{1}{N}J^*(\theta_k) + \langle \nabla_{\theta^i} J^*(\theta_k), \theta^i - \theta_k^i\rangle + \tau\|\theta^i - \theta_k^i\|_2^2, \\ \widetilde{D}^i(\theta^i, \theta_k, \tau) &= D^{i,*}(\theta_k^i) + \langle \nabla_{\theta^i} D^{i,*}(\theta_k^i), \theta^i - \theta_k^i\rangle + \tau\|\theta^i - \theta_k^i\|_2^2. \end{aligned}$$

Note that the constraint function is naturally decomposable into N parts. We also "manually" split the objective function into N parts, so that each agent could solve its own QCQP problem in a distributed manner. As before, we define

$$\begin{aligned} \overline{J}^{i,(k)}(\theta^i) &= (1-\rho_k)\cdot \overline{J}^{i,(k-1)}(\theta^i) + \rho_k \cdot \widetilde{J}^i(\theta^i, \theta_k, \tau), \\ \overline{D}^{i,(k)}(\theta^i) &= (1-\rho_k)\cdot \overline{D}^{i,(k-1)}(\theta^i) + \rho_k \cdot \widetilde{D}^i(\theta^i, \theta_k, \tau). \end{aligned}$$

With this surrogate functions, each agent then solves its own convex relaxation problem

$$\overline{\theta}_k^i = \underset{\theta^i}{\operatorname{argmin}} \ \overline{J}^{i,(k)}(\theta^i) \qquad \text{subject to} \qquad \overline{D}^{i,(k)}(\theta^i) \le D_0^i, \tag{5.22}$$

or, alternatively, solves for the feasibility problem if (5.22) is infeasible

$$\overline{\theta}_k^i = \underset{\theta^i,\alpha^i}{\operatorname{argmin}} \ \alpha^i \qquad \text{subject to} \qquad \overline{D}^{i,(k)}(\theta^i) \le D_0^i + \alpha^i.$$

This step can be implemented in a distributed manner for each agent and is more efficient than solving the overall problem with the overall parameter θ. Finally, the update rule for each agent i is as usual

$$\theta_{t+1}^i = (1 - \eta_k) \cdot \theta_k^i + \eta_k \cdot \overline{\theta}_k^i.$$

This process continues until convergence.

5.6 Experiment

We verify the effectiveness of the proposed algorithm through experiments. We focus on the LQR setting with a random initial state as discussed in Section 5.5.1. In this experiment we set $x \in \mathbb{R}^{15}$ and $u \in \mathbb{R}^8$. The initial state distribution is uniform on the unit cube: $x_0 \sim \mathcal{D} = \text{Uniform}\big([-1,1]^{15}\big)$. Each element of A and B is sampled independently from the standard normal distribution and scaled such that the eigenvalues of A are within the range $(-1, 1)$. We initialize F_0 as an all-zero matrix, and the choice of the constraint function and the value D_0 are such that (1) the constrained problem is feasible; (2) the solution of the unconstrained problem does not satisfy the constraint, i.e., the problem is not trivial; (3) the initial value F_0 is not feasible. The learning rates are set as $\eta_k = \frac{2}{3}k^{-3/4}$ and $\rho_k = \frac{2}{3}k^{-2/3}$. The conservative choice of step size is to avoid the situation where an eigenvalue of $A - BF$ runs out of the range $(-1, 1)$, and so the system is stable. [1]

[1] The code is available at `https://github.com/ming93/Safe_reinforcement_learning`

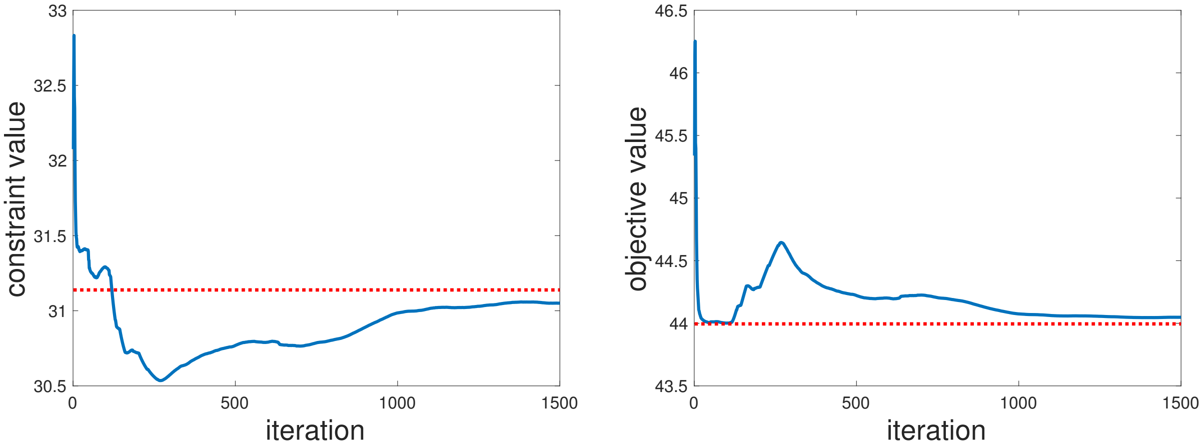
	min value	# iterations	approx. min value	approx. # iterations
Our method	30.689 ± 0.114	2001 ± 1172	30.694 ± 0.114	604.3 ± 722.4
Lagrangian	30.693 ± 0.113	7492 ± 1780	30.699 ± 0.113	5464 ± 2116

Table 5.1: Comparison of our method with Lagrangian method

Figure 5.1(a) and 5.1(b) show the constraint and objective value of one replicate in each iteration, respectively. The red horizontal line in Figure 5.1(a) is for D_0, while the horizontal line in Figure 5.1(b) is for the unconstrained minimum objective value. We can see from Figure 5.1(a) that we start from an infeasible point, and the problem becomes feasible after about 100 iterations. The objective value is in general decreasing after becoming feasible, but never lower than the unconstrained minimum, as shown in Figure 5.1(b).

Comparison with the Lagrangian method. We compare our proposed method with the usual Lagrangian method. For the Lagrangian method, we follow the algorithm proposed in [83] for safe reinforcement learning, which iteratively applies gradient descent on the parameter F and gradient ascent on the Lagrangian multiplier λ for the Lagrangian function until convergence.

Table 5.1 reports the comparison results with mean and standard deviation based on 50 replicates. In the second and third columns, we compare the minimum objective value and the number of iterations to achieve it. We also consider an approximate version, where we are satisfied with the result if the objective value exceeds less than 0.02% of the minimum value. The fourth and fifth columns show the comparison results for this approximate version. We can see that both methods achieve similar minimum objective values, but ours requires less number of policy updates, for both minimum and approximate minimum version.



(a) Constraint value $D(\theta_k)$ in each iteration.

(b) Objective value $J(\theta_k)$ in each iteration.

Figure 5.1: An experiment on constrained LQR problem. The iterate starts from an infeasible point and then becomes feasible and eventually converges.

REFERENCES

[1] Scott Aaronson. The learnability of quantum states. *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.*, 463(2088):3089–3114, 2007.

[2] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International Conference on Machine Learning*, pages 22–31, 2017.

[3] Leonard Adolphs. Non convex-concave saddle point optimization. Master's thesis, ETH Zurich, 2018.

[4] Alekh Agarwal, Sahand Negahban, and Martin J. Wainwright. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *The Annals of Statistics*, 40(2):1171–1197, apr 2012.

[5] J. Aitchison and S. D. Silvey. Maximum-likelihood estimation of parameters subject to restraints. *Ann. Math. Statist.*, 29:813–828, 1958.

[6] J. Akerboom, T.-W. Chen, T. J. Wardill, L. Tian, J. S. Marvin, S. Mutlu, N. C. Calderon, F. Esposti, B. G. Borghuis, X. R. Sun, A. Gordus, M. B. Orger, R. Portugues, F. Engert, J. J. Macklin, A. Filosa, A. Aggarwal, R. A. Kerr, R. Takagi, S. Kracun, E. Shigetomi, B. S. Khakh, H. Baier, L. Lagnado, S. S.-H. Wang, C. I. Bargmann, B. E. Kimmel, V. Jayaraman, K. Svoboda, D. S. Kim, E. R. Schreiter, and L. L. Looger. Optimization of a GCaMP calcium indicator for neural activity imaging. *Journal of Neuroscience*, 32(40):13819–13840, oct 2012.

[7] Magne Aldrin. Moderate projection pursuit regression for multivariate response data. *Computational statistics & data analysis*, 21(5):501–531, 1996.

[8] Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

[9] Takeshi Amemiya. *Advanced econometrics*. Harvard university press, 1985.

[10] Arash A Amini and Martin J Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. In *2008 IEEE International Symposium on Information Theory*, pages 2454–2458. IEEE, 2008.

[11] Arash A. Amini and Martin J. Wainwright. High-dimensional analysis of semidefinite relaxations for sparse principal components. *Ann. Statist.*, 37(5B):2877–2921, 2009.

[12] Yonatan Amit, Michael Fink, Nathan Srebro, and Shimon Ullman. Uncovering shared structures in multiclass classification. In *Proceedings of the 24th international conference on Machine learning*, pages 17–24. ACM, 2007.

[13] Haitham Bou Ammar, Rasul Tutunov, and Eric Eaton. Safe policy search for lifelong reinforcement learning with sublinear regret. In *International Conference on Machine Learning*, pages 2361–2369, 2015.

[14] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[15] Brian DO Anderson and John B Moore. *Optimal control: linear quadratic methods*. Courier Corporation, 2007.

[16] Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal Of Machine Learning Research*, 6:1817–1853, 2005.

[17] Donald WK Andrews. Hypothesis testing with a restricted parameter space. *Journal of Econometrics*, 84(1):155–199, 1998.

[18] Donald WK Andrews. Testing when a parameter is on the boundary of the maintained hypothesis. *Econometrica*, pages 683–734, 2001.

[19] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[20] Mohammad Taha Bahadori, Zemin Zheng, Yan Liu, and Jinchi Lv. Scalable interpretable multi-response regression via seed. *Technical report*, 2016.

[21] Yu Bai, Tengyang Xie, Nan Jiang, and Yu-Xiang Wang. Provably efficient q-learning with low switching cost. *arXiv preprint arXiv:1905.12849*, 2019.

[22] Sivaraman Balakrishnan, Mladen Kolar, Alessandro Rinaldo, and Aarti Singh. Recovering block-structured activations using compressive measurements. *Electron. J. Statist.*, 11(1):2647–2678, 2017.

[23] Sivaraman Balakrishnan, Mladen Kolar, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. Statistical and computational tradeoffs in biclustering. In *NIPS 2011 Workshop on Computational Trade-offs in Statistical Learning*, volume 4, 2011.

[24] Maria-Florina Balcan, Vitaly Feldman, and Csaba Szepesvári, editors. *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, volume 35 of *JMLR Workshop and Conference Proceedings*. JMLR.org, 2014.

[25] Ray Ball, Joseph Gerakos, Juhani T Linnainmaa, and Valeri Nikolaev. Accruals, cash flows, and operating profitability in the cross section of stock returns. *Journal of Financial Economics*, 121(1):28–45, 2016.

[26] Rina Foygel Barber and Wooseok Ha. Gradient descent with nonconvex constraints: local concavity determines convergence. *Technical report*, 2017.

[27] Rina Foygel Barber, Mladen Kolar, et al. Rocket: Robust confidence intervals via kendall's tau for transelliptical graphical models. *The Annals of Statistics*, 46(6B):3422–3450, 2018.

[28] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[29] Alexandre Belloni, Victor Chernozhukov, and Christian Hansen. Inference on treatment effects after selection among high-dimensional controls. *The Review of Economic Studies*, 81(2):608–650, 2014.

[30] Alexandre Belloni, Victor Chernozhukov, Christian Hansen, and Damian Kozbur. Inference in high-dimensional panel models with an application to gun control. *Journal of Business & Economic Statistics*, 34(4):590–605, 2016.

[31] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, pages 908–918, 2017.

[32] Quentin Berthet and Philippe Rigollet. Optimal detection of sparse principal components in high dimension. *Ann. Statist.*, 41(4):1780–1815, 2013.

[33] D. P. Bertsekas and J. N. Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 1, pages 560–564 vol.1, Dec 1995.

[34] Michael J Best and Nilotpal Chakravarti. Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, 47(1-3):425–439, 1990.

[35] Srinadh Bhojanapalli, Anastasios Kyrillidis, and Sujay Sanghavi. Dropping convexity for faster semi-definite optimization. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 530–582, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

[36] Srinadh Bhojanapalli, Behnam Neyshabur, and Nati Srebro. Global optimality of local search for low rank matrix recovery. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3873–3881. Curran Associates, Inc., 2016.

[37] Peter J Bickel, Chris AJ Klaassen, Peter J Bickel, Ya'acov Ritov, J Klaassen, Jon A Wellner, and YA'Acov Ritov. *Efficient and adaptive estimation for semiparametric models*, volume 4. Johns Hopkins University Press Baltimore, 1993.

[38] John R Birge, Ozan Candogan, Hongfan Chen, and Daniela Saban. Optimal commissions and subscriptions in networked markets. *Forthcoming in Manufacturing & Service Operations Management*, 2019.

[39] Aharon Birnbaum, Iain M. Johnstone, Boaz Nadler, and Debashis Paul. Minimax bounds for sparse PCA with noisy high-dimensional data. *Ann. Statist.*, 41(3):1055–1084, 2013.

[40] Daniel A Bloch, Tze Leung Lai, and Pascale Tubert-Bitter. One-sided tests in clinical trials with multiple endpoints. *Biometrics*, 57(4):1039–1047, 2001.

[41] Vivek S Borkar. Stochastic approximation with two time scales. *Systems & Control Letters*, 29(5):291–294, 1997.

[42] Craig Boutilier. Planning, learning and coordination in multiagent decision processes. In *Proceedings of the 6th conference on Theoretical aspects of rationality and knowledge*, pages 195–210. Morgan Kaufmann Publishers Inc., 1996.

[43] Steven J Bradtke. Reinforcement learning applied to linear quadratic regulation. In *Advances in neural information processing systems*, pages 295–302, 1993.

[44] Steven J Bradtke, B Erik Ydstie, and Andrew G Barto. Adaptive linear quadratic control using policy iteration. In *Proceedings of the American control conference*, volume 3, pages 3475–3475. Citeseer, 1994.

[45] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[46] Florentina Bunea, Yiyuan She, and Marten H. Wegkamp. Optimal selection of reduced rank estimators of high-dimensional matrices. *The Annals of Statistics*, 39(2):1282–1309, apr 2011.

[47] Florentina Bunea, Yiyuan She, and Marten H. Wegkamp. Joint variable and rank selection for parsimonious estimation of high-dimensional matrices. *Ann. Statist.*, 40(5):2359–2388, 2012.

[48] Samuel Burer and Renato D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Math. Program.*, 95(2, Ser. B):329–357, 2003. Computational semidefinite and second order cone programming: the state of the art.

[49] Samuel Burer and Renato D. C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Math. Program.*, 103(3, Ser. A):427–444, 2005.

[50] Adolf Buse. The likelihood ratio, wald, and lagrange multiplier tests: An expository note. *The American Statistician*, 36(3a):153–157, 1982.

[51] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, And Cybernetics-Part C: Applications and Reviews, 38 (2), 2008*, 2008.

[52] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.*, 20(4):1956–1982, 2010.

[53] Qi Cai, Zhuoran Yang, Jason D Lee, and Zhaoran Wang. Neural temporal-difference learning converges to global optima. *arXiv preprint arXiv:1905.10027*, 2019.

[54] T Tony Cai, Xiaodong Li, Zongming Ma, et al. Optimal rates of convergence for noisy sparse phase retrieval via thresholded wirtinger flow. *The Annals of Statistics*, 44(5):2221–2251, 2016.

[55] T. Tony Cai, Zongming Ma, and Yihong Wu. Sparse PCA: optimal rates and adaptive estimation. *Ann. Statist.*, 41(6):3074–3110, 2013.

[56] T. Tony Cai and Anru Zhang. ROP: matrix recovery via rank-one projections. *Ann. Statist.*, 43(1):102–138, 2015.

[57] Daniele Calandriello, Alessandro Lazaric, and Marcello Restelli. Sparse multi-task reinforcement learning. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 819–827. Curran Associates, Inc., 2014.

[58] Emmanuel J Candes, Yonina C Eldar, Thomas Strohmer, and Vladislav Voroninski. Phase retrieval via matrix completion. *SIAM review*, 57(2):225–251, 2015.

[59] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3):Art. 11, 37, 2011.

[60] Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, jun 2010.

[61] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009.

[62] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Trans. Inform. Theory*, 56(5):2053–2080, 2010.

[63] Jianfei Cao and Connor Dowd. Estimation and inference for synthetic control methods with spillover effects. *arXiv preprint arXiv:1902.07343*, 2019.

[64] Jianfei Cao, Chris Gu, and Yike Wang. Principal component and static factor analysis. In *Macroeconomic Forecasting in the Era of Big Data*, pages 229–266. Springer, 2020.

[65] Jianfei Cao and Shirley Lu. Synthetic control inference for staggered adoption: Estimating the dynamic effects of board gender diversity policies. *arXiv preprint arXiv:1912.06320*, 2019.

[66] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[67] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM J. Optim.*, 21(2):572–596, 2011.

[68] Olivier Chapelle, Pannagadatta Shivaswamy, Srinivas Vadrevu, Kilian Weinberger, Ya Zhang, and Belle Tseng. Multi-task learning for boosting with application to web search ranking. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1189–1198. ACM, 2010.

[69] Vasileios Charisopoulos, Yudong Chen, Damek Davis, Mateo Díaz, Lijun Ding, and Dmitriy Drusvyatskiy. Low-rank matrix recovery with composite optimization: good conditioning and rapid convergence. *arXiv preprint arXiv:1904.10020*, 2019.

[70] Jianhui Chen, Jiayu Zhou, and Jieping Ye. Integrating low-rank and group-sparse structures for robust multi-task learning. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 42–50. ACM, 2011.

[71] Kun Chen, Kung-Sik Chan, and Nils Chr. Stenseth. Reduced rank stochastic regression with a sparse singular value decomposition. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 74(2):203–221, nov 2011.

[72] Lisha Chen and Jianhua Z. Huang. Sparse reduced-rank regression for simultaneous dimension reduction and variable selection. *Journal of the American Statistical Association*, 107(500):1533–1545, oct 2012.

[73] Tianyi Chen, Kaiqing Zhang, Georgios B Giannakis, and Tamer Başar. Communication-efficient distributed reinforcement learning. *arXiv preprint arXiv:1812.03239*, 2018.

[74] You-Lin Chen, Mladen Kolar, and Ruey S Tsay. Tensor canonical correlation analysis. *arXiv preprint arXiv:1906.05358*, 2019.

[75] Yudong Chen. Incoherence-optimal matrix completion. *IEEE Trans. Inform. Theory*, 61(5):2909–2923, 2015.

[76] Yudong Chen, Srinadh Bhojanapalli, Sujay Sanghavi, and Rachel Ward. Coherent matrix completion. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 674–682, Bejing, China, 22–24 Jun 2014. PMLR.

[77] Yudong Chen, Ali Jalali, Sujay Sanghavi, and Constantine Caramanis. Low-rank matrix recovery from errors and erasures. *IEEE Transactions on Information Theory*, 59(7):4324–4337, jul 2013.

[78] Yudong Chen and Martin J. Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *Technical report*, 2015.

[79] Herman Chernoff. On the distribution of the likelihood ratio. *The Annals of Mathematical Statistics*, pages 573–578, 1954.

[80] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. Double/debiased machine learning for treatment and structural parameters. *The Econometrics Journal*, 21(1):C1–C68, 2018.

[81] Victor Chernozhukov, Christian Hansen, and Martin Spindler. Post-selection and post-regularization inference in linear models with many controls and instruments. *American Economic Review*, 105(5):486–90, 2015.

[82] Chih-Ping Chou and Peter M Bentler. Model modification in covariance structure modeling: A comparison among likelihood ratio, lagrange multiplier, and wald tests. *Multivariate Behavioral Research*, 25(1):115–136, 1990.

[83] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(167):1–167, 2017.

[84] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. *arXiv preprint arXiv:1805.07708*, 2018.

[85] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal Of Machine Learning Research*, 12:2493–2537, 2011.

[86] Ran Dai, Hyebin Song, Rina Foygel Barber, and Garvesh Raskutti. The bias of isotonic regression. *arXiv preprint arXiv:1908.04462*, 2019.

[87] Christoph Dann, Gerhard Neumann, and Jan Peters. Policy evaluation with temporal differences: A survey and comparison. *The Journal of Machine Learning Research*, 15(1):809–883, 2014.

[88] Mark A. Davenport and Justin Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):608–622, jun 2016.

[89] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *arXiv preprint arXiv:1710.01688*, 2017.

[90] Ruben Dezeure, Peter Bühlmann, and Cun-Hui Zhang. High-dimensional simultaneous inference with the bootstrap. *Test*, 26(4):685–719, 2017.

[91] Lijun Ding and Yudong Chen. The leave-one-out approach for matrix completion: Primal and dual analysis. *arXiv preprint arXiv:1803.07554*, 2018.

[92] Mathias Drton and Marloes H Maathuis. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4:365–393, 2017.

[93] Nelson Dunford and Jacob T Schwartz. *Linear operators part I: general theory*, volume 7. Interscience publishers New York, 1958.

[94] Robert F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.

[95] Robert F Engle. Wald, likelihood ratio, and lagrange multiplier tests in econometrics. *Handbook of econometrics*, 2:775–826, 1984.

[96] Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal Of Machine Learning Research*, 6:503–556, December 2005.

[97] Lawrence C Evans. An introduction to mathematical optimal control theory. *Lecture Notes, University of California, Department of Mathematics, Berkeley*, 2005.

[98] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi-task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117. ACM, 2004.

[99] Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B*, 70(5):849–911, 2008.

[100] Jianqing Fan, Jingjin Zhang, and Ke Yu. Vast portfolio selection with gross-exposure constraints. *Journal of the American Statistical Association*, 107(498):592–606, 2012.

[101] Jicong Fan, Lijun Ding, Yudong Chen, and Madeleine Udell. Factor group-sparse regularization for efficient low-rank matrix recovery. In *Advances in Neural Information Processing Systems*, pages 5105–5115, 2019.

[102] Ethan X Fang, Yang Ning, and Han Liu. Testing and confidence intervals for high dimensional proportional hazards models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(5):1415–1437, 2017.

[103] M. Fazel, H. Hindi, and S. Boyd. Rank minimization and applications in system theory. In *Proceedings of the 2004 American Control Conference*, volume 4, pages 3273–3278 vol.4, June 2004.

[104] M. Fazel, H. Hindi, and S.P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*. IEEE, 2001.

[105] Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pages 1466–1475, 2018.

[106] Guanhao Feng and Jingyu He. Factor investing: Hierarchical ensemble learning. *arXiv preprint arXiv:1902.01015*, 2019.

[107] Guanhao Feng, Jingyu He, and Nicholas G Polson. Deep learning for predicting asset returns. *arXiv preprint arXiv:1804.09314*, 2018.

[108] Jaime F Fisac, Anayo K Akametalu, Melanie N Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 2018.

[109] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[110] Zuguang Gao, Xudong Chen, and Tamer Başar. Controllability of conjunctive boolean networks with application to gene regulation. *IEEE Transactions on Control of Network Systems*, 5(2):770–781, 2017.

[111] Zuguang Gao, Xudong Chen, and Tamer Başar. Stability structures of conjunctive boolean networks. *Automatica*, 89:8–20, 2018.

[112] Zuguang Gao, Xudong Chen, Ji Liu, and Tamer Başar. Periodic behavior of a diffusion model over directed graphs. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 37–42. IEEE, 2016.

[113] Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

[114] Rong Ge, Jason D Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

[115] Rainer Gemulla, Erik Nijkamp, Peter J. Haas, and Yannis Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11*. ACM Press, 2011.

[116] Christopher Glynn, Jingyu He, Nicholas G Polson, and Jianeng Xu. Bayesian inference for polya inverse gamma models. *arXiv preprint arXiv:1905.12141*, 2019.

[117] Anna Goldenberg, Alice X Zheng, Stephen E Fienberg, Edoardo M Airoldi, et al. A survey of statistical network models. *Foundations and Trends® in Machine Learning*, 2(2):129–233, 2010.

[118] Larry Goldstein and Xiaohan Wei. Non-gaussian observations in nonlinear compressed sensing via stein discrepancies. *Information and Inference: A Journal of the IMA*, 8(1):125–159, 2018.

[119] Manuel Gomez-Rodriguez, Le Song, Hadi Daneshmand, and Bernhard Schölkopf. Estimating diffusion networks: Recovery conditions, sample complexity & soft-thresholding algorithm. *Journal of Machine Learning Research*, 2015.

[120] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[121] John D Gorman and Alfred O Hero. Lower bounds for parametric estimation with constraints. *IEEE Transactions on Information Theory*, 36(6):1285–1301, 1990.

[122] Christian Gourieroux, Alberto Holly, and Alain Monfort. Likelihood ratio test, wald test, and kuhn-tucker test in linear models with inequality constraints on the regression parameters. *Econometrica: journal of the Econometric Society*, pages 63–80, 1982.

[123] Ivo Grondman, Lucian Busoniu, Gabriel AD Lopes, and Robert Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1291–1307, 2012.

[124] David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Trans. Inform. Theory*, 57(3):1548–1566, 2011.

[125] Quanquan Gu, Zhaoran Wang Wang, and Han Liu. Low-rank and sparse structure pursuit via alternating minimization. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 600–609, Cadiz, Spain, 09–11 May 2016. PMLR.

[126] Thomas Gueuning and Gerda Claeskens. Confidence intervals for high-dimensional partially linear single-index models. *Journal of Multivariate Analysis*, 149:13–29, 2016.

[127] Wooseok Ha and Rina Foygel Barber. Alternating minimization and alternating descent over nonconvex sets. *Technical report*, 2017.

[128] Wooseok Ha, Haoyang Liu, and Rina Foygel Barber. An equivalence between stationary points for rank constraints versus low-rank factorizations. *arXiv preprint arXiv:1812.00404*, 2018.

[129] Benjamin Haeffele, Eric Young, and Rene Vidal. Structured low-rank matrix factorization: Optimality, algorithm, and applications to image processing. In Tony Jebara and Eric P. Xing, editors, *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 2007–2015. JMLR Workshop and Conference Proceedings, 2014.

[130] P Richard Hahn, Carlos M Carvalho, David Puelz, Jingyu He, et al. Regularization and confounding in linear regression for treatment effect estimation. *Bayesian Analysis*, 13(1):163–182, 2018.

[131] P Richard Hahn, Jingyu He, and Hedibert Lopes. Bayesian factor model shrinkage for linear iv regression with many instruments. *Journal of Business & Economic Statistics*, 36(2):278–287, 2018.

[132] P Richard Hahn, Jingyu He, and Hedibert F Lopes. Efficient sampling for gaussian linear regression with arbitrary priors. *Journal of Computational and Graphical Statistics*, 28(1):142–154, 2019.

[133] Daniel B Hall and Jens T Præstgaard. Order-restricted score tests for homogeneity in generalised linear and nonlinear mixed models. *Biometrika*, 88(3):739–751, 2001.

[134] Aaron K Han. Non-parametric analysis of a generalized regression model: the maximum rank correlation estimator. *Journal of Econometrics*, 35(2-3):303–316, 1987.

[135] Fang Han, Hongkai Ji, Zhicheng Ji, Honglang Wang, et al. A provable smoothing approach for high dimensional generalized regression with applications in genomics. *Electronic Journal of Statistics*, 11(2):4347–4403, 2017.

[136] Z. Harchaoui, M. Douze, M. Paulin, M. Dudik, and J. Malick. Large-scale image classification with trace-norm regularization. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2012.

[137] Wolfgang Karl Härdle, Marlene Müller, Stefan Sperlich, and Axel Werwatz. *Nonparametric and semiparametric models*. Springer Science & Business Media, 2012.

[138] Moritz Hardt. Understanding alternating minimization for matrix completion. In *55th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2014*, pages 651–660. IEEE Computer Soc., Los Alamitos, CA, 2014.

[139] Moritz Hardt, Raghu Meka, Prasad Raghavendra, and Benjamin Weitz. Computational limits for matrix completion. In Balcan et al. [24], pages 703–725.

[140] Moritz Hardt and Mary Wootters. Fast matrix completion without the condition number. In Balcan et al. [24], pages 638–678.

[141] Trevor Hastie, Rahul Mazumder, Jason D. Lee, and Reza Zadeh. Matrix completion and low-rank SVD via fast alternating least squares. *Journal Of Machine Learning Research*, 16:3367–3402, 2015.

[142] Jingyu He and P Richard Hahn. Stochastic tree ensembles for regularized nonlinear regression. *arXiv preprint arXiv:2002.03375*, 2020.

[143] Jingyu He, Saar Yalov, and P Richard Hahn. XBART: Accelerated Bayesian additive regression trees. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1130–1138, 2019.

[144] Matthias Hein and Thomas Bühler. An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse pca. In *Advances in Neural Information Processing Systems*, pages 847–855, 2010.

[145] David A Hirshberg and Stefan Wager. Debiased inference of average partial effects in single-index models. *arXiv preprint arXiv:1811.02547*, 2018.

[146] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[147] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[148] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[149] Marian Hristache, Anatoli Juditsky, and Vladimir Spokoiny. Direct estimation of the index coefficient in a single-index model. *Annals of Statistics*, pages 595–623, 2001.

[150] Cho-Jui Hsieh and Peder Olsen. Nuclear norm minimization via active subspace selection. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 575–583, Bejing, China, 22–24 Jun 2014. PMLR.

[151] Daniel Hsu, Sham M. Kakade, and Tong Zhang. Robust matrix decomposition with sparse corruptions. *IEEE Trans. Inform. Theory*, 57(11):7221–7234, 2011.

[152] Jessie Huang, Fa Wu, Doina Precup, and Yang Cai. Learning safe policies with expert guidance. *arXiv preprint arXiv:1805.08313*, 2018.

[153] Junzhou Huang and Tong Zhang. The benefit of group sparsity. *Annals of Statistics*, 38(4):1978–2004, 2010.

[154] David R. Hunter. Asymptotic tools, 2011.

[155] Hidehiko Ichimura. Semiparametric least squares (sls) and weighted sls estimation of single-index models. *Journal of Econometrics*, 58(1-2):71–120, 1993.

[156] Alan Julian Izenman. *Modern multivariate statistical techniques*. Springer Texts in Statistics. Springer, New York, 2008. Regression, classification, and manifold learning.

[157] Prateek Jain, Raghu Meka, and Inderjit S. Dhillon. Guaranteed rank minimization via singular value projection. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 937–945. Curran Associates, Inc., 2010.

[158] Prateek Jain and Praneeth Netrapalli. Fast exact matrix completion with finite samples. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 1007–1034. JMLR.org, 2015.

[159] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization (extended abstract). In *STOC'13—Proceedings of the 2013 ACM Symposium on Theory of Computing*, pages 665–674. ACM, New York, 2013.

[160] Jana Janková and Sara van de Geer. De-biased sparse pca: Inference and testing for eigenstructure of large covariance matrices. *arXiv preprint arXiv:1801.10567*, 2018.

[161] Jana Janková and Sara van de Geer. Inference in high-dimensional graphical models. *arXiv preprint arXiv:1801.08512*, 2018.

[162] Jana Jankova, Sara Van De Geer, et al. Confidence intervals for high-dimensional inverse covariance estimation. *Electronic Journal of Statistics*, 9(1):1205–1229, 2015.

[163] Jana Jankova, Sara Van De Geer, et al. Semiparametric efficiency bounds for high-dimensional models. *The Annals of Statistics*, 46(5):2336–2359, 2018.

[164] Majid Janzamin, Hanie Sedghi, and Anima Anandkumar. Score function features for discriminative learning: Matrix and tensor framework. *arXiv preprint arXiv:1412.2863*, 2014.

[165] Adel Javanmard and Jason D Lee. A flexible framework for hypothesis testing in high-dimensions. *arXiv preprint arXiv:1704.07971*, 2017.

[166] Adel Javanmard and Andrea Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909, 2014.

[167] Bo Jiang, Jun S Liu, et al. Variable selection for general index models via sliced inverse regression. *The Annals of Statistics*, 42(5):1751–1786, 2014.

[168] Iain M. Johnstone and Arthur Yu Lu. On consistency and sparsity for principal components analysis in high dimensions. *Journal of the American Statistical Association*, 104(486):682–693, 2009.

[169] Zheng Tracy Ke and Minzhe Wang. A new svd approach to optimal topic estimation. *arXiv preprint arXiv:1704.07016*, 2017.

[170] John L Kelley. *General topology.* Courier Dover Publications, 2017.

[171] Bryan Kelly and Seth Pruitt. Market expectations in the cross-section of present values. *The Journal of Finance*, 68(5):1721–1756, 2013.

[172] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

[173] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(Jul):2057–2078, 2010.

[174] Byol Kim, Song Liu, and Mladen Kolar. Two-sample inference for high-dimensional markov networks. *arXiv preprint arXiv:1905.00466*, 2019.

[175] Seyoung Kim and Eric P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 543–550, 2010.

[176] Maxwell L King and Murray D Smith. Joint one-sided tests of linear regression coefficients. *Journal of Econometrics*, 32(3):367–383, 1986.

[177] Maxwell L King and Ping X Wu. Locally optimal one-sided tests for multiparameter hypotheses. *Econometric Reviews*, 16(2):131–156, 1997.

[178] Chris AJ Klaassen and Nanang Susyanto. Semiparametrically efficient estimation of euclidean parameters under equality constraints. *Journal of Statistical Planning and Inference*, 201:120–132, 2019.

[179] Keith Knight and Wenjiang Fu. Asymptotics for lasso-type estimators. *Annals of Statistics*, 28(5):1356–1378, 2000.

[180] Anders Bredahl Kock and Haihan Tang. Uniform inference in high-dimensional dynamic panel data models with approximately sparse fixed effects. *Econometric Theory*, pages 1–65, 2018.

[181] David A Kodde and Franz C Palm. Wald criteria for jointly testing equality and inequality restrictions. *Econometrica: journal of the Econometric Society*, pages 1243–1248, 1986.

[182] Mladen Kolar, Sivaraman Balakrishnan, Alessandro Rinaldo, and Aarti Singh. Minimax localization of structural information in large noisy matrices. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 909–917, 2011.

[183] Mladen Kolar, John D. Lafferty, and Larry A. Wasserman. Union support recovery in multi-task learning. *Journal Of Machine Learning Research*, 12:2415–2435, 2011.

[184] Vladimir Koltchinskii, Karim Lounici, and Alexandre B. Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *Annals of Statistics*, 39(5):2302–2329, 2011.

[185] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[186] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, aug 2009.

[187] R Matthew Kretchmar. Parallel reinforcement learning. In *The 6th World Conference on Systemics, Cybernetics, and Informatics*. Citeseer, 2002.

[188] Akio Kudô. A multivariate analogue of the one-sided test. *Biometrika*, 50:403–418, 1963.

[189] Jonathan Lacotte, Yinlam Chow, Mohammad Ghavamzadeh, and Marco Pavone. Risk-sensitive generative adversarial imitation learning. *arXiv preprint arXiv:1808.04468*, 2018.

[190] Maksim Lapin, Bernt Schiele, and Matthias Hein. Scalable multitask representation learning for scene classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1434–1441, 2014.

[191] Alessandro Lazaric and Mohammad Ghavamzadeh. Bayesian multi-task reinforcement learning. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 599–606. Omnipress, 2010.

[192] Dennis Lee, Haoran Tang, Jeffrey O Zhang, Huazhe Xu, Trevor Darrell, and Pieter Abbeel. Modular architecture for starcraft ii with deep reinforcement learning. In *Fourteenth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2018.

[193] Jason D Lee, Dennis L Sun, Yuekai Sun, Jonathan E Taylor, et al. Exact post-selection inference, with application to the lasso. *The Annals of Statistics*, 44(3):907–927, 2016.

[194] Kiryung Lee and Yoram Bresler. ADMiRA: atomic decomposition for minimum rank approximation. *IEEE Trans. Inform. Theory*, 56(9):4402–4416, 2010.

[195] Mihee Lee, Haipeng Shen, Jianhua Z. Huang, and J. S. Marron. Biclustering via sparse singular value decomposition. *Biometrics*, 66(4):1087–1095, feb 2010.

[196] Hannes Leeb and Benedikt M. Pötscher. Model selection and inference: Facts and fiction. *Econ. Theory*, 21(01), feb 2005.

[197] Hannes Leeb and Benedikt M. Pötscher. Can one estimate the unconditional distribution of post-model-selection estimators? *Econ. Theory*, 24(02):338–376, Nov 2007.

[198] Hannes Leeb and Benedikt M. Pötscher. Sparse estimators and the oracle property, or the return of hodges' estimator. *Journal of Econometrics*, 142(1):201–211, jan 2008.

[199] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.

[200] Kathleen T Li. Statistical inference for average treatment effects estimated by synthetic control methods. *Journal of the American Statistical Association*, (just-accepted):1–40, 2019.

[201] Ker-Chau Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.

[202] Qiuwei Li, Zhihui Zhu, and Gongguo Tang. Geometry of factored nuclear norm regularization. *Technical report*, 2017.

[203] Xingguo Li, Tuo Zhao, Raman Arora, Han Liu, and Jarvis Haupt. Nonconvex sparse learning via stochastic optimization with progressive variance reduction. *Technical report*, 2016.

[204] Xingguo Li, Tuo Zhao, Raman Arora, Han Liu, and Jarvis Haupt. Stochastic variance reduced optimization for nonconvex sparse learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 917–925. JMLR.org, 2016.

[205] Yuxi Li. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*, 2017.

[206] Qian Lin, Zhigen Zhao, Jun S Liu, et al. On consistency and sparsity for sliced inverse regression in high dimensions. *The Annals of Statistics*, 46(2):580–610, 2018.

[207] An Liu, Vincent Lau, and Borna Kananian. Stochastic successive convex approximation for non-convex constrained stochastic optimization. *arXiv preprint arXiv:1801.08266*, 2018.

[208] Boyi Liu, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural proximal/trust region policy optimization attains globally optimal policy. *arXiv preprint arXiv:1906.10306*, 2019.

[209] Haoyang Liu and Rina Foygel Barber. Between hard and soft thresholding: optimal iterative thresholding algorithms. *arXiv preprint arXiv:1804.08841*, 2018.

[210] Zhang Liu and Lieven Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM J. Matrix Anal. Appl.*, 31(3):1235–1256, 2009.

[211] Andrés F López-Lopera, ST John, and Nicolas Durrande. Gaussian process modulated cox processes under linear inequality constraints. *arXiv preprint arXiv:1902.10974*, 2019.

[212] K. Lounici, M. Pontil, Alexandre B. Tsybakov, and Sara A. van de Geer. Oracle inequalities and optimal inference under group sparsity. *Annals of Statistics*, 39:2164–204, 2011.

[213] Jiarui Lu, Pixu Shi, and Hongzhe Li. Generalized linear models with linear constraints for microbiome compositional data. *Biometrics*, 75(1):235–244, 2019.

[214] Yao Lu and Valeri V Nikolaev. Expected loan loss provisioning: An empirical model. *Chicago Booth Research Paper*, (19-11), 2019.

[215] Zeng-Hua Lu. Halfline tests for multivariate one-sided alternatives. *Computational Statistics & Data Analysis*, 57(1):479–490, 2013.

[216] Zeng-Hua Lu. Extended maxt tests of one-sided hypotheses. *Journal of the American Statistical Association*, (just-accepted), 2015.

[217] X. Ma, L. Xiao, and W. H. Wong. Learning regulatory programs by threshold SVD regression. *Proceedings of the National Academy of Sciences*, 111(44):15675–15680, oct 2014.

[218] Zhuang Ma, Zongming Ma, and Tingni Sun. Adaptive estimation in two-way sparse reduced-rank regression. *Technical report*, 2014.

[219] Zongming Ma. Sparse principal component analysis and iterative thresholding. *Ann. Statist.*, 41(2):772–801, 2013.

[220] S Mei, B Cao, and J Sun. Encoding low-rank and sparse structures simultaneously in multi-task learning. techreport. Technical report, Microsoft Technical Report, 2012.

[221] Song Mei, Yu Bai, Andrea Montanari, et al. The landscape of empirical risk for nonconvex losses. *The Annals of Statistics*, 46(6A):2747–2774, 2018.

[222] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[223] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[224] Geert Molenberghs and Geert Verbeke. Likelihood ratio, score, and Wald tests in a constrained parameter space. *Amer. Statist.*, 61(1):22–27, 2007.

[225] Rémi Munos and Csaba Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(May):815–857, 2008.

[226] Sen Na, Mladen Kolar, and Oluwasanmi Koyejo. Estimating differential latent variable graphical models with applications to brain connectivity. *arXiv preprint arXiv:1909.05892*, 2019.

[227] Sen Na, Zhuoran Yang, Zhaoran Wang, and Mladen Kolar. High-dimensional varying index coefficient models via stein's identity. *Journal of Machine Learning Research*, 20(152):1–44, 2019.

[228] Boaz Nadler et al. Finite sample approximation results for principal component analysis: A matrix perturbation approach. *The Annals of Statistics*, 36(6):2791–2817, 2008.

[229] Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, et al. Massively parallel methods for deep reinforcement learning. *arXiv preprint arXiv:1507.04296*, 2015.

[230] Sahand Negahban and Martin J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *Annals of Statistics*, 39(2):1069–1097, 2011.

[231] 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.

[232] Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009.

[233] Sahand N Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

[234] Y. Nesterov. *Introductory Lectures on Convex Optimization.* Springer Us, 2013.

[235] Matey Neykov, Jun S Liu, and Tianxi Cai. L1-regularized least squares for support recovery of high dimensional single index models with gaussian designs. *Journal of Machine Learning Research*, 17(87):1–37, 2016.

[236] Matey Neykov, Zhaoran Wang, and Han Liu. Agnostic estimation for misspecified phase retrieval models. In *Advances in Neural Information Processing Systems*, pages 4089–4097, 2016.

[237] Yang Ning, Han Liu, et al. A general theory of hypothesis tests and confidence regions for sparse high dimensional models. *The Annals of Statistics*, 45(1):158–195, 2017.

[238] Yang Ning, Tianqi Zhao, Han Liu, et al. A likelihood ratio framework for high-dimensional semiparametric regression. *The Annals of Statistics*, 45(6):2299–2327, 2017.

[239] G. Obozinski, Martin J. Wainwright, and Michael I. Jordan. Support union recovery in high-dimensional multivariate regression. *Annals of Statistics*, 39(1):1–47, 2011.

[240] Dohyung Park, Anastasios Kyrillidis, Constantine Caramanis, and Sujay Sanghavi. Finding low-rank solutions to matrix problems, efficiently and provably. *arXiv preprint arXiv:1606.03168*, 2016.

[241] Santiago Paternain. *Stochastic Control Foundations of Autonomous Behavior*. PhD thesis, University of Pennsylvania, 2018.

[242] Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

[243] Michael D Perlman. One-sided testing problems in multivariate analysis. *The Annals of Mathematical Statistics*, pages 549–567, 1969.

[244] Michael D Perlman and Lang Wu. Some improved tests for multivariate one-sided hypotheses. *Metrika*, 64(1):23–39, 2006.

[245] Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Policy gradient in lipschitz markov decision processes. *Machine Learning*, 100(2-3):255–283, 2015.

[246] Yaniv Plan and Roman Vershynin. The generalized lasso with non-linear observations. *IEEE Transactions on information theory*, 62(3):1528–1537, 2016.

[247] Yaniv Plan, Roman Vershynin, and Elena Yudovina. High-dimensional estimation with geometric constraints. *Information and Inference: A Journal of the IMA*, 6(1):1–40, 2016.

[248] Eftychios A. Pnevmatikakis, Yuanjun Gao, Daniel Soudry, David Pfau, Clay Lacefield, Kira Poskanzer, Randy Bruno, Rafael Yuste, and Liam Paninski. A structured matrix factorization framework for large scale calcium imaging data analysis. *Technical report*, 2014.

[249] Benedikt M. Pötscher. Confidence sets based on sparse estimators are necessarily large. *Sankhyā*, 71(1, Ser. A):1–18, 2009.

[250] Benedikt M. Pötscher and Hannes Leeb. On the distribution of penalized maximum likelihood estimators: The LASSO, SCAD, and thresholding. *Journal of Multivariate Analysis*, 100(9):2065–2082, oct 2009.

[251] Jean Pouget-Abadie and Thibaut Horel. Inferring graphs from cascades: A sparse recovery framework. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 625–626, 2015.

[252] LA Prashanth and Mohammad Ghavamzadeh. Variance-constrained actor-critic algorithms for discounted and average reward mdps. *Machine Learning*, 105(3):367–417, 2016.

[253] Suba Rao. *Lectures on statistical inference.*

[254] Benjamin Recht. A simpler approach to matrix completion. *Journal Of Machine Learning Research*, 12:3413–3430, 2011.

[255] Benjamin Recht. A tour of reinforcement learning: The view from continuous control. *Annual Review of Control, Robotics, and Autonomous Systems*, 2018.

[256] Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, 2010.

[257] Benjamin Recht and Christopher Ré. Parallel stochastic gradient algorithms for large-scale matrix completion. *Math. Program. Comput.*, 5(2):201–226, 2013.

[258] Zhao Ren, Tingni Sun, Cun-Hui Zhang, Harrison H Zhou, et al. Asymptotic normality and optimalities in estimation of large gaussian graphical models. *The Annals of Statistics*, 43(3):991–1026, 2015.

[259] Chang-han Rhee and Peter W Glynn. Unbiased estimation with square root convergence for sde models. *Operations Research*, 63(5):1026–1043, 2015.

[260] Emile Richard, Pierre andre Savalle, and Nicolas Vayatis. Estimation of simultaneously sparse and low rank matrices. In John Langford and Joelle Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1351–1358, New York, NY, USA, 2012. ACM.

[261] Manuel Gomez Rodriguez, David Balduzzi, and Bernhard Schölkopf. Uncovering the temporal dynamics of diffusion networks. *arXiv preprint arXiv:1105.0697*, 2011.

[262] Alan J Rogers. Modified lagrange multiplier tests for problems with one-sided alternatives. *Journal of Econometrics*, 31(3):341–361, 1986.

[263] Angelika Rohde and Alexandre B. Tsybakov. Estimation of high-dimensional low-rank matrices. *Ann. Statist.*, 39(2):887–930, 2011.

[264] Andrzej Ruszczyński. Feasible direction methods for stochastic programming problems. *Mathematical Programming*, 19(1):220–229, 1980.

[265] Joris Scharpff, Diederik M Roijers, Frans A Oliehoek, Matthijs TJ Spaan, and Mathijs Michiel de Weerdt. Solving transition-independent multi-agent mdps with sparse interactions. In *AAAI*, pages 3174–3180, 2016.

[266] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pages 1889–1897, 2015.

[267] Gesualdo Scutari, Francisco Facchinei, Peiran Song, Daniel P Palomar, and Jong-Shi Pang. Decomposition by partial linearization: Parallel optimization of multi-agent systems. *IEEE Transactions on Signal Processing*, 62(3):641–656, 2013.

[268] Michael L Seltzer and Jasha Droppo. Multi-task learning in deep neural networks for improved phoneme recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*, pages 6965–6969. IEEE, 2013.

[269] Alexander Shapiro. Asymptotic theory of overparameterized structural models. *Journal of the American Statistical Association*, 81(393):142–149, 1986.

[270] Alexander Shapiro. Towards a unified theory of inequality constrained testing in multivariate analysis. *International Statistical Review/Revue Internationale de Statistique*, pages 49–62, 1988.

[271] Alexander Shapiro et al. Asymptotic properties of statistical estimators in stochastic programming. *The Annals of Statistics*, 17(2):841–858, 1989.

[272] Yiyuan She. Selective factor extraction in high dimensions. *Biometrika*, 104(1):97–110, 2017.

[273] Yiyuan She and Hoang Tran. On cross-validation for sparse reduced rank regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(1):145–161, 2019.

[274] Yoav Shechtman, Yonina C Eldar, Oren Cohen, Henry Nicholas Chapman, Jianwei Miao, and Mordechai Segev. Phase retrieval with application to optical imaging: a contemporary overview. *IEEE signal processing magazine*, 32(3):87–109, 2015.

[275] Robert P Sherman. The limiting distribution of the maximum rank correlation estimator. *Econometrica: Journal of the Econometric Society*, pages 123–137, 1993.

[276] Chengchun Shi, Rui Song, Zhongqian Chen, and Runze Li. Linear hypothesis testing for high dimensional generalized linear models. 2018.

[277] Pixu Shi, Anru Zhang, Hongzhe Li, et al. Regression analysis for microbiome compositional data. *The Annals of Applied Statistics*, 10(2):1019–1040, 2016.

[278] Mervyn J Silvapulle and Pranab Kumar Sen. *Constrained statistical inference: Order, inequality, and shape constraints*, volume 912. John Wiley & Sons, 2011.

[279] Mervyn J. Silvapulle and Paramsothy Silvapulle. A score test against one-sided alternatives. *Journal of the American Statistical Association*, 90(429):342–349, 1995.

[280] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[281] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

[282] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

[283] S. D. Silvey. The Lagrangian multiplier test. *Ann. Math. Statist.*, 30:389–407, 1959.

[284] Martin Slawski and Matthias Hein. Non-negative least squares for high-dimensional linear models: consistency and sparse recovery without regularization. *Electronic Journal of Statistics*, 7:3004–3056, 2013.

[285] Matthijs Snel and Shimon Whiteson. Multi-task reinforcement learning: Shaping and feature selection. In *Lecture Notes in Computer Science*, pages 237–248. Springer Berlin Heidelberg, 2012.

[286] Hyebin Song, Ran Dai, Garvesh Raskutti, and Rina Foygel Barber. Convex and non-convex approaches for statistical inference with noisy labels. *arXiv preprint arXiv:1910.02348*, 2019.

[287] Nathan Srebro, Jason Rennie, and Tommi S. Jaakkola. Maximum-margin matrix factorization. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1329–1336. MIT Press, 2005.

[288] Charles Stein, Persi Diaconis, Susan Holmes, Gesine Reinert, et al. Use of exchangeable pairs in the analysis of simulations. In *Stein's Method*, pages 1–25. Institute of Mathematical Statistics, 2004.

[289] Peng Sun, Xinghai Sun, Lei Han, Jiechao Xiong, Qing Wang, Bo Li, Yang Zheng, Ji Liu, Yongsheng Liu, Han Liu, et al. Tstarbots: Defeating the cheating level builtin ai in starcraft ii in the full game. *arXiv preprint arXiv:1809.07193*, 2018.

[290] Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Trans. Inform. Theory*, 62(11):6535–6579, 2016.

[291] Ying Sun, Prabhu Babu, and Daniel P Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65(3):794–816, 2016.

[292] Edward Susko. Likelihood ratio tests with boundary constraints using data-dependent degrees of freedom. *Biometrika*, page ast032, 2013.

[293] Nanang Susyanto, Chris AJ Klaassen, et al. Semiparametrically efficient estimation of constrained euclidean parameters. *Electronic Journal of Statistics*, 11(2):3120–3140, 2017.

[294] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[295] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[296] Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063, 2000.

[297] Gábor Takács, István Pilászy, Bottyán Németh, and Domonkos Tikk. Major components of the gravity recommendation system. *ACM SIGKDD Explorations Newsletter*, 9(2):80, dec 2007.

[298] Kean Ming Tan, Zhaoran Wang, Tong Zhang, Han Liu, and R Dennis Cook. A convex formulation for high-dimensional sparse sliced inverse regression. *Biometrika*, 105(4):769–782, 2018.

[299] Jonathan E. Taylor, Richard Lockhart, Robert J. Tibshirani, and Robert J. Tibshirani. Exact post-selection inference for forward stepwise and least angle regression. *ArXiv e-prints, arXiv:1401.3889*, 2014.

[300] Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

[301] Xiaoying Tian and Jonathan Taylor. Asymptotics of selective inference. *Scandinavian Journal of Statistics*, 44(2):480–499, 2017.

[302] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

[303] Robert J. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

[304] Ryan J Tibshirani, Jonathan Taylor, Richard Lockhart, and Robert Tibshirani. Exact post-selection inference for sequential regression procedures. *Journal of the American Statistical Association*, 111(514):600–620, 2016.

[305] Stephen Tu, Ross Boczar, Max Simchowitz, Mahdi Soltanolkotabi, and Ben Recht. Low-rank solutions of linear matrix equations via procrustes flow. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 964–973, New York, New York, USA, 20–22 Jun 2016. PMLR.

[306] Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. Safe exploration in finite markov decision processes with gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4312–4320, 2016.

[307] B. A. Turlach, W. N. Venables, and S. J. Wright. Simultaneous variable selection. *Technometrics*, 47(3):349–363, 2005.

[308] Yoshimasa Uematsu, Yingying Fan, Kun Chen, Jinchi Lv, and Wei Lin. Sofar: large-scale association network learning. *Technical report*, 2017.

[309] Sara Van de Geer, Peter Bühlmann, Ya'acov Ritov, Ruben Dezeure, et al. On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42(3):1166–1202, 2014.

[310] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

[311] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

[312] J. T. Vogelstein, A. M. Packer, T. A. Machado, T. Sippy, B. Babadi, R. Yuste, and L. Paninski. Fast nonnegative deconvolution for spike train inference from population calcium imaging. *Journal of Neurophysiology*, 104(6):3691–3704, jun 2010.

[313] Maria Vounou, Eva Janousova, Robin Wolz, Jason L. Stein, Paul M. Thompson, Daniel Rueckert, and Giovanni Montana. Sparse reduced-rank regression detects genetic associations with voxel-wise longitudinal phenotypes in alzheimer's disease. *NeuroImage*, 60(1):700–716, mar 2012.

[314] Van Vu. Singular vectors under random perturbation. *Random Struct. Alg.*, 39(4):526–538, may 2011.

[315] Vincent Vu and Jing Lei. Minimax rates of estimation for sparse pca in high dimensions. In *Artificial intelligence and statistics*, pages 1278–1286, 2012.

[316] Vincent Q Vu, Juhee Cho, Jing Lei, and Karl Rohe. Fantope projection and selection: A near-optimal convex relaxation of sparse pca. In *Advances in neural information processing systems*, pages 2670–2678, 2013.

[317] Vincent Q. Vu and Jing Lei. Minimax sparse principal subspace estimation in high dimensions. *Ann. Statist.*, 41(6):2905–2947, 2013.

[318] Hoi-To Wai, Zhuoran Yang, Zhaoran Wang, and Mingyi Hong. Multi-agent reinforcement learning via double averaging primal-dual optimization. *arXiv preprint arXiv:1806.00877*, 2018.

[319] Jialei Wang, Mladen Kolar, and Nathan Srebro. Distributed multi-task learning with shared representation. *Technical report*, March 2016.

[320] Jialei Wang, Mladen Kolar, and Nathan Srerbo. Distributed multi-task learning. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 751–760, Cadiz, Spain, 09–11 May 2016. PMLR.

[321] Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural policy gradient methods: Global optimality and rates of convergence. *arXiv preprint arXiv:1909.01150*, 2019.

[322] Zhaoran Wang, Huanran Lu, and Han Liu. Nonconvex statistical optimization: Minimax-optimal sparse pca in polynomial time. *Technical report*, 2014.

[323] Zheng Wang, Ming-Jun Lai, Zhaosong Lu, Wei Fan, Hasan Davulcu, and Jieping Ye. Orthogonal rank-one matrix pursuit for low rank matrix completion. *SIAM J. Sci. Comput.*, 37(1):A488–A514, 2015.

[324] Yuting Wei, Martin J Wainwright, Adityanand Guntuboyina, et al. The geometry of hypothesis testing over convex cones: Generalized likelihood ratio tests and minimax radii. *The Annals of Statistics*, 47(2):994–1024, 2019.

[325] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1113–1120. ACM, 2009.

[326] Min Wen and Ufuk Topcu. Constrained cross-entropy method for safe reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 7461–7471, 2018.

[327] Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. Multi-task reinforcement learning. In *Proceedings of the 24th international conference on Machine learning - ICML '07*. ACM Press, 2007.

[328] Frank A Wolak. Testing inequality constraints in linear econometric models. *Journal of econometrics*, 41(2):205–235, 1989.

[329] Shuo Xiang, Yunzhang Zhu, Xiaotong Shen, and Jieping Ye. Optimal exact least squares rank minimization. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*. ACM Press, 2012.

[330] Pan Xu, Jian Ma, and Quanquan Gu. Speeding up latent variable gaussian graphical model estimation via nonconvex optimizations. *Technical report*, 2017.

[331] Sijia Xu, Hongyu Kuang, Zhi Zhuang, Renjie Hu, Yang Liu, and Huyang Sun. Macro action selection with deep reinforcement learning in starcraft. *arXiv preprint arXiv:1812.00336*, 2018.

[332] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram. Multi-task learning for classification with dirichlet process priors. *Journal Of Machine Learning Research*, 8:35–63, 2007.

[333] Qi Yan, Jieping Ye, and Xiaotong Shen. Simultaneous pursuit of sparseness and rank structures for matrix decomposition. *Journal Of Machine Learning Research*, 16:47–75, 2015.

[334] Dan Yang, Zongming Ma, and Andreas Buja. A sparse singular value decomposition method for high-dimensional data. *J. Comput. Graph. Statist.*, 23(4):923–942, 2014.

[335] Fan Yang, Rina Foygel Barber, et al. Contraction and uniform convergence of isotonic regression. *Electronic Journal of Statistics*, 13(1):646–677, 2019.

[336] Fan Yang, Rina Foygel Barber, Prateek Jain, and John Lafferty. Selective inference for group-sparse linear models. In *Advances in Neural Information Processing Systems*, pages 2469–2477, 2016.

[337] Yang Yang, Gesualdo Scutari, Daniel P Palomar, and Marius Pesavento. A parallel decomposition method for nonconvex stochastic multi-agent optimization problems. *IEEE Transactions on Signal Processing*, 64(11):2949–2964, 2016.

[338] Zhuoran Yang, Krishna Balasubramanian, Zhaoran Wang, and Han Liu. Learning non-gaussian multi-index model via second-order stein's method. *Advances in Neural Information Processing Systems*, 30:6097–6106, 2017.

[339] Zhuoran Yang, Krishnakumar Balasubramanian, and Han Liu. High-dimensional non-gaussian single index models via thresholded score function estimation. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3851–3860. JMLR. org, 2017.

[340] Zhuoran Yang, Krishnakumar Balasubramanian, Princeton Zhaoran Wang, and Han Liu. Estimating high-dimensional non-gaussian multiple index models via stein's lemma. In *Advances in Neural Information Processing Systems*, pages 6097–6106, 2017.

[341] Zhuoran Yang, Yongxin Chen, Mingyi Hong, and Zhaoran Wang. On the global convergence of actor-critic: A case for linear quadratic regulator with ergodic cost. *arXiv preprint arXiv:1907.06246*, 2019.

[342] Zhuoran Yang, Lin F Yang, Ethan X Fang, Tuo Zhao, Zhaoran Wang, and Matey Neykov. Misspecified nonconvex statistical optimization for phase retrieval. *arXiv preprint arXiv:1712.06245*, 2017.

[343] Zhuoran Yang, Lin F Yang, Ethan X Fang, Tuo Zhao, Zhaoran Wang, and Matey Neykov. Misspecified nonconvex statistical optimization for sparse phase retrieval. *Mathematical Programming*, pages 1–27, 2019.

[344] Ming Yu. Penalized score test for high dimensional logistic regression. 2016.

[345] Ming Yu, Varun Gupta, and Mladen Kolar. Statistical inference for pairwise graphical models using score matching. In *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 2016.

[346] Ming Yu, Varun Gupta, and Mladen Kolar. Estimation of a low-rank topic-based model for information cascades. *arXiv preprint arXiv:1709.01919*, 2017.

[347] Ming Yu, Varun Gupta, and Mladen Kolar. An influence-receptivity model for topic based information cascades. *2017 IEEE International Conference on Data Mining (ICDM)*, pages 1141–1146, 2017.

[348] Ming Yu, Varun Gupta, and Mladen Kolar. An influence-receptivity model for topic based information cascades. *Technical report*, 2017.

[349] Ming Yu, Varun Gupta, and Mladen Kolar. Learning influence-receptivity network structure with guarantee. *arXiv preprint arXiv:1806.05730*, 2018.

[350] Ming Yu, Varun Gupta, and Mladen Kolar. Constrained high dimensional statistical inference. *arXiv preprint arXiv:1911.07319*, 2019.

[351] Ming Yu, Varun Gupta, and Mladen Kolar. Simultaneous inference for pairwise graphical models with generalized score matching. *arXiv preprint arXiv:1905.06261*, 2019.

[352] Ming Yu, Varun Gupta, Mladen Kolar, et al. Recovery of simultaneous low rank and two-way sparse coefficient matrices, a nonconvex approach. *Electronic Journal of Statistics*, 14(1):413–457, 2020.

[353] Ming Yu, Karthikeyan Natesan Ramamurthy, Addie Thompson, and Aurélie C Lozano. Simultaneous parameter learning and bi-clustering for multi-response models. *Frontiers in Big Data*, 2:27, 2019.

[354] Ming Yu, Addie M. Thompson, Karthikeyan Natesan Ramamurthy, Eunho Yang, and Aurélie C. Lozano. Multitask learning using task clustering with applications to predictive modeling and gwas of plant varieties. *Technical report*, 2017.

[355] Ming Yu, Zhuoran Yang, Mladen Kolar, and Zhaoran Wang. Convergent policy optimization for safe reinforcement learning. *arXiv preprint arXiv:1910.12156*, 2019.

[356] Ming Yu, Zhuoran Yang, Mengdi Wang, and Zhaoran Wang. Provable q-iteration with l infinity guarantees and function approximation.

[357] Ming Yu, Zhuoran Yang, Zhaoran Wang, Yang Ning, and Mladen Kolar. Asymptotic inference for high dimensional model under equality constraints. *Manuscript*, 2020.

[358] Ming Yu, Zhuoran Yang, Tuo Zhao, Mladen Kolar, and Zhaoran Wang. Provable gaussian embedding with one observation. In *Advances in Neural Information Processing Systems*, pages 6764–6774, 2018.

[359] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68:49–67, 2006.

[360] Ming Yuan, Ali Ekici, Zhaosong Lu, and Renato Monteiro. Dimension reduction and coefficient estimation in multivariate linear regression. *Journal of the Royal Statistical Society: Series B*, 69(3):329–346, 2007.

[361] Xiao-Tong Yuan and Tong Zhang. Truncated power method for sparse eigenvalue problems. *Journal Of Machine Learning Research*, 14:899–925, 2013.

[362] Cun-Hui Zhang and Stephanie S Zhang. Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):217–242, 2014.

[363] Kaiqing Zhang, Zhuoran Yang, Han Liu, Tong Zhang, and Tamer Başar. Finite-sample analyses for fully decentralized multi-agent reinforcement learning. *arXiv preprint arXiv:1812.02783*, 2018.

[364] Xianyang Zhang and Guang Cheng. Simultaneous inference for high-dimensional linear models. *Journal of the American Statistical Association*, 112(518):757–768, 2017.

[365] Xiao Zhang, Lingxiao Wang, and Quanquan Gu. A nonconvex free lunch for low-rank plus sparse matrix recovery. *Technical report*, 2017.

[366] Boxin Zhao, Y Samuel Wang, and Mladen Kolar. Direct estimation of differential functional graphical models. *arXiv preprint arXiv:1910.09701*, 2019.

[367] Tuo Zhao, Zhaoran Wang, and Han Liu. Nonconvex low rank matrix factorization via inexact first order oracle. *Advances in Neural Information Processing Systems*, 2015.

[368] Qinqing Zheng and John D. Lafferty. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 109–117. Curran Associates, Inc., 2015.

[369] Jiayu Zhou, Jun Liu, Vaibhav A. Narayan, and Jieping Ye. Modeling disease progression via multi-task learning. *NeuroImage*, 78:233 – 248, 2013.

[370] Rong Zhu and Sherry ZF Zhou. Testing inequality constraints in a linear regression model with spherically symmetric disturbances. *Journal of Systems Science and Complexity*, 27(6):1204–1212, 2014.

[371] Yunzhang Zhu, Xiaotong Shen, and Wei Pan. On high-dimensional constrained maximum likelihood inference. *Journal of the American Statistical Association*, pages 1–14, 2019.

[372] Yunzhang Zhu, Xiaotong Shen, and Changqing Ye. Personalized prediction and sparsity pursuit in latent factor models. *J. Amer. Statist. Assoc.*, 111(513):241–252, 2016.

[373] Zhihui Zhu, Qiuwei Li, Gongguo Tang, and Michael B. Wakin. Global optimality in low-rank matrix optimization. *Technical report*, 2017.

[374] Zhihui Zhu, Qiuwei Li, Gongguo Tang, and Michael B. Wakin. The global optimization geometry of nonsymmetric matrix factorization and sensing. *Technical report*, 2017.

[375] Yong Zhuang, Wei-Sheng Chin, Yu-Chin Juan, and Chih-Jen Lin. A fast parallel SGD for matrix factorization in shared memory systems. In *Proceedings of the 7th ACM conference on Recommender systems - RecSys '13*. ACM Press, 2013.

[376] Bumeng Zhuo and Chao Gao. Mixing time of metropolis-hastings for bayesian community detection. *arXiv preprint arXiv:1811.02612*, 2018.

[377] Hui Zou, Trevor Hastie, and Robert Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

[378] Hui Zou and Lingzhou Xue. A selective overview of sparse principal component analysis. *Proceedings of the IEEE*, 106(8):1311–1320, 2018.