STOCHASTIC METHODS IN MODEL ESTIMATION: NEW ALGORITHMS AND NEW PROPERTIES

A Dissertation
Presented to
The Academic Faculty

By

Yiling Luo

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
H. Milton Stewart School of Industrial and Systems Engineering
Department of Industrial Engineering

Georgia Institute of Technology

May 2023

STOCHASTIC METHODS IN MODEL ESTIMATION: NEW ALGORITHMS AND NEW PROPERTIES

Thesis committee:

Dr. Xiaoming Huo, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Yajun Mei, Co-Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Arkadi Nemirovski
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Vladimir Koltchinskii
School of Mathematics
Georgia Institute of Technology

Dr. Kai Zhang
Department of Statistics and Operations Research
University of North Carolina at Chapel Hill

Date approved: December 21, 2022

On ne naît pas femme, on le devient.

Simone De Veauvoir, Les Deuxième Sexe

To my daughter.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor Prof. Xiaoming Huo. I have been working with Prof. Huo since I entered this program. Throughout the time, I was deeply impressed by Prof. Huo's passion for statistical research and his mathematical intuition. He always treated us as researchers and collaborators instead of just students, which helped a lot to build our research ability. I still remember how Prof. Huo gave me a small project to start with, and let me gradually lead my own research during my Ph.D. journey. When I feel lost about what I am doing, he always encourages me to keep going and inspires me to explore different possibilities. With his help, I could grow from a student to an independent researcher.

I am also extremely grateful to my co-advisor Prof. Yajun Mei. I started working with Prof. Mei at the end of my first year of study. Prof. Mei is full of patience and has a lot of care for his students. I remember many times Prof. Mei encouraged me not to give up after a paper gets rejected, and we worked together to revise the paper and resubmit it.

Special thanks go to my committee members Prof. Arkadi Nemirovski, Prof. Vladimir Koltchinskii, and Prof. Kai Zhang. I took Prof. Nemirovski's *Nonlinear Optimization* and *Advanced Nonlinear Programming* classes. I enjoyed Prof. Nemirovski's teaching style very much, got shocked by his depth and breadth of knowledge, and learned a lot of optimization techniques. I also benefited from the discussion with Prof. Nemirovski on one of my research projects about the convergence rate. I got to know Prof. Koltchinskii by taking his *Linear Statistical Model* class. It was a very interesting class and covered the topic of high-dimensional statistics. Though I did not continue research in this area, it is motivating and points out a lot of interesting future topics. I was fortunate to have a small chat with Prof. Zhang when he visited Georgia Tech in Spring 2020. We discussed how to do research, which motivated me a lot as I had just started my Ph.D. study at that time.

I would also like to thank the advisor of my master's study, Prof. Daren Cline. I

worked with him on a research project during the second year of my master's. This was the first research project that I worked on, and it raised my interest in doing research on the intersection of optimization and statistics. Prof. Cline encouraged me to apply to a Ph.D. program to continue research and suggested the IE program at Georgia Tech to me. Without him, I would never think about starting this journey.

Thanks should also go to my collaborators and friends at Georgia Tech. Starting in the year of 2019, my four-year Ph.D. journey covers the whole COVID-19 era. A lot of campus activities have to go online, and it is truly a hard time. Fortunately, I was able to socialize with my friends, discuss our research projects, and have fun. I would like to express thanks to my lab mates: Yiling Xie, Yijin Ni, Yujie Zhao, Namjoon Suh, Tianyi Zhou, Andy Ko, and Eugene Ndiaye; my friends: Steven Ma, Bill Huang, Shangkun Wang, Qian Wang, Song Wei, Hanyu Zhang, and many others; and also four-legged friends: Sitong, Xiaofang, and Bobbi, etc.

I am also grateful for my family. It has been nearly four years since the last time I stayed with them due to the COVID quarantine policy in China. Though we have this physically long-distance, we still keep close interaction via video chat. I would especially thank my mother, Min Xie, who raised me alone since I was eleven. Life is not easy for her, but she still has a lot of love for her daughter.

Finally, I would like to thank my husband, Xiaobo He. We have made a lot of effort to stay together during the past eight years, and everything was worth it. Without his love and support, I could never make it this far. Now we are very excited to have our newborn baby girl, Xi He. With them, I will continue this life journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

We study the properties of applying stochastic algorithms to solve optimization problems in model estimation. In particular, we investigate the statistical properties of estimators that are based on some stochastic algorithms in Chapters 2, 3, 4, and 5; we propose a new stochastic algorithm and study its optimization property in Chapter 6.

We summarize the main contents in each chapter as follows. In Chapter 2, we explore a directional bias phenomenon in both stochastic gradient descent and gradient descent, and examine their implications for the resulting estimators. We would argue that the outcome from the stochastic gradient descent may lead to a better generalization error bound. In Chapter 3, we study a property of implicit regularization by a variance reduction version of the stochastic mirror descent algorithm. The phenomenon of implicit regularization by applying certain algorithms has attracted a lot of attention, and its existence with the variance reduction based stochastic algorithm is new. In Chapter 4, we establish the equivalence between the variance reduced stochastic mirror descents with a technique that has been developed in information theory – variance reduced stochastic natural gradient descent. The purpose of establishing such an equivalence is that the properties of both problems can automatically be shared with each other. In Chapter 5, we study a recent algorithm – ROOT-SGD – for the online learning problem, and we estimate the covariance of the estimator that is computed via the ROOT-SGD algorithm. Our covariance estimators quantify the uncertainty in the ROOT-SGD algorithm, which are useful for statistical inference. In Chapter 6, we study a constrained strongly convex problem – the entropic OT, and we propose a primal-dual stochastic algorithm with variance reduction to solve it. We show that the computational complexity of our algorithm is better than other first-order algorithms for solving the entropic OT.

CHAPTER 1

INTRODUCTION

In statistics and machine learning, empirical risk minimization (ERM) is widely used in model estimation, which corresponds to the following optimization problem:

$$\min_{\boldsymbol{\theta}} \left\{ F(\boldsymbol{\theta}) := \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{\theta}; \boldsymbol{x}_i) \right\}, \tag{1.1}$$

where $\{\boldsymbol{x}_i\}_{i=1,\ldots,n}$ are the observed samples and $\boldsymbol{\theta}$ is the parameter to estimate. For simplicity, we further denote $f(\boldsymbol{\theta}; \boldsymbol{x}_i)$ as $f_i(\boldsymbol{\theta})$.

Many optimization algorithms have been proposed to solve (1.1). There are two main categories of algorithms: stochastic algorithms and deterministic algorithms. The most popular stochastic algorithm is Stochastic Gradient Descent (SGD), which updates the parameter estimation as follows:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \nabla f_{i_t}(\boldsymbol{\theta}_t).$$

And a deterministic algorithm is the Gradient Descent (GD):

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \frac{1}{n} \sum_{i} \nabla f_i(\boldsymbol{\theta}_t).$$

Each step of the deterministic algorithm requires calculating the gradient of the summation, while each step of the stochastic algorithm only uses the gradient of one term in the summation. The computation for each step of the stochastic algorithm is only a fraction of that of the deterministic algorithm, so the stochastic algorithm can be advantageous for a large-scale problem.

Chapters 2, 3 and 4 analyze the properties of applying stochastic algorithms to solve

optimization problems under the framework of empirical risk minimization. The details about them are presented below.

In Chapter 2, we analyze the point estimator from the stochastic gradient descent (SGD) applied to the nonparametric regression model, the kernel regression in particular. We show that SGD has a solution path biased towards a certain direction that is different from that of the gradient descent (GD). We further use this phenomenon to explain why the SGD estimator has a better generalization performance. Our results extend those in the current literature for the linear regression model, and are supported by numerical experiments using simulated and real data sets.

In Chapter 3, we analyze the properties of variance-reduced stochastic mirror descent (VRSMD) applied to a finite-sum objective function. We prove the convergence rate of the VRSMD and show that it converges to an implicit regularized solution in linear regression. Our finding covers both overfitting and underfitting scenarios in linear regression; note that the latter case is not included in the current literature. We also show the implicit regularized solution from VRSMD can find a sparse ground truth with a small error. Such a theoretical guarantee for the sparsity of the mirror descent solution is also new to the literature.

In Chapter 4, we compare the variance-reduced stochastic natural gradient descent (VRSNGD) and VRSMD algorithms for estimating parameters in a statistical model and show that the two algorithms are equivalent under suitable conditions. The equivalence allows one to generalize the understanding of one algorithm to the other; for example, in Chapter 4 we prove the convergence rate of VRSNGD based on the convergence rate of VRSMD, and show the parameterization invariance of VRSMD based on such property of VRSNGD. Based on the implicit regularization property of VRSMD studied in Chapter 3, one may investigate if a similar property holds for VRSNGD.

We also study the model estimation problem in an online setting, which leads to a different formulation than the empirical risk minimization formulation in (1.1). This is reported in Chapter 5. In online learning, the data points come in a sequence, and we are not able

to store the data due to its large size. In this case, a stochastic algorithm updates the parameter estimate using only the most recent (one or two) estimate(s) and the current data point. In particular, we analyze a stochastic algorithm in online learning: the ROOT-SGD algorithm. The ROOT-SGD is a stochastic gradient-based algorithm with fast convergence and asymptotic normality. Our goal is to estimate the covariance of the estimator that is computed via the ROOT-SGD algorithm. For a general online convex learning optimization problem, we provide two covariance estimators for the estimate from the ROOT-SGD algorithm. Our first estimator uses the stochastic Hessian. This estimator converges to the true asymptotic covariance at a rate that is faster than those in the literature. Our second estimator is purely based on the parameter estimates from ROOT-SGD. We show that the second estimator is also an asymptotically consistent estimator. Our covariance estimators can be used for statistical inference in ROOT-SGD. For example, the covariance estimator can facilitate hypothesis testing in this setting; it can help build confidence intervals as well. We perform numerical studies to support the asymptotic consistency of our estimators.

Besides the empirical risk minimization problem, the stochastic gradient descent methods can be applied to other problems that have the summation structure (i.e., the objective function in the optimization problem is a finite sum; and the gradients can be computed for each term in the summation). In Chapter 6, we study a constrained optimization problem: Optimal Transport (OT). In particular, we propose a primal-dual stochastic algorithm with variance reduction for solving it, and we study the optimization property of our algorithm. We show that the computational complexity of our algorithm is the best among the state-of-the-art algorithms, that is, the convergence rate of our algorithm is better than all other first-order algorithms. We further speed up our stochastic algorithm by a mini-batch algorithm that enables parallel computing. There is a sublinear speed-up in the batch version of our algorithm. Numerical experiments support our theoretical findings.

In summary, we studied stochastic algorithms and their applications in machine learning. Various properties of stochastic algorithms are analyzed. For some ERM problems,

we provided a theoretical guarantee of the generalization error of the model estimate that is based on a stochastic algorithm. And for the online learning problems, we proposed new estimators of the covariance of the output of a stochastic algorithm. Furthermore, we showed that we can achieve consistent statistical inference in the online learning using our covariance estimators. For a particular constrained problem, we improved the stochastic algorithm to solve the optimization problem faster than the existing algorithms.

CHAPTER 2
THE DIRECTIONAL BIAS HELPS STOCHASTIC GRADIENT DESCENT TO GENERALIZE IN KERNEL REGRESSION MODELS

We study the Stochastic Gradient Descent (SGD) algorithm in nonparametric statistics: kernel regression in particular. The directional bias property of SGD, which is known in the linear regression setting, is generalized to the kernel regression. More specifically, we prove that SGD with moderate and annealing step size converges along the direction of the eigenvector that corresponds to the largest eigenvalue of the gram matrix. In addition, the Gradient Descent (GD) with a moderate or small step size converges along the direction that corresponds to the smallest eigenvalue. These facts are referred to as the directional bias properties; they may interpret how an SGD-computed estimator has a potentially smaller generalization error than a GD-computed estimator. The application of our theory is demonstrated by simulation studies and a case study that is based on the FashionMNIST dataset.

2.1 Introduction

The Stochastic Gradient Descent (SGD) is a popular optimization algorithm that has a wide range of applications, including generalized linear model in statistics and deep Neural Network in machine learning. One main advantage of the SGD is the computational scalability due to the low cost per iteration. Recent work also indicates that the SGD might also lead to outcomes that possess nice statistical properties under the linear regression framework, see [1].

In this chapter, we study the statistical properties of the SGD under nonparametric regression models. We focus on the Reproducing Kernel Hilbert Space (RKHS) model, which is popular in both statistics and machine learning communities and is often simply

referred to as the "kernel trick", see [2, 3]. The kernel method can be applied in various domains such as image processing [4] and text mining [5].

Our main approach is to analyze the directional bias of the SGD algorithm under the RKHS model, which can help us to explain why the outcome of SGD has good generalization properties. Directional bias, also referred to as implicit bias, means that an algorithm generates a solution path that is biased towards a certain direction, and it is also closely related to implicit regularization in deep learning [6]. Directional bias also means that the algorithm prefers some directions over the others even though they may have the same objective function value. For example, [7] shows the directional bias of SGD and GD in the linear regression model, and analyzes the relationship between the directional bias and the generalization error in such a setting. There is no directional bias result in kernel regression, but one may expect it to exist and to explain the generalization performance of the outcome of SGD in the kernel regression model.

The state-of-the-art result on the directional bias of SGD can be divided into two categories, based on their underlying techniques, mostly under the linear regression model. The first category is the stochastic gradient flow method where one assumes an infinitesimal step size in SGD and thus the parameter dynamic follows a stochastic differential equation, see [8, 9, 10]. The second category is to analyze the discrete SGD sequence, which in general requires a moderate step size such that the algorithm converges, see [7] for an example in linear regression. Our approach belongs to the second category.

We want to point out that there is more research to study the directional bias of the Gradient Descent (GD) than for the SGD. For instance, for Neural Networks in the 'lazy training' regime, [11] shows that GD converges in the direction of the smallest eigenvalue of the Neural Tangent Kernel; for high-dimension regression, [12, 13] shows that GD finds a sparse estimation.

Our contributions are two folded. First, we study the directional bias of (S)GD in a nonparametric regression model. Though the nonparametric regression is well studied

in statistics, the directional bias is a relatively new concept [7]. Second, we unify the conditions to guarantee the directional bias of GD and SGD sequences. The main condition is the diagonally dominant gram matrix, which covers a large class of kernel functions. To the best of our knowledge, our findings and techniques are not yet discovered in the current literature.

Our result can shed new light on deep learning [14]. By the state-of-the-art mathematical theory of Neural Networks (NN), kernel and/or nonparametric methods can approximate the functional space of neural networks, see for example the NTK theory [15], and the Radon bounded variance space description for ReLU NN [16]. These phenomena can lead to interesting future research.

Organization. The rest of this chapter is organized as follows: In Section 2.2, we give problem formulation, formalize the algorithms and state our assumption; In Section 2.3, we state our main theory, including the directional bias result and its implication for generalization; In Section 2.4, we perform numerical experiments; The detailed proof and experiments are deferred to Section 2.5, the technical details of this chapter.

2.2 Problem Formulation

In Section 2.2.1, we define the kernel regression; in Section 2.2.2, we present the SGD and GD algorithms; in Section 2.2.3, we state our assumption for later analysis. We also provide a simple example to justify the assumption.

2.2.1 Kernel Regression

Suppose we have n data pairs $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ generated from an unknown model $y = f(\boldsymbol{x}) + \xi$, where $\boldsymbol{x}_i \in \mathcal{X} \subset \mathcal{R}^p$, $y_i \in \mathcal{R}$ and ξ_i's are i.i.d. noise. The goal is to estimate the unknown model f from the data. One solution is to minimize the empirical risk function

$$\min_f \frac{1}{n} \sum_{i=1}^n \ell(y_i, f(\boldsymbol{x}_i)), \tag{2.1}$$

where ℓ is the loss function. A popular choice for regression task is the squared loss $\ell(y, \boldsymbol{x}) = \frac{1}{2}(y - f(\boldsymbol{x}))^2$.

One can see that problem (2.1) is not well-defined, as there are infinitely many solutions to $\forall i : f(\boldsymbol{x}_i) = y_i$, and some of them do not generalize for new test data. One way to fix it is to restrict $f \in \mathcal{H}$ and penalize $\|f\|_{\mathcal{H}}$ for smoothness, where $\mathcal{H}$ is a RKHS with reproducing kernel $K(\cdot, \cdot)$ and $\|\cdot\|_{\mathcal{H}}$ is the Hilbert norm. Adding these restrictions and applying Representer Theorem, problem (2.1) with the squared loss becomes

$$\min_{\boldsymbol{\alpha} \in \mathcal{R}^n} \frac{1}{2n} \sum_{i=1}^{n} (y_i - \boldsymbol{K}_i^T \boldsymbol{\alpha})^2 = \frac{1}{2n} \|\boldsymbol{y} - K\boldsymbol{\alpha}\|_2^2, \tag{2.2}$$

where $\boldsymbol{K}_i^T$ is the ith row of $K := K(X, X) = (K(\boldsymbol{x}_i, \boldsymbol{x}_j))_{i,j}$. For a parameter $\boldsymbol{\alpha}$, the corresponding estimator in $\mathcal{H}$ is $f(\cdot) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}_i, \cdot) := \boldsymbol{\alpha}^T K(\cdot, X)$.

Now when K is invertible, it is trivial that any algorithm on objective function (2.2) converges at the unique minimizer $\hat{\boldsymbol{\alpha}} = K(X, X)^{-1}\boldsymbol{y}$, so the RKHS functional estimator is

$$\hat{f}(\boldsymbol{x}) = K(\boldsymbol{x}, X)^T K(X, X)^{-1} \boldsymbol{y}, \tag{2.3}$$

where $K(\boldsymbol{x}, X)^T = (K(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, K(\boldsymbol{x}, \boldsymbol{x}_n))$. Estimator (2.3) is the minimum norm interpolant, i.e.,

$$\arg\min_{f \in \mathcal{H}} \{\|f\|_{\mathcal{H}} : f(\boldsymbol{x}_i) = y_i, i = 1, \ldots, n\},$$

whose properties are studied in [17].

In this chapter, we compare the convergence direction of SGD and GD to $\hat{\alpha}$. Specifically, we consider a two-stage SGD with a phase transition from a larger step size to a decreased step size. Note that this matches the training scheme people always use in practice for SGD algorithms: decreasing the step size after training for a few epochs. For that

purpose, in the following sections, we define the one-step SGD/GD update and state our assumptions and notations for analysis.

2.2.2 One step SGD/GD update

For objective function (2.2), denote the parameter estimation at tth step as α_t, then SGD update $\boldsymbol{\alpha}_{t+1}$ as

$$\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t - \eta_t(\boldsymbol{K}_{i_t}^T\boldsymbol{\alpha}_t - y_{i_t}) \cdot \boldsymbol{K}_{i_t}, \tag{2.4}$$

where i_t is uniformly random sampled from $\{1, \ldots, n\}$.

The GD update $\boldsymbol{\alpha}_{t+1}$ as

$$\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t - \frac{\eta_t}{n}K^T(K\boldsymbol{\alpha}_t - \boldsymbol{y}). \tag{2.5}$$

2.2.3 Assumptions and Notations

We state our assumption on the gram matrix as follows:

Assumption 2.1 (Diagonally Dominant gram matrix)**.** *Denote by $K = K(X, X)$ the gram matrix, we assume that K is diagonally dominant. Specifically, suppose w.l.o.g. that $K_{1,1} \geq K_{2,2} \geq \ldots \geq K_{n,n} > 0$, then we have for a small value τ that*

$$|K_{i,j}| \leq \tau \ll K_{n,n}, \forall i \neq j.$$

Remark 2.1. *A diagonally dominant gram matrix is common in kernel learning. Mathematically, one can justify that a gram matrix is diagonally dominant by imposing proper assumptions on the kernel function $K(\cdot, \cdot)$ and the data distribution. As an example, we show diagonal dominance for the bilinear kernel. One can check that other popular kernels can have diagonally dominant kernel matrices too.*

Remark 2.2. *Thinking of the kernel function as the inner product of high-dimensional features, the resulting gram matrix is diagonally dominant when the high-dimension features are sparse. This happens in a lot of practical problems [18, 19], for example, linear or string kernels being applied to text data [5], domain-specific kernels being applied to image retrieval [20] and bioinformatics [21], and the Global Alignment kernel being applied to most datasets [22, 23].*

Proposition 2.1 (Lemma 1 in [7])**.** *Consider the bilinear kernel $K(\boldsymbol{x}, \boldsymbol{x}') := \langle \boldsymbol{x}, \boldsymbol{x}' \rangle$. Assume the data $\boldsymbol{x}_i, i = 1, \ldots, n$, are i.i.d. uniformly distributed on the unit sphere S^{d-1}, where $d \gg n$. When $d \geq 4\log(2n^2/\delta)$ for some $\delta \in (0, 1)$. Then with probability at least $1 - \delta$, we have*

$$|K_{i,j}| = |\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle| < \tilde{\tau} := \tilde{\mathcal{O}}(1/\sqrt{d}) \ll K_{n,n} = 1, \forall i \neq j.$$

Though commonly exists, the diagonal dominance is undesired in classification and clustering tasks. It indicates that the data points are dissimilar to each other, which means not enough information for classification/clustering. There are some efforts for solving the issue of diagonal dominance in these cases, see for example [5, 24]. But for the regression task, the diagonal dominance, in other words, the dissimilarity of data points, may have benefits. One can find similar conditions such as Restricted Isometry Property and s-goodness that describes linearly dissimilar features in a regression literature [25, 26]. Such conditions are required for proving minimax optimality or exact recovery of a sparse signal in many settings. In our case, we adopt the dissimilarity concept and apply it to data points in high-dimensional nonlinear feature space. Later we will see that in our case, the directional bias drives SGD to select a good solution that generalizes well among all solutions of the same level of empirical loss. In this way, our SGD estimator benefits from the diagonal dominance.

Notations. We use the following notations throughout this chapter. For the gram matrix

K, let $K_{i,j}$ denote its (i,j)th element. Denote $\lambda_i = K_{i,i} = K(\boldsymbol{x}_i, \boldsymbol{x}_i)$, and assume w.l.o.g. that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$. Denote the ith column of K as $\boldsymbol{K}_i$, let $K_{-1} = [K_2, \ldots, K_n]$. Denote P_{-1} the projection onto column space of K_{-1}, and $P_1 = I - P_{-1}$. And denote $\gamma_1 \geq \ldots \geq \gamma_n > 0$ the eigenvalues of K in non-increasing order.

2.3 Main result

The main results are presented in two subsections: Section 2.3.1 states the directional bias results of SGD and GD estimators, respectively; Section 2.3.2 shows that certain directional bias leads to good generalization performance, and applies this result to show that an outcome from SGD potentially generalizes better than an outcome from GD.

2.3.1 Directional bias

By our assumption, K will be full rank, (S)GD on (2.2) converges at $\hat{\boldsymbol{\alpha}} = K^{-1}\boldsymbol{y}$. We are interested in the direction at which $\boldsymbol{\alpha}_t$ converges to $\hat{\boldsymbol{\alpha}}$, i.e., the quantity

$$\boldsymbol{b}_t := \boldsymbol{\alpha}_t - \hat{\boldsymbol{\alpha}}.$$

With assumption 2.1 that the gram matrix is diagonally dominant, we prove that a two-stage SGD has $\boldsymbol{b}_t$ converge in the direction that is aligned with the eigenvector associated with the largest eigenvalue of the gram matrix K.

Theorem 2.1 (directional bias of an SGD-based estimator)**.** *Assume Assumption 2.1 holds, run a two-stage SGD with a fixed step size for each stage: stage 1 with step size η_1 for steps $1, \ldots, k_1$, stage 2 with step size η_2 for steps $k_1 + 1, \ldots, k_2$, such that*

$$\frac{2}{\lambda_1^2 - C_1\sqrt{n}\tau} < \eta_1 < \frac{2}{\lambda_2^2 + C_2\sqrt{n}\tau},$$
$$\eta_2 < \frac{1}{\lambda_1^2 + C_3\sqrt{n}\tau},$$

where C_1, C_2, C_3 are constants. For a small $\epsilon > 0$ such that $n\tau < poly(\epsilon)$ there exists $k_1 = \mathcal{O}(\log \frac{1}{\epsilon})$ and k_2 such that

$$(1 - 2\epsilon)\gamma_1 \leq \mathbb{E}[\|K\boldsymbol{b}_{k_2}^{SGD}\|_2]/\mathbb{E}[\|\boldsymbol{b}_{k_2}^{SGD}\|_2] \leq \gamma_1.$$

That is, $\boldsymbol{b}_{k_2}^{SGD}$ is close to the direction of eigenvector corresponding to the largest eigenvalue of K.

Remark 2.3. *One should assume τ in Assumption 2.1 to be small enough for ϵ to be very small if one would like the resulting estimator $\boldsymbol{b}_{k_2}^{SGD}$ to have the direction that corresponds to the largest eigenvalue of K. Later we will see that if one only wants different directional bias of SGD and GD estimators, a moderate ϵ is allowed, the assumption on τ is not that strong.*

Next, we see the different convergence direction of GD.

Theorem 2.2 (directional bias of a GD-based estimator)**.** *Assume Assumption 2.1 holds, run GD with a fixed step size η such that*

$$\eta < n/(\lambda_1 + n\tau)^2.$$

For a $\epsilon' > 0$, let $k = \mathcal{O}(\log \frac{1}{\epsilon'})$, we have the GD estimator after k steps satisfying:

$$\gamma_n \leq \|K\boldsymbol{b}_k^{GD}\|_2/\|\boldsymbol{b}_k^{GD}\|_2 \leq \sqrt{1+\epsilon'}\gamma_n.$$

That is, $\boldsymbol{b}_k^{GD}$ is close to the direction that corresponds to the smallest eigenvalue of K.

Remark 2.4. *The assumption (on τ) is mild for differentiating the directional bias of SGD and GD. Comparing Theorem 2.1 and 2.2,when $\gamma_n < (1 - 2\epsilon)\gamma_1$, taking k large enough we have*

$$\frac{\|K\boldsymbol{b}_k^{GD}\|_2}{\|\boldsymbol{b}_k^{GD}\|_2} < \frac{\mathbb{E}\|K\boldsymbol{b}_{k_2}^{SGD}\|_2}{\mathbb{E}\|\boldsymbol{b}_{k_2}^{SGD}\|_2}.$$

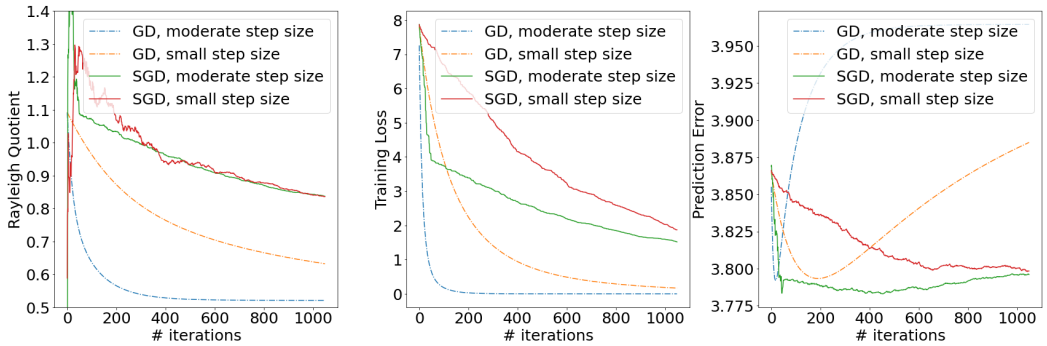


Figure 2.1: Kernel regression on synthetic data. The first plot shows directional bias by Rayleigh Quotient(RQ):= $\|K\boldsymbol{b}\|_2^2/\|\boldsymbol{b}\|_2^2$. The SGD indeed converges in the direction of a larger RQ, which matches our Theorems 2.1 and 2.2. In the third plot, we show the prediction error of the solution paths, and the SGD does have a lower prediction error than GD, even though GD has a smaller training loss than SGD. This supports Theorem 2.4.

That is, one may expect $\boldsymbol{b}_{k_2}^{SGD}$ *to be in the direction of larger eigenvalue compared with* $\boldsymbol{b}_k^{GD}$. *In the following subsection, we will see that the directional bias towards a larger eigenvalue of the kernel is good for generalization. That is, directional bias helps an SGD-based estimator to generalize.*

Though assumption 2.1 appears in Theorem 2.2, it is just used to bound the step size so that GD converges; the diagonally dominant structure of K is not required. Moreover, the choice of ϵ' is independent of τ, then for an arbitrarily small $\epsilon' > 0$, run GD long enough then the theorem will apply. The estimator $\boldsymbol{b}_k^{GD}$ can be arbitrarily close to the eigenvector that correspond to the smallest eigenvalue.

2.3.2 Effect of directional bias

In this subsection, the estimator that is biased towards the largest eigenvalue of the Hessian is shown to be the best for parameter estimation, see Theorem 2.3. Later we define a realizable problem setting of kernel regression where the generalization error depends on the parameter estimation error, and in this way, the directional bias helps an SGD-based estimator to generalize.

Theorem 2.3. *Consider minimizing the quadratic loss function*

$$L(\boldsymbol{w}) = \|A\boldsymbol{w} - \boldsymbol{y}\|_2^2.$$

Assume there is a ground truth $\boldsymbol{w}^*$ such that $\boldsymbol{y} = A\boldsymbol{w}^*$. For a fixed level of the quadratic loss, the parameter estimation error $\|\boldsymbol{w} - \boldsymbol{w}^*\|_2^2$ has a lower bound:

$$\forall \boldsymbol{w} \in \{\boldsymbol{w} : L(\boldsymbol{w}) = a\} : \|\boldsymbol{w} - \boldsymbol{w}^*\|_2^2 \geq a/\|A^T A\|_2.$$

Moreover, the equality is obtained when $\boldsymbol{w} - \boldsymbol{w}^*$ is in the direction of the eigenvector that corresponds to the largest eigenvalue of matrix $A^T A$.

Remark 2.5. *Theorem 2.3 implies that the directional bias towards the largest eigenvalue is good for parameter estimation. As discussed in Remark 2.4, the SGD-based estimator is biased towards a larger eigenvalue compared to the GD-based estimator, by Theorem 2.3 the SGD estimator potentialy better estimates the true parameter and thus generalizes better, which we will formalize later.*

Suppose $\exists f^* \in \mathcal{H}$ such that $y = f^*(\boldsymbol{x})$. Consider the generalization error $L_D(f) := \|f - f^*\|_{\mathcal{H}}^2$. For an algorithm output f^{alg}, we decompose its generalization error as:

$$\begin{aligned} & L_D(f^{\text{alg}}) - \inf_{f \in \mathcal{H}} L_D(f) \\ = & \underbrace{L_D(f^{\text{alg}}) - \inf_{f \in \mathcal{H}_s} L_D(f)}_{:=\Delta(f^{\text{alg}}),\ \text{estimation error}} + \underbrace{\inf_{f \in \mathcal{H}_s} L_D(f) - \inf_{f \in \mathcal{H}} L_D(f)}_{\text{approximation error}}, \end{aligned}$$

where $\mathcal{H}_s$ is the hypothesis class that the output of the algorithm is restricted to. By formulation (2.2), we have $\mathcal{H}_s$:

$$\mathcal{H}_s = \{f \in \mathcal{H} : f = \boldsymbol{\alpha}^T K(\cdot, X), \boldsymbol{\alpha} \in \mathcal{R}^n\}.$$

We define the a-level set of training loss:

$$\nu_a = \{f \in \mathcal{H}_s : f = \boldsymbol{\alpha}^T K(\cdot, X), \frac{1}{2n}\|K\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 = a\},$$

and denote $\Delta_a^* := \inf_{f \in \nu_a} \Delta(f)$.

Note that the approximation error can not be improved unless we change the hypothesis class, which is, changing the problem formulation in our case. So we just minimize the estimation error for estimators that are in the a-level set. One can check the estimation error is given by

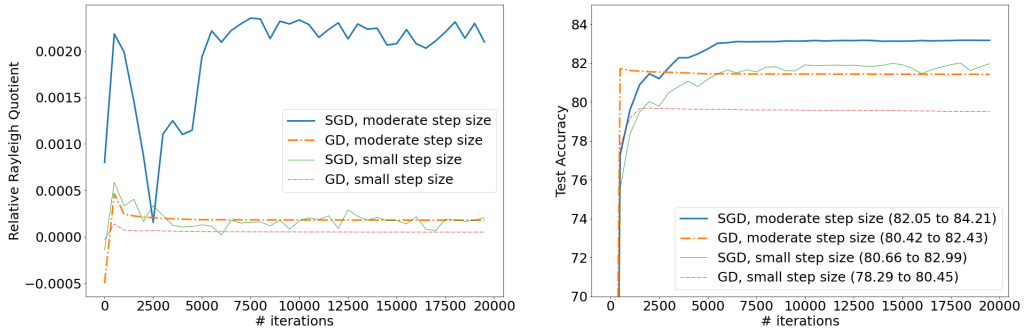
$$f \in \mathcal{H}_s : \Delta(f) = \boldsymbol{b}^T K \boldsymbol{b},$$

where $\boldsymbol{b} = \boldsymbol{\alpha} - \hat{\boldsymbol{\alpha}}$. Similar to Theorem 2.3, the estimation error is minimized when $\boldsymbol{b}$ is in the direction of the largest eigenvalue of K, so the directional bias towards a larger eigenvalue helps to generalize in kernel regression. We compare the estimation error of SGD and GD in following theorem.

Theorem 2.4 (Generalization performance)**.** *Follow Theorems 2.1 and 2.2, we have the following:*

- *The output of SGD has* $\mathbb{E}[\Delta^{1/2}(f^{SGD})] \leq (1 + 4\epsilon)(\Delta_a^*)^{1/2}$*, where* a *is such that* $\mathbb{E}[\|K\boldsymbol{\alpha}^{SGD} - y\|_2]^2 = 2na$ *and* ϵ *could be any positive small constant;*
- *The output of GD has* $\Delta(f^{GD}) \geq M\Delta_a^*$*, where* a *is the training loss of GD estimator, and* $M = \frac{\gamma_1}{\gamma_n}(1 - \epsilon') > 1$ *is a large constant.*

Remark 2.6. *This theorem indicates that* $\mathbb{E}[\Delta^{1/2}(f^{SGD})] \leq \Delta^{1/2}(f^{GD})$ *when* $1 + 4\epsilon \leq M^{1/2}$*. Taking* $\epsilon < (\sqrt{\gamma_1/\gamma_n} - 1)/4$ *in Theorem 2.1 and combining with Theorem 2.2 which states that* $\epsilon' \overset{k\to\infty}{\longrightarrow} 0$*, we will have* $1 + 4\epsilon \leq M^{1/2}$ *holds. In this way,* $\Delta(f^{SGD}) < \Delta(f^{GD})$ *with high probability. This finishes our claim that SGD generalizes better than GD.*



(a) Relative Rayleigh Quotient. (b) Test accuracy

Figure 2.2: The experiment of a small ResNet on FashionMNIST. In (a), we follow [7] to use the Relative Rayleigh Quotient(RRQ) as the measurement of the convergence direction. The SGD with a moderate step size has a higher RRQ than the GD with either a moderate step size or small step size, which supports the theory in Theorems 2.1 and 2.2. It is interesting to observe that SGD with a small step size has a different directional bias compared with SGD with a moderate step size, indicating that the directional bias studied in this chapter does not hold for a general SGD. In (b), we plot the testing accuracy from 20 repetitions of experiments, the test accuracy (inside bracket) of SGD with moderate step size is higher than the other cases, and we have Wilcoxon signed-rank test to confirm that the difference is significant at 0.01 level. The test accuracy validates Theorem 2.4.

2.4 Numeric Studies

Simulation. We simulate data from a nonlinear regression model with Gaussian additive noise as $y_i = \sum_{j=1}^{100} \sin(x_{i,j}) + \epsilon_i$, where $x_{i,j} \sim N(0,1)$ and $\epsilon_i \sim N(0, 0.01)$. We fit kernel regression using the polynomial kernel $K(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle + .01)^2$ on 10 training data and test the estimator on 5 testing data. We run both SGD and GD for two step size schemes: small step size, and moderate annealing step size. The results are in Fig. 2.1.

Real data experiment. We run a 6-layer ResNet [27] on FashionMNIST. The network structure is

$$\begin{aligned} \text{Input} &\Rightarrow 7 \times 7 \text{ Conv} \Rightarrow \text{BatchNorm} \Rightarrow \text{ReLU} \\ &\Rightarrow 3 \times 3 \text{ MaxPool} \Rightarrow \text{ResBlock1} \Rightarrow \text{ResBlock2} \\ &\Rightarrow \text{Global AvePool} \Rightarrow \text{FC} \Rightarrow \text{output}. \end{aligned}$$

We run SGD and GD for two step size schemes, similar to our simulation. There are $1,500$ training data and $10,000$ testing data in our experiment. The details of the data and algorithm implementation are deferred to Section 2.5.8.2. The result is in Fig. 2.2.

Remark 2.7. *The purposes of experiment using a Neural Network (Fig. 2.2) are: first, the Neural Network results support our finding on kernel regression, since Neural Network is related to kernel regression through NTK theory [15]; second, our experiment indicates that our result may be empirically true for the more general deep learning framework [14].*

Proof Sketch for Theorem 2.1

We show the proof sketch for a special case, the proof for general case is similar subject to some modifications. Consider the case when $K = diag(\lambda_1, \ldots, \lambda_n)$ where $\lambda_1 > \lambda_2 \geq \ldots \geq \lambda_n$, the first stage of SGD with step size $\eta_1 \in (\frac{2}{\lambda_1^2}, \frac{2}{\lambda_2^2})$ will have for the direction of the first data point:

$$(\boldsymbol{\alpha}_{t+1})_1 = (\boldsymbol{\alpha}_t)_1 - \eta_1\lambda_1(\lambda_1(\boldsymbol{\alpha}_t)_1 - \lambda_1(\hat{\boldsymbol{\alpha}})_1).$$

Thus

$$\begin{aligned}(\boldsymbol{\alpha}_{t+1} - \hat{\boldsymbol{\alpha}})_1 &= (1 - \eta_1\lambda_1^2)(\boldsymbol{\alpha}_t - \hat{\boldsymbol{\alpha}})_1 \\ \Longrightarrow |(\boldsymbol{\alpha}_{t+1} - \hat{\boldsymbol{\alpha}})_1| &= |1 - \eta_1\lambda_1^2||(\boldsymbol{\alpha}_t - \hat{\boldsymbol{\alpha}})_1| > |(\boldsymbol{\alpha}_t - \hat{\boldsymbol{\alpha}})_1|,\end{aligned}$$

while for the other data points, we have:

$$|(\boldsymbol{\alpha}_{t+1} - \hat{\boldsymbol{\alpha}})_i| = |1 - \eta_1\lambda_i^2||(\boldsymbol{\alpha}_t - \hat{\boldsymbol{\alpha}})_i| < |(\boldsymbol{\alpha}_t - \hat{\boldsymbol{\alpha}})_i|.$$

That is, the first stage of SGD does not converge in the direction corresponding to λ_1 and converges for other directions. After run the first stage long enough, we will have all directions sufficiently fitted to data except the first eigenvector. Now in second stage, we

decrease the step size so that the first eigenvector direction is fitted. Since it is the only direction that remains to be fitted, the estimator will converge in this direction.

Proof Sketch for Theorem 2.2

Denote the eigen decomposition of K:

$$K = G\Gamma G^T, \Gamma = diag(\gamma_1, \ldots, \gamma_n), G = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n].$$

Denote $\boldsymbol{w}_t := G^T(\boldsymbol{\alpha}_t - \hat{\boldsymbol{\alpha}})$, we can rewrite GD update in $\boldsymbol{w}_t$:

$$\begin{aligned} \boldsymbol{w}_{t+1} &= \boldsymbol{w}_t - \frac{\eta}{n}\Gamma^2\boldsymbol{w}_t = (I - \frac{\eta}{n}\Gamma^2)\boldsymbol{w}_t \\ \Longrightarrow (\boldsymbol{w}_t)_i &= (1 - \eta\gamma_i^2/n)^t(\boldsymbol{w}_0)_i. \end{aligned}$$

So, for $\eta < \frac{n}{\gamma_1^2}$, we have $|1 - \eta\gamma_1^2/n| \leq \ldots \leq |1 - \eta\gamma_n^2/n|$. The direction corresponding to larger eigenvalue is fitted faster, leaving the direction of smaller eigenvalue to be fitted later, which is the direction of convergence after several steps of GD.

2.5 Technique Details for Chapter 2

In this section, we give technique details for Chapter 2.

2.5.1 Background on RKHS

This section details the background on RKHS in two subsections. The first subsection includes notations, theorems, and an example of RKHS, the second section reduces the kernel regression in RKHS from infinite dimension to finite dimension, which gives our objective function (2.2) .

2.5.1.1 *Nonparametric model in RKHS*

In this subsection, we give the definition and notations for our model in RKHS, and its associated norms, basis, etc. The definitions are similar to those in [28].

Given n data pairs $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathcal{X} \subset \mathcal{R}^p$ and $y_i \in \mathcal{R}$, assume that y_is are associated with $\boldsymbol{x}_i$s through $f(\boldsymbol{x}_i)$, where $f(\cdot)$ is some unknown function in the reproducing kernel Hilbert space (RKHS) of functions $\mathcal{X} \to \mathcal{R}$, our goal is to estimate the function $f(\cdot)$ from the data.

Denote the RKHS where f lives as $\mathcal{H}$, with reproducing kernel $K : \mathcal{X} \times \mathcal{X} \to \mathcal{R}_+$ (which is known to us). And we associate the functions in $\mathcal{H}$ with probability measure $\mathbb{Q}$, assume w.l.o.g. that $\int_{\mathcal{R}^p} f(\boldsymbol{a}) d\mathbb{Q}(\boldsymbol{a}) = 0$. By Mercer's theorem, K has eigen-expansion:

$$K(\boldsymbol{a}, \boldsymbol{b}) = \sum_{j=1}^{\infty} \gamma_j \phi_j(\boldsymbol{a}) \phi_j(\boldsymbol{b})$$

Where $\{\phi_j\}_{j=1}^{\infty}$ are orthonormal basis in $\mathbb{L}^2(\mathbb{Q})$, w.r.t. the usual inner product in $\mathbb{L}^2(\mathbb{Q})$ as

$$\langle g(\cdot), h(\cdot) \rangle_{\mathbb{L}^2(\mathbb{Q})} = \int_{\mathcal{X}} g(\boldsymbol{a}) h(\boldsymbol{a}) d\mathbb{Q}(\boldsymbol{a})$$

Now for any $f \in \mathcal{H}$, we can expand $f(\cdot) = \sum_{j=1}^{\infty} c_j \phi_j(\cdot)$, where $c_j = \langle f(\cdot), \phi_j(\cdot) \rangle_{\mathbb{L}^2(\mathbb{Q})}$. And for $f(\cdot) = \sum_{j=1}^{\infty} c_j \phi_j(\cdot)$, $g(\cdot) = \sum_{j=1}^{\infty} c'_j \phi_j(\cdot)$, by Parseval's theorem

$$\langle f(\cdot), g(\cdot) \rangle_{\mathbb{L}^2(\mathbb{Q})} = \sum_{j=1}^{\infty} c_j c'_j$$

And we have another inner product that is defined for RKHS $\mathcal{H}$ as

$$\langle f(\cdot), g(\cdot) \rangle_{\mathcal{H}} = \sum_{j=1}^{\infty} \frac{c_j c'_j}{\gamma_j}$$

The reproducing property of RKHS says that $\forall f \in \mathcal{H}$, we have

$$\langle f(\cdot), K(\cdot, \boldsymbol{x})\rangle_{\mathcal{H}} = f(\boldsymbol{x})$$

Cubic Splines Formulate a RKHS. We go over an example of RKHS for better understanding. Consider the cubic spline of one dimension, we can show that the space of cubic splines is a RKHS. One can also find the cubic spline example in [29]. For more details on the relationship between polynomial smoothing splines and RKHS, one can check Section 1.2 of [3] .

Assume w.l.o.g. that $x_i \in \mathcal{X} = [0, 1] \subset \mathcal{R}$. The cubic spline f on $\mathcal{X}$ is continuous, has a continuous first-order derivative and square integrable second order derivative. By Taylor's theorem with remainder, we have

$$\begin{aligned} f(t) &= f(0) + tf'(0) + \int_0^t (t-u)f''(u)du \\ &= f(0) + tf'(0) + \int_0^1 (t-u)_+ f''(u)du \end{aligned}$$

where $(t-u)_+ = \max\{0, t-u\}$. Let $\mathcal{B}$ be the set of cubic splines f on $[0, 1]$ that satisfies the boundary condition $f(0) = f'(0) = 0$, then for $f \in \mathcal{B}$

$$f(t) = \int_0^1 (t-u)_+ f''(u)du$$

Let $G(t, u) = (t-u)_+$, then we claim that $\mathcal{B}$ is RKHS with reproducing kernel

$$K(s, t) = \int_0^1 G(s, u)G(t, u)du$$

and inner product

$$\langle f, g\rangle_{\mathcal{B}} = \int_0^1 f''(u)g''(u)du$$

as one can check the reproducing property

$$\begin{aligned}\langle f(\cdot), K(\cdot,t)\rangle_{\mathcal{B}} &= \int_0^1 \frac{\partial^2 K(u,t)}{\partial u^2} f''(u)du \\ &= \int_0^1 (t-u)_+ f''(u)du = f(t)\end{aligned}$$

2.5.1.2 Optimization problem considered

This subsection gives problem formulation of kernel regression. Given data pairs $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ and RKHS $\mathcal{H}$, consider a loss function ℓ which is selected according to how y is connected with $f(\boldsymbol{x})$, we may estimate the model by

$$\begin{aligned}&\min_{f\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^n \ell(y_i, f(\boldsymbol{x}_i)) \\ =&\min_{c_j} \frac{1}{n}\sum_{i=1}^n \ell(y_i, \sum_{j=1}^\infty c_j\phi_j(\boldsymbol{x}_i))\end{aligned} \tag{2.6}$$

Example for ℓ includes

- Squared error loss $\ell(y,f) = (y-f)^2$, which is usually used in regression;
- 0-1 loss $\ell(y,f) = \mathbf{1}(y*f>0)$, for binary classification;
- Logistic loss $\ell_(y,f) = \log(1+\exp(-y*f))$, also a loss function for classification, can be considered as a surrogate function of 0-1 loss, and is the same as negative log likelihood function in logistic regression.

Let us come back to the nonparametric model part, to control the model smoothness, the usual practice is to add a penalty to objective (2.6), result in

$$\min_{f\in\mathcal{H}} \sum_{i=1}^n \ell(y_i, f(\boldsymbol{x}_i)) + \lambda \text{pen}(f)$$

A popular choice of pen(f) is $\|f\|_{\mathcal{H}}^2$, or any strictly increasing function of $\|f\|_{\mathcal{H}}^2$. Such method is explicitly controlling the model smoothness, and by Representer Theorem, it has

solution of the form

$$f(\cdot) = \sum_{i=1}^{n} \alpha_i K(\cdot, \boldsymbol{x}_i) \tag{2.7}$$

Plug equation (2.7) into objective (2.6), we have the problem becomes

$$\min_{\alpha_{i'}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, \sum_{i'=1}^{n} \alpha_{i'} K(\boldsymbol{x}_i, \boldsymbol{x}_{i'}))$$

Which gives the formulation (2.2) under loss function $\ell(y, f(\boldsymbol{x})) = (y - f(\boldsymbol{x}))^2/2$.

2.5.2 Diagonal Dominance of Some Popular Kernels

In this section, we justify Assumption 2.1 by figuring out a problem setting where some popular kernels give a diagonal dominant gram matrix. For simplicity, we assume the following data distribution throughout this section:

$$\mathbf{x}_i \in \mathcal{R}^d, i = 1, \ldots, n, \text{ are normalized such that } \|\boldsymbol{x}_i\|_2^2 = 1; \tag{2.8a}$$

$$\text{The direction of } \boldsymbol{x}_i\text{s are i.i.d. uniformly distributed on the unit sphere } S^{d-1}; \tag{2.8b}$$

$$d \gg n \text{ (overparameterized setting).} \tag{2.8c}$$

Given assumption set (2.8), we can bound the inner product of data $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ with high probability as follows:

Lemma 2.1 (Lemma 1 in [7])**.** *Under assumption set* (2.8)*, let* $d \geq 4\log(2n^2/\delta)$ *for some* $\delta \in (0, 1)$*. Then with probability at least* $1 - \delta$*, we have*

$$|\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle| < \tilde{\tau} := \tilde{\mathcal{O}}(\frac{1}{\sqrt{d}}), \forall i \neq j$$

Proof. See proof of Lemma 1 in [7]. □

The bound on the inner product $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ induces bound on $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ for some popu-

lar kernels. We show the diagonal dominance for two groups of kernels in the following propositions, and list some examples for kernels in each group.

Proposition 2.2 (Inner product kernel)**.** *The inner product kernel is defined as a smooth transformation of inner product. We can write it as:*

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = g(\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle)$$

Assume assumptions (2.8) *hold, and assume the function* $g : [-1, 1] \to R$ *satisfies:*

$$g \text{ is convex}; \tag{2.9a}$$

$$g \text{ is } L-\text{smooth, that is, } \nabla g \text{ is } L-\text{Lipschitz continuous}; \tag{2.9b}$$

$$|g(0)| \leq c\tilde{\tau} \text{ for a constant } c, g'(0) \geq 0. \tag{2.9c}$$

we will have with probability $1 - \delta$

$$|K_{i,j}| \leq (c + g'(0))\tilde{\tau} + \frac{L}{2}\tilde{\tau}^2 \text{ for } i \neq j \tag{2.10}$$

where δ *and* $\tilde{\tau}$ *the same as in Lemma 2.1.When* $g'(0)\tilde{\tau} + \frac{L}{2}\tilde{\tau}^2 \ll g(1)$ *for a small enough* $\tilde{\tau}$*, the gram matrix is diagonal dominant.*

Proof. We have the following with probability at least $1 - \delta$ by Lemma 2.1. For any off-diagonal elements of K:

$$\begin{aligned} K_{i,j} &= g(\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle) \\ &\geq g(0) + g'(0)\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\ &\geq -(g'(0) + c)\tilde{\tau} \end{aligned}$$

and

$$\begin{aligned} K_{i,j} &= g(\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle) \\ &\leq g(0) + g'(0)\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + \frac{L}{2}\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle^2 \\ &\leq (c + g'(0))\tilde{\tau} + \frac{L}{2}\tilde{\tau}^2 \end{aligned}$$

Thus $|K_{i,j}| \leq (c + g'(0))\tilde{\tau} + \frac{L}{2}\tilde{\tau}^2$. □

Remark 2.8. *We list some examples of inner product kernels that give diagonal dominant kernel matrices:*

- ***Bilinear Kernel:*** $K(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{x}, \boldsymbol{x}' \rangle$*, then*

$$|K(\boldsymbol{x}_i, \boldsymbol{x}_j)| \leq \tilde{\tau} \ll K(\boldsymbol{x}_n, \boldsymbol{x}_n) = 1.$$

- ***Polynomial Kernel:*** $K(\boldsymbol{x}, \boldsymbol{x}') = (\langle \boldsymbol{x}, \boldsymbol{x}' \rangle + c)^m$ *for* $m \in \mathbb{N}$ *and* $c \sim \mathcal{O}(\tilde{\tau})$*, then by Proposition 2.2*

$$|K(\boldsymbol{x}_i, \boldsymbol{x}_j)| \leq (1 + m)\tilde{\tau} + \frac{m * \exp((m-1)\tilde{\tau})}{2}\tilde{\tau}^2$$

  when $(1 + m)\tilde{\tau} + \frac{m * \exp((m-1)\tilde{\tau})}{2}\tilde{\tau}^2 \ll (1 + c)^m$*, we have diagonal dominant gram matrix.*

- ***Hyperbolic Tangent Kernel (Sigmoid Kernel):*** $K(\boldsymbol{x}, \boldsymbol{x}') = \tanh(\alpha\langle \boldsymbol{x}, \boldsymbol{x}' \rangle + c)$*, where* $\alpha > 0, c \geq 0$*. Note that Hyperbolic Tangent Kernel does not satisfy all the assumptions in Proposition 2.2, one can still calculate*

$$|K(\boldsymbol{x}_i, \boldsymbol{x}_j)| \leq \tanh(\alpha\tilde{\tau} + c)$$

  and

$$K(\boldsymbol{x}_k, \boldsymbol{x}_k) = \tanh(\alpha + c)$$

When $\tanh(\alpha\tilde{\tau} + c) \ll \tanh(\alpha + c)$ *(which is the case when* α *is large, and* $c, \tilde{\tau}$ *are small enough), we have* $|K(\boldsymbol{x}_i, \boldsymbol{x}_j)| \ll K(\boldsymbol{x}_n, \boldsymbol{x}_n)$ *and the gram matrix is diagonal dominant.*

Proposition 2.3 (Radial Basis Function (RBF) kernel)**.** *Radial Basis Function kernel depends on two data points through their distance, which is of following form*

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2), \gamma > 0$$

Assume assumptions (2.8) *hold, when* $\gamma = -c_0 \log(\tilde{\tau})$ *for a constant* c_0*, we have with probability* $1 - \delta$

$$|K_{i,j}| \leq \tilde{\tau}^{2c_0(1-\tilde{\tau})} \ll K_{n,n} = 1, \textit{ for } i \neq j$$

where δ *and* $\tilde{\tau}$ *the same as in Lemma 2.1. That is, the Gram matrix is diagonal dominant.*

Proof. Bound off-diagonal terms of K by Lemma 2.1:

$$\begin{aligned} K_{i,j} &= \exp(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2) \\ &\leq \exp(2\gamma\tilde{\tau} - 2\gamma) \\ &= \tilde{\tau}^{2c_0(1-\tilde{\tau})} \end{aligned}$$

□

Remark 2.9. *We note some popular kernels that are related to Radial Basis Function kernel, and show that they lead to diagonal dominance:*

- ***Gaussian Kernel:*** $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_2^2}{2\sigma^2})$*. One can see that Gaussian Kernel is reparameterizing RBF Kernel by* $\gamma = 1/(2\sigma^2)$*. Thus Gaussian gram matrix is diagonal dominant when* $\sigma^2 \sim \mathcal{O}(-\frac{1}{\log(\tilde{\tau})})$*.*
- ***Laplace Kernel:*** $K(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|_2}{\sigma})$ *for* $\sigma > 0$*. The Laplace Kernel is*

very similar to Gaussian Kernel, and one can check by similar steps that when $\sigma \sim \mathcal{O}(-\frac{1}{\log(\tilde{\tau})})$*, Laplace gram matrix is diagonal dominant.*

2.5.3 Lemmas

This section includes two useful lemmas for characterizing the eigenvalues of a symmetric matrix.

Lemma 2.2 (Gershgorin circle theorem, restated for symmetric matrix)**.** *Let* $A \in \mathcal{R}^{n \times n}$ *be a symmetric matrix. Let* A_{ij} *be the entry in the* $i-$*th row and the* $j-$*th column. Let*

$$R_i(A) := \sum_{j \neq i} |A_{ij}|, i = 1, \ldots, n$$

Consider n Gershgorin discs

$$D_i(A) := \{z \in \mathcal{R}, |z - A_{ii}| \leq R_i(A)\}, i = 1, \ldots, n$$

The eigenvalues of A are in the union of Gershgorin discs

$$G(A) := \cup_{i=1}^{n} D_i(A)$$

Furthermore, if the union of k *of the* n *discs that comprise* $G(A)$ *forms a set* $G_k(A)$ *that is disjoint from the remaining* $n - k$ *discs, then* $G_k(A)$ *contains exactly* k *eigenvalues of* A*, counted according to their algebraic multiplicities.*

Proof. See [30], Chap 6.1, Theorem 6.1.1. □

Lemma 2.3 (Cauchy interlacing theorem, restated for symmetric matrix)**.** *Let* $B \in \mathcal{R}^{m \times m}$ *be a symmetric matrix, let* $\boldsymbol{y} \in \mathcal{R}^n$ *and* $a \in \mathcal{R}$*, and let* $A = \begin{bmatrix} B & \boldsymbol{y} \\ \boldsymbol{y}^T & a \end{bmatrix}$*. Then*

$$\lambda_1(A) \geq \lambda_1(B) \geq \lambda_2(A) \geq \ldots \geq \lambda_m(A) \geq \lambda_m(B) \geq \lambda_{m+1}(A).$$

Proof. See [30], Chap 4.3, Theorem 4.3.17. □

2.5.4 Spectrum of gram matrix

This section analyzes the eigen structure of the gram matrix.

Lemma 2.4 (Characterizing K^2)**.** *Under Assumption 2.1, we have*

$$\langle \boldsymbol{K}_i, \boldsymbol{K}_i \rangle \in [\lambda_i^2, \lambda_i^2 + (n-1)\tau^2] \tag{2.11}$$

$$|\langle \boldsymbol{K}_i, \boldsymbol{K}_j \rangle| \leq [2\lambda_1 + (n-2)\tau]\tau, i \neq j \tag{2.12}$$

Proof. For $\langle \boldsymbol{K}_i, \boldsymbol{K}_i \rangle$

$$\begin{aligned}\langle \boldsymbol{K}_i, \boldsymbol{K}_i \rangle &= K_{i,i}^2 + \sum_{l \neq i} K_{l,i}^2 \\ &\in [\lambda_i^2, \lambda_i^2 + (n-1)\tau^2]\end{aligned}$$

And for $\langle \boldsymbol{K}_i, \boldsymbol{K}_j \rangle, i \neq j$

$$\begin{aligned}|\langle \boldsymbol{K}_i, \boldsymbol{K}_j \rangle| &= |\sum_{l=1}^{n} K_{l,i} K_{l,j}| \\ &= |K_{i,i}K_{i,j} + K_{j,i}K_{j,j} + \sum_{l \neq i,j} K_{l,i}K_{l,j}| \\ &\leq |K_{i,i}K_{i,j}| + |K_{j,i}K_{j,j}| + \sum_{l \neq i,j} |K_{l,i}K_{l,j}| \\ &\leq [\lambda_i + \lambda_j]\tau + (n-2)\tau^2 \\ &\leq [2\lambda_1 + (n-2)\tau]\tau\end{aligned}$$

□

Lemma 2.5 (Eigenvalue of K)**.** *Under Assumption 2.1, we have*

$$\gamma_1 \leq \lambda_1 + n\tau \tag{2.13}$$

$$\gamma_n \geq \lambda_n - n\tau \tag{2.14}$$

If we further assume $\lambda_j + n\tau < \lambda_{j-1} - n\tau$, we will have

$$\gamma_{j-1} \geq \lambda_{j-1} - n\tau > \lambda_j + n\tau \geq \gamma_j.$$

Proof. Use Gershgorin circle theorem, calculate

$$R_i(K) = \sum_{j \neq i} |K_{i,j}| \leq n\tau$$

then

$$D_i(K) \subset [\lambda_i - n\tau, \lambda_i + n\tau].$$

By Gershgorin circle theorem, the lemma claim holds. □

Lemma 2.6 (Characterize $P_1 K$ and $P_{-1} K$)**.** *Recall our definition: P_{-1} is the projection on column space of $K_{-1} = [\boldsymbol{K}_2, \boldsymbol{K}_3, \ldots, \boldsymbol{K}_n]$, and $P_1 = I - P_{-1}$. We claim the following hold*

$$P_1 K = [P_1 \boldsymbol{K}_1, \mathbf{0}, \ldots, \mathbf{0}] \tag{2.15}$$

$$P_{-1} K = [P_{-1} \boldsymbol{K}_1, \boldsymbol{K}_2, \ldots, \boldsymbol{K}_n] \tag{2.16}$$

Assume τ is small enough that $n\tau \leq \mathcal{O}(1)$, $\lambda_n - n\tau \geq c_1 > 0$ and $\lambda_1 + n\tau \leq c_2$, let $c_3 := \frac{c_2}{c_1^2}$, then we have the following:

$$\|P_{-1} \boldsymbol{K}_1\|_2 \in \left[0, c_3(2\lambda_1 + (n-2)\tau)\sqrt{n}\tau\right] \tag{2.17}$$

$$\|P_1 \boldsymbol{K}_1\|_2 \in \left[\sqrt{\lambda_1^2 - c_3^2[2\lambda_1 + (n-2)\tau]^2 n\tau^2}, \lambda_1 + \sqrt{n}\tau\right]. \tag{2.18}$$

Proof. For $i \neq 1$, we calculate

$$P_{-1}\boldsymbol{K}_i = \boldsymbol{K}_i$$

$$P_1\boldsymbol{K}_i = \boldsymbol{K}_i - P_{-1}\boldsymbol{K}_i = \mathbf{0}$$

thus we have equations (2.15) and (2.16).

For $\|P_{-1}\boldsymbol{K}_1\|_2$:

$$\begin{aligned}
&\|P_{-1}\boldsymbol{K}_1\|_2 \\
=&\|K_{-1}(K_{-1}^TK_{-1})^{-1}K_{-1}^T\boldsymbol{K}_1\|_2 \\
\leq&\|K_{-1}(K_{-1}^TK_{-1})^{-1}\|_2\|K_{-1}^T\boldsymbol{K}_1\|_2
\end{aligned}$$

where

$$\begin{aligned}
&\|K_{-1}^T\boldsymbol{K}_1\|_2^2 \\
=&\sum_{i=2}^{n}(\boldsymbol{K}_i^T\boldsymbol{K}_1)^2 \\
\overset{(2.12)}{\leq}&(n-1)[2\lambda_1+(n-2)\tau]^2\tau^2 \\
\leq&[2\lambda_1+(n-2)\tau]^2n\tau^2
\end{aligned}$$

and $K_{-1}^TK_{-1}$ has all eigenvalues in $[\gamma_n^2, \gamma_1^2]$ by Cauchy interlacing theorem (Lemma 2.3), that is, all singular values of K_{-1} are in $[c_1, c_2]$ by our assumption. Then

$$\|K_{-1}(K_{-1}^TK_{-1})^{-1}\|_2 \leq \frac{c_2}{c_1^2} := c_3$$

So we have

$$\|P_{-1}\boldsymbol{K}_1\|_2 \leq \|K_{-1}(K_{-1}^TK_{-1})^{-1}\|_2\|K_{-1}^T\boldsymbol{K}_1\|_2$$

$$\leq c_3[2\lambda_1 + (n-2)\tau]\sqrt{n}\tau.$$

For $\|P_1\boldsymbol{K}_1\|_2$:

$$\begin{aligned}
&\|P_1\boldsymbol{K}_1\|_2 \leq \|\boldsymbol{K}_1\|_2 \\
\leq&(\lambda_1^2 + (n-1)\tau^2)^{.5} \\
\leq&\lambda_1 + \sqrt{n}\tau
\end{aligned}$$

and

$$\begin{aligned}
&\|P_1\boldsymbol{K}_1\|_2^2 \\
=&\|\boldsymbol{K}_1\|_2^2 - \|P_{-1}\boldsymbol{K}_1\|_2^2 \\
\geq&\lambda_1^2 - c_3^2[2\lambda_1 + (n-2)\tau]^2 n\tau^2.
\end{aligned}$$

□

Lemma 2.7 (Spectrum of $H_{-1} := P_{-1}KK^TP_{-1}$)**.** *Assume*

$$c_3^2[2\lambda_1 + (n-2)\tau]^2 n\tau^2 + 2[2\lambda_1 + (n-2)\tau]n\tau \leq \lambda_n^2$$

We have the following:

- 0 *is an eigenvalue of* H_{-1}*, corresponding eigenspace is the column space of* P_1*;*
- *Restricted to the column space of* P_{-1}*, the eigenvalues of* H_{-1} *are all in the interval:*

$$\left(\lambda_n^2 - [2\lambda_1 + (n-2)\tau]n\tau, \lambda_2^2 + [2\lambda_1 + (n-1)\tau]n\tau\right).$$

Proof. The first claim is by construction of P_1 and P_{-1}.

For the second claim, note that H_{-1} has the same eigenvalues as

$$H'_{-1} = (P_{-1}K)^T P_{-1}K$$

Now the diagonal entries of H'_{-1} are:

$$(H'_{-1})_{ii} = \|P_{-1}\boldsymbol{K}_i\|_2^2 = \begin{cases} \|P_{-1}\boldsymbol{K}_1\|_2^2 \leq c_3^2[2\lambda_1 + (n-2)\tau]^2 n\tau^2 & , i = 1 \\ \|\boldsymbol{K}_i\|_2^2 \in [\lambda_i^2, \lambda_i^2 + (n-1)\tau^2] & , i \neq 1 \end{cases}$$

And the off-diagonal entries of H'_{-1} are:

$$|(H'_{-1})_{ij}| = |\langle P_{-1}\boldsymbol{K}_i, P_{-1}\boldsymbol{K}_j\rangle| = |\langle \boldsymbol{K}_i, \boldsymbol{K}_j\rangle| \leq [2\lambda_1 + (n-2)\tau]\tau$$

To use Gershgorin circle theorem, calculate

$$R_i(H'_{-1}) = \sum_{j\neq i} |(H'_{-1})_{ij}| < [2\lambda_1 + (n-2)\tau]n\tau$$

Thus the Gershgorin discs:

$$D_1(H'_{-1}) \in (\|P_{-1}\boldsymbol{K}_1\|_2^2 - [2\lambda_1 + (n-2)\tau]n\tau, \|P_{-1}\boldsymbol{K}_1\|_2^2 + [2\lambda_1 + (n-2)\tau]n\tau)$$
$$D_i(H'_{-1}) \in (\|\boldsymbol{K}_i\|_2^2 - [2\lambda_1 + (n-2)\tau]n\tau, \|\boldsymbol{K}_i\|_2^2 + [2\lambda_1 + (n-2)\tau]n\tau)$$

when $c_3^2[2\lambda_1 + (n-2)\tau]^2 n\tau^2 + [2\lambda_1 + (n-2)\tau]n\tau \leq \lambda_n^2 - [2\lambda_1 + (n-2)\tau]n\tau$, the first Gershgorin discs does not intersect with the others, so we have $n-1$ nonzero eigenvalues in

$$\cup_{i=2}^n D_i(H'_{-1}) \subset \left(\lambda_n^2 - [2\lambda_1 + (n-2)\tau]n\tau, \lambda_2^2 + [2\lambda_1 + (n-1)\tau]n\tau\right).$$

□

2.5.5 Directional bias of SGD with moderate step size

This section gives formal proof of Theorem 2.1 and specifies the constants. The proof is done in four steps: Lemma 2.8 analyzes one update of SGD; Lemma 2.9 uses Lemma 2.8 to bound the first stage updates of SGD with moderate step size; Lemma 2.10 again uses Lemma 2.8, and bounds the second stage updates of SGD with small step size; finally, Theorem 2.5 combines Lemma 2.9 and Lemma 2.10 to formalize the directional bias of SGD, it is the same as Theorem 2.1, but restated using the constants defined therein.

Lemma 2.8 (One step update of SGD)**.** *Under Assumption 2.1, denote* $A_t := \mathbb{E}[\|P_1 \boldsymbol{b}_t\|_2]$, $B_t := \mathbb{E}[\|P_{-1} \boldsymbol{b}_t\|_2]$, *fix a constant* $c_4 \geq (\lambda_1 + \sqrt{n}\tau)(2\lambda_1 + (n-2)\tau)c_3$, *then we have:*

$$A_{t+1} \leq q_1(\eta) A_t + \xi(\eta) B_t \tag{2.19}$$

$$A_{t+1} \geq q_1(\eta) A_t - \xi(\eta) B_t \tag{2.20}$$

$$B_{t+1} \leq q_{-1}(\eta) B_t + \xi(\eta) A_t \tag{2.21}$$

where

$$q_1(\eta) = \frac{n-1}{n} + \frac{1}{n}|1 - \eta \|P_1 \boldsymbol{K}_1\|_2^2|$$

$$q_{-1}(\eta) = \sqrt{1 + \frac{\|K P_{-1} \boldsymbol{b}_t\|_2^2}{n \|P_{-1} \boldsymbol{b}_t\|_2^2} [\eta^2(\lambda_2^2 + (n-1)\tau^2) - 2\eta]}$$

$$\xi(\eta) = c_4 \eta n^{-1/2} \tau.$$

Proof. One step of SGD update is:

$$\boldsymbol{b}_{t+1} = \boldsymbol{b}_t - \eta \boldsymbol{K}_i \boldsymbol{K}_i^T \boldsymbol{b}_t = [I - \eta \boldsymbol{K}_i \boldsymbol{K}_i^T] \boldsymbol{b}_t$$

where i is uniformly random sample from $[1, \ldots, n]$.

For inequalities (2.19) and (2.20), check

$$\begin{aligned} P_1\boldsymbol{b}_{t+1} &= P_1[I-\eta\boldsymbol{K}_i\boldsymbol{K}_i^T]\boldsymbol{b}_t \\ &= P_1\boldsymbol{b}_t - \eta P_1\boldsymbol{K}_i\boldsymbol{K}_i^T(P_1+P_{-1})\boldsymbol{b}_t \\ &= [I-\eta P_1\boldsymbol{K}_i\boldsymbol{K}_i^T P_1]P_1\boldsymbol{b}_t - \eta[P_1\boldsymbol{K}_i\boldsymbol{K}_i^T P_{-1}]P_{-1}\boldsymbol{b}_t \end{aligned}$$

where $P_1\boldsymbol{K}_i$ and $P_1\boldsymbol{b}_t$ are in the same 1 dimensional linear space, thus

$$\begin{aligned} & P_1\boldsymbol{K}_i\boldsymbol{K}_i^T P_1 P_1\boldsymbol{b}_t \\ =&\|P_1\boldsymbol{K}_i\|_2\|P_1\boldsymbol{b}_t\|_2 \text{sign}(\langle P_1\boldsymbol{K}_i, P_1\boldsymbol{b}_t\rangle)P_1\boldsymbol{K}_i \\ =&\|P_1\boldsymbol{K}_i\|_2^2 P_1\boldsymbol{b}_t \\ \Rightarrow\quad & [I-\eta P_1\boldsymbol{K}_i\boldsymbol{K}_i^T P_1]P_1\boldsymbol{b}_t = [1-\eta\|P_1\boldsymbol{K}_i\|_2^2]P_1\boldsymbol{b}_t \end{aligned}$$

and

$$\begin{aligned} &\|[P_1\boldsymbol{K}_i\boldsymbol{K}_i^T P_{-1}]P_{-1}\boldsymbol{b}_t\|_2 \\ \le&\|P_1\boldsymbol{K}_i\|_2\|P_{-1}\boldsymbol{K}_i\|_2\|P_{-1}\boldsymbol{b}_t\|_2. \end{aligned}$$

Then

$$\mathbb{E}[\|P_1\boldsymbol{b}_{t+1}\|_2|\boldsymbol{b}_t] \le \mathbb{E}[|1-\eta\|P_1\boldsymbol{K}_i\|_2^2|]\|P_1\boldsymbol{b}_t\|_2 + \eta\mathbb{E}[\|P_1\boldsymbol{K}_i\|_2\|P_{-1}\boldsymbol{K}_i\|_2]\|P_{-1}\boldsymbol{b}_t\|_2 \tag{2.22}$$

$$\mathbb{E}[\|P_1\boldsymbol{b}_{t+1}\|_2|\boldsymbol{b}_t] \ge \mathbb{E}[|1-\eta\|P_1\boldsymbol{K}_i\|_2^2|]\|P_1\boldsymbol{b}_t\|_2 - \eta\mathbb{E}[\|P_1\boldsymbol{K}_i\|_2\|P_{-1}\boldsymbol{K}_i\|_2]\|P_{-1}\boldsymbol{b}_t\|_2 \tag{2.23}$$

where

$$\mathbb{E}[|1-\eta\|P_1\boldsymbol{K}_i\|_2^2|]$$

$$
\begin{aligned}
=&\frac{1}{n}\sum_{i=1}^{n}|1-\eta\|P_1\boldsymbol{K}_i\|_2^2| \\
=&\frac{n-1}{n}+\frac{1}{n}|1-\eta\|P_1\boldsymbol{K}_1\|_2^2| \\
:=&q_1(\eta)
\end{aligned}
$$

and

$$
\begin{aligned}
&\eta\mathbb{E}[\|P_1\boldsymbol{K}_i\|_2\|P_{-1}\boldsymbol{K}_i\|_2] \\
=&\eta\frac{1}{n}\sum_{i=1}^{n}\|P_1\boldsymbol{K}_i\|_2\|P_{-1}\boldsymbol{K}_i\|_2 \\
=&\frac{\eta}{n}\|P_1\boldsymbol{K}_1\|_2\|P_{-1}\boldsymbol{K}_1\|_2 \\
\leq&\frac{\eta}{n}(\lambda_1+\sqrt{n}\tau)c_3(2\lambda_1+(n-2)\tau)\sqrt{n}\tau \\
\leq&\frac{\eta}{n}c_4\sqrt{n}\tau:=\xi(\eta)
\end{aligned}
\tag{2.24}
$$

where the first inequality by upper bounds (2.17) and (2.18), second inequality by $n\tau \leq \mathcal{O}(1)$. Plug the term (2.24) into inequalities (2.22) and (2.23), take expectation on both sides, we get claims (2.19) and (2.20).

For inequality (2.21), check

$$
\begin{aligned}
P_{-1}\boldsymbol{b}_{t+1}&=P_{-1}[I-\eta\boldsymbol{K}_i\boldsymbol{K}_i^T]\boldsymbol{b}_t \\
&=[I-\eta P_{-1}\boldsymbol{K}_i\boldsymbol{K}_i^TP_{-1}]P_{-1}\boldsymbol{b}_t-\eta[P_{-1}\boldsymbol{K}_i\boldsymbol{K}_i^TP_1]P_1\boldsymbol{b}_t
\end{aligned}
$$

Then we have

$$
\begin{aligned}
&\mathbb{E}[\|P_{-1}\boldsymbol{b}_{t+1}\|_2|\boldsymbol{b}_t] \\
\leq&\frac{1}{n}\sum_{i=1}^{n}\|[I-\eta P_{-1}\boldsymbol{K}_i\boldsymbol{K}_i^TP_{-1}]P_{-1}\boldsymbol{b}_t\|_2+\eta\mathbb{E}[\|P_{-1}\boldsymbol{K}_i\|_2\|P_1\boldsymbol{K}_i\|_2]\|P_1\boldsymbol{b}_t\|_2
\end{aligned}
$$

$$\overset{(2.24)}{\leq} \frac{1}{n}\sum_{i=1}^{n} \|[I-\eta P_{-1}\boldsymbol{K}_i\boldsymbol{K}_i^T P_{-1}]P_{-1}\boldsymbol{b}_t\|_2 + \xi(\eta)\|P_1\boldsymbol{b}_t\|_2$$

where

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n} \|[I-\eta P_{-1}\boldsymbol{K}_i\boldsymbol{K}_i^T P_{-1}]P_{-1}\boldsymbol{b}_t\|_2 \\
=&\frac{1}{n}\sum_{i=1}^{n} \sqrt{\|[I-\eta P_{-1}\boldsymbol{K}_i\boldsymbol{K}_i^T P_{-1}]P_{-1}\boldsymbol{b}_t\|_2^2} \\
=&\frac{1}{n}\sum_{i=1}^{n} \sqrt{\|P_{-1}\boldsymbol{b}_t\|_2^2 + \eta^2\|P_{-1}\boldsymbol{K}_i\boldsymbol{K}_i^T P_{-1}P_{-1}\boldsymbol{b}_t\|_2^2 - 2\eta\langle P_{-1}\boldsymbol{b}_t, P_{-1}\boldsymbol{K}_i\boldsymbol{K}_i^T P_{-1}P_{-1}\boldsymbol{b}_t\rangle} \\
=&\frac{1}{n}\sum_{i=1}^{n} \sqrt{\|P_{-1}\boldsymbol{b}_t\|_2^2 + \eta^2(\boldsymbol{K}_i^T P_{-1}P_{-1}\boldsymbol{b}_t)^2\|P_{-1}\boldsymbol{K}_i\|_2^2 - 2\eta(\boldsymbol{K}_i^T P_{-1}P_{-1}\boldsymbol{b}_t)^2} \\
=&\frac{1}{n}\sum_{i=1}^{n} \sqrt{1+\frac{(\boldsymbol{K}_i^T P_{-1}P_{-1}\boldsymbol{b}_t)^2}{\|P_{-1}\boldsymbol{b}_t\|_2^2}(\eta^2\|P_{-1}\boldsymbol{K}_i\|_2^2-2\eta)}\|P_{-1}\boldsymbol{b}_t\|_2 \\
\leq&\sqrt{1+\frac{1}{n}\sum_{i=1}^{n}\frac{(\boldsymbol{K}_i^T P_{-1}P_{-1}\boldsymbol{b}_t)^2}{\|P_{-1}\boldsymbol{b}_t\|_2^2}(\eta^2\|P_{-1}\boldsymbol{K}_i\|_2^2-2\eta)}\|P_{-1}\boldsymbol{b}_t\|_2
\end{aligned}$$

where the last inequality from Jensen's inequality. Now the term

$$\begin{aligned}
&\frac{1}{n}\sum_{i=1}^{n}\frac{(\boldsymbol{K}_i^T P_{-1}P_{-1}\boldsymbol{b}_t)^2}{\|P_{-1}\boldsymbol{b}_t\|_2^2}(\eta^2\|P_{-1}\boldsymbol{K}_i\|_2^2-2\eta) \\
\overset{(2.11)}{\leq}&\frac{1}{n}\sum_{i=1}^{n}\frac{(\boldsymbol{K}_i^T P_{-1}P_{-1}\boldsymbol{b}_t)^2}{\|P_{-1}\boldsymbol{b}_t\|_2^2}\left[\eta^2(\lambda_2^2+(n-1)\tau^2)-2\eta\right] \\
=&\frac{\|KP_{-1}\boldsymbol{b}_t\|_2^2}{n\|P_{-1}\boldsymbol{b}_t\|_2^2}\left[\eta^2(\lambda_2^2+(n-1)\tau^2)-2\eta\right]
\end{aligned}$$

Let

$$q_{-1}(\eta) := \sqrt{1+\frac{\|KP_{-1}\boldsymbol{b}_t\|_2^2}{n\|P_{-1}\boldsymbol{b}_t\|_2^2}\left[\eta^2(\lambda_2^2+(n-1)\tau^2)-2\eta\right]}$$

combine all three inequalities above, take the expectation w.r.t. $\boldsymbol{b}_t$, we have claim (2.21).

□

Lemma 2.9 (Long run behavior of SGD with moderate step size)**.** *Assume $\boldsymbol{b}_0$ is away from*

0, $\lambda_n^2 > (2\lambda_1+(n-2)\tau)n\tau+c_4\sqrt{n}\tau$, $\lambda_2^2+c_6\sqrt{n}\tau < \lambda_1^2-c_5\sqrt{n}\tau$ *where* c_5, c_6 *are constants such that*

$$c_5 \geq c_3^2[2\lambda_1+(n-2)\tau]^2\sqrt{n}\tau + c_4$$

$$c_6 \geq \frac{\sqrt{n}\tau - c_4^2 n^{-.5}\tau/[\lambda_n^2-(2\lambda_1+(n-2)\tau)n\tau] + \lambda_2^2 c_4/[\lambda_n^2-[2\lambda_1+(n-2)\tau]n\tau]}{1-c_4\sqrt{n}\tau/[\lambda_n^2-(2\lambda_1+(n-2)\tau)n\tau]}$$

Consider first k_1 *steps of SGD updates with step size* η*:*

$$\frac{2}{\lambda_1^2-c_5\sqrt{n}\tau} < \eta < \frac{2}{\lambda_2^2+c_6\sqrt{n}\tau}$$

Fix a $\beta_0 \leq A_0$*, then for* $0<\epsilon<1$ *and* $0<\beta<\beta_0$ *such that* $\sqrt{n}\tau \leq poly(\epsilon\beta)$*, there exists* $k_1 = \mathcal{O}(\log\frac{1}{\epsilon\beta})$ *satisfying:*

- $B_{k_1} \leq \epsilon\beta$
- $A_{k_1} \leq \|\boldsymbol{b}_0\|_2 * \rho_1^{k_1} + \epsilon\beta/2$ *for some* $\rho_1 > 1$
- $A_k > \beta_0$ *for* $k = 0, \ldots, k_1$.

Proof. For this choice of η, denote $q_1 = q_1(\eta), q_{-1} = q_{-1}(\eta), \xi = \xi(\eta)$. By Lemma 2.4, we have

$$A_k \geq q_1 A_{k-1} - \xi B_{k-1}$$

$$\begin{bmatrix} A_k \\ B_k \end{bmatrix} \leq \begin{bmatrix} q_1 & \xi \\ \xi & q_{-1} \end{bmatrix} \begin{bmatrix} A_{k-1} \\ B_{k-1} \end{bmatrix}$$

Decompose the coefficient matrix as

$$\begin{bmatrix} q_1 & \xi \\ \xi & q_{-1} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \rho_1 & 0 \\ 0 & \rho_{-1} \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

Assume w.l.o.g. that $\sin\theta \geq 0$ (since otherwise we can take $\theta \to \theta + \pi$), then we have

$$\begin{aligned}
\begin{bmatrix} A_k \\ B_k \end{bmatrix} &\leq \begin{bmatrix} q_1 & \xi \\ \xi & q_{-1} \end{bmatrix}^k \begin{bmatrix} A_0 \\ B_0 \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \rho_1^k & 0 \\ 0 & \rho_{-1}^k \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} A_0 \\ B_0 \end{bmatrix} \\
&= \begin{bmatrix} A_0(\rho_1^k \cos^2\theta + \rho_{-1}^k \sin^2\theta) + B_0(\rho_1^k \cos\theta\sin\theta - \rho_{-1}^k \cos\theta\sin\theta) \\ B_0(\rho_{-1}^k \cos^2\theta + \rho_1^k \sin^2\theta) + A_0(\rho_1^k \cos\theta\sin\theta - \rho_{-1}^k \cos\theta\sin\theta) \end{bmatrix} \\
&= \begin{bmatrix} A_0\rho_1^k + (\rho_1^k - \rho_{-1}^k)\sin\theta(B_0\cos\theta - A_0\sin\theta) \\ B_0\rho_{-1}^k + (\rho_1^k - \rho_{-1}^k)\sin\theta(B_0\sin\theta + A_0\cos\theta) \end{bmatrix} \\
&\leq \begin{bmatrix} A_0\rho_1^k + |\rho_1^k - \rho_{-1}^k|\sin\theta\sqrt{B_0^2 + A_0^2} \\ B_0\rho_{-1}^k + |\rho_1^k - \rho_{-1}^k|\sin\theta\sqrt{B_0^2 + A_0^2} \end{bmatrix} \\
&= \begin{bmatrix} A_0\rho_1^k + |\rho_1^k - \rho_{-1}^k|\sin\theta\|\boldsymbol{b}_0\|_2 \\ B_0\rho_{-1}^k + |\rho_1^k - \rho_{-1}^k|\sin\theta\|\boldsymbol{b}_0\|_2 \end{bmatrix}
\end{aligned}$$

We claim the following holds:

$$0 < \rho_{-1} < 1 < \rho_1 \leq q_1 + \xi \tag{2.25a}$$

$$\rho_{-1}^{k_1}\|\boldsymbol{b}_0\|_2 \leq \epsilon\beta/2 \tag{2.25b}$$

$$\rho_1^{k_1}\|\boldsymbol{b}_0\|_2 \sin\theta \leq \epsilon\beta/2 \tag{2.25c}$$

$$(B_0 + \epsilon\beta_0/2)\xi < (q_1 - 1)\beta_0 \tag{2.25d}$$

which we check later. Using inequalities (2.25), we can upper bound B_{k_1} as

$$\begin{aligned}
B_{k_1} &\leq B_0\rho_{-1}^{k_1} + (\rho_1^{k1} - \rho_{-1}^{k_1})\sin\theta\|\boldsymbol{b}_0\|_2 \\
&\leq \|\boldsymbol{b}_0\|_2\rho_{-1}^{k_1} + \rho_1^{k1}\sin\theta\|\boldsymbol{b}_0\|_2
\end{aligned}$$

$$\overset{(2.25b),(2.25c)}{\leq} \epsilon\beta$$

In addition, for $k = 0, \ldots, k_1$

$$\begin{aligned} B_k &\leq B_0\rho_{-1}^k + (\rho_1^k - \rho_{-1}^k)\sin\theta\|\boldsymbol{b}_0\|_2 \\ &\leq B_0 + \rho_1^{k1}\sin\theta\|\boldsymbol{b}_0\|_2 \\ &\overset{(2.25c)}{\leq} B_0 + \epsilon\beta/2 \end{aligned}$$

We now lower bound A_k by mathematical induction. We have $A_0 \geq \beta_0$, assume $A_{k-1} > \beta_0$, then

$$\begin{aligned} A_k &\geq q_1 A_{k-1} - \xi B_{k-1} \\ &\geq q_1\beta_0 - \xi(B_0 + \epsilon\beta/2) \\ &\overset{(2.25d)}{>} q_1\beta_0 - (q_1 - 1)\beta_0 = \beta_0 \end{aligned}$$

For upper bound A_{k_1}, check

$$\begin{aligned} A_{k_1} &\leq A_0\rho_1^{k_1} + (\rho_1^{k_1} - \rho_{-1}^{k_1})\sin\theta\|\boldsymbol{b}_0\|_2 \\ &\leq \|\boldsymbol{b}_0\|_2\rho_1^{k_1} + \rho_1^{k_1}\sin\theta\|\boldsymbol{b}_0\|_2 \\ &\overset{(2.25c)}{\leq} \|\boldsymbol{b}_0\|_2\rho_1^{k_1} + \epsilon\beta/2 \end{aligned}$$

We have all lemma claims proved. Now it remains to check inequalities (2.25). First note that our choice of the upper bound on η guarantees that $q_{-1} < 1$.

For inequality (2.25a)**:** By Gershgorin circle theorem, it suffices to show $q_{-1} + \xi < 1, q_1 - \xi > 1$, then we have $\rho_1 \geq q_1 - \xi > 1$ and $\rho_{-1} \leq q_{-1} + \xi < 1$. In addition, we need the matrix to be positive definite so that $\rho_{-1} > 0$, we just need $q_1 q_{-1} > \xi^2$ to make

the matrix p.d..

$$
\begin{aligned}
& q_{-1} + \xi < 1 \\
\iff & \sqrt{1 + \frac{\|KP_{-1}\boldsymbol{b}_t\|_2^2}{n\|P_{-1}\boldsymbol{b}_t\|_2^2}\left[\eta^2(\lambda_2^2 + (n-1)\tau^2) - 2\eta\right]} < 1 - c_4\eta n^{-1/2}\tau \\
\Longleftarrow & \frac{\|KP_{-1}\boldsymbol{b}_t\|_2^2}{n\|P_{-1}\boldsymbol{b}_t\|_2^2}\left[\eta^2(\lambda_2^2 + (n-1)\tau^2) - 2\eta\right] < c_4^2\eta^2 n^{-1}\tau^2 - 2c_4\eta n^{-1/2}\tau \\
\overset{\text{Lemma 2.7}}{\Longleftarrow} & 2c_4\eta n^{-1/2}\tau \le c_4^2\eta^2 n^{-1}\tau^2 + \frac{\lambda_n^2 - [2\lambda_1 + (n-2)\tau]n\tau}{n}\left[2\eta - \eta^2(\lambda_2^2 + (n-1)\tau^2)\right] \\
\iff & \frac{\lambda_n^2 - [2\lambda_1 + (n-2)\tau]n\tau}{n}\left[\lambda_2^2 + (n-1)\tau^2\right]\eta - c_4^2 n^{-1}\tau^2\eta \\
& \qquad \le 2\frac{\lambda_n^2 - [2\lambda_1 + (n-2)\tau]n\tau}{n} - 2c_4 n^{-1/2}\tau \\
\Longleftarrow & \eta \le 2\frac{1 - c_4\sqrt{n}\tau/[\lambda_n^2 - [2\lambda_1 + (n-2)\tau]n\tau]}{\lambda_2^2 + (n-1)\tau^2 - c_4^2\tau^2/[\lambda_n^2 - [2\lambda_1 + (n-2)\tau]n\tau]} \\
\iff & \eta \le \frac{2}{\lambda_2^2 + \frac{(n-1)\tau^2 - c_4^2\tau^2/[\lambda_n^2 - [2\lambda_1 + (n-2)\tau]n\tau] + \lambda_2^2[c_4\sqrt{n}\tau/[\lambda_n^2 - [2\lambda_1 + (n-2)\tau]n\tau]]}{1 - c_4\sqrt{n}\tau/[\lambda_n^2 - [2\lambda_1 + (n-2)\tau]n\tau]}}
\end{aligned}
$$

which is true by our choice of η.

$$
\begin{aligned}
& q_1 - \xi > 1 \\
\iff & \frac{n-1}{n} + \frac{1}{n}|1 - \eta\|P_1\boldsymbol{K}_1\|_2^2| - c_4\eta n^{-1/2}\tau > 1 \\
\iff & 1 + c_4\eta\sqrt{n}\tau < |1 - \eta\|P_1\boldsymbol{K}_1\|_2^2| \\
\Longleftarrow & \eta\|P_1\boldsymbol{K}_1\|_2^2 - 1 > 1 + c_4\eta\sqrt{n}\tau \\
\Longleftarrow & \eta > \frac{2}{\|P_1\boldsymbol{K}_1\|_2^2 - c_4\sqrt{n}\tau} \\
\overset{(2.18)}{\Longleftarrow} & \eta > \frac{2}{\lambda_1^2 - c_3^2[2\lambda_1 + (n-2)\tau]^2 n\tau^2 - c_4\sqrt{n}\tau}
\end{aligned}
$$

which is true by our lower bound on η.

$$
\begin{aligned}
& q_1 q_{-1} > \xi^2 \\
\Longleftarrow & q_{-1}^2 \ge \xi
\end{aligned}
$$

$$
\begin{aligned}
&\Longleftrightarrow 1+\frac{\|KP_{-1}\boldsymbol{b}_t\|_2^2}{n\|P_{-1}\boldsymbol{b}_t\|_2^2}\left[\eta^2(\lambda_2^2+(n-1)\tau^2)-2\eta\right]>c_4\eta n^{-1/2}\tau\\
&\Longleftarrow 1-2\eta\frac{\lambda_2^2+(2\lambda_1+(n-1)\tau)n\tau}{n}>c_4\eta n^{-1/2}\tau\\
&\Longleftrightarrow \eta<\frac{n}{2\lambda_2^2+2(2\lambda_1+(n-1)\tau)n\tau+c_4\sqrt{n}\tau}
\end{aligned}
$$

which is true.

For inequality (2.25b)**:** It suffices to take

$$
k_1=\frac{\log(\epsilon\beta/(2\|\boldsymbol{b}_0\|_2))}{\log(\rho_{-1})}=\mathcal{O}(\log\frac{1}{\epsilon\beta})
$$

For inequality (2.25c)**:** We just need to show $\sin\theta<(\rho_{-1}/\rho_1)^{k_1}$. Calculate that $\xi/(q_1-q_{-1})=\frac{\cos\theta\sin\theta}{\cos^2\theta-\sin^2\theta}$, then

$$
\begin{aligned}
&\sin\theta<(\rho_{-1}/\rho_1)^{k_1}\\
\Longleftarrow&\xi/(q_1-q_{-1})<0.9(\rho_{-1}/\rho_1)^{k_1}\\
\Longleftarrow&\xi<0.9(q_1-q_{-1})(\frac{q_{-1}-\xi}{q_1+\xi})^{k_1}\\
\Longleftarrow&\xi<0.9(q_1-q_{-1})(\epsilon\beta/(2\|\boldsymbol{b}_0\|_2))^{1-\frac{\log(q_1+\xi)}{\log(q_{-1}-\xi)}}\\
\Longleftarrow&\xi<(q_1-1)poly(\epsilon\beta)\\
\Longleftarrow&\sqrt{n}\tau\leq poly(\epsilon\beta)
\end{aligned}
$$

For inequality (2.25d)**:** Suffice to show

$$\xi < \frac{(q_1 - 1)\beta_0}{B_0 + \epsilon\beta_0/2}$$
$$\Longleftarrow \sqrt{n}\tau \leq \mathcal{O}(1)$$

□

Lemma 2.10 (Long run behavior of SGD with small step size)**.** *Under the same notations and assumptions as Lemma 2.9 unless otherwise specified. Consider another $k_2 - k_1$ steps of SGD update with step size*

$$\eta' < \frac{1}{\lambda_1^2 + c_7\sqrt{n}\tau}$$

where the constant $c_7 \geq \sqrt{n}\tau + c_4$. Then we have for $k > k_1$:

- $B_k \leq \epsilon\beta$
- $A_k \leq \begin{cases} qA_{k-1} & , A_{k-1} > \beta \\ \beta & , A_{k-1} < \beta \end{cases}$

where $q := q_1(\eta') + \xi(\eta')\epsilon < 1$.

Proof. Denote $q_1' = q_1(\eta'), q_{-1}' = q_{-1}(\eta'), \xi' = \xi(\eta')$, then $q = q_1' + \xi'\epsilon$, denote $B = \|\boldsymbol{b}_0\|_2 * \rho_1^{k_1} + \epsilon\beta/2$. We have by proof of Lemma 2.9 that $q_{-1}' + \xi < 1$. We claim the following holds:

$$q < 1 \tag{2.26a}$$
$$\xi' B \leq (1 - q_1')\epsilon\beta \tag{2.26b}$$

Check inequality (2.26a):

$$q_1' + \xi'\epsilon < 1$$
$$\Longleftarrow \frac{n-1}{n} + \frac{1}{n}|1 - \eta'\|P_1\boldsymbol{K}_1\|_2^2| + c_4\eta' n^{-1/2}\tau < 1$$

$$
\begin{aligned}
&\Longleftrightarrow |1-\eta'\|P_1\boldsymbol{K}_1\|_2^2| < 1 - c_4\eta'\sqrt{n}\tau \\
&\Longleftrightarrow c_4\eta'\sqrt{n}\tau < \eta'\|P_1\boldsymbol{K}_1\|_2^2 < 2 - c_4\eta'\sqrt{n}\tau \\
&\Longleftrightarrow \begin{cases} c_4\sqrt{n}\tau < \|P_1\boldsymbol{K}_1\|_2^2 \\ \eta' < \frac{2}{\|P_1\boldsymbol{K}_1\|_2^2 + c_4\sqrt{n}\tau} \end{cases} \\
&\Longleftarrow \begin{cases} \sqrt{n}\tau < \mathcal{O}(1) \\ \eta' < \frac{2}{\lambda_1^2 + n\tau^2 + c_4\sqrt{n}\tau} \end{cases}
\end{aligned}
$$

which are true by assumption.

Check inequality (2.26b):

$$
\begin{aligned}
&\xi' B \le (1-q_1')\epsilon\beta \\
\Longleftrightarrow & c_4\eta' n^{-1/2}\tau(\|\boldsymbol{b}_0\|_2 * \rho_1^{k_1} + \epsilon\beta/2) \le (1 - \frac{n-1}{n} - \frac{1}{n}(1-\eta'\|P_1\boldsymbol{K}_1\|_2^2))\epsilon\beta \\
\Longleftrightarrow & c_4\eta'\sqrt{n}\tau(\|\boldsymbol{b}_0\|_2 * \exp(k_1)^{\log\rho_1} + \epsilon\beta/2) \le \eta'\|P_1\boldsymbol{K}_1\|_2^2\epsilon\beta \\
\Longleftarrow & \sqrt{n}\tau \le poly(\epsilon\beta).
\end{aligned}
$$

With (2.26) we can prove the lemma by mathematical induction. Suppose $B_{k-1} \le \epsilon\beta$ and $A_{k-1} \le A_{k_1} \le B$, then check

$$
\begin{aligned}
B_k &\le \xi' A_{k-1} + q_{-1}' B_{k-1} \\
&\le \xi' B + q_{-1}'\epsilon\beta \\
&\overset{(2.26b)}{\le} \epsilon\beta
\end{aligned}
$$

and

$$
\begin{aligned}
A_k &\le q_1' A_{k-1} + \xi' B_{k-1} \\
&\le q_1' A_{k-1} + \xi'\epsilon\beta
\end{aligned}
$$

$$\leq (q_1' + \xi'\epsilon)\max\{A_{k-1}, \beta\}$$
$$\leq \begin{cases} qA_{k-1} & , A_{k-1} > \beta \\ q\beta < \beta & , A_{k-1} < \beta \end{cases}.$$

□

We recap Theorem 2.1 using our notations in previous lemmas as follows:

Theorem 2.5 (Directional bias of the two-stage SGD)**.** *Use the two stage SGD scheme as defined in Lemma 2.9 and 2.10. Assume $n\tau < poly(\epsilon)$, then there exists $k_1 = \mathcal{O}(\log \frac{1}{\epsilon})$ and k_2 such that*

$$(1 - 2\epsilon)\gamma_1 \leq \frac{\mathbb{E}[\|K\boldsymbol{b}_{k_2}\|_2]}{\mathbb{E}[\|\boldsymbol{b}_{k_2}\|_2]} \leq \gamma_1$$

where γ_1 is the largest eigenvalue of K.

Proof. In Lemma 2.9 let $\beta = \beta_0$, then for $k_1 = \mathcal{O}(\log \frac{1}{\epsilon})$ we have $B_{k_1} \leq \epsilon\beta_0$. For the 2nd stage, by Lemma 2.10 we can early stop at k_2 such that $A_{k_2} \geq \beta_0$ and $A_{k_2+1} < \beta_0$. We then have

$$B_{k_2} \leq \epsilon\beta_0 \leq \epsilon A_{k_2}$$

Then we check

$$\begin{aligned}
&\frac{\mathbb{E}\|K\boldsymbol{b}_{k_2}\|_2}{\mathbb{E}\|\boldsymbol{b}_{k_2}\|_2} \\
=&\frac{\mathbb{E}\sqrt{\|KP_{-1}\boldsymbol{b}_{k_2}\|_2^2 + \|KP_1\boldsymbol{b}_{k_2}\|_2^2 + 2\langle \boldsymbol{K}_1^T P_{-1}\boldsymbol{b}_{k_2}, \boldsymbol{K}_1^T P_1 \boldsymbol{b}_{k_2}\rangle}}{\mathbb{E}\|\boldsymbol{b}_{k_2}\|_2} \\
\geq&\frac{\mathbb{E}\sqrt{\|KP_1\boldsymbol{b}_{k_2}\|_2^2 - 2\|P_{-1}\boldsymbol{K}_1\|_2\|P_1\boldsymbol{K}_1\|_2\|\boldsymbol{b}_{k_2}\|_2^2}}{\mathbb{E}\|\boldsymbol{b}_{k_2}\|_2} \\
\geq&\frac{\mathbb{E}\sqrt{\|KP_1\boldsymbol{b}_{k_2}\|_2^2} - \mathbb{E}\sqrt{2\|P_{-1}\boldsymbol{K}_1\|_2\|P_1\boldsymbol{K}_1\|_2\|\boldsymbol{b}_{k_2}\|_2^2}}{\mathbb{E}\|\boldsymbol{b}_{k_2}\|_2} \\
=&\frac{\mathbb{E}\|\boldsymbol{K}_1^T P_1\boldsymbol{b}_{k_2}\|_2 - \sqrt{2\|P_{-1}\boldsymbol{K}_1\|_2\|P_1\boldsymbol{K}_1\|_2}\mathbb{E}\|\boldsymbol{b}_{k_2}\|_2}{\mathbb{E}\|\boldsymbol{b}_{k_2}\|_2}
\end{aligned}$$

$$
\begin{aligned}
&\overset{(2.17),(2.18)}{\geq} \sqrt{\lambda_1^2 - c_3^2[2\lambda_1 + (n-2)\tau]^2 n\tau^2}\frac{\mathbb{E}\|P_1\boldsymbol{b}_{k_2}\|_2}{\mathbb{E}\|P_1\boldsymbol{b}_{k_2}\|_2 + \mathbb{E}\|P_{-1}\boldsymbol{b}_{k_2}\|_2} \\
&\qquad - \sqrt{2(\lambda_1 + \sqrt{n}\tau)(c_3(2\lambda_1 + (n-2)\tau)\sqrt{n}\tau)} \\
&\geq \sqrt{\lambda_1^2 - c_3^2[2\lambda_1 + (n-2)\tau]^2 n\tau^2}\frac{\beta_0}{\epsilon\beta_0 + \beta_0} \\
&\qquad - \sqrt{2(\lambda_1 + \sqrt{n}\tau)(c_3(2\lambda_1 + (n-2)\tau)\sqrt{n}\tau)} \\
&\overset{(2.13)}{\geq} (\gamma_1 - n\tau - c_3(2\lambda_1 + (n-2)\tau)\sqrt{n}\tau)(1-\epsilon) \\
&\qquad - \sqrt{2(\lambda_1 + \sqrt{n}\tau)(c_3(2\lambda_1 + (n-2)\tau)\sqrt{n}\tau)} \\
&\geq \gamma_1(1-\epsilon) - \gamma_1\epsilon \quad (\text{By } n\tau < poly(\epsilon)) \\
&= \gamma_1(1-2\epsilon)
\end{aligned}
$$

And the upper bound in the theorem is by definition of γ_1. □

2.5.6 Directional bias of GD with moderate or small step size

This section includes the proof of Theorem 2.2. We first rewrite the GD updates as linear combination of eigenvectors. Then the theorem is proved using the transformed variables and finally transformed back to original parameters.

The directional bias of GD does not require diagonal dominant gram matrix.

Reloading notations Denote the eigen decomposition of K:

$$
K = G\Gamma G^T, \Gamma = diag(\gamma_1, \ldots, \gamma_n), G = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n]
$$

where the eigenvectors $\boldsymbol{g}_i$'s are orthogonal. The GD update as

$$
\boldsymbol{\alpha}_{t+1} = \boldsymbol{\alpha}_t - \frac{\eta}{n}K(K\boldsymbol{\alpha}_t - \boldsymbol{y})
$$

Denote $\boldsymbol{w}_t := G^T(\boldsymbol{\alpha}_t - \hat{\boldsymbol{\alpha}})$, we can rewrite GD update in $\boldsymbol{w}_t$:

$$
\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \frac{\eta}{n}\Gamma^2\boldsymbol{w}_t = (I - \frac{\eta}{n}\Gamma^2)\boldsymbol{w}_t
$$

We recap Theorem 2.2 to make reading easier as follows:

Theorem 2.6 (Direction bias of GD)**.** *Assume $\boldsymbol{\alpha}_0$ is away from 0, $\lambda_n + 2n\tau < \lambda_{n-1}$, GD with step size:*

$$\eta < \frac{n}{(\lambda_1 + n\tau)^2}$$

For a small $\epsilon > 0$, take $k = \mathcal{O}(\log \frac{1}{\epsilon})$, we have

$$\gamma_n \leq \frac{\|K(\boldsymbol{\alpha}_k - \hat{\boldsymbol{\alpha}})\|_2}{\|\boldsymbol{\alpha}_k - \hat{\boldsymbol{\alpha}}\|_2} \leq \sqrt{1+\epsilon}\gamma_n$$

Proof. For $i = 1, \ldots, n$, we have

$$w_k^{(i)} = (1 - \eta\gamma_i^2/n)^k w_0^{(i)}$$

Denote $q_i = 1 - \eta\gamma_i^2/n$, then $0 < q_1 \leq \ldots \leq q_n < 1$ since

$$0 < \eta < \frac{n}{(\lambda_1 + n\tau)^2} \overset{(2.13)}{\leq} \frac{n}{\gamma_1^2} \leq \frac{n}{\gamma_i^2}$$

Since $\lambda_n + n\tau < \lambda_{n-1} - n\tau$, we have $\gamma_n < \gamma_{n-1}$ by lemma 2.3, it follows that $q_n > q_{n-1}$. Denote $q = q_{n-1}/q_n < 1$, then

$$\begin{aligned}
&\frac{\sum_{i=1}^{n-1}(w_k^{(i)})^2}{(w_k^{(n)})^2} \\
=&\frac{\sum_{i=1}^{n-1} q_i^{2k}(w_0^{(i)})^2}{q_n^{2k}(w_0^{(n)})^2} \\
\leq&\frac{\sum_{i=1}^{n-1} q_{n-1}^{2k}(w_0^{(i)})^2}{q_n^{2k}(w_0^{(n)})^2} \\
=&q^{2k}\frac{\sum_{i=1}^{n-1}(w_0^{(i)})^2}{(w_0^{(n)})^2}
\end{aligned}$$

Let $q^{2k} \leq \frac{\gamma_n^2 \epsilon (w_0^{(n)})^2}{\gamma_1^2 \sum_{i=1}^{n-1} (w_0^{(i)})^2} \iff k \geq \frac{1}{2} \frac{\log \frac{\gamma_n^2 \epsilon (w_0^{(n)})^2}{\gamma_1^2 \sum_{i=1}^{n-1} (w_0^{(i)})^2}}{\log q} = \mathcal{O}(\log \frac{1}{\epsilon})$, we have

$$\frac{\sum_{i=1}^{n-1} (w_k^{(i)})^2}{(w_k^{(n)})^2} \leq \frac{\gamma_n^2 \epsilon}{\gamma_1^2}$$

Thus

$$\begin{aligned}
\frac{\|K(\boldsymbol{\alpha}_k - \hat{\boldsymbol{\alpha}})\|_2^2}{\|\boldsymbol{\alpha}_k - \hat{\boldsymbol{\alpha}}\|_2^2} &= \frac{\|\Gamma \boldsymbol{w}_k\|_2^2}{\|\boldsymbol{w}_k\|_2^2} \\
&= \frac{\sum_{i=1}^{n} (w_k^{(i)})^2 \gamma_i^2}{\sum_{i=1}^{n} (w_k^{(i)})^2} \\
&= \frac{(w_k^{(n)})^2 \gamma_n^2}{\sum_{i=1}^{n} (w_k^{(i)})^2} + \frac{\sum_{i=1}^{n-1} (w_k^{(i)})^2 \gamma_i^2}{\sum_{i=1}^{n} (w_k^{(i)})^2} \\
&\leq \gamma_n^2 + \frac{\sum_{i=1}^{n-1} (w_k^{(i)})^2}{\sum_{i=1}^{n} (w_k^{(i)})^2} \gamma_1^2 \\
&\leq \gamma_n^2 + \gamma_1^2 \frac{\gamma_n^2}{\gamma_1^2} \epsilon = \gamma_n^2 (1 + \epsilon)
\end{aligned}$$

thus

$$\frac{\|K(\boldsymbol{\alpha}_k - \hat{\boldsymbol{\alpha}})\|_2}{\|\boldsymbol{\alpha}_k - \hat{\boldsymbol{\alpha}}\|_2} \leq \sqrt{\gamma_n^2 (1 + \epsilon)}$$

The lower bound of the theorem holds by definition of γ_n. □

2.5.7 Effect of directional bias

In this section, we provide the proof for theorems in Section 2.3.2. There are two theorems there, so we split this section into two subsections. Section 2.5.7.1 proves Theorem 2.3, for a general problem setting of squared error minimization, it provides a straightforward understanding for why directional bias towards the largest eigenvalue of the Hessian is good for generalization. Section 2.5.7.2 proves Theorem 2.4 by giving concrete generalization bounds of SGD and GD estimators in kernel regression.

2.5.7.1 *Proof of Theorem 2.3*

Denote $\boldsymbol{v} = \boldsymbol{w} - \boldsymbol{w}^*$, rewrite the objective function as

$$\begin{aligned} \min_{\boldsymbol{v}} \quad & \|\boldsymbol{v}\|_2^2 \\ s.t. \quad & \|A\boldsymbol{v}\|_2^2 = a \end{aligned}$$

Denote the eigen decomposition of $A^TA = Q\Gamma Q^T$ where $Q = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_n]$, $QQ^T = Q^TQ = I$ and $\Gamma = diag([\rho_1, \ldots, \rho_n])$, $\rho_1 \geq \ldots \geq \rho_n \geq 0$. Then

$$\|A\boldsymbol{v}\|_2^2 = \sum_{i=1}^{n} \rho_i (\boldsymbol{q_i}^T \boldsymbol{v})^2$$

So

$$\|A\boldsymbol{v}\|_2^2 \leq \rho_1 [\sum_{i=1}^{n} (\boldsymbol{q_i}^T \boldsymbol{v})^2] = \rho_1 \boldsymbol{v}^T Q Q^T \boldsymbol{v} = \rho_1 \|\boldsymbol{v}\|_2^2$$

The equality is achieved when $\boldsymbol{v}$ is in the direction of $\boldsymbol{q}_1$, and $\rho_1 = \|A^TA\|_2$. Take $L(\boldsymbol{w}) = \|A\boldsymbol{v}\|_2^2 = a$ then the theorem holds.

2.5.7.2 *Proof of Theorem 2.4*

Calculate Δ_a^*: Denote $f^* = \hat{\boldsymbol{\alpha}}^T K(\cdot, X) + \tilde{f}$, then we have for a $f \in \mathcal{H}_s$, $f = \boldsymbol{\alpha}^T K(\cdot, X)$, let $\boldsymbol{b} = \hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}$, then

$$\begin{aligned} & \|f^* - f\|_{\mathcal{H}}^2 \\ =& \|\boldsymbol{b}^T K(\cdot, X) + \tilde{f}\|_{\mathcal{H}}^2 \\ =& \|\boldsymbol{b}^T K(\cdot, X)\|_{\mathcal{H}}^2 + \|\tilde{f}\|_{\mathcal{H}}^2 + 2\langle \boldsymbol{b}^T K(\cdot, X), \tilde{f} \rangle_{\mathcal{H}} \end{aligned}$$

where we can check

$$\begin{aligned}\langle \boldsymbol{b}^T K(\cdot, X), \tilde{f} \rangle_{\mathcal{H}} &= \sum_{i=1} b_i \langle K(\cdot, \boldsymbol{x}_i), f^* - \hat{\boldsymbol{\alpha}}^T K(\cdot, X) \rangle_{\mathcal{H}} \\ &= \sum_{i=1} b_i [\langle K(\cdot, \boldsymbol{x}_i), f^* \rangle_{\mathcal{H}} - \langle K(\cdot, \boldsymbol{x}_i), \hat{\boldsymbol{\alpha}}^T K(\cdot, X) \rangle_{\mathcal{H}}] \\ &= \sum_{i=1} b_i [f^*(\boldsymbol{x}_i) - \hat{\boldsymbol{\alpha}}^T K(\boldsymbol{x}_i, X)] (\text{By reproducing property}) \\ &= \sum_{i=1} b_i [y_i - y_i] = 0\end{aligned}$$

And we further calculate that

$$\begin{aligned}\|\boldsymbol{b}^T K(\cdot, X)\|_{\mathcal{H}}^2 &= \langle \sum_{i=1}^n b_i K(\cdot, \boldsymbol{x}_i), \sum_{j=1}^n b_j K(\cdot, \boldsymbol{x}_j) \rangle_{\mathcal{H}} \\ &= \sum_{i,j=1}^n b_i b_j \langle K(\cdot, \boldsymbol{x}_i), K(\cdot, \boldsymbol{x}_j) \rangle \\ &= \sum_{i,j=1}^n b_i b_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{b}^T K \boldsymbol{b}\end{aligned}$$

That is,

$$L_D(f) = \boldsymbol{b}^T K \boldsymbol{b} + \|\tilde{f}\|_{\mathcal{H}}^2$$

and

$$\inf_{f \in \mathcal{H}_s} L_D(f) = \|\tilde{f}\|_{\mathcal{H}}^2$$

It follows that

$$\Delta(f) = L_D(f) - \inf_{f \in \mathcal{H}_s} L_D(f) = \boldsymbol{b}^T K \boldsymbol{b}$$

We claim that

$$\Delta_a^* = \min_{\boldsymbol{b}:\frac{1}{2n}\|K\boldsymbol{b}\|_2^2=a} \boldsymbol{b}^T K\boldsymbol{b} = \frac{1}{\gamma_1}\|K\boldsymbol{b}\|_2^2 = 2na/\gamma_1$$

where the equality is obtained when $\boldsymbol{b}$ is in the direction of the largest eigenvector of K. To see this, we check $\|K\boldsymbol{b}\|_2^2 \leq \gamma_1 \boldsymbol{b}^T K\boldsymbol{b}$. Recall the eigendecomposition of $K = G\Gamma G^T$ where $G = [\boldsymbol{g}_1, \ldots, \boldsymbol{g}_n]$ has orthogonal columns and $\Gamma = diag(\gamma_1, \ldots, \gamma_n)$. Then

$$\|K\boldsymbol{b}\|_2^2 = \sum_{i=1}^n \gamma_i^2 (\boldsymbol{g_i}^T \boldsymbol{b})^2$$

and

$$\boldsymbol{b}^T K\boldsymbol{b} = \sum_{i=1}^n \gamma_i (\boldsymbol{g_i}^T \boldsymbol{b})^2$$

So we have

$$\|K\boldsymbol{b}\|_2^2 \leq \gamma_1 [\sum_{i=1}^n \gamma_i (\boldsymbol{g_i}^T \boldsymbol{b})^2] = \gamma_1 \boldsymbol{b}^T K\boldsymbol{b}$$

This finishes our claim on Δ_a^*.

SGD output: By Theorem 2.1, the SGD output has

$$(1-2\epsilon)\gamma_1 \mathbb{E}[\|\boldsymbol{b}_{k_2}\|_2] \leq \mathbb{E}[\|K\boldsymbol{b}_{k_2}\|_2]$$

Thus

$$\begin{aligned}
\mathbb{E}[\Delta^{1/2}(f^{SGD})] &= \mathbb{E}\left[\sqrt{\sum_{i=1}^n \gamma_i (\boldsymbol{g_i}^T \boldsymbol{b}^{SGD})^2}\right] \\
&\leq \sqrt{\gamma_1}\mathbb{E}[(\sum_{i=1}^n (\boldsymbol{g_i}^T \boldsymbol{b}^{SGD})^2)^{1/2}] \\
&= \sqrt{\gamma_1}\mathbb{E}\|\boldsymbol{b}^{SGD}\|_2 \\
&\leq \sqrt{\gamma_1}\mathbb{E}[\|K\boldsymbol{b}^{SGD}\|_2]/[(1-2\epsilon)\gamma_1]
\end{aligned}$$

$$
\begin{aligned}
&= \sqrt{2na/\gamma_1}/(1-2\epsilon) \\
&= \frac{1}{1-2\epsilon}(\Delta_a^*)^{1/2} \\
&< (1+4\epsilon)(\Delta_a^*)^{1/2}
\end{aligned}
$$

last inequality by let $\epsilon < 1/4$.

GD output: By Theorem 2.2, the GD output has

$$
\frac{\|K\boldsymbol{b}^{GD}\|_2^2}{\|\boldsymbol{b}^{GD}\|_2^2} \leq (1+\epsilon')\gamma_n^2
$$

Thus

$$
\begin{aligned}
\Delta(f^{GD}) &= \sum_{i=1}^{n} \gamma_i (\boldsymbol{g_i}^T \boldsymbol{b}^{GD})^2 \\
&\geq \gamma_n \sum_{i=1}^{n} (\boldsymbol{g_i}^T \boldsymbol{b}^{GD})^2 \\
&= \gamma_n \|\boldsymbol{b}^{GD}\|_2^2 \\
&\geq \gamma_n \|K\boldsymbol{b}^{GD}\|_2^2 / [(1+\epsilon')\gamma_n^2] \\
&= 2na/[(1+\epsilon')\gamma_n] \\
&= \frac{\gamma_1}{(1+\epsilon')\gamma_n}\Delta_a^* \\
&> \frac{\gamma_1}{\gamma_n}(1-\epsilon')\Delta_a^* \\
&:= M\Delta_a^*
\end{aligned}
$$

where $M > 1$ by taking $\epsilon' < 1 - \gamma_n/\gamma_1$.

2.5.8 Experiments

We list the implementation details of the experiments in Section 2.4 and include more experiment results. For better presenting, we split into two subsections: Section 2.5.8.1 includes the details of simulation; Section 2.5.8.2 is about the NN experiment on Fashion-MNIST, including the data description, network structure, and algorithm details. We defer experiment results to Section 2.5.8.2.

2.5.8.1 Simulation

This subsection is corresponding to Figure 2.1.

Data Generation. The training data is simulated as follows: Set $n = 10, p = 100$, simulate $X_{n\times p}$ where elements of X are i.i.d. $N(0,1)$; denote ith row of X as $\boldsymbol{x}_i$, normalize $\boldsymbol{x}_i$ such that it has squared ℓ_2 norm in $[.49, 1]$; set $y_i = \sum_{j=1}^{p} \sin(x_{i,j}) + \epsilon_i$ where $\epsilon_i \overset{i.i.d.}{\sim} N(0, .01)$. The testing data is simulated in exactly the same way, except that we only simulate $n = 5$ testing data.

Kernel Function. We set the kernel function to be the polynomial kernel

$$K(\boldsymbol{x}_1, \boldsymbol{x}_2) = (\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle + .01)^2$$

SGD and GD implementation. Both SGD and GD is run for small and moderate step sizes. The moderate step size scheme for SGD is: $\eta_1 = .1$ for the first 50 steps, and $\eta_2 = .01$ for the next 1000 steps; for GD is: $\eta_1 = .5$ for the first 50 steps, and $\eta_2 = .05$ for the next 1000 steps. The small step size scheme for SGD is $\eta = 0.01$ for 1050 steps; for GD is $\eta = 0.05$ for 1050 steps. Note that the step size for SGD is a fraction of that for GD, this matches our Theorem 2.1 and 2.2 that the step size of GD is of magnitude $n/2$ times that of SGD.

2.5.8.2 *Neural Network on FashionMNIST*

This subsection is corresponding to Figure 2.2.

Dataset. The original FashionMNIST consist of $60,000$ training data and $10,000$ testing data. We randomly sample $1,500$ data from original training data for training, and use all $10,000$ original testing data for testing. All data entries are normalized to $[0, 1]$.

Network structure. we use a 6-layer ResNet-like [27] Neural Network, and the structure is as follows

$$\begin{aligned}\text{Input } &\Rightarrow 7 \times 7 \text{ Conv } \Rightarrow \text{BatchNorm} \Rightarrow \text{ReLU} \Rightarrow 3 \times 3 \text{ MaxPool} \\ &\Rightarrow \text{ResBlock1} \Rightarrow \text{ResBlock2} \Rightarrow \text{Global AvePool} \Rightarrow \text{FC} \Rightarrow \text{output}\end{aligned}$$

The Residual Blocks are as Figure $7.6.3$ in [31] (without 1×1 convolution). Note that each residual block contains two 3×3 convolutional layers, thus total number of layers is as stated.

Algorithm. We minimize the Cross Entropy Loss objective $L(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} l_i(\mathbf{w})$, where $l_i(\mathbf{w})$ is the loss function at ith sample. One step SGD is as follows:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \frac{1}{|I|}\sum_{i \in I} \nabla l_i(\boldsymbol{w}_t)$$

where I is a randomly sampled subset of $\{1, \ldots, n\}$ (uniform random sample without replacement). We choose the batch size $|I|$ to be 25.

One step GD is as follows:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \nabla L(\boldsymbol{w}_t)$$

Both SGD and GD are run using two settings of step sizes η_t. The moderate step size

setting is as follows:

$$\eta_t = \begin{cases} 0.2, & t = 1, \ldots, 5000 \\ 0.02, & t = 5001, \ldots, 20000 \end{cases}$$

And the small step size setting has $\eta_t = 0.02, t = 1, \ldots, 20000$.

Comparison of convergence direction. Since the loss surface is nonconvex and the Hessian varies, we follow [7] to measure the convergence direction by Relative Rayleigh Quotient(RRQ), which normalizes the Rayleigh Quotient by the maximum eigenvalue of the Hessian as follows

$$RRQ(\boldsymbol{w}) = \frac{\frac{\nabla L(\boldsymbol{w})^\top}{\|\nabla L(\boldsymbol{w})\|_2} \cdot \nabla^2 L(\boldsymbol{w}) \cdot \frac{\nabla L(\boldsymbol{w})}{\|\nabla L(\boldsymbol{w})\|_2}}{\|\nabla^2 L(\boldsymbol{w})\|_2}$$

where $L(w)$ is the loss function on the whole training set. A high RRQ indicates that the convergence direction of w is close to a larger eigenvector of the Hessian.

Comparison of test accuracy. We set 20 different random seeds. For each random seed, we run: SGD with moderate step size, GD with moderate step size, SGD with small step size, GD with small step size. For each algorithm, we evaluate its test accuracy once every 500 steps, and use the average of the last 5 values as its test accuracy. We list the test accuracy in Table 2.1.

Table 2.1: Test Accuracy

Experiment	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
SGD + moderate LR	83.69	82.95	82.37	82.05	83.4	83.16	83.72	83.29	83.28	83.23
GD + moderate LR	80.93	80.79	80.79	81.80	81.68	81.12	82.43	81.63	80.94	81.54
SGD + small LR	82.00	81.72	81.34	81.92	82.63	82.67	82.99	82.22	80.78	82.10
GD + small LR	78.88	78.71	78.49	79.3	80.45	79.78	80.15	79.66	79.54	79.68
Experiment	#11	#12	#13	#14	#15	#16	#17	#18	#19	#20
SGD + moderate LR	83.12	82.92	83.58	83.47	82.35	83.57	83.59	82.43	84.21	83.12
GD + moderate LR	82.41	81.56	81.42	80.86	81.23	81.25	81.82	80.42	81.80	82.12
SGD + small LR	82.62	80.66	82.01	81.01	81.32	81.66	82.12	80.78	82.28	82.48
GD + small LR	80.08	78.29	79.93	79.36	78.9	79.69	80.2	79.62	79.98	79.69

We also use one-side Wilcoxon signed-rank test to check if the test accuracy of different algorithm are significantly different, the result is in Table 2.2. All the p-values are significant at 0.01 level, so we reject the null hypothesis and conclude that the SGD with moderate step size has test accuracy significantly higher than all other algorithms.

Table 2.2: Wilcoxon signed-rank test result

Null Hypothesis on Test Accuracy	p-value
SGD + moderate LR $\leq$ GD + moderate LR	9.54×10^{-7}
SGD + moderate LR $\leq$ SGD + small LR	9.54×10^{-7}
SGD + moderate LR $\leq$ GD + small LR	9.54×10^{-7}

Additional experiments. We conduct more experiments using different step sizes. The initial step size is taken in $\{1, 0.5, 0.2, 0.1, 0.02, 0.01, 0.005, 0.001\}$, and the step size is divided by a factor of 10 after 5000 steps. The test accuracy is in Figure 2.3, where we see that SGD with step size 0.2 has the best test accuracy, and GD with step size 0.5 performs better than GD with any other step sizes, but is still worse than the best SGD.

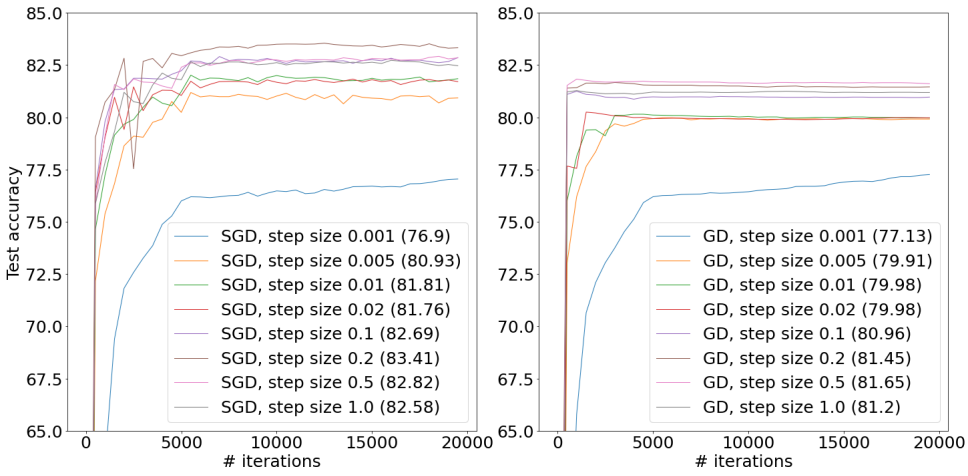


Figure 2.3: Use more step sizes in SGD/GD. The test accuracy is evaluated once every 500 iterations, and inside the bracket is the average of the last 5 test accuracy values.

CHAPTER 3

IMPLICIT REGULARIZATION PROPERTIES OF VARIANCE REDUCED STOCHASTIC MIRROR DESCENT

In machine learning and statistical data analysis, we often run into objective function that is a summation: the number of terms in the summation possibly is equal to the sample size, which can be enormous. In such a setting, the stochastic mirror descent (SMD) algorithm is a numerically efficient method—each iteration involving a very small subset of the data. The variance reduction version of SMD (VRSMD) can further improve SMD by inducing faster convergence. On the other hand, algorithms such as gradient descent and stochastic gradient descent have the implicit regularization property that leads to better performance in terms of the generalization errors. Little is known on whether such a property holds for VRSMD. We prove here that the discrete VRSMD estimator sequence converges to the minimum mirror interpolant in linear regression. This establishes the implicit regularization property for VRSMD. As an application of the above result, we derive a model estimation accuracy result in the setting when the true model is sparse. We use numerical examples to illustrate the empirical power of VRSMD.

3.1 Introduction

In statistics and machine learning, it is common to optimize an objective function that is a finite-sum. SMD efficiently optimizes such an objective by using a subset of data to do one step update of the variable/parameter. Further adopting the variance reduction technique to SMD, we get the VRSMD algorithm that enjoys fast convergence [32, 33].

The implicit regularization is a relatively new concept [2] that explains why a result of an algorithm generalizes well in some overparameterized models [2, 6]. It refers to the fact that an algorithm can automatically select a minimum norm solution, which is not ex-

plicitly induced by the objective function. There are works on implicit regularization for Gradient Descent [12, 13, 34, 35], Stochastic Gradient Descent [9, 36, 37, 10], and Stochastic Mirror Descent [38]. Considering the computational advantage of VRSMD compared to all the algorithms above, it would be even better if VRSMD also has the useful implicit regularization property.

From technical point of view, our work contains the following two results:

- In linear regression (including underfitting and overfitting), we show that the solution sequence of VRSMD converges to the minimum mirror interpolant, which is the implicit regularization property of VRSMD, and we also specify the convergence rate.

- In sparse regression, by choosing a proper mirror map, we show that the implicit regularization estimator finds the sparse true parameter with a small error. Moreover, compared with the deterministic algorithms in [12, 13], our algorithm is equally good in estimating a sparse truth while computationally faster, as supported by our experiments.

From the broad application point of view, our result shows that the Mirror Descent and its variants are useful to explore the low dimensional geometric structure from high dimensional data, which leads to nice generalization properties.

Notation. The following notations are used throughout this chapter. For a matrix $X \in \mathcal{R}^{n\times p}$, we denote by $\mathrm{col}(X) := \{\mathbf{u} \in \mathcal{R}^n : \exists \mathbf{v} \in \mathcal{R}^p, \mathbf{u} = X\mathbf{v}\}$ the column space of X, and we denote by $\mathcal{N}(X) := \{\mathbf{v} \in \mathcal{R}^p : X\mathbf{v} = \mathbf{0}\}$ the null space of X. For a vector $\mathbf{v} \in \mathcal{R}^p$, we use the definition of ℓ_p norm of $\mathbf{v}$ that $\|\mathbf{v}\|_p = (\sum_i |v_i|^p)^{1/p}$ for $p \geq 1$ and we denote the number of non-zero elements in $\mathbf{v}$ as $\|\mathbf{v}\|_0$. For a subset of indexes $I \subset \{1, \ldots, p\}$, we define $\mathbf{v}_I := (v_i)_{i\in I}$, and denote the cardinality of I as $|I|$. For a set $\mathcal{X} \subset \mathcal{R}^p$, define $P_{\mathcal{X}}\mathbf{v} = \mathrm{argmin}_{\mathbf{u}\in\mathcal{X}} \|\mathbf{u} - \mathbf{v}\|_2$. For two non-negative-valued functions $a(x)$ and $b(x)$, we denote $a(x) \sim \mathcal{O}(b(x))$ if there exists an absolute constant C such that

$a(x) \leq Cb(x)$; and we denote $a(x) \sim \Theta(b(x))$ if there are absolute constants c, C such that $cb(x) \leq a \leq Cb(x)$.

Organization. The rest of the chapter is as follows: In Section 3.2, we describe our problem formulation and algorithm; Section 3.3 states the main theory on the implicit regularization; Section 3.4 develops insight into the implicit regularization and establishes the sparse recovery property; Section 3.5 supports the theory on implicit regularization by simulations and experiments; The technique comparisons between literature, proofs, experiment details, and more experiments are included in Section 3.6.

3.2 Formulation and Algorithm

To better present the material, we split this section into two subsections. In Subsection 3.2.1 we formulate the optimization problem motivated from linear regression. In Subsection 3.2.2, we present the Variance Reduction Stochastic Mirror Descent (VRSMD) algorithm for solving such an optimization problem.

3.2.1 Formulation

Assume we observe data pairs $\{(\mathbf{x}_i, y_i) \in \mathcal{R}^p \times \mathcal{R}\}_{i=1}^n$, the goal is to predict the response y based on $\mathbf{x}$. Under the empirical risk minimization framework, we consider the general optimization problem of the form

$$\min_{\boldsymbol{\beta}} F(\boldsymbol{\beta}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\beta}; (\mathbf{x}_i, y_i)), \tag{3.1}$$

and we shorten $f_i(\boldsymbol{\beta}; (\mathbf{x}_i, y_i))$ as $f_i(\boldsymbol{\beta})$ to simplify the notation.

As a concrete example, for the linear regression model, the classical least squares method is to find coefficient $\boldsymbol{\beta}$ that minimizes the objective function

$$\min_{\boldsymbol{\beta}} F(\boldsymbol{\beta}) = \frac{1}{2n} \sum_{i=1}^{n} \left(\mathbf{x}_i^T \boldsymbol{\beta} - y_i\right)^2 = \frac{1}{2n} \|X\boldsymbol{\beta} - y\|_2^2, \tag{3.2}$$

where we denote $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^T \in \mathcal{R}^{n\times p}$, and $\mathbf{y} = [y_1, \ldots, y_n]^T \in \mathcal{R}^n$.

When problem (3.2) has non-unique solutions, it is important yet nontrivial to find a solution that has nice generalization property. It is well known that when running Gradient Descent algorithm with initialization $\boldsymbol{\beta}_0 = \mathbf{0}$ on (3.2), the corresponding solution is the minimal ℓ_2 norm solution among all solutions of (3.2), see [7]. Also, the properties of SMD are studied in [38]. It is unknown what happens to the solution if we run variants of SMD, for example, variance reduced SMD.

3.2.2 VRSMD Algorithm

Let us now present the main idea of the variance reduced stochastic mirror descent (VRSMD) algorithm as follows.

To understand why we need variance reduction, consider the Stochastic Mirror Descent(SMD) algorithm using a strictly convex and differentiable mirror map $\psi(\cdot)$. At step t, the SMD updates $\boldsymbol{\beta}_{t+1}$ such that

$$\nabla\psi(\boldsymbol{\beta}_{t+1}) = \nabla\psi(\boldsymbol{\beta}_t) - \eta_t \nabla f_{i_t}(\boldsymbol{\beta}_t),$$

where i_t is randomly sampled from $\{1, \ldots, n\}$. Now the term $\nabla f_{i_t}(\boldsymbol{\beta}_t)$ has $\mathbb{E}[\nabla f_{i_t}(\boldsymbol{\beta}_t)] = \nabla F(\boldsymbol{\beta}_t)$, so SMD has unbiased update compared to Mirror Descent, where the update is $\nabla\psi(\boldsymbol{\beta}_{t+1}) = \nabla\psi(\boldsymbol{\beta}_t) - \eta_t \nabla F(\boldsymbol{\beta}_t)$. However, in general $\mathrm{Var}[\nabla f_{i_t}(\boldsymbol{\beta}_t)] \neq 0$ for any $\boldsymbol{\beta}_t$, so we need $\eta_t \to 0$, which may lead to slow convergence.

Variance reduction addresses the issues above by replacing $\nabla f_{i_t}(\boldsymbol{\beta}_t)$ with term A_t such that

- $\mathbb{E}[A_t] = \mathbb{E}[\nabla f_{i_t}(\boldsymbol{\beta}_t)]$ to keep unbiased update;
- $\mathrm{Var}[A_t] < \mathrm{Var}[\nabla f_{i_t}(\boldsymbol{\beta}_t)]$ to control variance.

One choice of A_t is $A_t = \nabla f_{i_t}(\boldsymbol{\beta}_t) - B_t + \mathbb{E}[B_t]$, where B_t and $\nabla f_{i_t}(\boldsymbol{\beta}_t)$ are positively correlated with correlation coefficient $r > 0.5$ and $\mathrm{Var}[B_t] \approx \mathrm{Var}[\nabla f_{i_t}(\boldsymbol{\beta}_t)]$. For

this A_t one can check that $\mathbb{E}[A_t] = \mathbb{E}[\nabla f_{i_t}(\boldsymbol{\beta}_t)]$ and $\mathrm{Var}[A_t] = \mathrm{Var}[\nabla f_{i_t}(\boldsymbol{\beta}_t) - B_t] = \mathrm{Var}[\nabla f_{i_t}(\boldsymbol{\beta}_t)] - 2r\sqrt{\mathrm{Var}[\nabla f_{i_t}(\boldsymbol{\beta}_t)]\,\mathrm{Var}[B_t]} + \mathrm{Var}[B_t] < \mathrm{Var}[\nabla f_{i_t}(\boldsymbol{\beta}_t)]$. For a proper B_t such that $\mathrm{Var}(A_t) \overset{t\to\infty}{\longrightarrow} 0$, the algorithm converges for a fixed η.

For illustration purpose, we use the variance reduction technique in [39] to get the VRSMD Algorithm 3.1. However, one should note that this framework applies to other variance reduction methods such as SARAH [40] and SPIDER [41].

Algorithm 3.1: Variance Reduced Stochastic Mirror Descent (VRSMD)

Input: An objective function $F(\cdot) = \frac{1}{n}\sum_{i=1}^{n} f_i(\cdot)$, and a strictly convex and differentiable mirror map $\psi(\cdot)$;

Initialization: Initialize $\tilde{\boldsymbol{\beta}}^0$. Choose the step size η, outer iteration number S, inner iteration number m. Denote the estimator at tth inner iteration of sth outer iteration as $\boldsymbol{\beta}_t^s$. Set $\boldsymbol{\beta}_1^1 = \boldsymbol{\beta}_{m+1}^0 = \tilde{\boldsymbol{\beta}}^0$;

for *Outer iteration s = 1,. . . ,S* **do**

Calculate $\nabla F(\tilde{\boldsymbol{\beta}}^{s-1})$;

for *Inner iteration t = 1,. . . ,m* **do**

Randomly sample i_t from $\{1, \ldots, n\}$, calculate

$$\mathbf{v}_t^s = \nabla f_{i_t}(\boldsymbol{\beta}_t^s) - \nabla f_{i_t}(\tilde{\boldsymbol{\beta}}^{s-1}) + \nabla F(\tilde{\boldsymbol{\beta}}^{s-1}), \tag{3.3}$$

and update $\boldsymbol{\beta}_{t+1}^s$ such that

$$\nabla\psi(\boldsymbol{\beta}_{t+1}^s) = \nabla\psi(\boldsymbol{\beta}_t^s) - \eta\mathbf{v}_t^s. \tag{3.4}$$

Set $\tilde{\boldsymbol{\beta}}^s$ to be a uniform random sample from $\{\boldsymbol{\beta}_1^s, \ldots, \boldsymbol{\beta}_m^s\}$;

Option I: Set $\boldsymbol{\beta}_1^{s+1} = \boldsymbol{\beta}_{m+1}^s$;

Option II: Set $\boldsymbol{\beta}_1^{s+1} = \tilde{\boldsymbol{\beta}}^s$;

Option I: Output $\boldsymbol{\beta}_a$ chosen uniformly random from $\{\{\boldsymbol{\beta}_t^s\}_{t=1}^m\}_{s=1}^S$;

Option II: Output $\boldsymbol{\beta}_a = \tilde{\boldsymbol{\beta}}^S$.

Remark 3.1. *We note the complexity of VRSMD as follows: The total number of stochastic-first-order calls (i.e. SFO complexity) of Algorithm 3.1 is $\mathcal{O}(nS + mS)$. A popular choice of the inner loop number m is $\Theta(n)$, which leads to $\mathcal{O}(1)$ SFO complexity per inner loop.*

The variance reduction component of VRSMD is $\mathbf{v}_t^s$ in (3.3). Note that it is equivalent to the variance reduction scheme we talked above by taking $B_t = \nabla f_{i_t}(\tilde{\boldsymbol{\beta}}^{s-1})$. The conditions

we list there on B_t hold when $\boldsymbol{\beta}_t^s$ and $\tilde{\boldsymbol{\beta}}^{s-1}$ are close, which happens by taking moderate values of the inner iteration number m and the step size η.

We show that the VRSMD algorithm is a generalization of the SVRG algorithm in [39, 42]: For the special case of $\psi(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, we have $\nabla\psi(\boldsymbol{\beta}) = \boldsymbol{\beta}$, then (3.3) updates $\boldsymbol{\beta}_{t+1}^s$ as:

$$\boldsymbol{\beta}_{t+1}^s = \boldsymbol{\beta}_t^s - \eta \mathbf{v}_t^s,$$

which is the SVRG update, and VRSMD reduces to SVRG.

3.3 Implicit Regularization

In this section, we present the implicit regularization property of the VRSMD solution. To do so, it is necessary to first show that the VRSMD converges.

To begin with, we introduce some definitions that will be useful in our theory.

Definition 3.1 (L-smoothness)**.** *f is L-smooth with respect to $\|\cdot\|$ norm if there exists a constant $L > 0$ such that*

$$\|\nabla f(\mathbf{u}) - \nabla f(\mathbf{w})\|_* \leq L\|\mathbf{u} - \mathbf{w}\|, \forall \mathbf{u}, \mathbf{w},$$

where $\|\cdot\|_* := \max_{y:\|y\|=1}\langle y, \cdot\rangle$ is the dual norm of $\|\cdot\|$.

Definition 3.2 (α-strongly convex)**.** *f is α-strongly convex with respect to $\|\cdot\|$ norm if there exists a constant $\alpha > 0$ such that*

$$f(\mathbf{u}) \geq f(\mathbf{w}) + \nabla f(\mathbf{w})^T(\mathbf{u} - \mathbf{w}) + \frac{\alpha}{2}\|\mathbf{u} - \mathbf{w}\|^2, \forall \mathbf{u}, \mathbf{w}.$$

Definition 3.3 (Quadratic Growth (QG))**.** *Let $\mathcal{X}$ be the set of all minimizers of f. f satisfies QG condition w.r.t. $\|\cdot\|$ if*

$$\frac{\mu}{2}\|\mathbf{u} - P_{\mathcal{X}}\mathbf{u}\|^2 \leq f(\mathbf{u}) - f(P_{\mathcal{X}}\mathbf{u}), \forall \mathbf{u}. \tag{3.5}$$

Definition 3.4 (ϵ-solution)**.** *For the optimization problem*

$$Opt = \min_{x \in B} \{f(x) : g_i(x) \leq 0, 1 \leq i \leq m\}. \tag{3.6}$$

$x_\epsilon \in B$ *is called an* ϵ*-solution to* (3.6) *if*

$$f(x_\epsilon) - Opt \leq \epsilon,$$
$$g_i(x_\epsilon) \leq \epsilon, 1 \leq i \leq m.$$

Definition 3.5 (Restricted Eigenvalue (RE))**.** X *satisfies* (s, γ)*-RE condition if for any* $\boldsymbol{\beta}$ *such that* $\|\boldsymbol{\beta}\|_0 \leq s$ *we have*

$$\frac{\frac{1}{n}\|X\boldsymbol{\beta}\|_2^2}{\|\boldsymbol{\beta}\|_2^2} \geq \gamma.$$

Definition 3.6 (s-good)**.** *A matrix* $X_{n \times p}$ *is* s*-good if* $\exists \kappa < \frac{1}{2}$ *such that* $\forall \mathbf{u} \in \mathcal{N}(X) \subset \mathcal{R}^p$ *and* $\forall I \subset \{1, \ldots, p\}$ *with* $|I| \leq s$*, we have*

$$\|\mathbf{u}_I\|_1 \leq \kappa \|\mathbf{u}\|_1.$$

Next, let us present the convergence result of VRSMD:

Proposition 3.1. *Assume* $F(\cdot) = \frac{1}{n}\sum_{i=1}^n f_i(\cdot)$ *has every* $f_i(\cdot)$ *convex and* L*-smooth w.r.t. an arbitrary norm* $\|\cdot\|$*, and* $\psi(\cdot)$ *is* α*-strongly convex w.r.t.* $\|\cdot\|$*. Denote* $\boldsymbol{\beta}^* = \arg\min F(\cdot)$.

(a) *Run Option I of Algorithm 3.1 on* F *with* $\eta < \frac{\alpha}{24L}$*, then we have*

$$\begin{aligned} \mathbb{E}[F(\boldsymbol{\beta}_a) - F(\boldsymbol{\beta}^*)] \leq \frac{\alpha}{(\alpha\eta - 24L\eta^2)T} \times \\ \left[D_\psi(\boldsymbol{\beta}^*, \tilde{\boldsymbol{\beta}}^0) + \frac{12L\eta^2 m}{\alpha}(F(\tilde{\boldsymbol{\beta}}^0) - F(\boldsymbol{\beta}^*)) \right], \end{aligned} \tag{3.7}$$

where $T = m \cdot S$*, and* $D_\psi(\boldsymbol{\beta}^*, \tilde{\boldsymbol{\beta}}^0) := \psi(\boldsymbol{\beta}^*) - \psi(\tilde{\boldsymbol{\beta}}^0) - \langle \nabla\psi(\tilde{\boldsymbol{\beta}}^0), \boldsymbol{\beta}^* - \tilde{\boldsymbol{\beta}}^0 \rangle$ *is the Bregman divergence.*

(b) *If we further assume that* $F(\cdot)$ *satisfies the QG condition in* (3.5) *with constant* μ*,*

and that $\psi(\cdot)$ is ℓ-smooth, all w.r.t. $\|\cdot\|$, and also suppose that we run Option II of Algorithm 3.1 with a large enough m such that

$$\tau := \frac{12L\eta^2/\alpha + \ell/(m\mu)}{\eta - 12L\eta^2/\alpha} < 1, \tag{3.8}$$

then the VRSMD has a stronger linear convergence rate:

$$\mathbb{E}[F(\boldsymbol{\beta}_a) - F(\boldsymbol{\beta}^*)] \leq \tau^S [F(\tilde{\boldsymbol{\beta}}^0) - F(\boldsymbol{\beta}^*)]. \tag{3.9}$$

Remark 3.2. *We analyze the computational complexity implied by Proposition 3.1: In (a), let $m = n$ and take $\eta = \frac{\alpha}{48L}$, then we have*

$$\mathbb{E}[F(\boldsymbol{\beta}_a) - F(\boldsymbol{\beta}^*)] \leq \frac{96L}{\alpha T} \times \left[D_\psi(\boldsymbol{\beta}^*, \tilde{\boldsymbol{\beta}}^0) + \frac{\alpha n}{192L} (F(\tilde{\boldsymbol{\beta}}^0) - F(\boldsymbol{\beta}^*)) \right].$$

In this case, the number of gradient computations for achieving an ϵ-solution is $\mathcal{O}(\frac{L}{\epsilon} + \frac{n}{\epsilon})$.

Remark 3.3. *The assumption in (b) is moderate. Take $m = \frac{110L\ell}{\alpha\mu}$ and $\eta = \frac{\alpha}{36L}$, we have $\tau < 1$. This choice of m does not violate the $\mathcal{O}(1)$ SFO comlexity per iteration – take a good mirror map so that $\ell/\alpha = \mathcal{O}(1)$, and consider the most indicative case [39] where the condition number $L/\mu = n$, we have $m = \Theta(n)$.*

Our results in Proposition 3.1 are consistent with those for SVRG. In part (a), the $\mathcal{O}(1/T)$ convergence rate matches the rate in [42]; in part (b), the linear rate matches the rate in [39], while we reduce their strong convexity assumption to quadratic growth.

Finally, we are ready to present our main result on implicit regularization. In the following theorem, we show that VRSMD finds an ϵ-solution of the minimum mirror interpolation problem.

Theorem 3.1. *For the objective function in* (3.2)*, assume $\psi(\boldsymbol{\beta})$ is α-strongly convex w.r.t.*

$\|\cdot\|_2$, denote $L = \max_i \|\mathbf{x}_i\|_2^2$ and let s_m be the smallest nonzero singular value of X. The VRSMD algorithm converges to the minimum mirror map interpolant

$$\begin{aligned} \boldsymbol{\beta}^{\psi} := \operatorname{argmin}_{\boldsymbol{\beta}} \quad & \psi(\boldsymbol{\beta}) \\ s.t. \quad & F(\boldsymbol{\beta}) = \min_{\boldsymbol{\beta}'} F(\boldsymbol{\beta}'). \end{aligned} \tag{3.10}$$

We describe the convergence by the following ϵ-solution:

***(a)** Run Option I of Algorithm 3.1 with choice $\eta < \frac{\alpha}{24L}$ and initialization $\tilde{\boldsymbol{\beta}}^0$ such that $\nabla\psi(\tilde{\boldsymbol{\beta}}^0) \in \operatorname{col}(X^T)$, assume that the output $\boldsymbol{\beta}^a$ satisfies $\|\nabla\psi(\boldsymbol{\beta}^a)\|_2 \leq B$, then we will have*

$$\begin{aligned} \mathbb{E}[\psi(\boldsymbol{\beta}^a) - \psi(\boldsymbol{\beta}^{\psi})] \leq \frac{B}{s_m}\sqrt{\frac{\alpha}{(\alpha\eta - 24L\eta^2)T}}\times \\ \left[2nD_{\psi}(\boldsymbol{\beta}^{\psi}, \tilde{\boldsymbol{\beta}}^0) + \frac{24nL\eta^2 m}{\alpha}\left(F(\tilde{\boldsymbol{\beta}}^0) - F(\boldsymbol{\beta}^{\psi})\right)\right]^{.5}, \end{aligned} \tag{3.11}$$

which describes how far is the objective value at $\boldsymbol{\beta}^a$ away from the optimal solution to (3.10)*. Moreover, we have*

$$\begin{aligned} \mathbb{E}[F(\boldsymbol{\beta}^a) - F(\boldsymbol{\beta}^{\psi})] \leq \frac{\alpha}{(\alpha\eta - 8L\eta^2)T}\times \\ \left[D_{\psi}(\boldsymbol{\beta}^{\psi}, \tilde{\boldsymbol{\beta}}^0) + \frac{12L\eta^2 m}{\alpha}\left(F(\tilde{\boldsymbol{\beta}}^0) - F(\boldsymbol{\beta}^{\psi})\right)\right], \end{aligned} \tag{3.12}$$

which characterizes how much does $\boldsymbol{\beta}^a$ violates the constraints of (3.10)*. They together show that the VRSMD algorithm finds an ϵ-solution to* (3.10) *for $T = \mathcal{O}(\frac{1}{\epsilon} + \frac{1}{\epsilon^2})$.*

***(b)** Further assume that $\psi(\cdot)$ is ℓ-smooth w.r.t. $\|\cdot\|_2$ and*

$$\tau' = \frac{12L\eta^2/\alpha + \ell n/(ms_m^2)}{\eta - 12L\eta^2/\alpha} < 1. \tag{3.13}$$

Run Option II of Algorithm 3.1, we can show that:

$$\mathbb{E}[\psi(\boldsymbol{\beta}^a) - \psi(\boldsymbol{\beta}^{\psi})] \leq \frac{B(\tau')^{S/2}\sqrt{2n}}{s_m}\sqrt{F(\tilde{\boldsymbol{\beta}}^0) - F(\boldsymbol{\beta}^{\psi})},$$

$$\mathbb{E}[F(\boldsymbol{\beta}^a) - F(\boldsymbol{\beta}^\psi)] \leq (\tau')^S \left(F(\tilde{\boldsymbol{\beta}}^0) - F(\boldsymbol{\beta}^\psi)\right). \tag{3.14}$$

Remark 3.4. *We need to point out that the assumptions in Theorem 3.1 are moderate. For instance, for the assumption that* $\nabla\psi(\tilde{\boldsymbol{\beta}}^0) \in \mathrm{col}(X^T)$*, we can take* $\tilde{\boldsymbol{\beta}}^0 = (\nabla\psi)^{-1}(X^T\mathbf{a})$ *for any* $\mathbf{a}$*. One feasible choice is* $\mathbf{a} = \mathbf{0}$*, resulting in* $\tilde{\boldsymbol{\beta}}_0 = \arg\min_{\boldsymbol{\beta}\in R^p} \psi(\boldsymbol{\beta})$*. Since* ψ *is strongly convex, this minimizer is not hard to calculate, for example: we have*

$\cdot\psi(\cdot) = \|\cdot\|_q^q$ *or* $\psi(\cdot) = \|\cdot\|_q^2$ *for* $q > 1 \Rightarrow \arg\min \psi(\cdot) = \mathbf{0}$*;*

$\cdot\psi(\boldsymbol{\beta}) = \boldsymbol{\beta}^T H\boldsymbol{\beta}$ *for a positive definite* $H \Rightarrow \arg\min \psi(\cdot) = \mathbf{0}$*;*

$\cdot\psi(\boldsymbol{\beta}) = \sum_{i=1}^p \beta_i \log(\beta_i) - \beta_i \Rightarrow \arg\min \psi(\cdot) = \mathbf{1}$.

Remark 3.5. *As for the assumption in (b), we can take* $m = \frac{110L\ell n}{\alpha s_m^2}$ *and* $\eta = \frac{\alpha}{36L}$ *to get* $\tau' = (1 + 108/110)/2 < 1$*. Take a good mirror map such that* $\ell/\alpha = \mathcal{O}(1)$ *and assume* $L/s_m^2 = \mathcal{O}(1)$*, we further have* $m = \Theta(n)$*, so the algorithm can be implemented efficiently.*

It is useful to provide a high level understanding of Theorem 3.1. It implies that the discrete update of the VRSMD Algorithm on the unregularized objective (3.2) is an ϵ-solution of the regularized optimization problem (3.10), so it is an implicit regularization result. Furthermore, since (3.10) minimizes a strictly convex function over a convex set, the solution will be unique, thus the estimator from VRSMD must converge to this unique solution. The finding in Theorem 3.1 can be extended to other variants of SMD, which we omit here.

3.4 Further Understanding of Implicit Regularization

In this section, we provide a deeper understanding of our theoretical results in the previous section by analyzing two special mirror maps for linear regression model: one is $\psi(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_2^2/2$, the other is $\psi(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_{1+\delta}^{1+\delta}$ for small $\delta > 0$.

Let us first consider $\psi(\cdot) = \|\cdot\|_2^2/2$, where VRSMD reduces to SVRG algorithm in [39]. In this case, we further show that $\|\boldsymbol{\beta}^a - \boldsymbol{\beta}^\psi\|_2^2$ linearly converges to 0, where $\boldsymbol{\beta}^\psi$ is the minimum ℓ_2 norm solution $\boldsymbol{\beta}^\psi = (X^TX)^+X^T\mathbf{y} = X^+\mathbf{y}$:

Corollary 3.1. *Denote* $L = \max_i \|\mathbf{x}_i\|_2^2$*, take* $\psi(\cdot) = \|\cdot\|_2^2/2$*, let* $\tilde{\boldsymbol{\beta}}^0 = \boldsymbol{\beta}_m^0 = \mathbf{0}$*,* $\eta_t = \eta < 1/(24L)$*, and assume*

$$\tau'' = \frac{12L\eta^2 + n/(ms_m^2)}{\eta - 12L\eta^2} < 1. \tag{3.15}$$

Run Option II of Algorithm 3.1 on (3.2)*, we have*

$$\mathbb{E}\|\boldsymbol{\beta}^a - X^+\mathbf{y}\|_2^2 \leq \frac{(\tau'')^S}{s_m^2}\|P_{\mathrm{col}(X)}\mathbf{y}\|_2^2. \tag{3.16}$$

Next, let us consider the mirror map $\psi(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_{1+\delta}^{1+\delta}$ for a small $\delta > 0$ in VRSMD, which leads to a sparse solution. Assume that $\|\boldsymbol{\beta}_t^s\|_\infty \leq K$ for a large enough K throughout the updates, we then have $\psi(\boldsymbol{\beta}_t^s)$ is $\frac{(1+\delta)\delta}{K^{1-\delta}}$-strongly convex and $\|\nabla\psi(\boldsymbol{\beta}^a)\|_2 \leq \sqrt{p}(1+\delta)K^\delta$. By Theorem 3.1 we have the estimator sequence converges to the penalized solution

$$\boldsymbol{\beta}^{(\delta)} = \mathrm{argmin}_{\boldsymbol{\beta}} \left\{ \|\boldsymbol{\beta}\|_{1+\delta}^{1+\delta} : X\boldsymbol{\beta} = P_{\mathrm{col}(X)}\mathbf{y} \right\}. \tag{3.17}$$

This choice of mirror map has $\lim_{\delta\to 0}\|\boldsymbol{\beta}\|_{1+\delta}^{1+\delta} = \|\boldsymbol{\beta}\|_1$ and the solution $\boldsymbol{\beta}^{(0)} := \mathrm{argmin}_{\boldsymbol{\beta}}\{\|\boldsymbol{\beta}\|_1 : X\boldsymbol{\beta} = P_{\mathrm{col}(X)}\mathbf{y}\}$ is sparse. Thus we expect that for small δ, $\boldsymbol{\beta}^{(\delta)}$ is close to $\boldsymbol{\beta}^{(0)}$ and recovers a sparse true parameter.

We now provide a rigorous argument for the sparse recovery. Assume that the data is generated by $\mathbf{y} = X\boldsymbol{\beta}^o$ for a sparse $\boldsymbol{\beta}^o$. In the following theorem we have $\boldsymbol{\beta}^{(\delta)}$ accurately recovers $\boldsymbol{\beta}^o$ when the design matrix X satisfies some proper conditions.

Theorem 3.2 (Sparse Recovery)**.** *Under the sparse setting defined above, denote* $s = \|\boldsymbol{\beta}^o\|_0$*. Assume that the design matrix* X *satisfies* (s,γ)*-RE condition and is* s*-good with constant* $\kappa < \frac{1}{2}$*. For any* $\xi > 0$*, if we choose*

$$\delta \leq \frac{\log\left(1 + \frac{(1-2\kappa)\sqrt{n\gamma}}{\sqrt{s}\|\mathbf{y}\|_2}\xi\right)}{\log p - \log\left(1 + \frac{(1-2\kappa)\sqrt{n\gamma}}{\sqrt{s}\|\mathbf{y}\|_2}\xi\right)}, \tag{3.18}$$

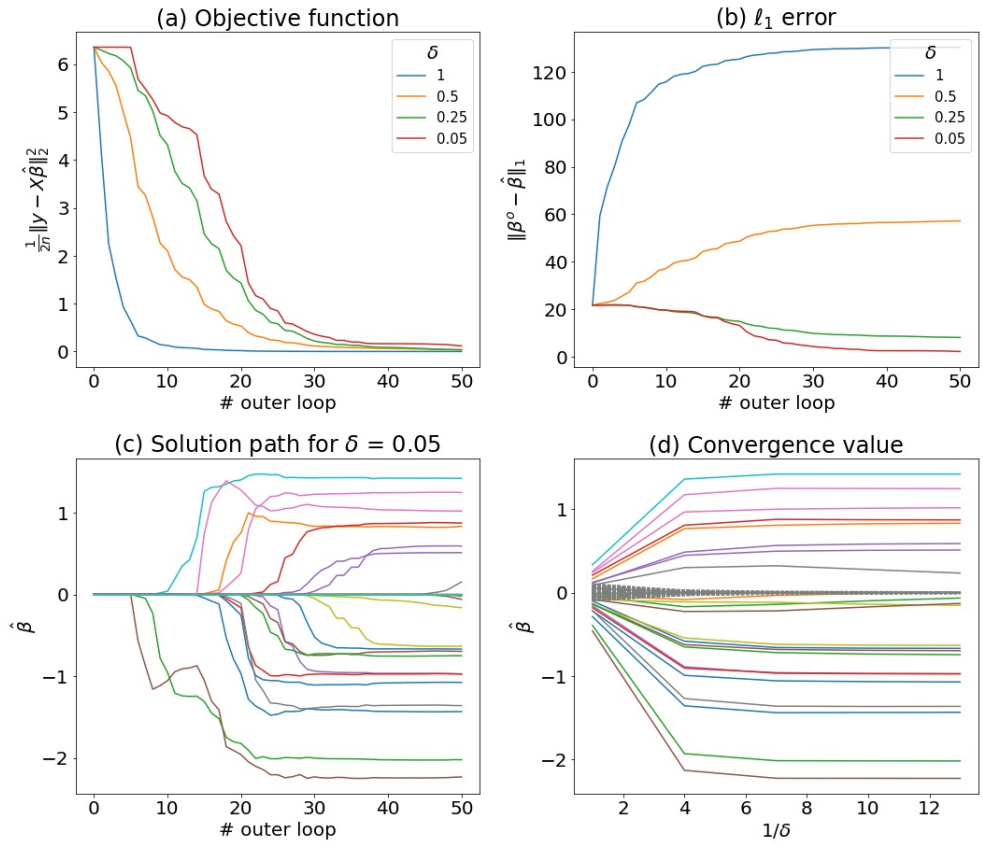


Figure 3.1: Run VRSMD Algorithm on simulated noiseless data. (a) the squared error objective function converges quickly to 0. In (b), the ℓ_1 estimation error converges to a smaller value for smaller δ, indicating the nearly exact recovery of the sparse signal, which supports Theorem 3.2. In (c), for $\psi(\cdot) = \|\cdot\|_{1.05}^{1.05}$, the VRSMD estimator converges to the true parameter values. In (d), for a smaller δ, the convergence value of VRSMD estimator is closer to ground truth.

we have

$$\|\boldsymbol{\beta}^{(\delta)} - \boldsymbol{\beta}^o\|_1 \leq \xi. \tag{3.19}$$

By Theorem 3.2, the estimator $\boldsymbol{\beta}^{(\delta)}$ estimates the sparse truth with a small error ξ. In this way, VRSMD algorithm 3.1 achieves near sparse recovery via implicit regularization.

3.5 Numerical Experiment

Simulation. We generate data by a sparse model as follows: Set $n = 1000, p = 5000 > n$ for the design matrix X. Simulate $X = \Sigma^{1/2}W$ where the entries of W are i.i.d. $N(0, 1)$

and $\Sigma = 0.5 * \mathrm{diag}(\mathbf{1}_n) + 0.5 * \mathbf{1}_{n\times n}$. The true parameter $\boldsymbol{\beta}^o \in \mathcal{R}^p$ has its first 30 entries sampled from i.i.d. $N(0,1)$ and the rest entries set to 0. Compute responses $\mathbf{y} = X\boldsymbol{\beta}^o$.

We then run VRSMD on objective function (3.2) for this simulated X and $\mathbf{y}$. For a range of δ, set the mirror map as $\psi(\cdot) = \|\cdot\|_{1+\delta}^{1+\delta}$, and run VRSMD with initialization $\tilde{\boldsymbol{\beta}}^0 = \mathbf{0}$, step size $\eta = 0.0002$, outer iteration number 50 and inner iteration number $1000 = n$. The result is in Fig. 3.1.

Experiment on RNA dataset. We use the gene expression cancer RNA-Seq data set[1] for experiment. The data consists of 801 observations, each of dimension $20,531$. Randomly split the data into 600 training data and 201 testing data. Run VRSMD algorithm on training data using mirror function $\psi = \|\cdot\|_{1.1}^{1.1}$ where initialization $\tilde{\boldsymbol{\beta}}^0 = \mathbf{0}$, step size $\eta = 0.015$, inner iteration number 400 and outer iteration number determined by 5-fold cross validation (i.e. early stopping). We also compare VRSMD with the Hadamard GD [13, 12], which also has implicit regularization for sparsity. The result is in Fig. 3.2.

Proof sketch for Theorem 3.1

It is easy to check that $\mathbf{v}_t^s \in \mathrm{col}\left(X^T\right)$. Then by $\nabla\psi(\tilde{\boldsymbol{\beta}}^0) \in \mathrm{col}(X^T)$, we immediately have $\nabla\psi\left(\boldsymbol{\beta}^a\right) \in \mathrm{col}\left(X^T\right)$. Combine this key observation with the fact that

$$\psi(\boldsymbol{\beta}^a) - \psi(\boldsymbol{\beta}^\psi) \le \langle \nabla\psi(\boldsymbol{\beta}^a), \boldsymbol{\beta}^a - \boldsymbol{\beta}^\psi\rangle,$$

we have

$$\begin{aligned}
&\mathbb{E}[\psi(\boldsymbol{\beta}^a) - \psi(\boldsymbol{\beta}^\psi)] \le \mathbb{E}\langle \nabla\psi(\boldsymbol{\beta}^a), \boldsymbol{\beta}^a - \boldsymbol{\beta}^\psi\rangle \\
\le& B\mathbb{E}\|P_{\mathrm{col}(X^T)}(\boldsymbol{\beta}^a - \boldsymbol{\beta}^\psi)\|_2 \le \frac{B}{s_m}\sqrt{\mathbb{E}\|X\boldsymbol{\beta}^a - X\boldsymbol{\beta}^\psi\|_2^2} \\
=& \frac{B}{s_m}\sqrt{2n\mathbb{E}(F(\boldsymbol{\beta}^a) - F(\boldsymbol{\beta}^\psi))}. \qquad (3.20)
\end{aligned}$$

[1] https://archive.ics.uci.edu/ml/datasets/gene+expression+cancer+RNA-Seq

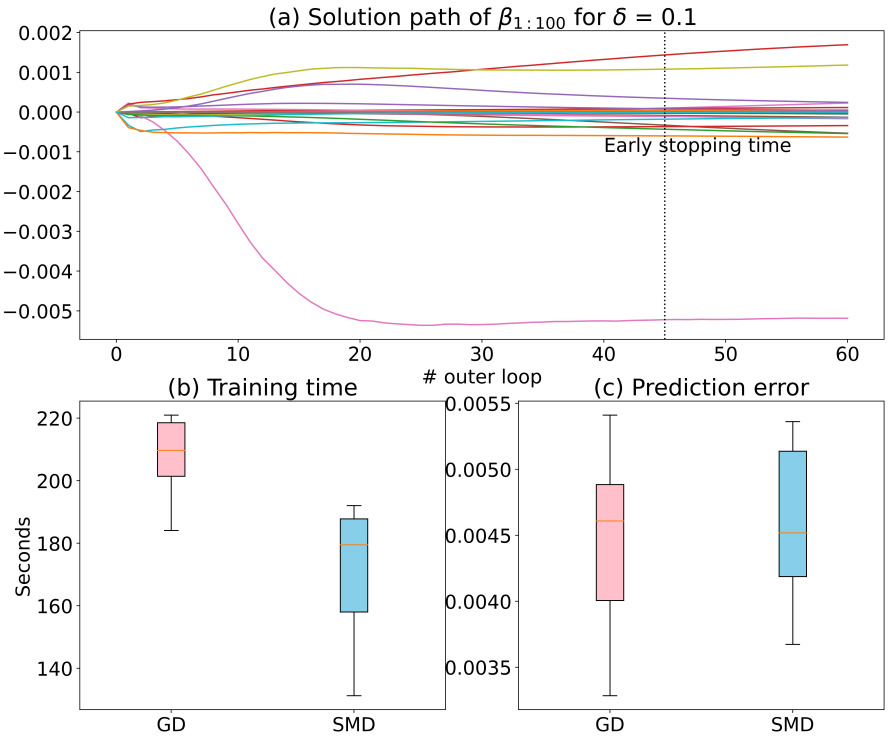


Figure 3.2: Run VRSMD Algorithm on RNA dataset. In (a), we plot the solution path of the first 100 entries of $\boldsymbol{\beta}$, and it shows that the early stopped estimator is sparse. In (b) and (c), we compare the performance of VRSMD with Hadamard GD. Plot (b) shows that VRSMD trains faster than Hadamard GD, which is tested significant by one-sided Wilcoxon signed-rank test (p-value = 9.77×10^{-4}). Plot (c) shows that the two algorithms have same prediction error on testing data, for which we test by two-sided Wilcoxon signed-rank test (p-value = 0.56).

Apply Proposition 3.1 to (3.20) we have the desired inequalities.

3.6 Technique Details for Chapter 3

In this section, we provide the technique details in Chapter 3.

3.6.1 Comparison with the VRSMD work [32]

Besides the implicit regularization property that is emphasised in this chapter, the variance reduced stochastic mirror descent (VRSMD) itself is an algorithm of independent interest to the optimization community. We notice a concurrent work that also applies variance reduction to SMD [32]. Our work is independently done without been affected by their

work, and we have different focuses. The high level difference of their work and this one is that: their main goal is the convergence analysis, while this work focus on the interesting implicit regularization property and its implication of VRSMD. Nevertheless, we can compare the overlapping part of our work with theirs, which is the convergence analysis. Here we do it from three perspectives: assumptions, measurement of convergence, and the the number of stochastic gradient $\nabla f_i(x)$ (i.e. SFO calls) required to achieve a ϵ convergence:

Difference in assumption: There are 4 main difference in assumptions:

1. They assume their objective function to be $\min \frac{1}{n}\sum_i f_i(x)+g(x)$, where g is possibly non-smooth but convex and f_i is possibly non-convex but smooth. Our formulation does not have the non-smooth component, and we assume f_i to be convex. In their work, they assume the proximal operator in g can be efficiently solved to handle the non-smoothness. And they use different measurement for convergence compared to our work due to non-convexity.

2. Their assumptions on the strong convexity of mirror map and the smoothness of the objective f_is are with respect to ℓ_2 norm. Our assumptions of the strong convexity and smoothness are for an arbitrary norm. Considering that the mirror descent is a nonEuclidean variant of gradient descent, assume a nonEuclidean norm would be more suitable.

3. They assume that the f_i has bounded variance as $E_i[\|\nabla f_i(x) - \nabla f(x)\|_2^2] \leq \sigma^2$. This assumption is not required in our work.

4. They assume the PL inequality (w.r.t. the ℓ_2 norm) holds to achieve a linear convergence rate. We prove the same rate, but under Quadratic Growth (QC) condition (w.r.t. an arbitrary norm). When considering the ℓ_2 norm, the PL condition implies the QC condition [43], thus their assumption is stronger than our assumption here.

Difference in measuring convergence: Since the objective in [32] is nonconvex, the convergence is evaluated for the stability gap $E[\|\nabla f(x)\|_2^2]$. Our works applies to a convex objective, so the convergence is evaluated for the optimization gap $E[f(x)] - \min_x f(x)$.

Comparing the SFO complexity: Unify all other notations and conditions in our

work and theirs, we compare the SFO complexity required to achieve a ϵ stability gap, i.e. $E[\|\nabla f(x)\|_2^2] \leq \epsilon$:

For the general case, our SFO is $\mathcal{O}(\frac{1}{\epsilon} + \frac{n}{\epsilon})$, and theirs is $\mathcal{O}(\min(\frac{n}{\epsilon}, \frac{\sigma^2}{\epsilon^2}) + \frac{1}{\epsilon})$. Ignoring the bounded variance assumption by taking $\sigma = \infty$, then their rate is the same as ours;

For the special case where we have linear convergence rate, our SFO is $\mathcal{O}(n \log \frac{1}{\epsilon} + \frac{1}{\mu} \log \frac{1}{\epsilon})$, and theirs is $\mathcal{O}(\min(n, \frac{\sigma^2}{\mu\epsilon}) \frac{1}{\mu} \log \frac{1}{\epsilon} + \frac{1}{\mu} \log \frac{1}{\epsilon})$. Again taking $\sigma = \infty$, our SFO complexity is better than theirs, possibly because the difference in problem setting.

3.6.2 Lemmas

We first list some lemmas that will be used in the proof. For each lemma, we either give its reference or show the proof.

Lemma 3.1 ([44]). *For a L-smooth function f with respect to $\|\cdot\|$, we have:*

$$f(\mathbf{u}) - f(\mathbf{w}) - \nabla f(\mathbf{w})^T(\mathbf{u} - \mathbf{w}) \leq \frac{L}{2}\|\mathbf{u} - \mathbf{w}\|^2. \tag{3.21}$$

Lemma 3.2 (Theorem 2.1.5 in [45]). *For a L-smooth function f with respect to $\|\cdot\|$, we have:*

$$\|\nabla f(\mathbf{u}) - \nabla f(\mathbf{w})\|_*^2 \leq 2L[f(\mathbf{u}) - f(\mathbf{w}) - \langle \nabla f(\mathbf{w}), \mathbf{u} - \mathbf{w}\rangle]. \tag{3.22}$$

And especially, when $\nabla f(\mathbf{w}) = 0$:

$$\|\nabla f(\mathbf{u})\|_*^2 \leq 2L[f(\mathbf{u}) - f(\mathbf{w})]. \tag{3.23}$$

Lemma 3.3. *For the problem $\min_{\boldsymbol{\beta}} F(\boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{\beta})$, suppose all f_is are convex and $L-$smooth w.r.t. norm $\|\cdot\|$, denote $\boldsymbol{\beta}^* = \operatorname{argmin}_{\boldsymbol{\beta}} F(\boldsymbol{\beta})$, then for i chosen uniformly random from $\{1, \ldots, n\}$, the term $\mathbf{v} = \nabla f_i(\boldsymbol{\beta}) - \nabla f_i(\tilde{\boldsymbol{\beta}}) + \nabla F(\tilde{\boldsymbol{\beta}})$ have:*

$$\mathbb{E}\|\mathbf{v}\|_*^2 \leq 12L[F(\boldsymbol{\beta}) - F(\boldsymbol{\beta}^*) + F(\tilde{\boldsymbol{\beta}}) - F(\boldsymbol{\beta}^*)]. \tag{3.24}$$

Proof.

$$
\begin{aligned}
\mathbb{E}\|\mathbf{v}\|_*^2 &= \mathbb{E}\|\nabla f_i(\boldsymbol{\beta}) - \nabla f_i(\tilde{\boldsymbol{\beta}}) + \nabla F(\tilde{\boldsymbol{\beta}})\|_*^2 \\
&\leq 3\mathbb{E}\|\nabla f_i(\boldsymbol{\beta}) - \nabla f_i(\boldsymbol{\beta}^*)\|_*^2 + 3\mathbb{E}\|\nabla f_i(\boldsymbol{\beta}^*) - \nabla f_i(\tilde{\boldsymbol{\beta}})\|_*^2 + 3\|\nabla F(\tilde{\boldsymbol{\beta}})\|_*^2 \\
&= 3\mathbb{E}\|\nabla f_i(\boldsymbol{\beta}) - \nabla f_i(\boldsymbol{\beta}^*)\|_*^2 + 3\mathbb{E}\|\nabla f_i(\boldsymbol{\beta}^*) - \nabla f_i(\tilde{\boldsymbol{\beta}})\|_*^2 + 3\|\nabla F(\tilde{\boldsymbol{\beta}}) - \nabla F(\boldsymbol{\beta}^*)\|_*^2 \\
&\overset{(3.22)}{\leq} 6\mathbb{E}\{L[f_i(\boldsymbol{\beta}) - f_i(\boldsymbol{\beta}^*) - \langle \nabla f_i(\boldsymbol{\beta}^*), \boldsymbol{\beta} - \boldsymbol{\beta}^* \rangle]\} \\
&\quad + 6\mathbb{E}\{L[f_i(\tilde{\boldsymbol{\beta}}) - f_i(\boldsymbol{\beta}^*) - \langle \nabla f_i(\boldsymbol{\beta}^*), \tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^* \rangle]\} \\
&\quad + 6\{L[F(\tilde{\boldsymbol{\beta}}) - F(\boldsymbol{\beta}^*) - \langle \nabla F(\boldsymbol{\beta}^*), \tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}^* \rangle]\} \\
&= 6L[F(\boldsymbol{\beta}) - F(\boldsymbol{\beta}^*)] + 12L[F(\tilde{\boldsymbol{\beta}}) - F(\boldsymbol{\beta}^*)] \\
&\leq 12L[F(\boldsymbol{\beta}) - F(\boldsymbol{\beta}^*) + F(\tilde{\boldsymbol{\beta}}) - F(\boldsymbol{\beta}^*)]
\end{aligned}
$$

□

Lemma 3.4. *For the linear regression objective function* $F(\boldsymbol{\beta}) = \|X\boldsymbol{\beta} - y\|_2^2/(2n) = \frac{1}{n}\sum_{i=1}^n f_i(\boldsymbol{\beta})$ *where* $f_i(\boldsymbol{\beta}) = \left(\mathbf{x}_i^T\boldsymbol{\beta} - y_i\right)^2/2$*, the term* $\mathbf{v} = \nabla f_i(\boldsymbol{\beta}) - \nabla f_i(\tilde{\boldsymbol{\beta}}) + \nabla F(\tilde{\boldsymbol{\beta}})$ *is in* $\mathrm{col}(X^T)$.

Proof. Check

$$
\nabla f_i(\boldsymbol{\beta}) = \mathbf{x}_i(\mathbf{x}_i^T\boldsymbol{\beta} - y_i) = X^T \begin{bmatrix} \mathbf{0}_{i-1} \\ 1 \\ \mathbf{0}_{n-i} \end{bmatrix} (\mathbf{x}_i^T\boldsymbol{\beta} - y_i) \in \mathrm{col}(X^T).
$$

And $\nabla F(\tilde{\boldsymbol{\beta}}) = X^T\left[\frac{1}{n}(X\tilde{\boldsymbol{\beta}} - \mathbf{y})\right] \in \mathrm{col}(X^T)$. Thus $\mathbf{v} = \nabla f_i(\boldsymbol{\beta}) - \nabla f_i(\tilde{\boldsymbol{\beta}}) + \nabla F(\tilde{\boldsymbol{\beta}}) \in \mathrm{col}(X^T)$. □

Lemma 3.5. *For a matrix* $X \in \mathcal{R}^{n\times p}$*, for any* $\mathbf{u} \in \mathrm{col}(X^T)$*, we have*

$$
\|X\mathbf{u}\|_2^2 \geq s_m^2\|\mathbf{u}\|_2^2 \tag{3.25}
$$

where s_m^2 is the smallest non-zero eigenvalue of X^TX.

Proof. Consider the SVD of $X^T = Q\Sigma V$ where $Q^TQ = QQ^T = I_p$ and $V^TV = VV^T = I_n$

$$\Sigma = \begin{bmatrix} \mathrm{diag}(\mathbf{s})_m & \mathbf{0}_{m\times(n-m)} \\ \mathbf{0}_{(p-m)\times m} & \mathbf{0}_{(p-m)\times(n-m)} \end{bmatrix}_{p\times n}$$

with $\mathbf{s} = [s_1, \ldots, s_m]$, $s_1 > \ldots > s_m > 0$. And write $Q = [\mathbf{q}_1, \ldots, \mathbf{q}_p]$, then for any $\mathbf{u} \in \mathrm{col}(X^T)$, we can write $\mathbf{u} = \sum_{i=1}^m w_i\mathbf{q}_i$ for some $w_1, \ldots, w_m$, and $\|\mathbf{u}\|_2^2 = \sum_{i=1}^m w_i^2$.

Then

$$\begin{aligned} X\mathbf{u} &= V^T\Sigma^TQ^T(\sum_{i=1}^m w_i\mathbf{q}_i) \\ &= V^T\Sigma^T \begin{bmatrix} w_1 \\ \vdots \\ w_m \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ &= V^T \begin{bmatrix} s_1w_1 \\ \vdots \\ s_mw_m \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \end{aligned}$$

Thus

$$\|X\mathbf{u}\|_2^2 = \|V^T \begin{bmatrix} s_1 w_1 \\ \vdots \\ s_m w_m \\ 0 \\ \vdots \\ 0 \end{bmatrix} \|_2^2 = \sum_{i=1}^{m} s_i^2 w_i^2 \geq s_m^2 \sum_{i=1}^{m} w_i^2 = s_m^2 \|\mathbf{u}\|_2^2.$$

□

Lemma 3.6. *If f is $\alpha-$strongly convex w.r.t. $\|\cdot\|$, then*

$$f(\mathbf{u}) \leq f(\mathbf{w}) + \nabla f(\mathbf{w})^T(\mathbf{u}-\mathbf{w}) + \frac{1}{2\alpha}\|\nabla f(\mathbf{u}) - \nabla f(\mathbf{w})\|_*^2. \tag{3.26}$$

Proof. Consider $g(\mathbf{v}) = f(\mathbf{v}) - \mathbf{v}^T\nabla f(\mathbf{w})$, then g is also $\alpha-$strongly convex, and it's easy to check that g has a unique minimizer $\mathbf{w}$.

Next, by $\alpha-$strong convexity of g, for any $\mathbf{u}$ we have

$$g(\mathbf{v}) \geq g(\mathbf{u}) + \nabla g(\mathbf{u})^T(\mathbf{v}-\mathbf{u}) + \frac{\alpha}{2}\|\mathbf{v}-\mathbf{u}\|^2. \tag{3.27}$$

Minimizing both side of (3.27) w.r.t. $\mathbf{v}$, we have

$$\begin{aligned} g(\mathbf{w}) = \min_{\mathbf{v}} g(\mathbf{v}) &\geq \min_{\mathbf{v}} g(\mathbf{u}) + \nabla g(\mathbf{u})^T(\mathbf{v}-\mathbf{u}) + \frac{\alpha}{2}\|\mathbf{v}-\mathbf{u}\|^2 \\ &\geq \min_{\mathbf{v}} g(\mathbf{u}) - \|\nabla g(\mathbf{u})\|_*\|\mathbf{v}-\mathbf{u}\| + \frac{\alpha}{2}\|\mathbf{v}-\mathbf{u}\|^2 \\ &\geq g(\mathbf{u}) - \frac{1}{2\alpha}\|\nabla g(\mathbf{u})\|_*^2. \end{aligned}$$

That is,

$$f(\mathbf{w}) - \mathbf{w}^T\nabla f(\mathbf{w}) \geq f(\mathbf{u}) - \mathbf{u}^T\nabla f(\mathbf{w}) - \frac{1}{2\alpha}\|\nabla f(\mathbf{u}) - \nabla f(\mathbf{w})\|_*^2.$$

Rearrange the terms we get (3.26). □

Lemma 3.7. *If f is $\alpha-$strongly convex w.r.t. $\|\cdot\|$, then*

$$[\nabla f(\mathbf{u}) - \nabla f(\mathbf{w})]^T[\mathbf{u} - \mathbf{w}] \leq \frac{1}{\alpha}\|\nabla f(\mathbf{u}) - \nabla f(\mathbf{w})\|_*^2. \tag{3.28}$$

Proof. Apply (3.26) twice to get:

$$f(\mathbf{u}) \leq f(\mathbf{w}) + \nabla f(\mathbf{w})^T(\mathbf{u} - \mathbf{w}) + \frac{1}{2\alpha}\|\nabla f(\mathbf{u}) - \nabla f(\mathbf{w})\|_*^2, \tag{3.29}$$

$$f(\mathbf{w}) \leq f(\mathbf{u}) + \nabla f(\mathbf{u})^T(\mathbf{w} - \mathbf{u}) + \frac{1}{2\alpha}\|\nabla f(\mathbf{u}) - \nabla f(\mathbf{w})\|_*^2. \tag{3.30}$$

Sum (3.29) and (3.30) we have:

$$\begin{aligned} 0 &\leq (\nabla f(\mathbf{w}) - f(\mathbf{u}))^T(\mathbf{u} - \mathbf{w}) + \frac{1}{\alpha}\|\nabla f(\mathbf{u}) - \nabla f(\mathbf{w})\|_*^2 \\ \Longrightarrow &[\nabla f(\mathbf{u}) - \nabla f(\mathbf{w})]^T[\mathbf{u} - \mathbf{w}] \leq \frac{1}{\alpha}\|\nabla f(\mathbf{u}) - \nabla f(\mathbf{w})\|_*^2. \end{aligned}$$

□

Lemma 3.8 ([43])**.** *If f satisfies PL inequality with respect to ℓ_2 norm $\|\cdot\|_2$, then f satisfies the Quadratic Growth (QG) condition w.r.t. $\|\cdot\|_2$ for the same constant μ.*

3.6.3 Proof for Proposition 3.1

Proof. Consider the Bregman divergence

$$\begin{aligned} &\mathbb{E}[D_\psi(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{t+1}^s)] \\ =&\mathbb{E}[\psi(\boldsymbol{\beta}^*) - \psi(\boldsymbol{\beta}_{t+1}^s) - \nabla\psi(\boldsymbol{\beta}_{t+1}^s)^T(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}^s)] \\ =&\mathbb{E}[\psi(\boldsymbol{\beta}^*) - \psi(\boldsymbol{\beta}_{t+1}^s) - [\nabla\psi(\boldsymbol{\beta}_t^s) - \eta\mathbf{v}_t^s]^T(\boldsymbol{\beta}^* - \boldsymbol{\beta}_t^s + \boldsymbol{\beta}_t^s - \boldsymbol{\beta}_{t+1}^s)] \\ =&\mathbb{E}[\psi(\boldsymbol{\beta}^*) - \psi(\boldsymbol{\beta}_t^s) + \psi(\boldsymbol{\beta}_t^s) - \psi(\boldsymbol{\beta}_{t+1}^s) - \nabla\psi(\boldsymbol{\beta}_t^s)^T(\boldsymbol{\beta}^* - \boldsymbol{\beta}_t^s) \\ &+ \nabla\psi(\boldsymbol{\beta}_t^s)^T(\boldsymbol{\beta}_{t+1}^s - \boldsymbol{\beta}_t^s) + (\eta\mathbf{v}_t^s)^T(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}^s)] \end{aligned}$$

$$=\mathbb{E}[D_\psi(\boldsymbol{\beta}^*, \boldsymbol{\beta}_t^s) - D_\psi(\boldsymbol{\beta}_{t+1}^s, \boldsymbol{\beta}_t^s) + (\eta \mathbf{v}_t^s)^T(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}^s)]. \tag{3.31}$$

The last term in (3.31):

$$\begin{aligned}
&\mathbb{E}[(\mathbf{v}_t^s)^T(\boldsymbol{\beta}^* - \boldsymbol{\beta}_{t+1}^s)] \\
=&\mathbb{E}[(\mathbf{v}_t^s)^T(\boldsymbol{\beta}^* - \boldsymbol{\beta}_t^s + \boldsymbol{\beta}_t^s - \boldsymbol{\beta}_{t+1}^s)] \\
=&\mathbb{E}[\mathbb{E}[(\mathbf{v}_t^s)^T(\boldsymbol{\beta}^* - \boldsymbol{\beta}_t^s)|\boldsymbol{\beta}_t^s]] + \mathbb{E}[(\mathbf{v}_t^s)^T(\boldsymbol{\beta}_t^s - \boldsymbol{\beta}_{t+1}^s)] \\
=&\mathbb{E}[(\nabla F(\boldsymbol{\beta}_t^s)^T(\boldsymbol{\beta}^* - \boldsymbol{\beta}_t^s)] + \frac{1}{\eta}\mathbb{E}[(\nabla\psi(\boldsymbol{\beta}_t^s) - \nabla\psi(\boldsymbol{\beta}_{t+1}^s))^T(\boldsymbol{\beta}_t^s - \boldsymbol{\beta}_{t+1}^s)] \\
\leq&\mathbb{E}[(F(\boldsymbol{\beta}^*) - F(\boldsymbol{\beta}_t^s)] + \frac{1}{\eta}\mathbb{E}[(\nabla\psi(\boldsymbol{\beta}_t^s) - \nabla\psi(\boldsymbol{\beta}_{t+1}^s))^T(\boldsymbol{\beta}_t^s - \boldsymbol{\beta}_{t+1}^s)] \\
\overset{(3.28)}{\leq}&\mathbb{E}[(F(\boldsymbol{\beta}^*) - F(\boldsymbol{\beta}_t^s)] + \frac{1}{\eta\alpha}\mathbb{E}[\|\nabla\psi(\boldsymbol{\beta}_t^s) - \nabla\psi(\boldsymbol{\beta}_{t+1}^s)\|_*^2] \\
=&\mathbb{E}[(F(\boldsymbol{\beta}^*) - F(\boldsymbol{\beta}_t^s)] + \frac{\eta}{\alpha}\mathbb{E}[\|\mathbf{v}_t^s\|_*^2].
\end{aligned}$$

Thus we have

$$\begin{aligned}
&\mathbb{E}[D_\psi(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{t+1}^s)] \\
\leq&\mathbb{E}[D_\psi(\boldsymbol{\beta}^*, \boldsymbol{\beta}_t^s)] - \mathbb{E}[D_\psi(\boldsymbol{\beta}_{t+1}^s, \boldsymbol{\beta}_t^s)] + \eta\mathbb{E}[F(\boldsymbol{\beta}^*) - F(\boldsymbol{\beta}_t^s)] + \frac{\eta^2}{\alpha}\mathbb{E}[\|\mathbf{v}_t^s\|_*^2] \\
\overset{(3.24)}{\leq}&\mathbb{E}[D_\psi(\boldsymbol{\beta}^*, \boldsymbol{\beta}_t^s)] - \mathbb{E}[D_\psi(\boldsymbol{\beta}_{t+1}^s, \boldsymbol{\beta}_t^s)] \\
&\quad + (\eta - \frac{12L\eta^2}{\alpha})\mathbb{E}[F(\boldsymbol{\beta}^*) - F(\boldsymbol{\beta}_t^s)] + \frac{12L\eta^2}{\alpha}\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s-1}) - F(\boldsymbol{\beta}^*)] \\
\leq&\mathbb{E}[D_\psi(\boldsymbol{\beta}^*, \boldsymbol{\beta}_t^s)] + \left(\frac{24L\eta^2}{\alpha} - \eta\right)\mathbb{E}[F(\boldsymbol{\beta}_t^s) - F(\boldsymbol{\beta}^*)] \\
&\quad + \frac{12L\eta^2}{\alpha}\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s-1}) - F(\boldsymbol{\beta}^*)] - \frac{12L\eta^2}{\alpha}\mathbb{E}[F(\boldsymbol{\beta}_t^s) - F(\boldsymbol{\beta}^*)].
\end{aligned} \tag{3.32}$$

Sum (3.32) for $t = 1, \ldots, m$, we have

$$\mathbb{E}[D_\psi(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{m+1}^s)] \leq \mathbb{E}[D_\psi(\boldsymbol{\beta}^*, \boldsymbol{\beta}_1^s)] + \sum_{t=1}^{m}\left(\frac{24L\eta^2}{\alpha} - \eta\right)\mathbb{E}[F(\boldsymbol{\beta}_t^s) - F(\boldsymbol{\beta}^*)]$$

$$+\frac{12L\eta^2 m}{\alpha}\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s-1})-F(\boldsymbol{\beta}^*)]-\frac{12L\eta^2}{\alpha}\sum_{t=1}^{m}\mathbb{E}[F(\boldsymbol{\beta}_t^s)-F(\boldsymbol{\beta}^*)]. \tag{3.33}$$

For option I, $\boldsymbol{\beta}_1^s = \boldsymbol{\beta}_{m+1}^{s-1}$, $\tilde{\boldsymbol{\beta}}^s$ is a uniform random sample from $\{\boldsymbol{\beta}_1^s,\ldots,\boldsymbol{\beta}_m^s\}$, (3.33) becomes:

$$\begin{aligned}\mathbb{E}[D_\psi(\boldsymbol{\beta}^*,\boldsymbol{\beta}_{m+1}^s)] \leq& \mathbb{E}[D_\psi(\boldsymbol{\beta}^*,\boldsymbol{\beta}_{m+1}^{s-1})]+\sum_{t=1}^{m}\left(\frac{24L\eta^2}{\alpha}-\eta\right)\mathbb{E}[F(\boldsymbol{\beta}_t^s)-F(\boldsymbol{\beta}^*)]\\ &+\frac{12L\eta^2 m}{\alpha}\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s-1})-F(\boldsymbol{\beta}^*)]-\frac{12L\eta^2 m}{\alpha}\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s})-F(\boldsymbol{\beta}^*)].\end{aligned}$$

That is,

$$\begin{aligned}&\left(\eta-\frac{24L\eta^2}{\alpha}\right)\sum_{t=1}^{m}\mathbb{E}[F(\boldsymbol{\beta}_t^s)-F(\boldsymbol{\beta}^*)]\\ \leq&\mathbb{E}[D_\psi(\boldsymbol{\beta}^*,\boldsymbol{\beta}_{m+1}^{s-1})]+\frac{12L\eta^2 m}{\alpha}\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s-1})-F(\boldsymbol{\beta}^*)]\\ &-\mathbb{E}[D_\psi(\boldsymbol{\beta}^*,\boldsymbol{\beta}_{m+1}^{s})]-\frac{12L\eta^2 m}{\alpha}\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s})-F(\boldsymbol{\beta}^*)].\end{aligned} \tag{3.34}$$

Thus we can define Lyapunov function

$$P^s := \mathbb{E}[D_\psi(\boldsymbol{\beta}^*,\boldsymbol{\beta}_{m+1}^s)]+\frac{12L\eta^2 m}{\alpha}\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^s)-F(\boldsymbol{\beta}^*)]\geq 0.$$

Then

$$\begin{aligned}\mathbb{E}[F(\boldsymbol{\beta}_a)-F(\boldsymbol{\beta}^*)] &= \frac{1}{T}\sum_{s=1}^{S}\sum_{t=1}^{m}\mathbb{E}[F(\boldsymbol{\beta}_t^s)-F(\boldsymbol{\beta}^*)]\\ &\overset{(3.34)}{\leq}\frac{1}{T}\sum_{s=1}^{S}\frac{1}{\eta-\frac{24L\eta^2}{\alpha}}[P^{s-1}-P^s]\\ &\leq\frac{\alpha}{(\alpha\eta-24L\eta^2)T}P^0\end{aligned}$$

For Option II, $\boldsymbol{\beta}_1^s = \tilde{\boldsymbol{\beta}}^{s-1}$, (3.33) becomes:

$$\begin{aligned}\mathbb{E}[D_\psi(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{m+1}^s)] \leq & \mathbb{E}[D_\psi(\boldsymbol{\beta}^*, \tilde{\boldsymbol{\beta}}^{s-1})] + \sum_{t=1}^{m}\left(\frac{24L\eta^2}{\alpha} - \eta\right)\mathbb{E}[F(\boldsymbol{\beta}_t^s) - F(\boldsymbol{\beta}^*)] \\ & + \frac{12L\eta^2 m}{\alpha}\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s-1}) - F(\boldsymbol{\beta}^*)] - \frac{12L\eta^2 m}{\alpha}\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s}) - F(\boldsymbol{\beta}^*)] \\ = & \mathbb{E}[D_\psi(\boldsymbol{\beta}^*, \tilde{\boldsymbol{\beta}}^{s-1})] + m\left(\frac{12L\eta^2}{\alpha} - \eta\right)\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s}) - F(\boldsymbol{\beta}^*)] \\ & + \frac{12L\eta^2 m}{\alpha}\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s-1}) - F(\boldsymbol{\beta}^*)]. \end{aligned} \tag{3.35}$$

Since ψ is $\ell-$smooth and F satisfies QG condition, $\boldsymbol{\beta}^*$ is any minimum point of F, we can take $\boldsymbol{\beta}^*$ as required by Lemma 3.8 to get

$$\begin{aligned} D_\psi(\boldsymbol{\beta}^*, \tilde{\boldsymbol{\beta}}^{s-1}) &= \psi(\boldsymbol{\beta}^*) - \psi(\tilde{\boldsymbol{\beta}}^{s-1}) - \nabla\psi(\tilde{\boldsymbol{\beta}}^{s-1})^T(\boldsymbol{\beta}^* - \tilde{\boldsymbol{\beta}}^{s-1}) \\ &\overset{(3.21)}{\leq} \frac{\ell}{2}\left\|\boldsymbol{\beta}^* - \tilde{\boldsymbol{\beta}}^{s-1}\right\|^2 \\ &\overset{QG \text{ of } f}{\leq} \frac{\ell}{\mu}[F(\tilde{\boldsymbol{\beta}}^{s-1}) - F(\boldsymbol{\beta}^*)]. \end{aligned} \tag{3.36}$$

Plug (3.36) into (3.35) we have

$$\begin{aligned} 0 \leq \mathbb{E}[D_\psi(\boldsymbol{\beta}^*, \boldsymbol{\beta}_{m+1}^s)] \leq & \left(\frac{12L\eta^2 m}{\alpha} - m\eta\right)\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s}) - F(\boldsymbol{\beta}^*)] \\ & + \left(\frac{12L\eta^2 m}{\alpha} + \frac{\ell}{\mu}\right)\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s-1}) - F(\boldsymbol{\beta}^*)]. \end{aligned}$$

That is,

$$\begin{aligned} \mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s}) - F(\boldsymbol{\beta}^*)] &\leq \frac{\frac{12L\eta^2}{\alpha} + \frac{\ell}{m\mu}}{\eta - \frac{12L\eta^2}{\alpha}}\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s-1}) - F(\boldsymbol{\beta}^*)] \\ &= \tau\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^{s-1}) - F(\boldsymbol{\beta}^*)]. \end{aligned}$$

Thus,

$$\mathbb{E}[F(\tilde{\boldsymbol{\beta}}^S) - F(\boldsymbol{\beta}^*)] \leq \tau^S \mathbb{E}[F(\tilde{\boldsymbol{\beta}}^0) - F(\boldsymbol{\beta}^*)]$$

and by $\boldsymbol{\beta}^a = \tilde{\boldsymbol{\beta}}^S$ we get the desired bound.

□

3.6.4 Proof for Theorem 3.1

Proof. We first check that every f_i is $L-$smooth w.r.t. $\|\cdot\|_2$.

$$\begin{aligned}
\|\nabla f_i(\mathbf{u}) - \nabla f_i(\mathbf{w})\|_2 &= \|(\mathbf{x}_i^T\mathbf{u} - y_i)\mathbf{x}_i - (\mathbf{x}_i^T\mathbf{w} - y_i)\mathbf{x}_i\|_2 \\
&= \|\mathbf{x}_i^T(\mathbf{u} - \mathbf{w})\mathbf{x}_i\|_2 \\
&= |\mathbf{x}_i^T(\mathbf{u} - \mathbf{w})| * \|\mathbf{x}_i\|_2 \\
&\leq \|\mathbf{x}_i\|_2\|\mathbf{u} - \mathbf{w}\|_2\|\mathbf{x}_i\|_2 \\
&\leq (\max_i \|x_i\|_2^2)\|\mathbf{u} - \mathbf{w}\|_2.
\end{aligned}$$

First, we prove part (a) of Theorem 3.1. When $\psi(\boldsymbol{\beta})$ is $\alpha-$strongly convex w.r.t. $\|\cdot\|_2$, we can apply part (a) of Theorem 1, take $\boldsymbol{\beta}^* = \boldsymbol{\beta}^\psi$ to get

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{2n}\|X\boldsymbol{\beta}^a - \mathbf{y}\|_2^2 - \frac{1}{2n}\|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2\right] \\
\leq &\frac{\alpha}{(\alpha\eta - 24L\eta^2)T}\left[D_\psi(\boldsymbol{\beta}^\psi, \tilde{\boldsymbol{\beta}}^0) + \frac{12L\eta^2 m}{\alpha}\left(\frac{1}{2n}\|X\tilde{\boldsymbol{\beta}}^0 - \mathbf{y}\|_2^2 - \frac{1}{2n}\|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2\right)\right].
\end{aligned} \tag{3.37}$$

For the L.H.S. of (3.37) we have

$$\begin{aligned}
&\|X\boldsymbol{\beta}^a - \mathbf{y}\|_2^2 - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2 \\
=&\|X\boldsymbol{\beta}^a - X\boldsymbol{\beta}^\psi + X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2 - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2
\end{aligned}$$

$$
\begin{aligned}
=&\|X\boldsymbol{\beta}^a - X\boldsymbol{\beta}^\psi\|_2^2 + \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2 + 2\langle X\boldsymbol{\beta}^a - X\boldsymbol{\beta}^\psi, P_{\mathrm{col}(X)}\mathbf{y} - \mathbf{y}\rangle - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2 \\
=&\|X\boldsymbol{\beta}^a - X\boldsymbol{\beta}^\psi\|_2^2 + 2\langle X(\boldsymbol{\beta}^a - \boldsymbol{\beta}^\psi), -P_{\mathcal{N}(X^T)}\mathbf{y}\rangle \\
=&\|X\boldsymbol{\beta}^a - X\boldsymbol{\beta}^\psi\|_2^2 = \|X\boldsymbol{\beta}^a - P_{\mathrm{col}(X)}\mathbf{y}\|_2^2.
\end{aligned}
$$

Thus (3.37) becomes

$$
\begin{aligned}
&\mathbb{E}[\|X\boldsymbol{\beta}^a - X\boldsymbol{\beta}^\psi\|_2^2] = \mathbb{E}[\|X\boldsymbol{\beta}^a - P_{\mathrm{col}(X)}\mathbf{y}\|_2^2] \\
\leq&\frac{\alpha}{(\alpha\eta - 24L\eta^2)T}\left[2nD_\psi(\boldsymbol{\beta}^\psi, \tilde{\boldsymbol{\beta}}^0) + \frac{12L\eta^2 m}{\alpha}\left(\|X\tilde{\boldsymbol{\beta}}^0 - \mathbf{y}\|_2^2 - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2\right)\right].
\end{aligned} \tag{3.38}
$$

Since $\nabla\psi(\tilde{\boldsymbol{\beta}}^0) = \nabla\psi(\boldsymbol{\beta}_m^0) \in \mathrm{col}(X^T)$, by Lemma 3.4 we will have $\nabla\psi(\boldsymbol{\beta}_t^s) \in \mathrm{col}(X^T)$ for all s and t, thus

$$
\nabla\psi(\boldsymbol{\beta}^a) \in \mathrm{col}(X^T). \tag{3.39}
$$

Then

$$
\begin{aligned}
&\mathbb{E}[\psi(\boldsymbol{\beta}^a) - \psi(\boldsymbol{\beta}^\psi)] \\
\leq&\mathbb{E}\langle\nabla\psi(\boldsymbol{\beta}^a), \boldsymbol{\beta}^a - \boldsymbol{\beta}^\psi\rangle \\
\overset{(3.39)}{=}&\mathbb{E}\langle\nabla\psi(\boldsymbol{\beta}^a), P_{\mathrm{col}(X^T)}(\boldsymbol{\beta}^a - \boldsymbol{\beta}^\psi)\rangle \\
\leq&\mathbb{E}[\|\nabla\psi(\boldsymbol{\beta}^a)\|_2\|P_{\mathrm{col}(X^T)}(\boldsymbol{\beta}^a - \boldsymbol{\beta}^\psi)\|_2] \\
\leq&B * \mathbb{E}\|P_{\mathrm{col}(X^T)}(\boldsymbol{\beta}^a - \boldsymbol{\beta}^\psi)\|_2 \\
\overset{(3.25)}{\leq}&\frac{B}{s_m}\mathbb{E}\|XP_{\mathrm{col}(X^T)}(\boldsymbol{\beta}^a - \boldsymbol{\beta}^\psi)\|_2 \\
=&\frac{B}{s_m}\mathbb{E}\|X\boldsymbol{\beta}^a - X\boldsymbol{\beta}^\psi\|_2 \\
\leq&\frac{B}{s_m}(\mathbb{E}\|X\boldsymbol{\beta}^a - X\boldsymbol{\beta}^\psi\|_2^2)^{1/2} \\
\overset{(3.38)}{\leq}&\frac{B}{s_m}\sqrt{\frac{\alpha}{(\alpha\eta - 24L\eta^2)T}\left[2nD_\psi(\boldsymbol{\beta}^\psi, \tilde{\boldsymbol{\beta}}^0) + \frac{12L\eta^2 m}{\alpha}\left(\|X\tilde{\boldsymbol{\beta}}^0 - \mathbf{y}\|_2^2 - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2\right)\right]}.
\end{aligned}
$$

Thus the $\boldsymbol{\beta}^a$ from SMD will have

$$\mathbb{E}[\psi(\boldsymbol{\beta}^a) - \psi(\boldsymbol{\beta}^\psi)]$$
$$\leq \frac{B}{s_m}\sqrt{\frac{\alpha}{(\alpha\eta - 24L\eta^2)T}\left[2nD_\psi(\boldsymbol{\beta}^\psi, \tilde{\boldsymbol{\beta}}^0) + \frac{12L\eta^2 m}{\alpha}\left(\|X\tilde{\boldsymbol{\beta}}^0 - \mathbf{y}\|_2^2 - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2\right)\right]} \tag{3.40}$$

$$\mathbb{E}[\|X\boldsymbol{\beta}^a - \mathbf{y}\|_2^2 - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2]$$
$$\leq \frac{\alpha}{(\alpha\eta - 24L\eta^2)T}\left[2nD_\psi(\boldsymbol{\beta}^\psi, \tilde{\boldsymbol{\beta}}^0) + \frac{12L\eta^2 m}{\alpha}\left(\|X\tilde{\boldsymbol{\beta}}^0 - \mathbf{y}\|_2^2 - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2\right)\right]. \tag{3.41}$$

Set the R.H.S. of (3.40) and (3.41) to ϵ to get an $\epsilon-$solution, we can solve

$$T \geq \frac{B^2}{s_m^2\epsilon^2}\frac{\alpha}{(\alpha\eta - 24L\eta^2)}\left[2nD_\psi(\boldsymbol{\beta}^\psi, \tilde{\boldsymbol{\beta}}^0) + \frac{12L\eta^2 m}{\alpha}\left(\|X\tilde{\boldsymbol{\beta}}^0 - \mathbf{y}\|_2^2 - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2\right)\right]$$
$$T \geq \frac{1}{\epsilon}\frac{\alpha}{(\alpha\eta - 24L\eta^2)}\left[2nD_\psi(\boldsymbol{\beta}^\psi, \tilde{\boldsymbol{\beta}}^0) + \frac{12L\eta^2 m}{\alpha}\left(\|X\tilde{\boldsymbol{\beta}}^0 - \mathbf{y}\|_2^2 - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2\right)\right].$$

That is, $T \sim \mathcal{O}(\frac{1}{\epsilon} + \frac{1}{\epsilon^2})$.

For part (b) of Theorem 3.1, we first need to show that $F(\cdot)$ satisfies PL inequality w.r.t. $\|\cdot\|_2$. Remember $F(\boldsymbol{\beta}) = \frac{1}{2n}\|X\boldsymbol{\beta} - y\|_2^2$, then

$$\begin{aligned}
\frac{1}{2}\|\nabla F(\boldsymbol{\beta})\|_2^2 &= \frac{1}{2}\|\frac{1}{n}X^T(X\boldsymbol{\beta} - y)\|_2^2 \\
&= \frac{1}{2n^2}\|X^T(X\boldsymbol{\beta} - P_{\text{col}(X)}y)\|_2^2 \\
&\overset{(3.25)}{\geq} \frac{s_m^2}{2n^2}\|X\boldsymbol{\beta} - P_{\text{col}(X)}y\|_2^2 \\
&= \frac{s_m^2}{2n^2}\|X\boldsymbol{\beta} - y + P_{\mathcal{N}(X^T)}y\|_2^2 \\
&= \frac{s_m^2}{2n^2}[\|X\boldsymbol{\beta} - y\|_2^2 - \|P_{\mathcal{N}(X^T)}y\|_2^2] \\
&= \frac{s_m^2}{n}[F(\boldsymbol{\beta}) - F(\boldsymbol{\beta}^\psi)].
\end{aligned}$$

Thus $F(\cdot)$ satisfies PL inequality with constant $\frac{s_m^2}{n}$. By Lemma 3.8, $F(\cdot)$ satisfies QG with

the same constant, so part (b) of Theorem 1 applies here, and we get

$$\mathbb{E}[F(\boldsymbol{\beta}_a) - F(\boldsymbol{\beta}^\psi)] \leq (\tau')^S [F(\tilde{\boldsymbol{\beta}}^0) - F(\boldsymbol{\beta}^\psi)].$$

where

$$\tau' = \frac{\frac{12L\eta^2}{\alpha} + \frac{\ell n}{ms_m^2}}{\eta - \frac{12L\eta^2}{\alpha}},$$

By the similar statement as the first part of the proof, we have

$$\begin{aligned}
&\mathbb{E}[\psi(\boldsymbol{\beta}^a) - \psi(\boldsymbol{\beta}^\psi)] \\
\leq &\frac{B}{s_m}[2n\mathbb{E}[F(\boldsymbol{\beta}^a) - F(\boldsymbol{\beta}^\psi)]]^{.5} \\
\leq &\frac{B\sqrt{2n}(\tau')^{S/2}}{s_m}[F(\tilde{\boldsymbol{\beta}}^0) - F(\boldsymbol{\beta}^\psi)]^{.5}.
\end{aligned}$$

Thus

$$\mathbb{E}[\psi(\boldsymbol{\beta}^a) - \psi(\boldsymbol{\beta}^\psi)] \leq \frac{B(\tau')^{S/2}}{s_m}\sqrt{\|X\tilde{\boldsymbol{\beta}}^0 - \mathbf{y}\|_2^2 - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2}$$

$$\mathbb{E}[\|X\boldsymbol{\beta}^a - \mathbf{y}\|_2^2 - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2] \leq (\tau')^S \left(\|X\tilde{\boldsymbol{\beta}}^0 - \mathbf{y}\|_2^2 - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2\right).$$

□

3.6.5 Proof of Corollary 3.1

Proof. For the specific choice of $\psi(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, we have

$$\nabla^2\psi(\cdot) = I.$$

That is, $\psi(\cdot)$ is $1-$smooth and $1-$strongly convex, for

$$\tau'' = \frac{12L\eta^2 + \frac{n}{ms_m^2}}{\eta - 12L\eta^2} < 1,$$

we can apply part (b) of Theorem 2 to get

$$\begin{aligned}\mathbb{E}[\|X\boldsymbol{\beta}^a - X\boldsymbol{\beta}^\psi\|_2^2 &\leq (\tau'')^S \left(\|X\tilde{\boldsymbol{\beta}}^0 - \mathbf{y}\|_2^2 - \|X\boldsymbol{\beta}^\psi - \mathbf{y}\|_2^2\right) \\ &= (\tau'')^S \left(\|\mathbf{y}\|_2^2 - \|P_{\mathcal{N}(X^T)}\mathbf{y}\|_2^2\right) = (\tau'')^S \|P_{\mathrm{col}(X)}\mathbf{y}\|_2^2.\end{aligned}$$

Then since $\nabla\psi(\boldsymbol{\beta}^a) = \boldsymbol{\beta}^a \in \mathrm{col}(X^T)$ and $\boldsymbol{\beta}^\psi = (X^TX)^+X^T\mathbf{y} = X^T(XX^T)^+\mathbf{y} \in \mathrm{col}(X^T)$, we have

$$\mathbb{E}\|\boldsymbol{\beta}_a - \boldsymbol{\beta}^\psi\|_2^2 \overset{(3.25)}{\leq} \frac{1}{s_m^2}\mathbb{E}\|X\boldsymbol{\beta}^a - X\boldsymbol{\beta}^\psi\|_2^2 \leq \frac{\tau^S}{s_m^2}\|P_{\mathrm{col}(X)}\mathbf{y}\|_2^2.$$

□

3.6.6 Proof of Theorem 3.2

To prove Theorem 3.2, we will need a few preliminaries.

Lemma 3.9. *Denote* $\boldsymbol{\beta}^{(0)} = \mathrm{argmin}_{\boldsymbol{\beta}}\{\|\boldsymbol{\beta}\|_1 : X\boldsymbol{\beta} = P_{\mathrm{col}(X)}\mathbf{y}\}$ *and* $\boldsymbol{\beta}^{(\delta)} = \mathrm{argmin}_{\boldsymbol{\beta}}\{\|\boldsymbol{\beta}\|_{1+\delta}^{1+\delta} : X\boldsymbol{\beta} = P_{\mathrm{col}(X)}\mathbf{y}\}$*, we have*

$$\|\boldsymbol{\beta}^{(\delta)}\|_1 \leq p^{\frac{\delta}{1+\delta}}\|\boldsymbol{\beta}^{(0)}\|_1 \tag{3.42}$$

Proof.

$$\begin{aligned}\|\boldsymbol{\beta}^{(\delta)}\|_1 &= \sum_{i=1}^p 1 * |\beta_i^{(\delta)}| \\ &\leq \left(\sum_{i=1}^p 1^{\frac{1+\delta}{\delta}}\right)^{\frac{\delta}{1+\delta}} * \left(\sum_{i=1}^p |\beta_i^{(\delta)}|^{1+\delta}\right)^{\frac{1}{1+\delta}} \\ &= p^{\frac{\delta}{1+\delta}}\|\boldsymbol{\beta}^{(\delta)}\|_{1+\delta} \\ &\leq p^{\frac{\delta}{1+\delta}}\|\boldsymbol{\beta}^{(0)}\|_{1+\delta} \\ &\leq p^{\frac{\delta}{1+\delta}}\left(\sum_{i=1}^p \|\boldsymbol{\beta}^{(0)} \circ \mathbf{e}_i\|_{1+\delta}\right)\end{aligned}$$

$$= p^{\frac{\delta}{1+\delta}} \left(\sum_{i=1}^{p} |\beta_i^{(0)}| \right)$$
$$= p^{\frac{\delta}{1+\delta}} \|\boldsymbol{\beta}^{(0)}\|_1$$

where the first inequality holds by Hölder's inequality, the second inequality follows from $\boldsymbol{\beta}^{(\delta)}$ minimizes $\|\boldsymbol{\beta}\|_{1+\delta}^{1+\delta}$ thus minimizes $\|\boldsymbol{\beta}\|_{1+\delta}$ among all $\boldsymbol{\beta} \in \mathcal{B} := \{\boldsymbol{\beta} : X\boldsymbol{\beta} = P_{\mathrm{col(X)}}\mathbf{y}\}$, and the third inequality follows from triangle inequality. □

Lemma 3.10. *Assume* $\mathbf{y} = X\boldsymbol{\beta}^o$ *where* $\boldsymbol{\beta}^o \leq s$ *and* X *is* $s-$*good with parameter* $\kappa < 1/2$. *Use the notation in Lemma 3.9 that* $\boldsymbol{\beta}^{(0)} = \mathrm{argmin}_{\boldsymbol{\beta}}\{\|\boldsymbol{\beta}\|_1 : X\boldsymbol{\beta} = \mathbf{y}\}$, *we have* $\boldsymbol{\beta}^{(0)} = \boldsymbol{\beta}^o$.

Proof. $\boldsymbol{\beta}^o$ is a feasible solution to $\min_{\boldsymbol{\beta}}\{\|\boldsymbol{\beta}\|_1 : X\boldsymbol{\beta} = \mathbf{y}\}$. Denote $\hat{\mathbf{u}} = \boldsymbol{\beta}^o - \boldsymbol{\beta}^{(0)}$, we have

$$\begin{aligned} & \left(\|\boldsymbol{\beta}_S^{(0)}\|_1 + \|\boldsymbol{\beta}_{S^c}^{(0)}\|_1\right) = \|\boldsymbol{\beta}^{(0)}\|_1 \leq \|\boldsymbol{\beta}^o\|_1 = \|\boldsymbol{\beta}_S^o\|_1 \\ \Longrightarrow \quad & \|\boldsymbol{\beta}_{S^c}^{(0)}\|_1 \leq \|\boldsymbol{\beta}_S^o\|_1 - \|\boldsymbol{\beta}_S^{(0)}\|_1 \leq \|\boldsymbol{\beta}_S^o - \boldsymbol{\beta}_S^{(0)}\|_1 \\ \Longleftrightarrow \quad & \|\hat{\mathbf{u}}_{S^c}\|_1 \leq \|\hat{\mathbf{u}}_S\|_1. \end{aligned} \tag{3.43}$$

On the other hand,

$$X\hat{\mathbf{u}} = X\boldsymbol{\beta}^o - X\boldsymbol{\beta}^{(0)} = \mathbf{y} - \mathbf{y} = \mathbf{0}.$$

That is, $\hat{\mathbf{u}} \in \mathcal{N}(X)$, then by X is $s-$good, we have for $\kappa < \frac{1}{2}$

$$\begin{aligned} & \|\hat{\mathbf{u}}_S\|_1 \leq \kappa \|\hat{\mathbf{u}}\|_1 \\ \Longrightarrow & \|\hat{\mathbf{u}}_S\|_1 \leq \frac{\kappa}{1-\kappa} \|\hat{\mathbf{u}}_{S^c}\|_1. \end{aligned} \tag{3.44}$$

Combine (3.43) and (3.44) we get

$$\|\hat{\mathbf{u}}_{S^c}\|_1 \leq \tfrac{\kappa}{1-\kappa} \|\hat{\mathbf{u}}_{S^c}\|_1$$

$$\begin{aligned} &\implies && \|\hat{\mathbf{u}}_{S^c}\|_1 = 0 \\ &\overset{(3.44)}{\implies} && \|\hat{\mathbf{u}}_S\|_1 = 0. \end{aligned}$$

Thus $\hat{\mathbf{u}} = \mathbf{0}$, $\boldsymbol{\beta}^{(0)} = \boldsymbol{\beta}^o$. □

Lemma 3.11. *Assume $X \in \mathcal{R}^{n\times p}$ is $s-$ good with parameter $\kappa < 1/2$, then $\forall \boldsymbol{\beta} \in \mathcal{R}^p$ and $I \subset \{1,\dots,p\}$ with $|I| \le s$ we have*

$$\|\boldsymbol{\beta}_I\|_1 \le \frac{\sqrt{s} + \kappa\sqrt{p}}{s_m}\|X\boldsymbol{\beta}\|_2 + \kappa\|\boldsymbol{\beta}\|_1$$

where s_m^2 is the smallest nonzero eigenvalue of X^TX.

Proof.

$$\begin{aligned} \|\boldsymbol{\beta}_I\|_1 &\le \|(P_{\mathrm{col}(X^T)}\boldsymbol{\beta})_I\|_1 + \|(P_{\mathcal{N}(X)}\boldsymbol{\beta})_I\|_1 \\ &\le \sqrt{s}\|(P_{\mathrm{col}(X^T)}\boldsymbol{\beta})_I\|_2 + \kappa\|P_{\mathcal{N}(X)}\boldsymbol{\beta}\|_1 \\ &\le \sqrt{s}\|P_{\mathrm{col}(X^T)}\boldsymbol{\beta}\|_2 + \kappa(\|P_{\mathcal{N}(X)}\boldsymbol{\beta} + P_{\mathrm{col}(X^T)}\boldsymbol{\beta}\|_1 + \|P_{\mathrm{col}(X^T)}\boldsymbol{\beta}\|_1) \\ &\overset{(3.25)}{\le} \frac{\sqrt{s}}{s_m}\|XP_{\mathrm{col}(X^T)}\boldsymbol{\beta}\|_2 + \kappa\left(\|\boldsymbol{\beta}\|_1 + \frac{\sqrt{p}}{s_m}\|XP_{\mathrm{col}(X^T)}\boldsymbol{\beta}\|_2\right) \\ &= \frac{\sqrt{s} + \kappa\sqrt{p}}{s_m}\|X\boldsymbol{\beta}\|_2 + \kappa\|\boldsymbol{\beta}\|_1 \end{aligned}$$

where the first inequality holds by triangle inequality, the second inequality holds by Cauchy-Schwartz inequality and s-goodness of X, the third inequality again uses the triangle inequality. □

Theorem 3.3. *[Theorem 1.3.1 in [46]] Consider the following optimization problem with any norm $\|\cdot\|$*

$$\begin{aligned} \hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \quad & \|\boldsymbol{\beta}\|_1 \\ s.t. \quad & \|X\boldsymbol{\beta} - \mathbf{y}\| \le \Delta \end{aligned} \tag{3.45}$$

Assume $\exists \alpha \geq 0$ *and* $\kappa' < 1/2$ *that* $\forall \boldsymbol{\beta} \in \mathcal{R}^p$ *and* $\forall I \subset \{1, \ldots, p\}$ *with* $|I| \leq s$*:*

$$\|\boldsymbol{\beta}_I\|_1 \leq \alpha \|X\boldsymbol{\beta}\| + \kappa' \|\boldsymbol{\beta}\|_1$$

Suppose $\tilde{\boldsymbol{\beta}}$ *is an approximate solution to* (3.45) *such that*

$$\|\tilde{\boldsymbol{\beta}}\|_1 \leq \|\hat{\boldsymbol{\beta}}\|_1 + \nu$$
$$\|X\tilde{\boldsymbol{\beta}} - \mathbf{y}\| \leq \Delta + \epsilon.$$

Then for any nearly $s-$*sparse and feasible* $\boldsymbol{\beta}'$ *for* (3.45)*, where near* $s-$*sparsity means* $\exists$ $s-$*sparse* $\boldsymbol{\beta}^{\boldsymbol{s}}$ *such that* $\|\boldsymbol{\beta}' - \boldsymbol{\beta}^{\boldsymbol{s}}\|_1 \leq v$*, we have*

$$\|\tilde{\boldsymbol{\beta}} - \boldsymbol{\beta}'\|_1 \leq \frac{2\alpha(2\Delta + \epsilon) + 2v + \nu}{1 - 2\kappa'}.$$

We are ready to prove Theorem 3.2.

Proof for Theorem 3.2. By Lemma 3.9 we have

$$\|\boldsymbol{\beta}^{(\delta)}\|_1 \leq \|\boldsymbol{\beta}^{(0)}\|_1 + \left(p^{\frac{\delta}{1+\delta}} - 1\right) \|\boldsymbol{\beta}^{(0)}\|_1$$

And by Lemma 3.11

$$\|\boldsymbol{\beta}_I\|_1 \leq \frac{\sqrt{s} + \kappa\sqrt{p}}{s_m} \|X\boldsymbol{\beta}\|_2 + \kappa \|\boldsymbol{\beta}\|_1$$

Thus we can apply Theorem 3.3 by setting $\Delta = 0$, and $\|\cdot\| = \|\cdot\|_2$, then $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(0)}$. Take $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(\delta)}$, $\boldsymbol{\beta}' = \boldsymbol{\beta}^o$, then $\nu = \left(p^{\frac{\delta}{1+\delta}} - 1\right) \|\boldsymbol{\beta}^{(0)}\|_1$, $\epsilon = 0$, $v = 0$, $\alpha = \frac{\sqrt{s} + \kappa\sqrt{p}}{s_m}$, $\kappa' = \kappa$, then we have

$$\|\boldsymbol{\beta}^{(\delta)} - \boldsymbol{\beta}^o\|_1 \leq \frac{\left(p^{\frac{\delta}{1+\delta}} - 1\right) \|\boldsymbol{\beta}^{(0)}\|_1}{1 - 2\kappa} = \frac{\left(p^{\frac{\delta}{1+\delta}} - 1\right) \|\boldsymbol{\beta}^o\|_1}{1 - 2\kappa} \tag{3.46}$$

where the equality is from Lemma 3.10.

We can further bound $\|\boldsymbol{\beta}^o\|_1$. By $(s,\gamma) - RE$ condition we have

$$\|\boldsymbol{\beta}^o\|_2^2 \leq \frac{\|X\boldsymbol{\beta}^o\|_2^2}{n\gamma} = \frac{\|\mathbf{y}\|_2^2}{n\gamma}.$$

Thus

$$\|\boldsymbol{\beta}^o\|_1 \leq \sqrt{s}\|\boldsymbol{\beta}^o\|_2 \leq \sqrt{\frac{s}{n\gamma}}\|\boldsymbol{y}\|_2. \tag{3.47}$$

Plug (3.47) into (3.46), we have

$$\begin{aligned}
\|\boldsymbol{\beta}^{(\delta)} - \boldsymbol{\beta}^o\|_1 &\leq \frac{\left(p^{\frac{\delta}{1+\delta}} - 1\right)}{1-2\kappa}\sqrt{\frac{s}{n\gamma}}\|\boldsymbol{y}\|_2 \\
&\leq \frac{\left(p^{\frac{\log\left(1+\frac{(1-2\kappa)\sqrt{n\gamma}}{\sqrt{s}\|\mathbf{y}\|_2}\xi\right)}{\log p}} - 1\right)}{1-2\kappa}\sqrt{\frac{s}{n\gamma}}\|\boldsymbol{y}\|_2 \\
&= \frac{\left(1 + \frac{(1-2\kappa)\sqrt{n\gamma}}{\sqrt{s}\|\mathbf{y}\|_2}\xi - 1\right)}{1-2\kappa}\sqrt{\frac{s}{n\gamma}}\|\boldsymbol{y}\|_2 = \xi.
\end{aligned}$$

□

CHAPTER 4

INFORMATION GEOMETRY OF VARIANCE REDUCED STOCHASTIC NATURAL GRADIENT AND MIRROR DESCENT

In this chapter, we propose two algorithms for estimating parameter in an exponential family. To accommodate the nonEuclidean structure that naturally exists in parameter estimation of exponential family, we closely investigates the Natural Gradient Descent and Mirror Descent algorithms. A stochastic version of these two algorithms are analyzed in spirit of fast computation, and variance reduction technique is applied to achieve convergence. Our finding reveals that the two algorithms are essentially equivalent. This allows us to generalize the properties of one algorithm to the other. Based on this, we prove the linear convergence of variance reduced stochastic natural gradient descent (VRSNGD) using the convergence rate of variance reduced stoachstic mirror descent (VRSMD) that is available in literature. We also get thorough understanding for VRSMD such as parameterization invariant as a direct consequence of such property in VRSNGD. In this way, we can make our VRSMD algorithm robust. Experiments are done to support our theorems. Keeping our finding in mind, more research could be done along this line for understanding the properties of nonEuclidean variance reduced algorithms, for example for understanding the role of mirror map in VRSMD and the choice of Riemannian metric in VRSNGD.

4.1 Introduction

A lot of parameters estimation problems can be considered as estimating the parameters from an exponential family. In this draft we analyze the batch learning problem for the exponential family under the empirical risk minimization framework, where the loss function we considered is the negative log likelihood loss. Specifically, we use generalizations of the stochastic gradient descent method to accommodate the special geometry structure of

the exponential model. In the mean time, the variance reduction technique is applied to our stochastic algorithm to speed convergence.

As mentioned above, the algorithms studied in this draft include variance reduced stochastic natural gradient descent (VRSNGD) and variance reduced stochastic mirror descent (VRSMD). Both the variance reduction technique and mirror descent/ natural gradient descent are well used in practice, yet the properties of the algorithms have not been fully discovered. Mirror descent and natural gradient descent have different origins and show different ideas to accommodate the geometry structure, but [47] finds that they can be essentially equivalent. It is interesting yet unknown to us whether or not the equivalency also holds for the variance reduced stochastic versions of these two algorithm. Note that the variance reduced algorithms consists of inner and outer loops, and is more complicated to analysis compared to the vanilla stochastic versions. In this chapter we carefully investigates both inner and outer loop of VRSMD and VRSNGD, and figure out the conditions under which they are equivalent.

The equivalency result is useful in at least two perspectives: First, it allows us to generalize our findings on one algorithm to the other, for example, we show the linear convergence of VRSNGD as a consequence of the linear convergence of VRSMD, and the parameterization invariance of VRSMD as a consequence of VRSNGD, both are desired properties in learning. Second, though the VRSNGD algorithm naturally accommodates the nonEuclidean structure of a statistical problem, it requires computing the inverse of the metric tensor for each update, which is expensive in both computation and space. Using its equivalent form of VRSMD we can replace such matrix inversion by approximately solving the proximal step in VRSMD, thus is more efficient while still enjoy the advantage of properly accommodates the nonEuclidean structure. Such kind of computational advantage can also be observed for VRSMD compared with the Iteratively Re-weighted Least Squares (IRLS) when applied to a Generalized Linear Model (GLM). One can check the details in Section 4.

4.1.1 Exponential family

Consider a parameter estimation problem as follows: we have data $\{\boldsymbol{z}_i\}_{i=1,\dots,n}$ from a natural parameter exponential family with density

$$p(\boldsymbol{z}|\boldsymbol{\theta}) = \tilde{h}(\boldsymbol{z}) \exp(\langle \boldsymbol{\theta}, \boldsymbol{z} \rangle - G(\boldsymbol{\theta})). \tag{4.1}$$

One can check that for a random variable Z from the density (4.1), one has $\mathbb{E}[Z] = \nabla G(\theta)$ and $\mathrm{Var}(Z) = \nabla^2 G(\theta)$. The parameterization problem is formulated as Maximum Likelihood Estimation (MLE):

$$\boldsymbol{\theta}^* = \mathrm{argmax}_{\boldsymbol{\theta}} \prod_{i=1}^{n} p(\boldsymbol{z}_i|\boldsymbol{\theta}), \tag{4.2}$$

or equivalently, the Empirical Risk Minimization (ERM) using negative log likelihood loss:

$$\boldsymbol{\theta}^* = \mathrm{argmin}_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} [-\log p(\boldsymbol{z}_i|\boldsymbol{\theta})] := \frac{1}{n} \sum_{i=1}^{n} l_i(\boldsymbol{\theta}). \tag{4.3}$$

We list some common distributions that belong to the exponential family, in their natural parameter form, as given in Table 4.1. One may find a thorough explanation and more examples for exponential families in [48, 49]. Next, the Generalized Linear Model (GLM)

Table 4.1: Exponential Family

Family	Natural parameter	$G(\boldsymbol{\theta})$
Normal($\boldsymbol{\theta}, I_p$)	$\boldsymbol{\theta}$	$\|\boldsymbol{\theta}\|_2^2/2$
Bernoulli(p)	$\theta = \log\left(\frac{p}{1-p}\right)$	$\log(\exp(\theta) + 1)$
Poisson(λ)	$\theta = \log \lambda$	$\exp(\theta)$
Beta(α, β)	$\boldsymbol{\theta} = (\alpha, \beta)^T$	$\log\Gamma(\theta_1) + \log\Gamma(\theta_2) - \log\Gamma(\theta_1 + \theta_2)$
Multinomial($\boldsymbol{\pi}$), where $\boldsymbol{\pi} \in \Delta^m$	$\boldsymbol{\theta} = \left(\log\left(\frac{\pi_i}{1-\sum_{j=1}^{m-1}\pi_j}\right)\right)_{i=1,\dots,m-1}$	$\log(1 + \sum_{i=1}^{m-1} \exp(\theta_i))$

comes from the exponential family. To begin with, consider the linear regression model

(LM) where data is generated by

$$y = \langle \boldsymbol{x}, \boldsymbol{\alpha} \rangle + \epsilon, \tag{4.4}$$

where $\epsilon \sim N(0, \sigma^2)$ for a known variance σ^2. Denote $\theta = \langle \boldsymbol{x}, \boldsymbol{\alpha} \rangle$, then $[y|\theta]$ is the normal distribution and belongs to the exponential family.

The GLM generalize LM as follows: consider the conditional mean $\boldsymbol{\mu}$ of $\boldsymbol{y}$ given $\boldsymbol{x}$ and $\boldsymbol{\gamma}$, where $\boldsymbol{\gamma}$ are unknown parameter, assume that we have an invertible *response function* h such that

$$\boldsymbol{\mu}|\boldsymbol{x}, \boldsymbol{\gamma} = \boldsymbol{h}(\langle \boldsymbol{x}, \boldsymbol{\gamma} \rangle), \tag{4.5}$$

and $[\boldsymbol{y}|\boldsymbol{\mu}]$ is from an exponential family with mean $\boldsymbol{\mu}$. In GLM literature, it is common to define the *link function* as $\boldsymbol{g} := \boldsymbol{h}^{-1}$. We further denote the dependency of $\boldsymbol{\mu}$ on the natural parameter $\boldsymbol{\theta}$ as $\boldsymbol{\mu}(\boldsymbol{\theta})$, then we have

$$\langle \boldsymbol{x}, \boldsymbol{\gamma} \rangle = \boldsymbol{g}(\boldsymbol{\mu}(\boldsymbol{\theta})) := \boldsymbol{q}^{-1}(\boldsymbol{\theta}). \tag{4.6}$$

Note that the link function $\boldsymbol{g}$ in (4.6) is not uniquely determined, and one should select a proper link function for GLM. One choice of link function is the *canonical link function* $\boldsymbol{g} = \boldsymbol{\mu}^{-1}$, or equivalently, the *canonical response function* $\boldsymbol{h} = \boldsymbol{\mu}$. When using the canonical link, one has $\boldsymbol{\theta} = \langle \boldsymbol{x}, \boldsymbol{\gamma} \rangle$. In Table 4.2 we list some GLMs, their associated exponential families, and the canonical response function $\boldsymbol{\mu}$, one can refer to [50] for more details and examples on GLM. Then given data $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1,\dots,n}$, the MLE/ERM of GLM is given by minimizing the negative log-likelihood as:

$$\operatorname{argmin}_{\boldsymbol{\gamma}}(-\ell(\boldsymbol{\gamma})) := \sum_{i=1}^{n} \left[G(\boldsymbol{q}(\langle \boldsymbol{x}_i, \boldsymbol{\gamma} \rangle)) - \langle \boldsymbol{y}_i, \boldsymbol{q}(\langle \boldsymbol{x}_i, \boldsymbol{\gamma} \rangle) \rangle - \log(\tilde{h}(\boldsymbol{y}_i)) \right]. \tag{4.7}$$

Table 4.2: GLM from Exponential Family

GLM	Exponential Family	$\boldsymbol{\mu}(\boldsymbol{\theta})$
Logistic Regression	Bernoulli	$1/(1+\exp(-\theta))$
Multinomial Regression	Multinomial	$\left(\left(\frac{\exp(\theta_i)}{1+\sum_{j=1}^{m-1}\exp(\theta_j)}\right)_{i=1,\ldots,m-1}, \frac{1}{1+\sum_{j=1}^{m-1}\exp(\theta_j)}\right)^T$
Poisson Regression	Poisson	$\exp(\theta)$

4.1.2 Gradient Descent and the Generalization

Although the main purpose of this draft is to discuss the usefulness of two algorithms in the exponential family, the first-order methods introduced in the following apply to more general objective functions other than negative log likelihood, and may be of independent interest. To avoid confusion, we rewrite the objective function as a finite-sum problem as follows

$$.F(\theta) = \frac{1}{n}\sum_{i=1}^{n} f_i(\theta) \tag{4.8}$$

When taking $f_i = l_i$ where l_i as in (4.3) the negative log density w.r.t. sample $\boldsymbol{z}_i$, we have the objective function become the negative log likelihood function.

A popular way to minimize (4.8) is Gradient Descent, the parameter is updated as:

$$\theta_{t+1} = \theta_t - \eta_t \nabla F(\theta_t), \tag{4.9}$$

where $(\eta_t)_{t=0}^{\infty}$ is the step size sequence. The Gradient Descent update (4.9) is in the steepest descent direction when the parameter θ is in the Euclidean space. However, in many applications, the space of parameters is non-Euclidean, and the GD direction is no longer the steepest. To make the updates more efficient in existence of the non-Euclidean structure, generalizations of GD have been developed, and we go over two of the main generalizations in the following subsections.

4.1.2.1 Natural Gradient Descent

The Natural Gradient Descent, developed by Amari [51], is the generalization of Gradient Descent when the parameter lies in a Riemannian manifold. It selects the steepest descent direction along the manifold and thus updates efficiently in the non-Euclidean space.

Consider a p-dimensional Riemannian manifold $(\mathcal{M}, \mathcal{H})$, where $\mathcal{M} \subset R^p$ is the parameter space, i.e., $\theta \in \mathcal{M}$. $\mathcal{H}(\theta)$ is a positive definite metric tensor that may depend on θ, and we denote $h_{ij}(\theta)$ be the ith row jth column of $\mathcal{H}(\theta)$. For a point $\theta \in \mathcal{M}$, the squared length of an infinite-small incremental $d\theta$ from θ to $\theta + d\theta$ along the Riemannian manifold is given by

$$|d\theta|^2 = \sum_{i,j=1}^{p} h_{ij}(\theta) d\theta_i d\theta_j. \tag{4.10}$$

By Theorem 1 in [51], the steepest descent direction of a function $F(\theta) : \mathcal{M} \to \mathcal{R}$ along the Riemannian manifold is given by:

$$-\tilde{\nabla} F(\theta) := -\mathcal{H}^{-1}(\theta) \nabla F(\theta). \tag{4.11}$$

Using this steepest descent direction, we have the Natural Gradient Descent updates as:

$$\theta_{t+1} = \theta_t - \eta_t \tilde{\nabla} F(\theta_t) = \theta_t - \eta_t \mathcal{H}^{-1}(\theta_t) \nabla F(\theta_t). \tag{4.12}$$

Note that for $(\mathcal{M}, \mathcal{H}) = (\mathcal{R}^p, I_{p \times p})$ the Riemannian manifold reduces to the Euclidean space and the Natural Gradient Descent reduces to GD.

The statistical model such as the Exponential family naturally forms a geometric manifold. When the parameter space of θ is geometrically regular (see definition in [52]), the Riemannian metric tensor of the parameter space is defined by the Fisher information matrix [53, 52]. This indicates that we can apply Natural Gradient Descent to optimize the statistical model in the steepest direction.

It is worth noting that, in practice a statistical model may be formulated such that the

parameter is not homeomorphic to a R^p space, thus is not geometrically regular. For example, when $\theta \in S^{p-1}$ the unit sphere in R^p or when θ is in other (essentially) low dimensional space such as the Swiss roll. Then in this case one should reparameterize θ into low dimension, then apply the Natural Gradient Descent algorithm.

4.1.2.2 Mirror Descent

Another popular generalization of Gradient descent is Mirror descent developed by Nemirovski and Yudin [54]. It takes into account the non-Euclidean structure of the parameter space by replacing ℓ_2 distance, which is a good measure of distance in Euclidean space, by some other proximity function that implies non-Euclidean geometry, with details as follows:

Note that the Gradient Descent update (4.9) can be rewritten as following:

$$\theta_{t+1} = \arg\min_{\theta} \left\{ \langle \theta, \nabla F(\theta_t) \rangle + \frac{1}{2\eta_t} \|\theta - \theta_t\|_2^2 \right\}, \tag{4.13}$$

which is an ℓ_2 penalized problem. Now in a non-Euclidean space, the distance between θ and θ_t maybe not properly measured by the ℓ_2 distance. To solve this issue we may replace the $\|\cdot\|_2^2$ term by some proximity function. A standard choice is the Bergman divergence D_ψ of a differentiable convex function ψ, resulting in proximal updates

$$\theta_{t+1} = \arg\min_{\theta} \left\{ \langle \theta, \nabla F(\theta_t) \rangle + \frac{1}{\eta_t} D_\psi(\theta, \theta_t) \right\}, \tag{4.14}$$

where the Bregman divergence:

$$D_\psi(\theta, \theta') = \psi(\theta) - \psi(\theta') - \langle \nabla\psi(\theta'), \theta - \theta' \rangle. \tag{4.15}$$

We note that the Bregman divergence corresponds to the Kullback-Leibler divergence between two distributions in an exponential family, thus may work well in dealing with the

non-Eudlidean structure in parameter estimation for an exponential family. For more details on the connection between Bregman divergence and exponential family, see [55, 56, 57].

Go back to the MD update (4.14), we observe that when ψ is strictly convex, $\nabla\psi$ is uniquely defined, and we can rewrite (4.14) as

$$\nabla\psi(\theta_{t+1}) = \nabla\psi(\theta_t) - \eta_t \nabla F(\theta_t). \tag{4.16}$$

Note that for $\psi = \|\cdot\|_2^2/2$, we can calculate $\nabla\psi(\theta) = \theta$, and $D(\theta, \theta') = \|\theta - \theta'\|_2^2/2$, in this case Mirror Descent reduces to Gradient Descent.

4.1.2.3 Connection between Natural Gradient Descent and Mirror Descent

Though seems to be different in updating formula, the Mirror Descent is shown to be equivalent to Natural Gradient Descent by [47, 58]. Concretely, the equivalency result is as follows:

Theorem 4.1 (Theorem 1 in [47], equivalence for any step size)**.** *The Mirror Descent step* (4.14) *with parameter space* Θ *is equivalent to the Natural Gradient Descent* (4.12) *step along the dual Riemannian manifold* $(\Theta^*, \nabla^2\phi)$*, where* Θ^* *is the dual space of* Θ *and* $\phi(\cdot) := \sup_\theta\{\langle\theta, \cdot\rangle - \psi(\theta)\}$ *is the dual function of* ψ*.*

Theorem 4.2 (Section 12.1.6 in [58], equivalence for infinitesimal step size)**.** *Consider the step size* $\eta \to 0$ *in* (4.14)*, then the Mirror Descent step is equivalent to the Natural Gradient Descent* (4.12) *step with metric tensor* $\nabla^2\psi$*.*

4.1.3 Stochastic Gradient Descent and Variance Reduction

The Gradient Descent and its generalizations in the previous section all require calculating

$$\nabla F(\theta_t) = \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\theta_t) \tag{4.17}$$

at each step, and the computation cost is $\mathcal{O}(n)$ times the cost for calculating each $\nabla f_i(\theta_t)$. When n is large, each update is expensive. To reduce the computation per step, the Stochastic Gradient Descent is widely used in place of Gradient Descent, where the update is:

$$\theta_{t+1} = \theta_t - \eta_t \nabla f_{i_t}(\theta_t), \tag{4.18}$$

where i_t is uniformly random sampled from $\{1, \ldots, n\}$. Note that each SGD step (4.18) performs an unbiased update compared to GD update (4.9) in the sense that $E[\nabla f_{i_t}(\theta_t)|\theta_t] = \nabla F(\theta_t)$, while the SGD step only requires $\mathcal{O}(1)$ calculation of ∇f_i.

Similar to the Stochastic Gradient Descent, we can write the Stochastic Natural Gradient Descent:

$$\theta_{t+1} = \theta_t - \eta_t \tilde{\nabla} f_{i_t}(\theta_t) = \theta_t - \eta_t \mathcal{H}^{-1}(\theta_t) \nabla f_{i_t}(\theta_t), \tag{4.19}$$

and the Stochastic Mirror Descent:

$$\nabla \psi(\theta_{t+1}) = \nabla \psi(\theta_t) - \eta_t \nabla f_{i_t}(\theta_t). \tag{4.20}$$

Despite its fast computation per step, the stochastic algorithm may converge slowly, which negates its overall performance. We now get an intuitive reason for the slow convergence. In general, $\theta^* = \arg\min F(\theta)$ does not simultaneously minimize all f_i's, and $Var(\nabla f_i(\theta)) \neq 0$ for all θ, where we consider randomness w.r.t. the index i. In this case, we need $\eta_t \to 0$ to make the stochastic algorithm converge, which slows the convergence.

The variance reduction is a useful technique to accelerate convergence for SGD [39]. We briefly introduce the idea of variance reduction here and leave the formal variance reduced stochastic natural gradient descent and variance reduced stochastic mirror descent algorithms to the next section. Now let us see how variance reduction works: for a fixed step size $\eta_t = \eta$, variance reduction method replaces $\nabla f_{i_t}(\boldsymbol{\theta}_t)$ by a term A_t such that

- $\mathbb{E}[A_t] = \mathbb{E}[\nabla f_{i_t}(\boldsymbol{\theta}_t)] = \nabla F(\theta_t)$ to keep unbiased update;

- $\mathrm{Var}[A_t] < \mathrm{Var}[\nabla f_{i_t}(\boldsymbol{\theta}_t)]$ to control variance.

To satisfy the conditions above, one method is to take $A_t = \nabla f_{i_t}(\boldsymbol{\theta}_t) - B_t + \mathbb{E}[B_t]$, where B_t and $\nabla f_{i_t}(\boldsymbol{\theta}_t)$ are positively correlated with correlation coefficient $r > 0.5$ and $\mathrm{Var}[B_t] \approx \mathrm{Var}[\nabla f_{i_t}(\boldsymbol{\theta}_t)]$. Using this A_t, we can check $\mathbb{E}[A_t] = \mathbb{E}[\nabla f_{i_t}(\boldsymbol{\theta}_t)] - \mathbb{E}[B_t] + \mathbb{E}[B_t] = \mathbb{E}[\nabla f_{i_t}(\boldsymbol{\theta}_t)]$ and $\mathrm{Var}[A_t] = \mathrm{Var}[\nabla f_{i_t}(\boldsymbol{\theta}_t) - B_t] = \mathrm{Var}[\nabla f_{i_t}(\boldsymbol{\theta}_t)] - 2r\sqrt{\mathrm{Var}[\nabla f_{i_t}(\boldsymbol{\theta}_t)]\,\mathrm{Var}[B_t]} + \mathrm{Var}[B_t] < \mathrm{Var}[\nabla f_{i_t}(\boldsymbol{\theta}_t)]$. Such replacement of A_t for $\nabla f_{i_t}(\boldsymbol{\theta}_t)$ can be done for stochastic mirror descent (4.20). For stochastic natural gradient descent (4.19), the idea of replacing $\tilde{\nabla} f_{i_t}(\boldsymbol{\theta}_t)$ for variance reduction is nearly identical by comparing the formula in (4.18) and (4.19), therefore, we omit the details.

4.2 Main Algorithm

In this section, we provide two variance reduced algorithms: the variance reduced stochastic natural gradient descent (VRSNGD) and variance reduced stochastic mirror descent (VRSMD). In these two algorithms, we use the variance reduction term similar to that in [39], but one should note that the variance reduction terms in SARAH [40] and SPIDER[41] also work.

The variance reduction components of VRSNGD and VRSMD are $\tilde{\mathbf{v}}_t^s$ in (4.21) and $\mathbf{v}_t^s$ in (4.23), respectively. Take $\mathbf{v}_t^s$ as an example, we can see it reduce variance using variance reduction scheme we talked above by taking $B_t = \nabla f_{i_t}(\tilde{\boldsymbol{\theta}}^{s-1})$. To satisfy the conditions we list there on B_t, we may expect that $\boldsymbol{\theta}_t^s$ and $\tilde{\boldsymbol{\theta}}^{s-1}$ are close, which happens if we take the number of inner iterations m and the step size η to be not too large such that the parameter estimator does not change too much within one inner loop. In this case, $\nabla f_{i_t}(\boldsymbol{\theta}_t^s)$ and $\nabla f_{i_t}(\tilde{\boldsymbol{\theta}}^{s-1})$ are highly positively correlated and have similar variance. This explains how the variance reduction in VRSMD works. For VRSNGD, the explanation is nearly identical and we omit the details.

Just as Gradient Descent is a special case of Natural Gradient Descent and Mirror Descent, the VRSNGD and VRSMD can reduce to the Gradient Descent version of variance

Algorithm 4.1: Variance Reduced Stochastic Natural Gradient Descent (VRSNGD)

Input: An objective function $F(\cdot) = \frac{1}{n}\sum_{i=1}^{n} f_i(\cdot)$, the metric tensor $\mathcal{H}(\theta)$ of the Riemannian manifold;

Initialization: Initialize $\tilde{\boldsymbol{\theta}}^0$. Choose the step size η, outer iteration number S, inner iteration number m. Denote the estimator at tth inner iteration of sth outer iteration as $\boldsymbol{\theta}_t^s$. Set $\boldsymbol{\theta}_1^1 = \boldsymbol{\theta}_{m+1}^0 = \tilde{\boldsymbol{\theta}}^0$;

for *Outer iteration s = 1,. . . ,S* **do**

Calculate $\tilde{\nabla} F(\tilde{\boldsymbol{\theta}}^{s-1}) = \mathcal{H}^{-1}(\tilde{\boldsymbol{\theta}}^{s-1})\nabla F(\tilde{\boldsymbol{\theta}}^{s-1})$;

for *Inner iteration t = 1,. . . ,m* **do**

Randomly sample i_t from $\{1, \ldots, n\}$, calculate

$$\begin{aligned}\tilde{\mathbf{v}}_t^s &= \tilde{\nabla} f_{i_t}(\boldsymbol{\theta}_t^s) - \tilde{\nabla} f_{i_t}(\tilde{\boldsymbol{\theta}}^{s-1}) + \tilde{\nabla} F(\tilde{\boldsymbol{\theta}}^{s-1}) \\ &= \mathcal{H}^{-1}(\boldsymbol{\theta}_t^s)\nabla f_{i_t}(\boldsymbol{\theta}_t^s) - \mathcal{H}^{-1}(\tilde{\boldsymbol{\theta}}^{s-1})\nabla f_{i_t}(\tilde{\boldsymbol{\theta}}^{s-1}) + \mathcal{H}^{-1}(\tilde{\boldsymbol{\theta}}^{s-1})\nabla F(\tilde{\boldsymbol{\theta}}^{s-1}),\end{aligned} \tag{4.21}$$

update $\boldsymbol{\theta}_{t+1}^s$ as

$$\boldsymbol{\theta}_{t+1}^s = \boldsymbol{\theta}_t^s - \eta\tilde{\mathbf{v}}_t^s. \tag{4.22}$$

Set $\tilde{\boldsymbol{\theta}}^s$ to be a uniform random sample from $\{\boldsymbol{\theta}_1^s, \ldots, \boldsymbol{\theta}_m^s\}$;

Option I: Set $\boldsymbol{\theta}_1^{s+1} = \boldsymbol{\theta}_{m+1}^s$;

Option II: Set $\boldsymbol{\theta}_1^{s+1} = \tilde{\boldsymbol{\theta}}^s$;

Option I: Output $\boldsymbol{\theta}_a$ chosen uniformly random from $\{\{\boldsymbol{\theta}_t^s\}_{t=1}^m\}_{s=1}^S$;

Option II: Output $\boldsymbol{\theta}_a = \tilde{\boldsymbol{\theta}}^S$.

reduced stochastic algorithm, namely the SVRG in [39], as follows. In VRSNGD, when the metric tensor is $I_{p\times p}$, we have $\tilde{v}_t^s = v_t^s$ for v_t^s defined in (4.23), and the algorithm is the same as SVRG. In VRSMD, for the choice of mirror map being $\psi(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, we have $\nabla\psi(\boldsymbol{\theta}) = \boldsymbol{\theta}$, then (4.24) updates $\boldsymbol{\theta}_{t+1}^s$ as:

$$\boldsymbol{\theta}_{t+1}^s = \boldsymbol{\theta}_t^s - \eta\mathbf{v}_t^s,$$

which is the SVRG update, and VRSMD reduces to SVRG.

Algorithm 4.2: Variance Reduced Stochastic Mirror Descent (VRSMD)

Input: An objective function $F(\cdot) = \frac{1}{n}\sum_{i=1}^{n} f_i(\cdot)$, and a strictly convex and differentiable mirror map $\psi(\cdot)$;

Initialization: Initialize $\tilde{\boldsymbol{\theta}}^0$. Choose the step size η, outer iteration number S, inner iteration number m. Denote the estimator at tth inner iteration of sth outer iteration as $\boldsymbol{\theta}_t^s$. Set $\boldsymbol{\theta}_1^1 = \boldsymbol{\theta}_{m+1}^0 = \tilde{\boldsymbol{\theta}}^0$;

for *Outer iteration s = 1,. . . ,S* **do**

Calculate $\nabla F(\tilde{\boldsymbol{\theta}}^{s-1})$;

for *Inner iteration t = 1,. . . ,m* **do**

Randomly sample i_t from $\{1, \ldots, n\}$, calculate

$$\mathbf{v}_t^s = \nabla f_{i_t}(\boldsymbol{\theta}_t^s) - \nabla f_{i_t}(\tilde{\boldsymbol{\theta}}^{s-1}) + \nabla F(\tilde{\boldsymbol{\theta}}^{s-1}), \tag{4.23}$$

and update $\boldsymbol{\theta}_{t+1}^s$ such that

$$\boldsymbol{\theta}_{t+1}^s = \arg\min_{\theta} \left\{ \langle \theta, \mathbf{v}_t^s \rangle + \frac{1}{\eta} D_\psi(\theta, \boldsymbol{\theta}_t^s) \right\}. \tag{4.24}$$

Set $\tilde{\boldsymbol{\theta}}^s$ to be a uniform random sample from $\{\boldsymbol{\theta}_1^s, \ldots, \boldsymbol{\theta}_m^s\}$;

Option I: Set $\boldsymbol{\theta}_1^{s+1} = \boldsymbol{\theta}_{m+1}^s$;

Option II: Set $\boldsymbol{\theta}_1^{s+1} = \tilde{\boldsymbol{\theta}}^s$;

Option I: Output $\boldsymbol{\theta}_a$ chosen uniformly random from $\{\{\boldsymbol{\theta}_t^s\}_{t=1}^m\}_{s=1}^S$;

Option II: Output $\boldsymbol{\theta}_a = \tilde{\boldsymbol{\theta}}^S$.

4.3 Theoretic Result

In this section, we analyze the properties of VRSNGD and VRSMD, especially for ERM objectives in the exponential family. Those properties could explain from an optimization perspective for why the algorithms are desired for parameter estimation in the exponential family. To begin with, we build the equivalence of the two algorithms, which helps understanding both algorithms.

4.3.1 VRSNGD and VRSMD are Equivalent

Similar to the equivalence of MD and ND, we have the equivalence of VRSNGD and VRSMD divided into two regimes: one for an arbitrary step size, in this case we show in Theorem 4.3 VRSMD is equivalent to VRSNGD in the dual space; another for an infinites-

imal step size, for which we show in Corollary 4.1 VRSMD is also equivalent to VRSNGD in the original space.

We first review the concept of convex conjugate function, which is the key to the main result. For a function $G(x) : \mathcal{X} \to R$, the convex conjugate on the dual space $\mathcal{X}^*$ is defined as:

$$G^*(x_*) := \sup_x \{\langle x, x_* \rangle - G(x)\}. \tag{4.25}$$

By Fenchel–Moreau theorem, for a convex and lower semi-continuous G, we have $(G^*)^* = G$. If we further assume that G is strictly convex and differentiable, denote $g = \nabla G$ and $g^* = \nabla(G^*)$, for x_* we have the supremum of (4.25) obtained at x such that $g(x) = x_*$, similarly, we have $g^*(x_*) = x$, that is, $g^{-1} = g^*$. We denote by $x_* = g(x)$ the dual coordinate system to x.

Theorem 4.3 (Equivalence of VRSMD and VRSNGD, arbitrary step size)**.** *Assume in the VRSMD algorithm 4.2 we have the objective function $F : \Theta \to R$ and a strictly convex twice differentiable mirror map ψ. Denote by Θ^* the dual space of Θ and by ϕ the convex conjugate of ψ, then Algorithm 4.2 is equivalent to VRSNGD algorithm 4.1 along the dual Riemannian manifold $(\Theta^*, \nabla^2\phi)$.*

Proof. For the VRSMD update (4.24), we can write it in the dual variable $\theta_* = \nabla\psi(\theta)$:

$$(\boldsymbol{\theta}_{t+1}^s)_* = (\boldsymbol{\theta}_t^s)_* - \eta \mathbf{v}_t^s, \tag{4.26}$$

where

$$\mathbf{v}_t^s = \nabla_\theta f_{i_t}(\boldsymbol{\theta}_t^s) - \nabla_\theta f_{i_t}(\tilde{\boldsymbol{\theta}}^{s-1}) + \nabla_\theta F(\tilde{\boldsymbol{\theta}}^{s-1}). \tag{4.27}$$

Now for $\nabla_\theta f_i$, we have by chain rule that:

$$\begin{aligned} &\nabla_{\theta_*} f_i(\nabla\phi(\theta_*)) = \nabla^2\phi(\theta_*)\nabla_\theta f_i(\nabla\phi(\theta_*)) \\ \Rightarrow &\nabla_\theta f_i(\nabla\phi(\theta_*)) = (\nabla^2\phi(\theta_*))^{-1}\nabla_{\theta_*} f_i(\nabla\phi(\theta_*)) = \tilde{\nabla}_{\theta_*} f_i(\nabla\phi(\theta_*)). \end{aligned}$$

Similarly, we have for $\nabla_\theta F$:

$$\nabla_\theta F(\nabla\phi(\theta_*)) = \tilde{\nabla}_{\theta_*} F(\nabla\phi(\theta_*)).$$

I.e., we can rewrite $\mathbf{v}_t^s$ as

$$\mathbf{v}_t^s = \tilde{\nabla}_{\theta_*} f_{i_t}(\nabla\phi(\boldsymbol{\theta}_{t*}^s)) - \tilde{\nabla}_{\theta_*} f_{i_t}(\nabla\phi(\tilde{\boldsymbol{\theta}}_*^{s-1})) + \tilde{\nabla}_{\theta_*} F(\nabla\phi(\tilde{\boldsymbol{\theta}}_*^{s-1})). \tag{4.28}$$

The updating scheme by (4.26) and (4.28) is VRSNGD algorithm 4.1 in the dual Riemannian manifold $(\Theta^*, \nabla^2\phi)$. Or equivalently, rewrite $g_i(\cdot) := f_i(\nabla\phi(\cdot))$, then $G = \frac{1}{n}\sum_{i=1}^n g_i$ is defined on Θ^*. One can consider the VRSMD on F as a VRSNGD algorithm for minimizing $G(\cdot) = \frac{1}{n}\sum_{i=1}^n g_i(\cdot)$ along the dual Riemannian manifold $(\Theta^*, \nabla^2\phi)$. □

Corollary 4.1 (Equivalence of VRSMD and VRSNGD, infinitesimal step size)**.** *Assume in the VRSMD algorithm 4.2 we have a strongly convex twice differentiable mirror map ψ. Consider the step size $\eta \to 0$, then Algorithm 4.2 is equivalent to VRSNGD algorithm 4.1 where the Riemannian metric is $\nabla^2\psi$.*

Proof. We rewrite the VRSMD update (4.24) as follows to simplify notation:

$$\nabla\psi(\boldsymbol{\theta}_{t+1}) = \nabla\psi(\boldsymbol{\theta}_t) - \eta\mathbf{v}_t, \tag{4.29}$$

where

$$\mathbf{v}_t = \nabla_\theta f_i(\boldsymbol{\theta}_t) - \nabla_\theta f_i(\tilde{\boldsymbol{\theta}}) + \nabla_\theta F(\tilde{\boldsymbol{\theta}}). \tag{4.30}$$

Since ψ is strongly convex, $\nabla\psi$ is invertible, and one has

$$\begin{aligned}
\boldsymbol{\theta}_{t+1} &= (\nabla\psi)^{-1}(\nabla\psi(\boldsymbol{\theta}_t) - \eta\mathbf{v}_t) \\
&= (\nabla\psi)^{-1}(\nabla\psi(\boldsymbol{\theta}_t)) - \eta\nabla((\nabla\psi)^{-1}(\boldsymbol{\theta}_t))\mathbf{v}_t + \mathcal{O}(\eta^2) \\
&= \boldsymbol{\theta}_t - \eta(\nabla^2\psi(\boldsymbol{\theta}_t))^{-1}\mathbf{v}_t + \mathcal{O}(\eta^2)
\end{aligned}$$

$$
\begin{aligned}
&= \boldsymbol{\theta}_t - \eta(\nabla^2\psi(\boldsymbol{\theta}_t))^{-1}(\nabla_\theta f_i(\boldsymbol{\theta}_t) - \nabla_\theta f_i(\tilde{\boldsymbol{\theta}}) + \nabla_\theta F(\tilde{\boldsymbol{\theta}})) + \mathcal{O}(\eta^2) \\
&= \boldsymbol{\theta}_t - \eta(\nabla^2\psi(\boldsymbol{\theta}_t)^{-1}\nabla_\theta f_i(\boldsymbol{\theta}_t) - \nabla^2\psi(\tilde{\boldsymbol{\theta}})^{-1}\nabla_\theta f_i(\tilde{\boldsymbol{\theta}}) + \nabla^2\psi(\tilde{\boldsymbol{\theta}})^{-1}\nabla_\theta F(\tilde{\boldsymbol{\theta}})) \\
&\quad - \eta(\nabla^2\psi(\boldsymbol{\theta}_t)^{-1} - \nabla^2\psi(\tilde{\boldsymbol{\theta}})^{-1})(\nabla_\theta F(\tilde{\boldsymbol{\theta}}) - \nabla_\theta f_i(\tilde{\boldsymbol{\theta}})) + \mathcal{O}(\eta^2) \\
&= \boldsymbol{\theta}_t - \eta(\nabla^2\psi(\boldsymbol{\theta}_t)^{-1}\nabla_\theta f_i(\boldsymbol{\theta}_t) - \nabla^2\psi(\tilde{\boldsymbol{\theta}})^{-1}\nabla_\theta f_i(\tilde{\boldsymbol{\theta}}) + \nabla^2\psi(\tilde{\boldsymbol{\theta}})^{-1}\nabla_\theta F(\tilde{\boldsymbol{\theta}})) + \mathcal{O}(\eta^2) \\
&\overset{\eta\to 0}{=} \boldsymbol{\theta}_t - \eta(\nabla^2\psi(\boldsymbol{\theta}_t)^{-1}\nabla_\theta f_i(\boldsymbol{\theta}_t) - \nabla^2\psi(\tilde{\boldsymbol{\theta}})^{-1}\nabla_\theta f_i(\tilde{\boldsymbol{\theta}}) + \nabla^2\psi(\tilde{\boldsymbol{\theta}})^{-1}\nabla_\theta F(\tilde{\boldsymbol{\theta}})),
\end{aligned}
$$

which is the VRSNGD update with metric tensor $\nabla^2\psi(\cdot)$. □

4.3.2 Consequence of equivalence

In this section, we discuss some consequences and implications of the equivalence result. The implications are bidirectional: on the one hand, in Subsection 4.3.2.1, we use the convergence rate of VRSMD to induce the convergence rate of VRSNGD; on the other hand, in Subsection 4.3.2.2, we use parameterization invariance of VRSNGD to derive the condition for VRSMD to be parameterization invariant.

4.3.2.1 Convergence Rate of VRSNGD

In Chapter 3, we show that VRSMD converges linearly under suitable conditions, where the conditions on the objective function and mirror map are easily quantified. Using the equivalence result, we can translate the conditions to VRSNGD and show its linear convergence, which may be not that easy by directly studying VRSNGD.

Before going to the result, we review some necessary concepts.

Definition 4.1 (L-smoothness)**.** *f is $L-$smooth if there exists a constant $L > 0$ such that*

$$\|\nabla f(\mathbf{u}) - \nabla f(\mathbf{w})\|_2 \leq L\|\mathbf{u} - \mathbf{w}\|_2, \forall \mathbf{u}, \mathbf{w}.$$

Definition 4.2 ($\alpha-$strongly convex)**.** *f is $\alpha-$strongly convex if there exists a constant $\alpha >$*

0 *such that*

$$f(\mathbf{u}) \geq f(\mathbf{w}) + \nabla f(\mathbf{w})^T(\mathbf{u}-\mathbf{w}) + \frac{\alpha}{2}\|\mathbf{u}-\mathbf{w}\|_2^2, \forall \mathbf{u}, \mathbf{w}.$$

Definition 4.3 (PL inequality)**.** *f satisfies the Polyak-Łojasiewicz (PL) inequality if there exists $\mu > 0$ such that for $\mathbf{u}^* \in \arg\min f(\mathbf{u})$*

$$\frac{1}{2}\|\nabla f(\mathbf{u})\|_2^2 \geq \mu[f(\mathbf{u}) - f(\mathbf{u}^*)], \forall \mathbf{u}. \tag{4.31}$$

We recap the convergence rate of VRSMD in the following lemma.

Lemma 4.1 (Convergence rate of VRSMD)**.** *Assume $F(\cdot) = \frac{1}{n}\sum_{i=1}^n f_i(\cdot)$ has every $f_i(\cdot)$ convex and $L-$smooth, and denote $\hat{\boldsymbol{\theta}}$ a minimizer of $F(\cdot)$; suppose $\psi(\cdot)$ is $\alpha-$strongly convex.*

***(a)** Run Algorithm 4.2 using Option I on $F(\cdot)$ with $\eta < \frac{\alpha}{8L}$, the output $\boldsymbol{\theta}_a$ will have:*

$$\mathbb{E}[F(\boldsymbol{\theta}_a) - F(\hat{\boldsymbol{\theta}})] \leq \frac{\alpha}{(\alpha\eta - 8L\eta^2)T}\left[D_\psi(\hat{\boldsymbol{\theta}}, \tilde{\boldsymbol{\theta}}^0) + \frac{4L\eta^2 m}{\alpha}(F(\tilde{\boldsymbol{\theta}}^0) - F(\hat{\boldsymbol{\theta}}))\right], \tag{4.32}$$

where $T = m \cdot S$, and D_ψ is the Bregman divergence.

***(b)** If we further assume that $F(\cdot)$ satisfies the Polyak-Łojasiewicz (PL) inequality in* (4.31) *with constant μ, and that $\psi(\cdot)$ is $\ell-$smooth, run Algorithm 4.2 using Option II and further assume that m is large enough such that*

$$\tau = \frac{4L\eta^2/\alpha + \ell/(m\mu)}{\eta - 4L\eta^2/\alpha} < 1, \tag{4.33}$$

we can prove a stronger linear convergence rate:

$$\mathbb{E}[F(\boldsymbol{\theta}_a) - F(\hat{\boldsymbol{\theta}})] \leq \tau^S[F(\tilde{\boldsymbol{\theta}}^0) - F(\hat{\boldsymbol{\theta}})]. \tag{4.34}$$

As a consequence, we have the convergence rate of VRSNGD, but before that we list two key identities that will be useful. First, for dual Bregman divergence D_{G^*} we have its

relationship with the primal D_G as the following lemma.

Lemma 4.2 (Duality of Bregman divergence)**.** *Assume G is strictly convex and differentiable. For the dual Bregman divergence*

$$D_{G^*}(x_*, x'_*) = G^*(x_*) - G^*(x'_*) - \langle \nabla G^*(x'_*), x_* - x'_* \rangle, \tag{4.35}$$

we have

$$D_{G^*}(x_*, x'_*) = D_G(g^*(x'_*), g^*(x_*)), \tag{4.36}$$

$$D_G(x, x') = D_{G^*}(g(x'), g(x)). \tag{4.37}$$

Proof. Notice that

$$G^*(x_*) = \langle x, g(x) \rangle - G(x). \tag{4.38}$$

We can check (4.36):

$$\begin{aligned}
& D_{G^*}(x_*, x'_*) \\
=& G^*(x_*) - G^*(x'_*) - \langle g^*(x'_*), x_* - x'_* \rangle \\
\overset{(4.38)}{=}& \langle x, g(x) \rangle - G(x) - \langle x', g(x') \rangle + G(x') - \langle x', g(x) - g(x') \rangle \\
=& G(x') - G(x) - \langle x' - x, g(x) \rangle \\
=& D_G(x', x) = D_G(g^*(x'_*), g^*(x_*)).
\end{aligned}$$

(4.37) can be checked in the similar way. □

Next, the curvature of G and its conjugate G^* are related as follows.

Lemma 4.3 (Fenchel duality for strong convexity and smoothness [59])**.** *For a closed $\alpha-$strongly convex G, the conjugate G^* is $1/\alpha-$smooth; and for a convex and $L-$smooth G, the conjugate G^* is $1/L-$strongly convex.*

Corollary 4.2 (Convergence rate of VRSNGD along dual Riemmannian manifold)**.** *Under the same set of assumptions as Lemma 4.1, denote ϕ the convex conjugate of ψ and the dual coordinate $\theta = \nabla\phi(\theta^*)$, run Algorithm 4.1 along the dual Riemannian manifold $(\Theta^*, \nabla^2\phi)$ we have:*

***(a)** Use Option I with $\eta < \frac{\alpha}{8L}$, the output $\boldsymbol{\theta}_a^*$ will have:*

$$\begin{aligned}&\mathbb{E}[F(\nabla\phi(\boldsymbol{\theta}_a^*)) - F(\nabla\phi(\hat{\boldsymbol{\theta}}^*))] \\ \leq&\frac{\alpha}{(\alpha\eta - 8L\eta^2)T}\left[D_\phi(\tilde{\boldsymbol{\theta}}^{0*}, \hat{\boldsymbol{\theta}}^*) + \frac{4L\eta^2 m}{\alpha}(F(\nabla\phi(\tilde{\boldsymbol{\theta}}^{0*})) - F(\nabla\phi(\hat{\boldsymbol{\theta}}^*)))\right],\end{aligned} \tag{4.39}$$

where $T = m \cdot S$.

***(b)** Assume that $F(\cdot)$ satisfies the Polyak-Łojasiewicz (PL) inequality with constant μ, and that $\psi(\cdot)$ is $\ell-$smooth, use Option II and further assume that m is large enough such that*

$$\tau = \frac{4L\eta^2/\alpha + \ell/(m\mu)}{\eta - 4L\eta^2/\alpha} < 1, \tag{4.40}$$

then we have a linear convergence rate:

$$\mathbb{E}[F(\nabla\phi(\boldsymbol{\theta}_a^*)) - F(\nabla\phi(\hat{\boldsymbol{\theta}}^*))] \leq \tau^S[F(\nabla\phi(\tilde{\boldsymbol{\theta}}^{0*})) - F(\nabla\phi(\hat{\boldsymbol{\theta}}^*))]. \tag{4.41}$$

Proof. The corollary holds by the equivalence result in Theorem 4.3, after plugging the dual coordinate and Bregman divergence duality in Lemma 4.2 into Lemma 4.1. □

We can rewrite the conditions of Corollary 4.2 in terms of the dual coordinate to get a simplified form as follows.

Corollary 4.3 (Convergence rate of VRSNGD in a different form)**.** *Consider the objective function $G(\cdot) = \frac{1}{n}\sum_{i=1}^n g_i(\cdot)$, given a strictly convex and twice differentiable function ϕ, run Algorithm 4.1 on G where the metric tensor given by $\nabla^2\phi$ (which is a valid positive definite metric tensor for Riemannian manifold by strict convexity of ϕ). Denote by ψ the convex conjugate of ϕ, assume $g_i(\nabla\psi(\cdot))$ are convex and $L-$smooth and suppose $\phi(\cdot)$ is*

$1/\alpha-$*smooth. Denote* $\hat{\boldsymbol{\theta}}^*$ *a minimizer of* $G(\cdot)$*, we have the following*

(a) *Using Option I with* $\eta < \frac{\alpha}{8L}$*, the output* $\boldsymbol{\theta}_a^*$ *will have:*

$$\mathbb{E}[G(\boldsymbol{\theta}_a^*) - G(\hat{\boldsymbol{\theta}}^*)] \leq \frac{\alpha}{(\alpha\eta - 8L\eta^2)T}\left[D_\phi(\tilde{\boldsymbol{\theta}}^{0*}, \hat{\boldsymbol{\theta}}^*) + \frac{4L\eta^2 m}{\alpha}(G(\tilde{\boldsymbol{\theta}}^{0*}) - G(\hat{\boldsymbol{\theta}}^*))\right], \tag{4.42}$$

where $T = m \cdot S$*.*

(b) *If we further assume that* $G(\nabla\psi(\cdot))$ *satisfies the Polyak-Łojasiewicz (PL) inequality in* (4.31) *with constant* μ*, and that* $\phi(\cdot)$ *is* $1/\ell-$*strongly convex, run Algorithm 4.1 using Option II and further assume that* m *is large enough such that*

$$\tau = \frac{4L\eta^2/\alpha + \ell/(m\mu)}{\eta - 4L\eta^2/\alpha} < 1, \tag{4.43}$$

we have a linear convergence rate:

$$\mathbb{E}[G(\boldsymbol{\theta}_a^*) - G(\hat{\boldsymbol{\theta}}^*)] \leq \tau^S[G(\tilde{\boldsymbol{\theta}}^{0*}) - G(\hat{\boldsymbol{\theta}}^*)]. \tag{4.44}$$

Proof. Denote $f_i(\cdot) = g_i(\nabla\psi(\cdot))$ and $F(\cdot) = G(\nabla\psi(\cdot))$, the dual coordinate implies that $f_i(\nabla\phi(\cdot)) = g_i(\cdot)$ and $F(\nabla\phi(\cdot)) = G(\cdot)$. Thus the VRSNGD algorithm on $G(\theta^*)$ is equivalent to VRSNGD on $F(\theta)$ along the dual Riemannian manifold $\theta = \nabla\phi(\theta^*)$.

By corollary assumption, f_is and F satisfy all assumptions in Corollary 4.2. By Lemma 4.3, ϕ is $1/\alpha-$smooth implies that ψ is $\alpha-$strongly convex, and the $1/\ell$ strong convexity of ϕ implies $\ell-$smoothness of ψ, all assumptions of Corollary 4.2 hold. Rewriting Corollary 4.2 we get the corollary claims. □

4.3.2.2 Parameterzation invariance of VRSMD

Parameterization invariance property is one of the main selling property of Natural Gradient Descent. It is not hard to conjecture that the same property holds for VRSNGD. In

this subsection, we first extent the parameterization invariance result of Natural Gradient to VRSNGD. We then show that the VRSMD for the exponential family is also parameterization invariant by its equivalence to a VRSNGD.

Proposition 4.1 (Parameterization Invariance of VRSNGD)**.** *Suppose we have a change of variable $\theta = \zeta(\gamma)$ where ζ is a smooth invertible function with upper bounded Hessian norm, denote by $J_\zeta(\gamma)$ be the Jacobian of ζ. Suppose we have*

$$J_\zeta^T \mathcal{H}_\theta J_\zeta = \mathcal{H}_\gamma, \forall \theta, \tag{4.45}$$

then the VRSNGD updates in θ using metric tensor $\mathcal{H}_\theta$ is approximately equivalent to the update in γ using metric tensor $\mathcal{H}_\gamma$. Furthermore, for an infinitesimal step size η and a fixed inner iteration number m, the smooth path of updating in θ and γ are equivalent.

Proof. For ease of notation, we can express one-step update of θ as following without confusion:

$$\begin{aligned}
\theta &\leftarrow \theta + d\theta \\
d\theta &= -\eta[\mathcal{H}_\theta^{-1}\nabla f_i(\theta) - \mathcal{H}_{\tilde{\theta}}^{-1}\nabla f_i(\tilde{\theta}) + \mathcal{H}_{\tilde{\theta}}^{-1}\nabla F(\tilde{\theta})],
\end{aligned}$$

and update of γ:

$$\begin{aligned}
\gamma &\leftarrow \gamma + d\gamma \\
d\gamma &= -\eta[\mathcal{H}_\gamma^{-1}\nabla_\gamma f_i(\zeta(\gamma)) - \mathcal{H}_{\tilde{\gamma}}^{-1}\nabla_\gamma f_i(\zeta(\tilde{\gamma})) + \mathcal{H}_{\tilde{\gamma}}^{-1}\nabla_\gamma F(\zeta(\tilde{\gamma}))].
\end{aligned}$$

We can check

$$\begin{aligned}
&\zeta(\gamma + d\gamma) \\
=&\zeta(\gamma) + J_\zeta(\gamma)d\gamma + \mathcal{O}(\|d\gamma\|_2^2) \text{ (Since the Hessian of } \zeta \text{ is upper bounded)}
\end{aligned}$$

$$
\begin{aligned}
=&\zeta(\gamma) - \eta J_\zeta(\gamma)[\mathcal{H}_\gamma^{-1}\nabla_\gamma f_i(\zeta(\gamma)) - \mathcal{H}_{\tilde{\gamma}}^{-1}\nabla_\gamma f_i(\zeta(\tilde{\gamma})) + \mathcal{H}_{\tilde{\gamma}}^{-1}\nabla_\gamma F(\zeta(\tilde{\gamma}))] + \mathcal{O}(\|d\gamma\|_2^2)\\
=&\zeta(\gamma) - \eta[J_\zeta(\gamma)\mathcal{H}_\gamma^{-1}J_\zeta(\gamma)^T\nabla f_i(\theta) + J_\zeta(\gamma)\mathcal{H}_{\tilde{\gamma}}^{-1}J_\zeta(\tilde{\gamma})(\nabla F(\tilde{\theta}) - \nabla f_i(\tilde{\theta}))] + \mathcal{O}(\|d\gamma\|_2^2)\\
\overset{(4.45)}{=}&\zeta(\gamma) - \eta[\mathcal{H}_\theta^{-1}\nabla f_i(\theta) + \mathcal{H}_{\tilde{\theta}}^{-1}(\nabla F(\tilde{\theta}) - \nabla f_i(\tilde{\theta}))\\
&\quad + (J_\zeta(\gamma) - J_\zeta(\tilde{\gamma}))\mathcal{H}_{\tilde{\gamma}}^{-1}J_\zeta(\tilde{\gamma})(\nabla F(\tilde{\theta}) - \nabla f_i(\tilde{\theta}))] + \mathcal{O}(\|d\gamma\|_2^2)\\
=&\theta + d\theta + \mathcal{O}(\eta m\|d\gamma\|_2 + \|d\gamma\|_2^2) = \theta + d\theta + \mathcal{O}(\eta^2).
\end{aligned}
$$

That is, $\zeta(\gamma+d\gamma) \approx \theta+d\theta$. For the infinitesimal step size η, we can ignore the second order term in η, i.e., $\mathcal{O}(\eta^2)$, and conclude that the solution path in θ and γ are equivalent. □

Remark 4.1. *Note that we require the step size η to be infinitesimal to get the exactly equivalent path result. When the step size is small, we can only show that the trajectories under different parameterizations are close. This is reasonable, as in the original work of [60] the exact parameterization invariance of Natural Gradient Descent is subject to infinitesimal step size.*

Remark 4.2. *We also require a specific formulation of the Riemannian manifold as in* (4.45)*, which is satisfied by Fisher Information metric for a statistic model [60].*

Next, we show that when using a proper mirror function, the VRSMD is equivalent to VRSNGD where the metric tensor is Fisher Information. This indicates that if we adopt a corresponding mirror map after reparameterization, the solution path of VRSMD is essentially invariant.

Corollary 4.4 (Parameterization Invariance of VRSMD)**.** *Consider the exponential family in natural parameters where the density is given by*

$$
p(y|\theta) = \tilde{h}(y)\exp(\langle\theta, y\rangle - G(\theta)), \tag{4.46}
$$

where G is a strictly convex differentiable function. Suppose we have observations $\{y_1, \ldots, y_n\}$ from the exponential family and want to estimate θ via maximum likelihood estimation, or

equivalently, minimizing the negative log-likelihood function

$$l(\theta) = -\sum_{i=1}^{n}[\log \tilde{h}(y_i) + \langle \theta, y_i \rangle - G(\theta)]. \tag{4.47}$$

Consider the VRSMD Algorithm 4.2 on (4.47) *using mirror map $G(\cdot)$, where the step size $\eta \to 0$ and the inner loop size m is a fixed constant, we have the VRSMD equivalent to VRSNGD where tensor metric is the Fisher Information in θ.*

Furthermore, the VRSMD is invariant to reparameterization as follows: consider reparameterization γ such that $g(\gamma) = \theta$, where g is an invertible function with invertible Jacobian ∇g. Denote the inverse of g as g^{-1}. Then run VRSMD in γ with mirror map ψ such that

$$\nabla^2 \psi(\gamma) = \nabla g(\gamma)^T \nabla^2_\theta G(\gamma) \nabla g(\gamma) \tag{4.48}$$

is equivalent to run VRSMD in θ with mirror map G.

Proof. Denote by $\mathcal{I}_\theta$ the Fisher Information in θ. For first part of the theorem, by Corollary 4.2 we need to check

$$\nabla^2 G(\theta) = \mathcal{I}_\theta. \tag{4.49}$$

We can verify (4.49) as follows:

$$\begin{aligned} \mathcal{I}_\theta &= -E\left[\frac{\partial^2}{\partial \theta^2} \log p(y|\theta) \middle| \theta\right] \\ &= -E\left[\frac{\partial^2}{\partial \theta^2}[\langle \theta, y \rangle - G(\theta)] \middle| \theta\right] \\ &= -E\left[\frac{\partial}{\partial \theta}[y - \nabla G(\theta)] \middle| \theta\right] \\ &= \nabla^2 G(\theta). \end{aligned} \tag{4.50}$$

For second part of the theorem, we need to show that the claimed VRSMD in γ is also equivalent to a VRSNGD where the tensor metric is the Fisher information. Once the above is checked, the parameterization invariance property automatically holds by Proposition

4.1. Again by Corollary 4.1, the goal is to show

$$\nabla^2\psi(\gamma) = \mathcal{I}_\gamma. \tag{4.51}$$

Check

$$\begin{aligned}
\mathcal{I}_\gamma &= -E\left[\frac{\partial^2}{\partial\gamma^2}\log p(y|\gamma)\middle|\gamma\right] \\
&= -E\left[\frac{\partial^2}{\partial\gamma^2}[\langle g(\gamma), y\rangle - G(g(\gamma))]\middle|\gamma\right] \\
&= -E\left[\frac{\partial}{\partial\gamma}[\nabla g(\gamma)^T y - \nabla g(\gamma)^T \nabla_\theta G(g(\gamma))]\middle|\gamma\right] \\
&= -E\left[\langle \nabla^2 g(\gamma), y - \nabla_\theta G(g(\gamma))\rangle - \nabla g(\gamma)^T \nabla_\theta^2 G(g(\gamma))\nabla g(\gamma)\right] \\
&= \nabla g(\gamma)^T \nabla_\theta^2 G(g(\gamma))\nabla g(\gamma) \overset{(4.48)}{=} \nabla^2\psi(\gamma).
\end{aligned}$$

So the parameterization invariant of VRSMD holds. □

Remark 4.3. *Similarly, for GLM model in* (4.7) *where* $y_i \in \mathbb{R}$*, treat* y_i *as a random variable* Y_i *in the likelihood, one can check the Fisher Information in* $\boldsymbol{\gamma}$

$$\begin{aligned}
\mathcal{I}_{\boldsymbol{\gamma}} &= \mathrm{Var}(\frac{\partial\ell(\boldsymbol{\gamma})}{\partial\boldsymbol{\gamma}}|\boldsymbol{\gamma}) \\
&= \mathrm{Var}(\sum_{i=1}^n \left[(y_i - G'(q(\boldsymbol{x}_i^T\boldsymbol{\gamma})))q'(\boldsymbol{x}_i^T\boldsymbol{\gamma})\boldsymbol{x}_i\right]|\boldsymbol{\gamma}) \\
&= \sum_{i=1}^n \left[G''(q(\boldsymbol{x}_i^T\boldsymbol{\gamma}))(q'(\boldsymbol{x}_i^T\boldsymbol{\gamma}))^2\boldsymbol{x}_i\boldsymbol{x}_i^T\right].
\end{aligned}$$

Thus for the GLM with link function g*, the VRSMD using mirror map* ψ *such that*

$$\nabla^2\psi = \mathcal{I}_{\boldsymbol{\gamma}} = \sum_{i=1}^n G''(q(\boldsymbol{x}_i^T\boldsymbol{\gamma}))q'(\boldsymbol{x}_i^T\boldsymbol{\gamma})^2\boldsymbol{x}_i\boldsymbol{x}_i^T \tag{4.52}$$

is equivalent to VRSNGD with Fisher information metric, where we note that ψ *is implicitly determined by* g *through* $q = (g\circ\mu)^{-1}$*. When subject to this choice of mirror function, the VRSMD has its solution path essentially invariant to a smooth invertible change in the link*

function. We note two link functions such that the mirror function in (4.52) *has closed form solution:*

- *The link function such that* $G''(q(\cdot))q'(\cdot)^2 \equiv 1$*, then* $\psi(\boldsymbol{\gamma}) = \sum_{i=1}^n (\boldsymbol{\gamma}^T \boldsymbol{x}_i)^2/2$*. Note that for this choice of link function, the asymptotic variance of* $\hat{\boldsymbol{\gamma}}_{MLE}$ *is* $(\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^T)^{-1}$*. This constant variance is appealing;*
- *The canonical link function such that* q *is identity map, then* $\psi(\boldsymbol{\gamma}) = \sum_{i=1}^n G(\boldsymbol{\gamma}^T \boldsymbol{x}_i)$*.*

4.4 Simulation Study

In this section, we run simulation study on Poisson regression. There are two goals: first, show the computation advantage of VRSMD algorithm compared with the Natural Gradient Descent (also called Iteratively Reweighted Least Squares (IRLS) method) that is a standard method for GLM; second, check the invariant property of VRSMD to the change of link function as mentioned in Note 4.3 for Corollary 4.4.

Our data simulation is as follows: For sample size n and parameter size p, we simulate the design matrix $X \in R^{n \times p}$ where all elements of X are i.i.d. $N(0,1)$, except the first column, which is vector of 1s. We simulate $\boldsymbol{\gamma}^*$ from $N(0, I_p)$, and calculate $\boldsymbol{\theta} = X\boldsymbol{\gamma}^*$. Finally, for $i = 1, \ldots, n$, we simulate y_i from $Poisson(\exp(\theta_i))$.

The VRSMD and NGD(IRLS) are compared for minimizing the negative likelihood objective function with canonical link function for Poisson GLM, which is

$$\text{argmin}_{\boldsymbol{\gamma}} \frac{1}{n} \sum_{i=1}^n \left[\exp(\langle \boldsymbol{x}_i, \boldsymbol{\gamma} \rangle) - y_i \langle \boldsymbol{x}_i, \boldsymbol{\gamma} \rangle \right]. \tag{4.53}$$

First, we compare VRSMD and IRLS for underparameterized case (n = 400, p = 10) and overparamterized case (n = 10, p = 15), respectively. We run VRSMD and IRLS using the same initialization $\boldsymbol{\gamma} = \mathbf{1}_p$. The result is in Figure 4.1. In first plot we compare the algorithms for underparameterized case. The VRSMD is run for 100 outer loops times 100 inner loops with step size $5 * 10^{-6}$, where the mirror function in VRSMD is set to $\|\cdot\|_2^2/2$.

For IRLS we use the closed form update as $\gamma_{t+1} = \gamma_t - I(\gamma_t)^{-1}\nabla F(\gamma_t)$. The IRLS is updated 100 times. One can see that IRLS converges faster than VRSMD, which is not surprising by the IRLS nature. However the IRLS takes about 0.09 seconds for 100 updates, while VRSMD takes about 0.42 seconds for 10000 updates, thus the per step computation of VRSMD is faster. For the overparameterized case, IRLS could not be applied since the Fisher information matrix is not invertible, while the VRSMD still converges to an optimal value as shown by the second plot in Figure 4.1. This illustrats the computation advantage of VRSMD over IRLS.

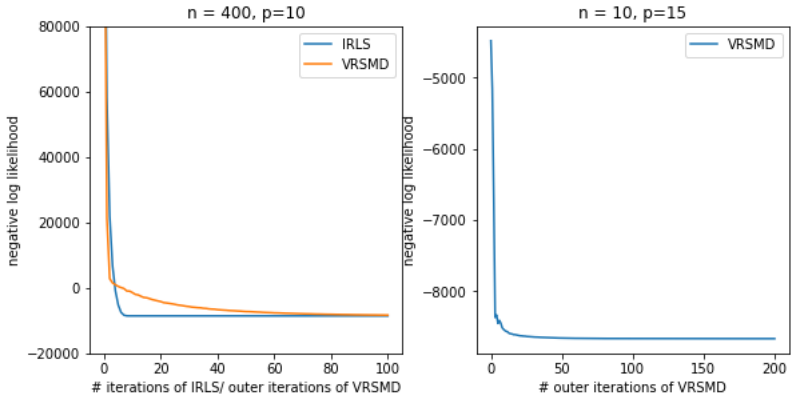


Figure 4.1: Compare VRSMD and IRLS

Next, we check the equivalency of VRSMD and VRSNGD. The metric tensor in VRSNGD is set to be the Fisher information and step size is chosen to be $5 * 10^{-5}$. In first plot the mirror map in VRSMD is chosen according to Corollary 4.1 and the step size the same as VRSNGD; in second plot the mirror map in VRSMD is set to be $\|\cdot\|_2^2/2$ and step size is set to $2 * 10^{-7}$ to make the convergence speed comparable. We are not able to show the exact parameter vector sequences in a plot since it is high dimension, so we observe the path of the negative log likelihood instead. We conclude from first plot of Figure 4.2 that the likelihood path of VRSMD and VRSNGD are almost equivalent to each other when Corollary 4.1 assumptions are satisfied, which indicates that the solution paths should also be equivalent. On the contrary, the second plot in Figure 4.2 indicates that VRSMD with

an arbitrary mirror map is not equivalent to VRSNGD, no matter what step size we choose since the shape of the likelihood paths are very different.

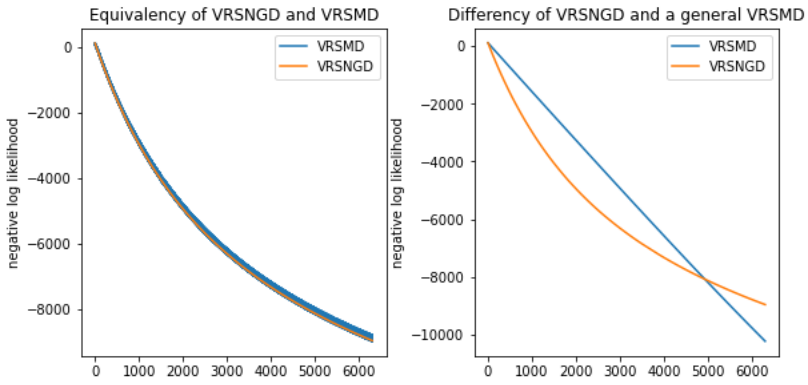


Figure 4.2: Equivalency of VRSMD and VRSNGD

Finally, we check that VRSMD is invariant to the link function by comparing the convergence of the likelihood function. We consider two link functions: canonical link function and the link function such that $G''(q(\cdot))q'(\cdot)^2 \equiv 1$. That is, $q(\cdot) = 2\log(\cdot/2)$. We call the link function for this case by special link function. The results are listed in Figure 4.3. We can see that when VRSMD has mirror map adjusted to link function, the convergence in the negative log likelihood and thus the solution path are closer than when VRSMD just use a fixed mirror map regardless of the link function. Note that even we adjust the mirror map according to link function, the solution path are not exactly the same since we cannot take step size to be infinitesimal in practice.

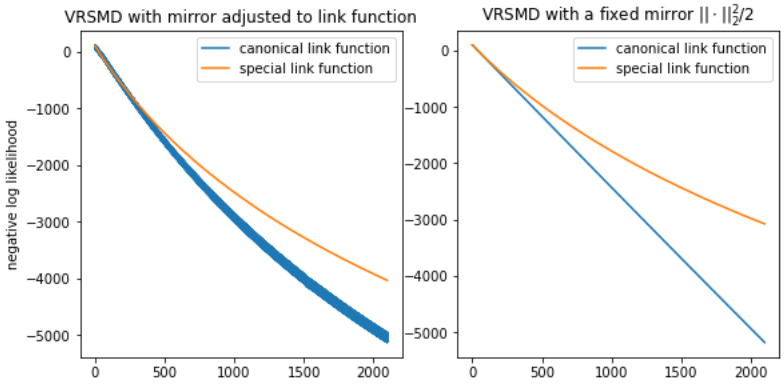


Figure 4.3: Change the Link Function

CHAPTER 5

COVARIANCE ESTIMATORS FOR THE ROOT-SGD ALGORITHM IN ONLINE LEARNING

Online learning naturally arises in a lot of statistical and machine learning problems. The most wildly used family of methods in online learning is the stochastic first-order algorithm. Among this family of algorithms, there is a recently developed algorithm – Recursive One-Over-T SGD (ROOT-SGD). ROOT-SGD is advantageous over the previous algorithms in that it converges at a non-asymptotically faster rate, and its estimator further achieves an asymptotically normal distribution so that one can calibrate its uncertainty. However, the uncertainty measurement, in particular the covariance, in the ROOT-SGD depends on the unknown population risk function and thus cannot be directly applied to measure the uncertainty. To fill this gap, we develop two estimators for the asymptotic covariance of ROOT-SGD. Using our covariance estimators, one will be able to conduct statistical inference for the ROOT-SGD estimator. Our first estimator is based on the idea of plug-in. For each unknown component in the asymptotic covariance, we estimate it by its empirical counterpart. The plug-in estimator converges to the true asymptotic covariance at a rate $\mathcal{O}(1/\sqrt{t})$, where t is the number of data samples. Despite its quick convergence, the plug-in estimator has the limitation that it relies on the Hessian of the stochastic loss function, which might be unavailable in some cases. Moreover, the computation of the plug-in estimator is expensive. Our second estimator is a Hessian-free estimator that overcomes the above disadvantages. The Hessian-free estimator uses the technique of random-scaling, and we show that it is an asymptotically consistent estimator for the true covariance. Observations in our numerical experiments are consistent with our theorems.

5.1 Introduction

In statistics, a parameter estimation problem often leads to an optimization problem. Let $\boldsymbol{\theta}^*$ be the true model parameters. In many widely-adopted models, $\boldsymbol{\theta}^*$ is the minimizer of a convex population risk function $F := \mathbb{E}_{\boldsymbol{x}\sim P} f(\boldsymbol{\theta}; \boldsymbol{x})$, where $\boldsymbol{x} \in \mathcal{R}^d$ denotes the random sample from distribution P, and $f(\boldsymbol{\theta}; \boldsymbol{x})$ is the loss function at $\boldsymbol{x}$. This is the assumption that we uphold throughout this chapter. Mathematically, we have that

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathcal{R}^p} F(\boldsymbol{\theta}). \tag{5.1}$$

In *offline learning*, one is given a fixed number of samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$, for parameter estimation. In general, the population risk function F is unknown, so people use the empirical risk function as an estimate for F. This gives us the Empirical Risk Minimization (ERM) objective:

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{t}\sum_{i=1}^{t} f(\boldsymbol{\theta}; \boldsymbol{x}_i). \tag{5.2}$$

Despite its popularity in statistics and machine learning, it is clear from (5.2) that when the data size is large, working with ERM will be both computational and memory expensive. In this case, the online learning framework can be considered.

In *Online learning*, one has i.i.d. data samples $\{\boldsymbol{x}_i, i = 1, 2, \ldots\}$ come in a sequence. Given the data stream, one need to sequentially update the current parameter estimate $\widehat{\boldsymbol{\theta}}_i$ based on the past estimate $\widehat{\boldsymbol{\theta}}_{i-1}$ and the incoming data sample $\boldsymbol{x}_i$. The goal is that the parameter estimate will approximate $\boldsymbol{\theta}^*$ in (5.1) more accurately as we get more and more data.

The most well-known algorithm in online learning is the Robbins-Monro algorithm, also known as the Stochastic Gradient Descent (SGD) [61]. The SGD algorithm updates

the parameter estimate as follows:

$$\widehat{\boldsymbol{\theta}}_i = \widehat{\boldsymbol{\theta}}_{i-1} - \eta_i \nabla_{\boldsymbol{\theta}} f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i), \tag{5.3}$$

where $\nabla_{\boldsymbol{\theta}} f(\cdot;\cdot)$ is the first-order derivative of $f(\cdot;\cdot)$ with respect to the first argument; we also simplify $\nabla_{\boldsymbol{\theta}} f(\cdot;\cdot)$ as $\nabla f(\cdot;\cdot)$ in the rest of this chapter. And η_i is the step size. In general, one needs to take a diminishing sequence of step sizes such that $\eta_i \overset{i\to\infty}{\to} 0$ to make SGD converge. Note that the SGD updates the parameter estimate using only the first-order derivative of the stochastic risk function, i.e., $\nabla_{\boldsymbol{\theta}} f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i)$. One refers to the algorithms that only use the stochastic first-order information for parameter update as stochastic first-order algorithms.

In the family of stochastic first-order algorithms, a lot of advancements have been made over the SGD algorithm by using the techniques such as averaging [62], acceleration [63, 64], and variance reduction [39]. There are two main goals in those algorithms: the value of the objective function converges to the minimum quickly, as illustrated by a tight upper bound of the quantity $\mathbb{E}F(\widehat{\boldsymbol{\theta}}_t) - F(\boldsymbol{\theta}^*)$, where $\widehat{\boldsymbol{\theta}}_t$ is the estimator from the algorithm given t data samples; the estimator $\widehat{\boldsymbol{\theta}}_t$ converges in distribution, and one can develop the theoretical expression of such a distribution. Recently, there is a new algorithm — ROOT-SGD [65] — that achieves both goals. We will discuss ROOT-SGD in the following.

Here we review the properties of ROOT-SGD. The non-asymptotic and asymptotic convergence of ROOT-SGD has been established for a strongly-convex population risk function in [65]. The non-asymptotic convergence result says that the value of the objective function in (5.1) converges at rate $\mathcal{O}(1/t)$ for the sample size t, which matches the optimal statistical error up to a constant factor. The asymptotic convergence result says that the estimator $\widehat{\boldsymbol{\theta}}_t$ converges in distribution to a multivariate normal distribution. In particular, one has $\sqrt{t}(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*) \overset{d}{\to} \mathcal{N}(0, \Sigma)$, where the theoretical expression of the asymptotic covariance Σ has been derived.

The asymptotic covariance of the ROOT-SGD estimator Σ is unknown since it depends on the unknown data distribution P and the unknown population risk function F. However, the asymptotic covariance is important for uncertainty quantification. We develop two estimators for the asymptotic covariance, Σ, in ROOT-SGD. We show that our estimators are consistent. Using our estimators, one has reliable uncertainty measurement. With such uncertainty measurement, one will be able to conduct statistical inference, e.g., hypothesis testing for the true parameter $\boldsymbol{\theta}^*$, or constructing confidence intervals.

We now give a high-level summary of the two estimators and discuss their pros and cons. Our first estimator uses the plug-in technique. For each component in Σ, we plug in the data samples to get the empirical version of it. We also made several adjustments to the empirical estimator so that it is well-defined and can be computed in an online fashion. Our final plug-in estimator converges to the true asymptotic covariance at rate $\mathcal{O}(1/\sqrt{t})$, where t is the number of data samples. The plug-in estimator has the advantage of quick convergence. However, it has the limitation of high computation burden and requires the stochastic Hessian. Thus the plug-in estimator might be hard or even impossible to compute in some cases. Our second estimator uses the random-scaling technique from the martingale theory. It only requires the ROOT-SGD point estimator sequences, so the estimator is Hessian-free. We call it a Hessian-free estimator. The Hessian-free estimator can also be computed in online way and is asymptotically consistent. When compared with the plug-in estimator, the Hessian-free estimator has the advantage of lower computational cost. However, the convergence rate of the Hessian-free estimator is unknown. Thus there is no guarantee of a quick convergence.

5.1.1 Existing Covariance Estimators

There is no work for estimating the covariance of the ROOT-SGD algorithm. But for other online stochastic algorithms, such as the Averaging SGD (ASGD) algorithm, there is some literature on covariance estimation. In this section, we review the existing covariance

estimation techniques. We also review the covariance estimators for the ASGD algorithm in detail.

We first quickly review the ASGD algorithm and its asymptotic distribution. Then we summarize the techniques in the current literature for covariance estimation in online stochastic algorithms and their detailed application in ASGD algorithms. These techniques motivate us to develop covariance estimators for ROOT-SGD.

The ASGD estimator is as follows. Recall that SGD updates as

$$\widehat{\boldsymbol{\theta}}_i = \widehat{\boldsymbol{\theta}}_{i-1} - \eta_i \nabla_{\boldsymbol{\theta}} f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i),$$

then ASGD estimator, denote $\bar{\boldsymbol{\theta}}_i$'s, initialize as $\bar{\boldsymbol{\theta}}_1 = \widehat{\boldsymbol{\theta}}_1$ and update as

$$\bar{\boldsymbol{\theta}}_i = \frac{i-1}{i}\bar{\boldsymbol{\theta}}_{i-1} + \frac{1}{i}\widehat{\boldsymbol{\theta}}_i. \tag{5.4}$$

One may notice that such update is equivalent to $\bar{\boldsymbol{\theta}}_i = \frac{1}{i}\sum_{j=1}^{i}\widehat{\boldsymbol{\theta}}_j$, i.e., the ASGD estimator is the averaging of all past SGD estimators. In equation (5.4), the ASGD estimator is updated recursively: the current estimator $\bar{\boldsymbol{\theta}}_i$ only depends on the most recent estimator $\bar{\boldsymbol{\theta}}_{i-1}$ and the current SGD estimator $\widehat{\boldsymbol{\theta}}_i$. Such recursive update avoids storing all past SGD estimators, so the ASGD is memory efficient and compatible with the online learning scheme.

The asymptotic distribution of the ASGD estimator is established for a special decreasing scheme of step size as follows. When $\eta_i = \eta i^{-\alpha}$ for $\alpha \in (1/2, 1)$, Theorem 2 of [62] shows the asymptotic distribution of ASGD estimator as:

$$\sqrt{t}(\bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*) \xrightarrow{d} \mathcal{N}(0, \Sigma_{ASGD}), \tag{5.5}$$

where the asymptotic covariance matrix $\Sigma_{ASGD} = A^{-1}SA^{-1}$ for $A = \nabla^2 F(\boldsymbol{\theta}^*)$, $S = \mathbb{E}_{x\sim P}[\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x})\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x})^T]$.

In the current literature, there are two classes of estimators for the asymptotic covariance based on the information required. We review these two classes for the ASGD algorithm.

The first class is the *plug-in* estimator [66, 67], which requires computing the stochastic Hessian $\nabla^2_{\boldsymbol{\theta}} f(\cdot;\cdot)$. The plug-in estimator estimates each component in the asymptotic covariance by their empirical counterparts. For example, in the asymptotic covariance $\Sigma_{ASGD} = A^{-1}SA^{-1}$, the plug-in estimator replaces A and S by their estimators. Since Σ_{ASGD} includes $A = \nabla^2 F(\cdot)$, the plug-in estimator then requires $\nabla^2_{\boldsymbol{\theta}} f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i)$ to compute the empirical counterpart of A.

The plug-in covariance estimator for the ASGD is as follows. For $\Sigma_{ASGD} = A^{-1}SA^{-1}$, [66] estimates A and S by their empirical counterparts as:

$$\widehat{A}_t = (1/t)\sum_{i=1}^{t} \nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i), \quad \widehat{S}_t = (1/t)\sum_{i=1}^{t} \nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i)\nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i)^T. \tag{5.6}$$

Note that $\widehat{A}_t$ and $\widehat{S}_t$ plug in the estimator sequence $\widehat{\boldsymbol{\theta}}_i$'s from SGD, instead of the ASGD sequence $\bar{\boldsymbol{\theta}}_i$'s. The plug-in estimator for Σ_{ASGD} is $\widehat{\Sigma}_{t,ASGD} = (\widehat{A}_t)^{-1}\widehat{S}_t(\widehat{A}_t)^{-1}$. Both $\widehat{A}_t$ and $\widehat{S}_t$ can be constructed in an online fashion, so this plug-in covariance estimator is an online estimator. Paper [66] further shows that the plug-in estimator $\widehat{\Sigma}_{t,ASGD}$ converges to Σ_{ASGD} as

$$\mathbb{E}\|\Sigma_{ASGD} - \widehat{\Sigma}_{t,ASGD}\|_2 \lesssim t^{-\alpha/2}. \tag{5.7}$$

Since we have $\alpha \in (1/2, 1)$, the convergence rate is strictly slower than $\mathcal{O}(t^{-1/2})$.

The second class is the *Hessian-free* estimator [66, 68, 67, 69, 70, 71] which does not use the stochastic Hessian $\nabla^2_{\boldsymbol{\theta}} f(\cdot;\cdot)$. The Hessian-free estimator can be computed purely based on the estimator sequences ($\widehat{\boldsymbol{\theta}}_i$'s or $\bar{\boldsymbol{\theta}}_i$'s in the ASGD algorithm). The idea is similar to that of sample covariance: Suppose $\boldsymbol{a}_i, i = 1, \ldots, t$ are i.i.d. samples from a distribution with finite second moment, then $(1/t)\sum_{i=1}^{t}(\boldsymbol{a}_i - \bar{\boldsymbol{a}}_t)(\boldsymbol{a}_i - \bar{\boldsymbol{a}}_t)^T$ is an asymptotically consistent estimator for the population covariance, where $\bar{\boldsymbol{a}}_t = (1/t)\sum_{i=1}^{t}\boldsymbol{a}_i$ is the sample

mean. Then, for $\widehat{\boldsymbol{\theta}}_i$ (or $\bar{\boldsymbol{\theta}}_i$), one can also construct a sample covariance estimator by treating each $\widehat{\boldsymbol{\theta}}_i$ (or $\bar{\boldsymbol{\theta}}_i$) as a sample. However, since $\widehat{\boldsymbol{\theta}}_i$ and $\widehat{\boldsymbol{\theta}}_{i-1}$ are highly correlated, a vanilla sample covariance as $(1/t)\sum_{i=1}^{t}(\widehat{\boldsymbol{\theta}}_i - \bar{\boldsymbol{\theta}}_t)(\widehat{\boldsymbol{\theta}}_i - \bar{\boldsymbol{\theta}}_t)^T$ will not be asymptotically consistent. To solve this issue, certain modifications will be required. We review them in the following.

There are two types of Hessian-free estimators for ASGD, which use different ways to deal with the autocorrelation structure in the $\{\widehat{\boldsymbol{\theta}}_i\}$ sequence.

The first is the batch-mean estimator that uses batch mean as decorrelated samples in place of the SGD estimators $\widehat{\boldsymbol{\theta}}_i$'s to calculate the sample covariance [66, 68]. The idea of batch-mean originates from the covariance estimation for time-homogeneous Markov-chain, which is an auto-correlated sample sequence [72, 73]. The batch-mean estimator is constructed as follows: Divide $\{\widehat{\boldsymbol{\theta}}_i\}_{i=1,\ldots,t}$ in to m batches as $\underbrace{\widehat{\boldsymbol{\theta}}_1,\ldots,\widehat{\boldsymbol{\theta}}_{b_1}}$, $\underbrace{\widehat{\boldsymbol{\theta}}_{b1+1},\ldots,\widehat{\boldsymbol{\theta}}_{b_1+b_2}}$, $\ldots,\underbrace{\widehat{\boldsymbol{\theta}}_{b_1+\ldots+b_{m-1}+1},\ldots,\widehat{\boldsymbol{\theta}}_{b_1+\ldots+b_m}}$ where $b_j > 0$ is the batch size of the jth batch, and $\sum_{j=1}^{m} b_j = t$. Denote the batch mean for jth batch as $\bar{\boldsymbol{\theta}}_j^b := (1/b_j)\sum_{i=b_1+\ldots+b_{j-1}+1}^{b_1+\ldots+b_j}\widehat{\boldsymbol{\theta}}_i$. Finally, the batch-mean covariance estimator is calculated as the (weighted) sample covariance of batch means $\bar{\boldsymbol{\theta}}_j^b$'s, denote $\widehat{\Sigma}_{t,BM}$. When the batch size is large enough, the batch means are sufficiently decorrelated, thus the $\widehat{\Sigma}_{t,BM}$ will be a consistent estimator of Σ_{ASGD} as shown by [66, 68].

The second is the random-scaling estimator that uses the estimators $\widehat{\boldsymbol{\theta}}_i$'s (or $\bar{\boldsymbol{\theta}}_i$'s for ASGD algorithm) to calculate sample covariance, but carefully analyze how the sample autocorrelation affects the asymptotic distribution [67, 70]. The random-scaling technique is also referred to as robust testing [71, 74]. For the ASGD algorithm, [70] computes a random-scaling estimator as $\bar{V}_{t,RS} = \frac{1}{t^2}\sum_{i=1}^{t} i^2(\bar{\boldsymbol{\theta}}_i - \bar{\boldsymbol{\theta}}_t)(\bar{\boldsymbol{\theta}}_i - \bar{\boldsymbol{\theta}}_t)^T$. Compared with a consistent covariance estimator $\widehat{\Sigma}_t$ for Σ_{ASGD} such that $\sqrt{t}(\widehat{\Sigma}_t)^{-1/2}(\bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*) \xrightarrow{d} \mathcal{N}(0, I)$, one instead has that $\sqrt{t}(\bar{V}_{t,RS})^{-1/2}(\bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)$ converges in distribution to another random variable. [70] further shows that such random variable is well-defined and independent of the specific optimization problem (i.e., does not depend on $f(\cdot;\cdot)$ or P). In this way, one can construct a consistent covariance estimator of $\sqrt{t}(\bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)$ using the random-scaling

estimator $\bar{V}_{t,RS}$.

5.1.2 Overview of ROOT-SGD Algorithm

In this section, we review the ROOT-SGD algorithm and the asymptotic distribution of the estimator from the ROOT-SGD algorithm.

The ROOT-SGD algorithm [65] applies the idea of variance reduction [39, 75, 40, 41] to online learning, so it converges fast for a fixed step size. The ROOT-SGD algorithm is given in Algorithm 5.1.

Algorithm 5.1: ROOT-SGD [65]

Initialization: $\widehat{\boldsymbol{\theta}}_0$; set v_1 to be $\nabla f(\widehat{\boldsymbol{\theta}}_0; \boldsymbol{x}_1)$; choose a step size η and burn-in period B;

for *i = 1,...,B-1* **do**

$$\boldsymbol{v}_i = \nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i) + \frac{i-1}{i}(\boldsymbol{v}_{i-1} - \nabla f(\widehat{\boldsymbol{\theta}}_{i-2}; \boldsymbol{x}_i)),$$
$$\widehat{\boldsymbol{\theta}}_i = \widehat{\boldsymbol{\theta}}_{i-1}.$$

for *i = B,...,t* **do**

$$\boldsymbol{v}_i = \nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i) + \frac{i-1}{i}(\boldsymbol{v}_{i-1} - \nabla f(\widehat{\boldsymbol{\theta}}_{i-2}; \boldsymbol{x}_i)),$$
$$\widehat{\boldsymbol{\theta}}_i = \widehat{\boldsymbol{\theta}}_{i-1} - \eta \boldsymbol{v}_i.$$

Output: $\widehat{\theta}_t$.

We explain why ROOT-SGD is a variance-reduced algorithm. In Algorithm 5.1, the intermediate quantity $\boldsymbol{v}_i$ serves as a variance-reduced estimator of the population gradient. The estimator $\boldsymbol{v}_i$ is developed as follows. One starts with an estimator $\boldsymbol{v}_i'$ of $\nabla F(\widehat{\boldsymbol{\theta}}_{i-1})$ such that the error $\boldsymbol{v}_i' - \nabla F(\widehat{\boldsymbol{\theta}}_{i-1})$ is the average of the errors $\{\nabla f(\widehat{\boldsymbol{\theta}}_{j-1}; \boldsymbol{x}_j) - \nabla F(\widehat{\boldsymbol{\theta}}_{j-1})\}_{j=1,\ldots,i}$. Mathematically,

$$\boldsymbol{v}_i' - \nabla F(\widehat{\boldsymbol{\theta}}_{i-1}) = \frac{1}{i}\sum_{j=1}^{i}(\nabla f(\widehat{\boldsymbol{\theta}}_{j-1}; \boldsymbol{x}_j) - \nabla F(\widehat{\boldsymbol{\theta}}_{j-1})). \tag{5.8}$$

Then the term $\boldsymbol{v}_i'$ is an unbiased estimator of $\nabla F(\widehat{\boldsymbol{\theta}}_{i-1})$. Moreover, assume the conditional

covariance $\mathrm{Cov}(\nabla f(\widehat{\boldsymbol{\theta}}_{j-1}; \boldsymbol{x}_j)|\widehat{\boldsymbol{\theta}}_{j-1}) \prec \Sigma'$ almost surely, then $\mathrm{Cov}(\boldsymbol{v}_i') \prec \Sigma'/i$, which is a reduced covariance. One can further rewrite (5.8) so that $\boldsymbol{v}_i' - \nabla F(\widehat{\boldsymbol{\theta}}_{i-1})$ only depends on $\boldsymbol{v}_{i-1}' - \nabla F(\widehat{\boldsymbol{\theta}}_{i-2})$ and $\nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i) - \nabla F(\widehat{\boldsymbol{\theta}}_{i-1})$ as follows:

$$\begin{aligned}\boldsymbol{v}_i' - \nabla F(\widehat{\boldsymbol{\theta}}_{i-1}) &= \frac{1}{i}(\nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i) - \nabla F(\widehat{\boldsymbol{\theta}}_{i-1})) + \frac{1}{i}\sum_{j=1}^{i-1}(\nabla f(\widehat{\boldsymbol{\theta}}_{j-1}; \boldsymbol{x}_j) - \nabla F(\widehat{\boldsymbol{\theta}}_{j-1})) \\ &= \frac{1}{i}(\nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i) - \nabla F(\widehat{\boldsymbol{\theta}}_{i-1})) + \frac{i-1}{i}(\boldsymbol{v}_{i-1}' - \nabla F(\widehat{\boldsymbol{\theta}}_{i-2})). \end{aligned} \tag{5.9}$$

Since $\nabla F(\widehat{\boldsymbol{\theta}}_{i-1})$ and $\nabla F(\widehat{\boldsymbol{\theta}}_{i-2})$ in (5.9) are unknown, [65] replaces them with the unbiased estimators $\nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i)$ and $\nabla f(\widehat{\boldsymbol{\theta}}_{i-2}; \boldsymbol{x}_i)$, respectively. This gives us the gradient estimator in the ROOT-SGD algorithm that

$$\boldsymbol{v}_i = \nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i) + \frac{i-1}{i}(\boldsymbol{v}_{i-1} - \nabla f(\widehat{\boldsymbol{\theta}}_{i-2}; \boldsymbol{x}_i)). \tag{5.10}$$

Then the ROOT-SGD performs update on parameter $\widehat{\boldsymbol{\theta}}_i$ using $\boldsymbol{v}_i$.

We now review the asymptotic distribution of ROOT-SGD in [65]. Our covariance estimators are all based on this asymptotic distribution. Under certain regularity conditions (see details in Lemma 5.4), for a proper step size η and burn-in period B, [65] proves that

$$\sqrt{t}(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*) \xrightarrow{d} \mathcal{N}(0, \Sigma) \tag{5.11}$$

for $\Sigma = A^{-1}(S + \mathbb{E}[\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)])A^{-1}$, where $\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}) = \nabla^2 f(\boldsymbol{\theta}; \boldsymbol{x}) - \nabla^2 F(\boldsymbol{\theta})$, $A = \nabla^2 F(\boldsymbol{\theta}^*)$, $S = \mathbb{E}_{x\sim P}[\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x})\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x})^T]$ and Λ is by solving the following matrix equation in Λ (a.k.a. *modified Lyapunov equation*):

$$\Lambda A + A\Lambda - \eta\mathbb{E}[\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)] - \eta A\Lambda A = \eta S. \tag{5.12}$$

5.1.3 Convergence of ROOT-SGD for Two Examples

In this section, we explain the convergence property of the ROOT-SGD algorithm for two learning examples: the normal mean estimation and the natural parameter estimation for the exponential family model.

Normal Mean Estimation. We now describe the normal mean estimation problem and show the convergence of the ROOT-SGD for the normal mean estimation.

Consider the problem of estimating the mean vector of multivariate normal distribution. Suppose the data $\boldsymbol{x}_i \overset{i.i.d.}{\sim} N(\boldsymbol{\theta}^*; I_p)$, let the loss function be the negative log-likelihood:

$$f(\boldsymbol{\theta}; \boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{x}\|_2^2.$$

Then $\nabla f(\boldsymbol{\theta}; \boldsymbol{x}) = \boldsymbol{\theta} - \boldsymbol{x}$. It is not hard to check that the ROOT-SGD update is:

$$\widehat{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\theta}}_{t-1} - \eta(\widehat{\boldsymbol{\theta}}_{t-1} - \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{x}_i) = (1-\eta)\widehat{\boldsymbol{\theta}}_{t-1} + \frac{\eta}{t}\sum_{i=1}^{t} \boldsymbol{x}_i. \tag{5.13}$$

We now compare the ROOT-SGD estimator with the common estimator of sample mean $\bar{\boldsymbol{x}}_t = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{x}_i$, which is a consistent estimator for the true mean vector $\boldsymbol{\theta}^*$. Then (5.13) can be rewritten as:

$$\widehat{\boldsymbol{\theta}}_t - \bar{\boldsymbol{x}}_t = (1-\eta)(\widehat{\boldsymbol{\theta}}_{t-1} - \bar{\boldsymbol{x}}_t). \tag{5.14}$$

Thus, for the step size $\eta \in (0, 1]$, the current estimator will be closer to the sample mean than the last estimator. When the sample mean $\bar{\boldsymbol{x}}_t$ is close to the true parameter $\boldsymbol{\theta}^*$, we will have nearly linear convergence of the ROOT-SGD estimator $\widehat{\boldsymbol{\theta}}_t$. Our calculation above explicitly shows that the ROOT-SGD converges fast.

Exponential Family Model. We now describe the problem of parameter estimation in the exponential family model and show the convergence of the ROOT-SGD algorithm in

this case.

Consider estimating the natural parameter $\boldsymbol{\theta} \in R^p$ in an exponential family model, where the probability density function is given by:

$$L(\boldsymbol{x}; \boldsymbol{\theta}) = h(\boldsymbol{x}) \exp[\langle \boldsymbol{\theta}, \boldsymbol{T}(\boldsymbol{x}) \rangle - B(\boldsymbol{\theta})].$$

Let the risk function be the negative log-likelihood $f(\boldsymbol{\theta}; \boldsymbol{x}) = -\langle \boldsymbol{\theta}, \boldsymbol{T}(\boldsymbol{x}) \rangle + B(\boldsymbol{\theta})$, then the gradient is $\nabla f(\boldsymbol{\theta}; \boldsymbol{x}) = -\boldsymbol{T}(\boldsymbol{x}) + \nabla B(\boldsymbol{\theta})$.

Then we write the ROOT-SGD update for the exponential family model. For the intermediate quantity $\boldsymbol{v}_t$, we can solve its recursion relationship to get:

$$\begin{aligned}
\boldsymbol{v}_t &= \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \boldsymbol{T}(\boldsymbol{x}_t) + \frac{t-1}{t}(\boldsymbol{v}_{t-1} - \nabla B(\widehat{\boldsymbol{\theta}}_{t-2}) + \boldsymbol{T}(\boldsymbol{x}_t)) \\
&= \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \frac{1}{t}\boldsymbol{T}(\boldsymbol{x}_t) + \frac{t-1}{t}(\boldsymbol{v}_{t-1} - \nabla B(\widehat{\boldsymbol{\theta}}_{t-2})) \\
&= \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{T}(\boldsymbol{x}_i),
\end{aligned}$$

The parameter update using such $\boldsymbol{v}_t$ is

$$\widehat{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\theta}}_{t-1} - \eta \boldsymbol{v}_t = \widehat{\boldsymbol{\theta}}_{t-1} - \eta \left(\nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{T}(\boldsymbol{x}_i) \right). \tag{5.15}$$

Finally, we compare $\widehat{\boldsymbol{\theta}}_t$ with the maximum likelihood estimator (MLE). The MLE is equivalent to the estimator in (5.2), and it is asymptotically efficient. In particular, the MLE converges to the true parameter at rate $\mathcal{O}(1/t)$. For exponential family model, the MLE $\boldsymbol{\theta}_t^*$ satisfies

$$\nabla B(\boldsymbol{\theta}_t^*) = \frac{1}{t}\sum_{i=1}^{t} \boldsymbol{T}(\boldsymbol{x}_i).$$

Thus the ROOT-SGD update in (5.15) can be rewritten as

$$\widehat{\boldsymbol{\theta}}_t = \widehat{\boldsymbol{\theta}}_{t-1} - \eta \left(\nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \nabla B(\boldsymbol{\theta}_t^*) \right).$$

Under some regularity conditions on $B(\cdot)$, we have Proposition 5.1 holds. By Proposition 5.1, when the MLE $\boldsymbol{\theta}_t^*$ is close to the true parameter $\boldsymbol{\theta}^*$, we will have nearly linear convergence of the ROOT-SGD estimator. In this way, we have a straightforward explanation for the convergence of ROOT-SGD that could not be applied to SGD.

Proposition 5.1. *When $B(\boldsymbol{\theta})$ is μ-strongly convex and l-smooth for $0 < \mu < l$, we can take $\eta \in (0, \frac{2\mu}{l(\mu+l)}]$, so that $\exists \alpha \in (0, 1/\eta)$ s.t.*

$$\|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t^*\|_2 \leq (1 - \eta\alpha)\|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2. \tag{5.16}$$

The proof for Proposition 5.1 is in Section 5.5.1.

Notations and Organization. We define some notations that we will use throughout this chapter. For matrices $X \in \mathbb{R}^{n\times o}, Y \in \mathbb{R}^{p\times q}$: let $X \otimes Y$ denote the standard Kronecker product; let $\|X\|_2$ be the operator norm of X and $\|X\|_F$ be the Frobenius norm of X; denote the vectorization of X as $\mathrm{Vec}(X) = (X_{11}, ..., X_{n1}, X_{12}, ..., X_{n2}, ..., X_{1o}, ..., X_{no})^T$. For two nonnegative real values $a(t)$ and $b(t)$, denote $a(t) \lesssim b(t)$ if $\exists c_1 > 0$ such that $a(t) \leq c_1 b(t)$.

The remainder of this chapter is organized as follows. In Section 5.2, we propose our plug-in estimator for asymptotic covariance and show the convergence of the plug-in estimator to the true covariance. In Section 5.3, we propose our Hessian-free estimator for the asymptotic covariance and prove its asymptotic consistency. In Section 5.4, some numerical experiments are reported.

5.2 A Plug-in Estimator and Its Convergence

In this section, we describe our plug-in covariance estimator in the ROOT-SGD algorithm, and we prove the convergence rate of the aforementioned plug-in covariance estimator.

5.2.1 A Plug-in Estimator with Thresholding

Our goal is to provide an online estimator for the asymptotic covariance of ROOT-SGD. Recall that the asymptotic covariance is $\Sigma = A^{-1}(S + \mathbb{E}[\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)])A^{-1}$. For the plug-in estimator, we assume that we have access to the stochastic second order term $\nabla^2 f(\widehat{\boldsymbol{\theta}}_{t-1}; \boldsymbol{x}_t)$.

There are two steps in the plug-in estimator: plug-in and thresholding. The plug-in step approximates each component in Σ by its empirical version, that is, estimate A, S and $\mathbb{E}[\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)]$ using the parameter estimates and data samples; the thresholding step performs spectral thresholding on the empirical estimators. We will explain spectral thresholding in detail later in our thresholding step.

Plug-in Step. For $A, S, \mathbb{E}[\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)]$ in the asymptotic covariance, we approximate them by their empirical counterparts as follows. For

$$A = \nabla^2 F(\boldsymbol{\theta}^*),$$

we approximate it by

$$\widehat{A}_t = \frac{1}{t-B}\sum_{i=B+1}^{t} \nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i). \tag{5.17}$$

For

$$S = \mathbb{E}_{x\sim P}[\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x})\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x})^T],$$

we approximate it by

$$\widehat{S}_t = \frac{1}{t-B}\sum_{i=B+1}^{t} \nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i)\nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i)^T. \tag{5.18}$$

For

$$\mathbb{E}[\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)] = \mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\Lambda\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})] - A\Lambda A$$

we approximate it by

$$\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)\Lambda\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)-A\Lambda A. \tag{5.19}$$

In the estimator (5.19), we need to further estimate Λ. Since Λ is the solution of (5.12), we can replace each unknown term in (5.12) by their empirical estimators, and solve the "perturbed version" of (5.12) in Λ as:

$$\Lambda\widehat{A}_t+\widehat{A}_t\Lambda-\eta\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)\Lambda\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)=\eta\widehat{S}_t, \tag{5.20}$$

and we denote the solution as $\widehat{\Lambda}_t$. Vectorize (5.20) we have

$$\begin{aligned}\eta\,\mathrm{Vec}(\widehat{S}_t)=&(\widehat{A}_t\otimes I)\,\mathrm{Vec}(\widehat{\Lambda}_t)+(I\otimes\widehat{A}_t)\,\mathrm{Vec}(\widehat{\Lambda}_t)\\&-\eta\underbrace{\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)\otimes\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)}_{:=\widehat{P}_t}\mathrm{Vec}(\widehat{\Lambda}_t).\end{aligned}$$

That is,

$$\mathrm{Vec}(\widehat{\Lambda}_t)=\eta\{\widehat{A}_t\otimes I+I\otimes\widehat{A}_t-\eta\widehat{P}_t\}^{-1}\,\mathrm{Vec}(\widehat{S}_t). \tag{5.21}$$

Using the estimators $\widehat{A}_t$,$\widehat{S}_t$ and $\widehat{\Lambda}_t$ in equations (5.17), (5.18), (5.21), the plug-in estimator of the asymptotic covariance $\Sigma=A^{-1}(S+\mathbb{E}[\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)])A^{-1}=A^{-1}(\frac{1}{\eta}\Lambda A+\frac{1}{\eta}A\Lambda-A\Lambda A)A^{-1}$ is

$$\widehat{\Sigma}_t=\widehat{A}_t^{-1}\left(\frac{1}{\eta}\widehat{\Lambda}_t\widehat{A}_t+\frac{1}{\eta}\widehat{A}_t\widehat{\Lambda}_t-\widehat{A}_t\widehat{\Lambda}_t\widehat{A}_t\right)\widehat{A}_t^{-1}. \tag{5.22}$$

We add notes on the computation of the plug-in estimator as follows. First, the plug-in estimator is an online estimator. Since each of $\widehat{A}_t,\widehat{P}_t,\widehat{S}_t$ can be computed in an online fashion, the estimator $\widehat{\Lambda}_t$ in (5.21) is also an online estimator. The plug-in estimator in (5.22) then can be computed with the online estimators $\widehat{A}_t$ and $\widehat{\Lambda}_t$. The plug-in estimator is thus online. Second, among all components in the plug-in estimator, the term $\widehat{\Lambda}_t$ in (5.21) is

most heavy in computation. Luckily, there are some cases in that we can avoid computing $\widehat{\Lambda}_t$. For example, when $\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*) = 0$ (which holds for the exponential family model), one has $\Sigma = A^{-1}SA^{-1}$. In this case, the plug-in estimator is simply $\widehat{\Sigma}_t = \widehat{A}_t^{-1}\widehat{S}_t\widehat{A}_t^{-1}$. Avoiding computing $\widehat{\Lambda}_t$ can reduce the computational burden for those cases.

Thresholding Step. Note that in equations (5.21) and (5.22) we need to invert $\widehat{A}_t \otimes I + I \otimes \widehat{A}_t - \eta\widehat{P}_t$ and $\widehat{A}_t$, which are random quantities that might be poorly conditioned or even non-invertible. To solve this issue, we do spectral thresholding for $\widehat{A}_t$ and $\widehat{P}_t$.

We briefly explain the spectral thresholding technique as follows. For a positive semi-definite matrix M, spectral thresholding involves two steps. First, one performs eigen-decomposition on M, denote $M = UDU^T$. Second, depending on whether one needs upper thresholding or lower thresholding on the matrix M, one performs upper or lower truncation on the diagonal matrix D. For example, when upper thresholding using the threhold value C, one computes a diagonal matrix $\widetilde{D}$ such that $(\widetilde{D})_{i,i} = \min(C, D_{i,i})$. On the contrary, when lower thresholding using the threhold value c, one computes a diagonal matrix $\widetilde{D}$ such that $(\widetilde{D})_{i,i} = \max(c, D_{i,i})$. The thresholded matrix is $M = U\widetilde{D}U^T$.

For our case, we do lower thresholding for $\widehat{A}_t$ using threshold value δ and denote the thresholded matrix as $\widetilde{A}_t$. We do upper thresholding for $\widehat{P}_t$ using threshold value δ' and denote the thresholded matrix as $\widetilde{P}_t$. The choice of thresholding parameters δ, δ' will be discussed in Theorem 5.1.

With the thresholded estimator for A and P, we now give the thresholded plug-in covariance estimator. Similar to equation (5.21), we first develop the thresholded estimator for Λ:

$$\mathrm{Vec}(\widetilde{\Lambda}_t) = \eta\{\widetilde{A}_t \otimes I + I \otimes \widetilde{A}_t - \eta\widetilde{P}_t\}^{-1}\,\mathrm{Vec}(\widehat{S}_t).$$

Then the thresholded plug-in estimator for the asymptotic covariance is

$$\widetilde{\Sigma}_t = \widetilde{A}_t^{-1}\left(\frac{1}{\eta}\widetilde{\Lambda}_t\widetilde{A}_t + \frac{1}{\eta}\widetilde{A}_t\widetilde{\Lambda}_t - \widetilde{A}_t\widetilde{\Lambda}_t\widetilde{A}_t\right)\widetilde{A}_t^{-1}. \tag{5.23}$$

The thresholding is usually not required in practice, as $\widehat{A}_t$ and $\widehat{P}_t$ are close to A and P with high probability. The thresholding is just a theoretical guarantee that the matrix inverse in our estimator is doable.

5.2.2 Convergence of Our Plug-in Estimator

In this section, we show that the thresholded plug-in estimator is consistent. In particular, we bound $\mathbb{E}\|\Sigma - \widetilde{\Sigma}_t\|_F$ and show that it converges to 0 as $t \to \infty$. We also specify the convergence rate in t. For a specific problem, such as linear regression, one can further analyze the dependence of the convergence rate on the dimension of the problem.

To establish the consistency of the plug-in estimators, we need some regularity conditions on the objective function. These conditions are mild, as we show that some common learning examples, such as linear regression, logistic regression, and exponential family model, all satisfy those conditions; see details in Section 5.5.3. We now define these conditions.

Assumption 5.1 (γ-strong convexity and L_1-smoothness)**.** *Assume that $F(\boldsymbol{\theta})$ is twice continuously differentiable, γ-strongly convex and L_1-smooth:*

$$F(\boldsymbol{\theta}_1) \geq F(\boldsymbol{\theta}_2) + \nabla F(\boldsymbol{\theta}_2)^T(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) + \frac{\gamma}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2, \forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2;$$

$$F(\boldsymbol{\theta}_1) \leq F(\boldsymbol{\theta}_2) + \nabla F(\boldsymbol{\theta}_2)^T(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2) + \frac{L_1}{2}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2, \forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2.$$

Assumption 5.2 (Finite covariance at optimality)**.** *Assume $\nabla f(\boldsymbol{\theta}; \boldsymbol{x})$ is unbiased and has finite covariance at $\boldsymbol{\theta}^*$ (w.r.t. randomness in x) as follows:*

$$\mathbb{E}_{x\sim P}[\nabla f(\boldsymbol{\theta}; \boldsymbol{x})] = \nabla F(\boldsymbol{\theta}),$$
$$\mathbb{E}_{x\sim P}[\|\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x}) - \nabla F(\boldsymbol{\theta}^*)\|_2^2] = \sigma_1^2 < \infty.$$

Assumption 5.3 (Lipschitz stochastic noise)**.** *Denote* $\delta(\boldsymbol{\theta};\boldsymbol{x}) := \nabla f(\boldsymbol{\theta};\boldsymbol{x}) - \nabla F(\boldsymbol{\theta})$*. Assume that*

$$\mathbb{E}\|\delta(\boldsymbol{\theta}_1;\boldsymbol{x}) - \delta(\boldsymbol{\theta}_2;\boldsymbol{x})\|_2^2 \leq L_2^2\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2, \forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2. \tag{5.24}$$

Assumption 5.4. *Assume the smoothness of stochastic gradients and moments as:* $\forall \boldsymbol{\theta} \in \mathcal{R}^p$*, we have*

$$\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^2 \leq L_3^2\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2,$$

and

$$\mathbb{E}\|\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^4 \leq l_4^4,$$
$$\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^4 \leq L_4^4.$$

Assumption 5.5 (L_5-smoothness)**.** *Assume that* $\nabla f(\boldsymbol{\theta};\boldsymbol{x})$ *is mean-squared Lipchitz continuous:*

$$\mathbb{E}\|\nabla f(\boldsymbol{\theta}_1;\boldsymbol{x}) - \nabla f(\boldsymbol{\theta}_2;\boldsymbol{x})\|_2^2 \leq L_5^2\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2, \forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2.$$

Remark 5.1. *Assumption 5.5 is implied by Assumptions 5.1 and 5.3. When Assumptions 5.1 and 5.3 hold, we must have Assumption 5.5 hold for* $L_5^2 = L_1^2 + L_2^2$*. See the proof in Section 5.5.2.*

Assumption 5.6. *Assume the Kronecker product of the Hessian is continuous and smooth around* $\boldsymbol{\theta}^*$*:*

$$\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2 \leq L_6\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2 + L_6'\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2,$$

and has bounded covariance at $\boldsymbol{\theta}^*$*:*

$$\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) - \mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})]\|_2^2 \leq L_7.$$

Remark 5.2. *Assumption 5.6 is implied by Assumption 5.4. When Assumption 5.4 holds, Assumption 5.6 also holds for* $L_6 = 2L_3L_4^2$, $L_6' = L_3^2$ *and* $L_7 = L_4^4$. *See proof in Section 5.5.2.*

We show that the thresholded plug-in estimator is consistent in the following theorem:

Theorem 5.1 (Asymptotically consistent estimator for Σ)**.** *Under Assumptions 5.1 to 5.6, there exists constants* c_1, c_2 *such that run ROOT-SGD for* $\eta < \min\Big(c_1(\frac{\gamma}{L_2^2} \wedge \frac{1}{L_1} \wedge \frac{\gamma^{1/3}}{L_4^{4/3}}), 2\delta/L_4^2\Big)$, *burn-in period* $B = \left\lceil \frac{c_2}{\gamma\eta} \right\rceil$ *and take the thresholding parameter in* $\widetilde{\Sigma}_t$ *be such that* $\delta < \lambda_{\min}(A)$ *and* $\delta' = L_4^2$, *we have*

$$\mathbb{E}\|\Sigma - \widetilde{\Sigma}_t\|_F \lesssim \left[\|S\|_F C_p^2 + \sqrt{p} C_p^4\right] / \sqrt{t},$$

where $C_p \lesssim \max\{\sigma_1/\gamma, L_3, \sqrt{L_4}, l_4, \sqrt{L_5}, L_6\}$. *Thus, for a fixed* C_p, *when sample size* $t \to \infty$, *we have* $\mathbb{E}\|\Sigma - \widetilde{\Sigma}_t\|_F \to 0$.

Remark 5.3. *In Theorem 5.1, we can further analyze how the upper bound depends on the parameter dimension for a specific learning problem.*

For example, consider the linear regression model. We have data $\boldsymbol{x}_i^T = (\boldsymbol{a}_i^T, b_i)$, *where* $\boldsymbol{a}_i$*'s are the vectors of explanatory variables, and* b_i*'s are the responses. Assume* $\boldsymbol{a}_i \overset{i.i.d.}{\sim} N(0, I_p)$, *and there is a true parameter* $\boldsymbol{\theta}^*$ *such that* $b_i = \boldsymbol{a}_i^T\boldsymbol{\theta}^* + \epsilon_i$ *for* $\epsilon_i \overset{i.i.d.}{\sim} N(0,1)$. *Here the additive noises* ϵ_i*'s are independent of* $\boldsymbol{a}_i$*'s. Use the squared loss*

$$f(\boldsymbol{\theta}; \boldsymbol{x}) = \frac{1}{2}(\boldsymbol{a}^T\boldsymbol{\theta} - b)^2.$$

As shown in Section 5.5.3.2, we have $C_p \lesssim \sqrt{p}$ *and* $\|S\|_F = \left\|\mathbb{E}[\epsilon^2 \boldsymbol{a}\boldsymbol{a}^T]\right\|_F = \|I_p\|_F = \sqrt{p}$. *Thus the bound in Theorem 5.1 becomes*

$$\mathbb{E}\|\Sigma - \widetilde{\Sigma}_t\|_F \lesssim p^{5/2}/\sqrt{t}.$$

The proof of Theorem 5.1 is in Section 5.5.4.

By Theorem 5.1, our thresholded plug-in estimator $\widetilde{\Sigma}_t$ is asymptotically consistent. Combined with the asymptotic distribution of the parameter estimator $\widehat{\boldsymbol{\theta}}_t$ in (5.11) that $\sqrt{t}(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*) \xrightarrow{d} \mathcal{N}(0, \Sigma)$, we can build asymptotically exact confidence interval. We summarize this result in the following corollary:

Corollary 5.1 (Asymptotically exact confidence interval)**.** *Under assumptions of Theorem 5.1, when the parameter dimension p is fixed and sample size $t \to \infty$, we have*

$$P\left((\widehat{\boldsymbol{\theta}}_t)_j - z_{q/2} * (\widetilde{\Sigma}_t)_{j,j}^{1/2}/\sqrt{t} \leq \boldsymbol{\theta}_j^* \leq (\widehat{\boldsymbol{\theta}}_t)_j + z_{q/2} * (\widetilde{\Sigma}_t)_{j,j}^{1/2}/\sqrt{t}\right) \to 1 - q.$$

With our plug-in covariance estimator proposed and the asymptotic convergence proved in this section, we compare our plug-in covariance estimator with that for the ASGD algorithm. The comparison details are in Section 5.5.7.1. To summarize the comparison, our plug-in estimator has faster convergence than that from the ASGD algorithm. However, the computation of our plug-in estimator is heavier than that of the ASGD, except for the special case that $\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) \equiv \nabla^2 F(\boldsymbol{\theta}^*)$.

5.3 A Hessian-free Estimator and Its Asymptotic Consistency

In this section, we propose a covariance estimator for ROOT-SGD that uses only the parameter estimation sequence. The plug-in estimator in the previous section uses the stochastic Hessian, which is unknown or hard to compute for some practical problems. Our covariance estimator in this section does not have this issue: it uses only the parameter estimation and thus is Hessian-free.

5.3.1 A Hessian-free Estimator

Our Hessian-free estimator uses the random-scaling technique. In this section, we review the random-scaling technique in the ASGD algorithm and discuss the covariance estimator derived by the random-scaling technique. We further propose a random-scaling estimator

for the ROOT-SGD algorithm.

The random-scaling technique, when applied to the ASGD algorithm, gives the following result. For the ASGD estimator $\bar{\boldsymbol{\theta}}_t$, [70] shows that

$$\frac{\sqrt{t}\boldsymbol{w}^T(\bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)}{\sqrt{\boldsymbol{w}^T\bar{V}_t\boldsymbol{w}}} \xrightarrow{d} \frac{W_1}{\sqrt{\int_0^1 (W_r - rW_1)^2 dr}}, \tag{5.25}$$

where $\boldsymbol{w}$ is any given direction satisfying $\boldsymbol{w} \neq \mathbf{0}$, W_r is the standard Wiener process and $\bar{V}_t = \frac{1}{t^2}\sum_{i=1}^t i^2(\bar{\boldsymbol{\theta}}_i - \bar{\boldsymbol{\theta}}_t)(\bar{\boldsymbol{\theta}}_i - \bar{\boldsymbol{\theta}}_t)^T$. The term $\bar{V}_t$ is similar to a (weighted) covariance estimator when we treat $\bar{\boldsymbol{\theta}}_i$'s as samples, so we refer to it as the random-scaling estimator. And we refer to the L.H.S. of (5.25), i.e. the $\frac{\sqrt{t}\boldsymbol{w}^T(\bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)}{\sqrt{\boldsymbol{w}^T\bar{V}_t\boldsymbol{w}}}$, as the random-scaling quantity. Now the random-scaling quantity converges in distribution to $\frac{W_1}{\sqrt{\int_0^1 (W_r - rW_1)^2 dr}}$ for any given direction $\boldsymbol{w}$. Then $\mathrm{Var}\left(\frac{W_1}{\sqrt{\int_0^1 (W_r - rW_1)^2 dr}}\right) * \bar{V}_t$ is an estimator for the asymptotic covariance of $\sqrt{t}(\bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)$, and we refer to such an estimator as the random-scaling covariance estimator.

The random-scaling technique can be applied to other online algorithms. For example, a similar result as (5.25) has been proved for the zero-order Kiefer-Wolfowitz algorithm by Theorem 4.4 of [67]. These results indicate the usefulness of the random-scaling technique in different algorithms. For the ROOT-SGD estimator, the random-scaling technique may also apply.

We now follow the random-scaling quantity in (5.25) to propose a random-scaling quantity for the ROOT-SGD algorithm. Observing (5.25), the random-scaling quantity essentially treats the averaging sequence $\bar{\boldsymbol{\theta}}_i$'s as samples and calculate the sample covariance. The asymptotic distribution in (5.25) relies on a substantial extension of the asymptotic normality of the averaging sequence $\sqrt{t}(\bar{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)$ in [62] to the random function of $\sqrt{t}(\bar{\boldsymbol{\theta}}_{\lfloor rt \rfloor} - \boldsymbol{\theta}^*)$ for $r \in [0, 1]$. Back to the ROOT-SGD algorithm, [65] shows that the asymptotic normality holds for the estimators $\widehat{\boldsymbol{\theta}}_t$, instead of the averaging estimator. Then ROOT-SGD counterpart of (5.25) should treat $\widehat{\boldsymbol{\theta}}_i$'s as samples instead of the average $\bar{\boldsymbol{\theta}}_i$'s.

Thus, the random-scaling quantity for the ROOT-SGD is given by:

$$\frac{\sqrt{t}\boldsymbol{w}^T(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)}{\sqrt{\boldsymbol{w}^T V_t \boldsymbol{w}}}, \tag{5.26}$$

where $V_t = \frac{1}{t^2}\sum_{i=1}^t i^2(\widehat{\boldsymbol{\theta}}_i - \widehat{\boldsymbol{\theta}}_t)(\widehat{\boldsymbol{\theta}}_i - \widehat{\boldsymbol{\theta}}_t)^T$. We refer to V_t as the random-scaling estimator. In the next section, we prove the asymptotic distribution of (5.26) and explain how to use it to develop an asymptotically consistent covariance estimator for $\boldsymbol{\theta}^*$.

5.3.2 Consistency of Our Hessian-free Estimator

In this section, we prove the asymptotic distribution of the random-scaling quantity in (5.26), develop an asymptotically consistent covariance estimator based on the random-scaling quantity, and explain how to use the random-scaling estimator to conduct statistical inference for the ROOT-SGD algorithm.

The asymptotic distribution of (5.26) is proved in the following theorem:

Theorem 5.2. *Suppose Assumptions 5.1 to 5.4 hold. Then there exists constants c_1, c_2, take the step size $\eta \in \left(0, c_1(\frac{\gamma}{L_2^2} \wedge \frac{1}{L_1} \wedge \frac{\gamma^{1/3}}{L_4^{4/3}})\right)$ and burn-in period $B = \left\lceil \frac{c_2}{\gamma\eta} \right\rceil$ in the ROOT-SGD algorithm, we have for any given direction $\boldsymbol{w} \neq \mathbf{0}$ that*

$$\frac{\sqrt{t}\boldsymbol{w}^T(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)}{\sqrt{\boldsymbol{w}^T V_t \boldsymbol{w}}} \xrightarrow{d} \frac{W_1}{\sqrt{\int_0^1 (W_r - rW_1)^2 dr}}, \tag{5.27}$$

where $V_t = \frac{1}{t^2}\sum_{i=1}^t i^2(\widehat{\boldsymbol{\theta}}_i - \widehat{\boldsymbol{\theta}}_t)(\widehat{\boldsymbol{\theta}}_i - \widehat{\boldsymbol{\theta}}_t)^T$, and W_r is the standard Wiener process (i.e., $W_r \sim N(0, r)$).

The proof of Theorem 5.2 is in Section 5.5.6.

Remark 5.4. *We add a note on the computation of the random-scaling quantity in* (5.27)*. The random-scaling estimator $V_t = \frac{1}{t^2}\sum_{i=1}^t i^2(\widehat{\boldsymbol{\theta}}_i - \widehat{\boldsymbol{\theta}}_t)(\widehat{\boldsymbol{\theta}}_i - \widehat{\boldsymbol{\theta}}_t)^T$ can be efficiently com-*

puted online. To see this, consider the following decomposition.

$$V_t = \frac{1}{t^2}\sum_{i=1}^{t} i^2 \widehat{\boldsymbol{\theta}}_i \widehat{\boldsymbol{\theta}}_i^T - \frac{1}{t^2}(\sum_{i=1}^{t} i^2 \widehat{\boldsymbol{\theta}}_i)\widehat{\boldsymbol{\theta}}_t^T - \frac{1}{t^2}\widehat{\boldsymbol{\theta}}_t(\sum_{i=1}^{t} i^2 \widehat{\boldsymbol{\theta}}_i^T) + \frac{t(t+1)(2t+1)}{6t^2}\widehat{\boldsymbol{\theta}}_t \widehat{\boldsymbol{\theta}}_t^T,$$

in which $\frac{1}{t^2}\sum_{i=1}^{t} i^2 \widehat{\boldsymbol{\theta}}_i \widehat{\boldsymbol{\theta}}_i^T$ *and* $\frac{1}{t^2}\sum_{i=1}^{t} i^2 \widehat{\boldsymbol{\theta}}_i$ *are efficiently computed online without requiring storing all past estimators* $\widehat{\boldsymbol{\theta}}_i$*'s. In this way, the random-scaling quantity* $\frac{\sqrt{t}\boldsymbol{w}^T(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)}{\sqrt{\boldsymbol{w}^T V_t \boldsymbol{w}}}$ *can be computed online.*

Remark 5.5. *We explain how to drive a consistent covariance estimator from Theorem 5.2 and how to use it for statistical inference as follows. Denote the asymptotic limit* $\frac{W_1}{\sqrt{\int_0^1 (W_r - rW_1)^2 dr}}$ *as* X_{rs}*. Then* X_{rs} *is a universal random variable that does not depend on the optimization problem we study. We refer to* X_{rs} *as the random-scaling variable. Using this notation,* $\mathrm{Var}\,(X_{rs}) * \boldsymbol{w}^T V_t \boldsymbol{w}$ *is an estimator for the variance of* $\sqrt{t}\boldsymbol{w}^T(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)$*, and* $\widehat{\Sigma}_{t,rs} := (\mathrm{Var}\,(X_{rs})) * V_t$ *is a random-scaling covariance estimator for the covariance of* $\sqrt{t}(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)$*. By Theorem 5.2,* $\widehat{\Sigma}_{t,rs}$ *is asymptotically consistent. When calculating the random-scaling covariance estimator* $\widehat{\Sigma}_{t,rs}$*, we can use the estimation in [76] that* $\mathrm{Var}(X_{rs}) \approx 11.177513184$*. Moreover, [76, 74] compute the quantiles for the random variable* $X_{rs} = \left(W_1 / \sqrt{\int_0^1 (W_r - rW_1)^2 dr}\right)$ *based on direct integration. Using their computed quantiles, one can develop asymptotically consistent statistical inference for the parameter estimation, for example, do statistical testing for* $\boldsymbol{\theta}^*$ *or build the confidence interval.*

We compare our random-scaling estimator with that from the ASGD algorithm. The comparison details are in Section 5.5.7.2. The computation costs of the two estimators are the same. Both estimators are asymptotically consistent, but the convergence speed is not proven. Since the ROOT-SGD converges faster than ASGD, there might be an advantage to use the random-scaling estimator from the ROOT-SGD in practice.

5.4 Numeric Studies

In this section, we perform numerical studies using synthetic data and a real dataset. In particular, in Section 5.4.1, we show the performance of the confidence intervals constructed based on our covariance estimators for linear regression and logistic regression; in Section 5.4.2, we visualize the confidence intervals based on our covariance estimator using the hand-written digit image example.

5.4.1 Simulation

In this section, we simulate some examples of linear regression and logistic regression and compare the confidence intervals of these examples based on the plug-in and Hessian-free covariance estimators. The goal is to check if the confidence intervals have the coverage probability converging to the nominal value as the sample size increases.

Models. In linear regression model, we have data $\boldsymbol{x}^T = (\boldsymbol{a}^T, b) \in R^d \times R$, where $\boldsymbol{a}$ is the vector of explanatory variable and b is the response variable. The data is generated by the true parameter $\boldsymbol{\theta}^*$ as $b = \boldsymbol{a}^T\boldsymbol{\theta}^* + \epsilon$, where ϵ is a zero-mean r.v. that is independent of $\boldsymbol{a}$. We use the squared loss for the linear regression task

$$f(\boldsymbol{\theta}; \boldsymbol{x}) = \frac{1}{2}(\boldsymbol{a}^T\boldsymbol{\theta} - b)^2.$$

In logistic regression model, we have data $\boldsymbol{x}^T = (\boldsymbol{a}^T, b) \in R^d \times \{-1, 1\}$. The data is generated by the true parameter $\boldsymbol{\theta}^*$ as $P(b = 1) = \frac{1}{1+\exp(-\langle \boldsymbol{a}, \boldsymbol{\theta}^* \rangle)}$. Take the negative log-likelihood as the risk function

$$f(\boldsymbol{\theta}; \boldsymbol{x}) = \log(1 + \exp(-b\langle \boldsymbol{a}, \boldsymbol{\theta} \rangle)).$$

Data Generation. The linear regression and logistic regression data are generated as follows. We first set a data dimension d, and let the explanatory variables $\boldsymbol{a}_i$'s to be i.i.d.

$N(\mathbf{0}_d, I_d)$. For both models, we set the true parameter $\boldsymbol{\theta}^*$ to be d equally spaced values in $[0, 1]$. We generate the response variable b_i's based on $\boldsymbol{a}_i$'s for linear regression and logistic regression cases, respectively. In particular, in linear regression, we let the additive noises ϵ_i's be i.i.d. standard normal random variables.

Algorithm Implementation. We compare the confidence intervals built based on our covariance estimators for ROOT-SGD with those based on the plug-in covariance estimator [66], the non-overlapping batch-mean covariance estimator [68], and the random-scaling estimator [70] for ASGD. So we implement the ROOT-SGD and ASGD algorithms; the implementation details are as follows. Both algorithms are initialized as $\widehat{\boldsymbol{\theta}}_0 = \mathbf{0}$ in linear regression and logistic regression. The total number of samples (i.e., the algorithm updates) is set to $250,000$. For ROOT-SGD, we take $\eta = 10^{-3}$ for linear regression and $\eta = 5 * 10^{-3}$ for logistic regression. The burn-in period is set to $B = 1000$ for both cases. For ASGD, we take $\eta_0 = 0.5$ and $\alpha = 0.505$, which are the same as the choices in [68].

Confidence Interval Computation. We compare confidence intervals induced by different covariance estimators. In particular, we compute the 95% confidence intervals for each dimension of the parameter estimation and check how they cover the true parameters. For ROOT-SGD algorithm, denote the plug-in covariance estimator as $\widehat{\Sigma}_{t,root-pi}$ and rewrite the random-scaling estimator V_t in (5.27) as $V_{t,root}$ to distinguish it with the random-scaling estimator from ASGD. For ASGD algorithm, denote the plug-in covariance estimator from [66] as $\widehat{\Sigma}_{t,asgd-pi}$, denote the non-overlapping batch-mean covariance estimator from [68] as $\widehat{\Sigma}_{t,asgd-bm}$, and denote the random-scaling estimator in [70] (i.e., $\bar{V}_t$ in (5.25)) as $V_{t,asgd}$. Let $\widehat{\boldsymbol{\theta}}_{t,root}$ and $\widehat{\boldsymbol{\theta}}_{t,asgd}$ be the point estimators from ROOT-SGD and ASGD after t updates, respectively. The 95% confidence intervals based on each estimator are as follows:

- ROOT-SGD algorithm, plug-in estimator: $(\widehat{\boldsymbol{\theta}}_{t,root})_i \pm z_{0.975} * (\widehat{\Sigma}_{t,root-pi})_{i,i}/\sqrt{t}$;
- ROOT-SGD algorithm, random-scaling estimator: $(\widehat{\boldsymbol{\theta}}_{t,root})_i \pm q_{rs,0.975} * (V_{t,root})_{i,i}/\sqrt{t}$;
- ASGD algorithm, plug-in estimator: $(\widehat{\boldsymbol{\theta}}_{t,asgd})_i \pm z_{0.975} * (\widehat{\Sigma}_{t,asgd-pi})_{i,i}/\sqrt{t}$;
- ASGD algorithm, batch-mean estimator: $(\widehat{\boldsymbol{\theta}}_{t,asgd})_i \pm z_{0.975} * (\widehat{\Sigma}_{t,asgd-bm})_{i,i}/\sqrt{t}$;

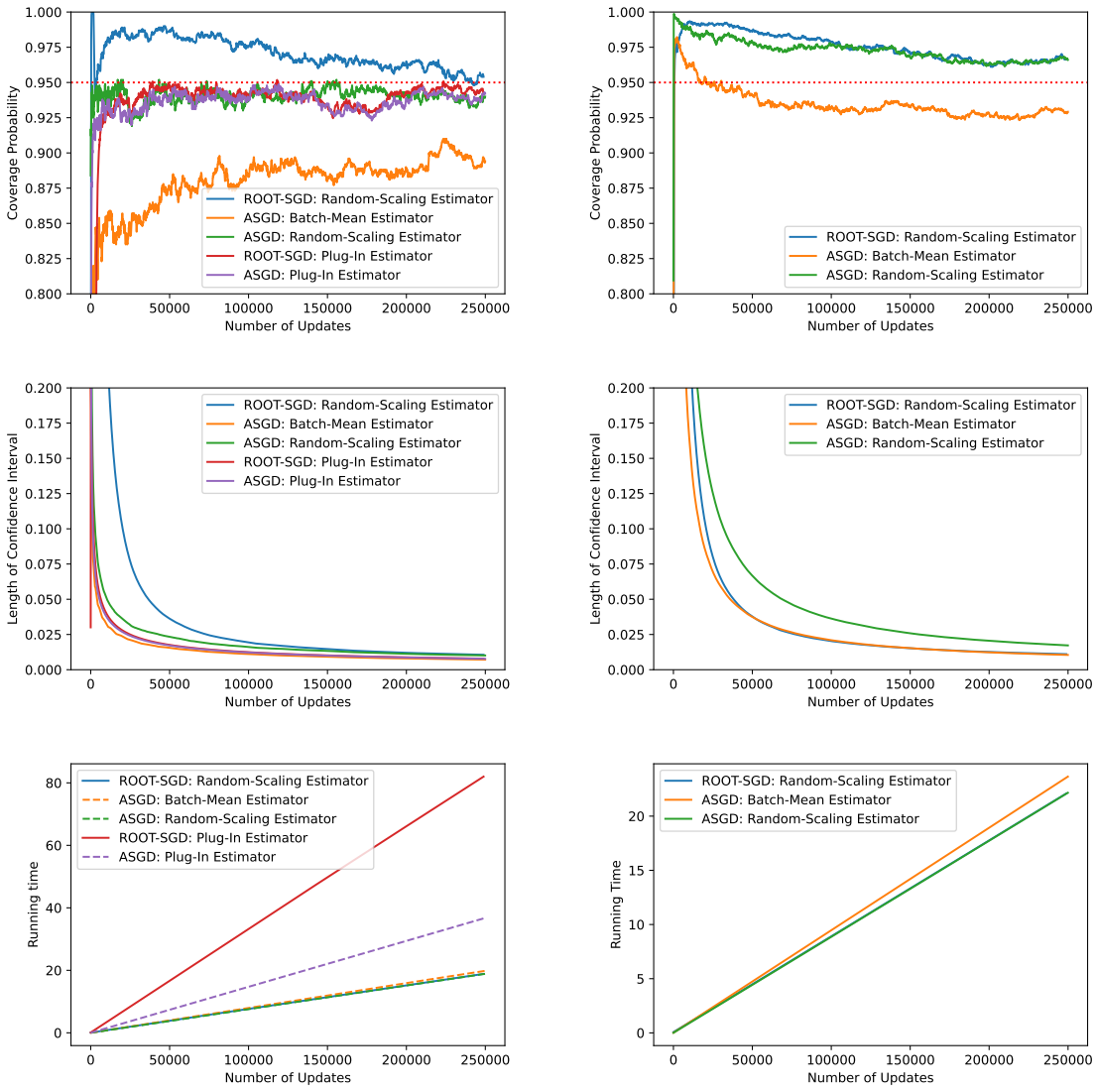
- ASGD algorithm, random-scaling estimator: $(\widehat{\boldsymbol{\theta}}_{t,asgd})_i \pm q_{rs,0.975} * (V_{t,asgd})_{i,i}/\sqrt{t}$;

where $z_{0.975}$ is the 97.5% percentile for the standard normal random variable, and $q_{rs,0.975}$ is the 97.5% percentile for $X_{rs} = \left(W_1/\sqrt{\int_0^1 (W_r - rW_1)^2 dr}\right)$, i.e., the limiting distribution of the random-scaling quantity in Theorem 5.2. By [76, 74], we have $q_{rs,0.975} = 6.747$.

Experiment Results. We run 200 repetitions of experiments for each estimator and compare the average coverage probability and length of the confidence interval from 200 runs. The results of the linear regression experiment are shown in Figure 5.1, where we have the two dimension settings: $d = 5$ and $d = 20$. Comparing the confidence intervals, we have the following.

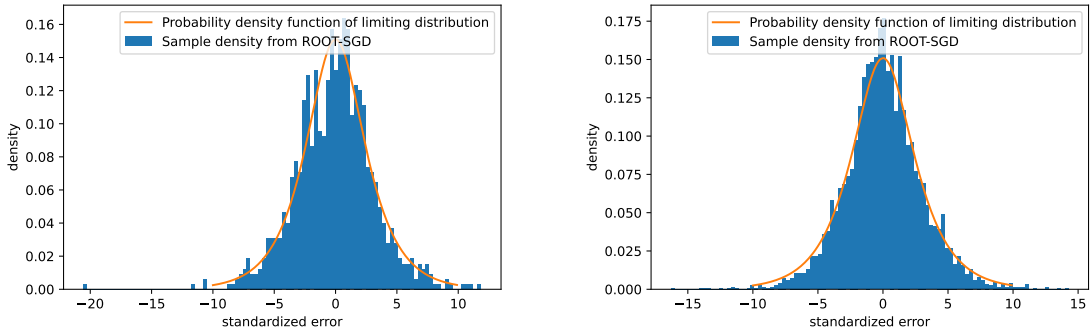
- For $d = 5$, all confidence intervals (CI) have coverage probability converging to the nominal value of 0.95, except the one corresponding to the ASGD batch-mean estimator. The coverage probability of the CI from the ROOT-SGD random-scaling estimator is higher than the counterpart from ASGD and is closer to the nominal coverage rate. For the confidence intervals from plug-in estimators for ROOT-SGD and ASGD, we observe that they have the same confidence interval length. However, the coverage probability of CI from ROOT-SGD is higher than that of ASGD and is closer to the nominal coverage rate. Thus the CIs from ROOT-SGD are better than those from ASGD. These observations show that the statistical inference of ROOT-SGD is more accurate than that of ASGD, possibly thanks to the fast convergence of ROOT-SGD.

- For $d = 20$, due to the large computation of the plug-in estimators for both ROOT-SGD and ASGD, we do not compute them. We compare the coverage of CIs from different Hessian-free estimators and conclude that the coverage probability of the CI from the ASGD batch mean estimator does not reach the nominal rate. The coverage probabilities of the CIs from both ROOT-SGD and ASGD random-scaling estimators are higher than the nominal rate. However, the CI length of the ROOT-SGD random-



(a) d = 5 (b) d = 20

Figure 5.1: Comparison of the confidence intervals in linear regression. The first row is the coverage probability against the number of updates, with a red dashed line denoting the nominal coverage rate of 0.95. The second row is the average length of the confidence interval. The last row is the total computation time to update the covariance estimator and the confidence interval.



(a) d = 5 (b) d = 20

Figure 5.2: Density of the Random Scaling Quantity.

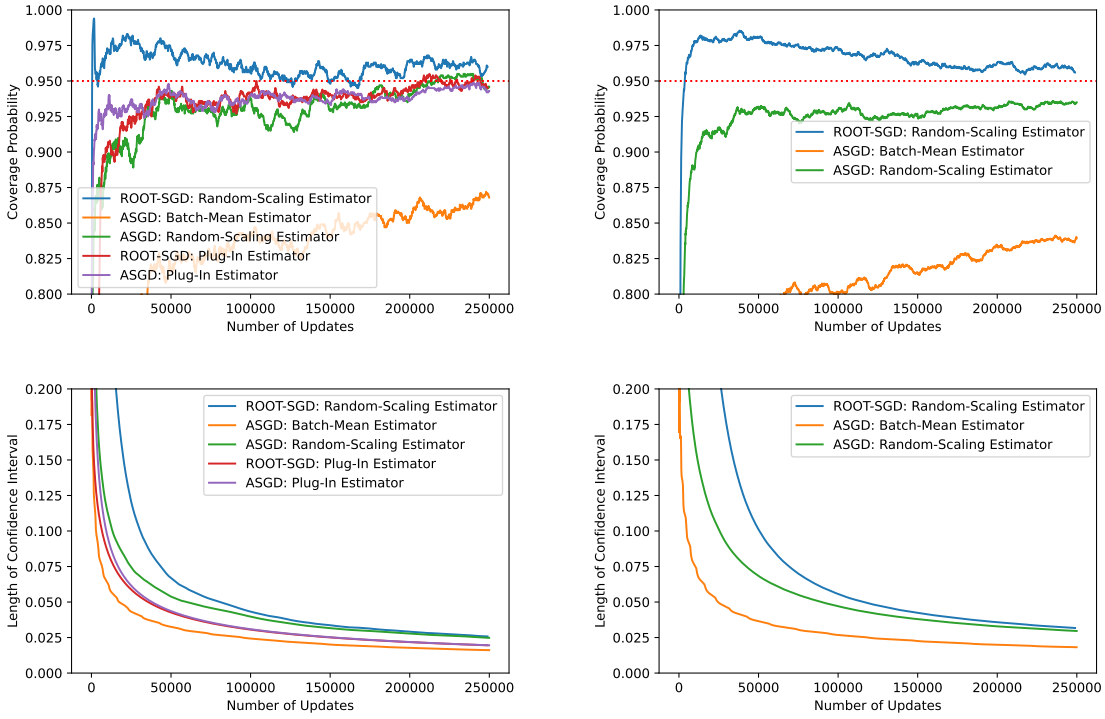
scaling estimator is smaller than that of ASGD, which shows the advantage of the ROOT-SGD algorithm in building the confidence interval.

- In the running time comparison, we can see that the plug-in estimators take more time than the Hessian-free estimators (including the batch mean estimator and the random-scaling estimator), which is not surprising due to the second-order nature of the plug-in estimators. The random-scaling estimators from ROOT-SGD and SGD take almost the same time, as they have lines on top of each other. The batch-mean estimator takes slightly more time than the random-scaling estimator.

For the linear regression example, we further check the empirical distribution of the random-scaling quantity of the ROOT-SGD algorithm (i.e., the l.h.s. of (5.27)) to see if it matches the theoretic limiting distribution X_{rs} (i.e., the r.h.s. of (5.27)). In particular, we let $\boldsymbol{w}$ be the standard basis vectors in the random-scaling quantity. The result is shown in Figure 5.2. For both $d = 5$ and $d = 20$, we can see that the empirical distribution is close to the theoretical limiting distribution.

The experiment result for logistic regression are in Figure 5.3. For $d = 5$, all confidence intervals (CI) have the coverage probability converge to the nominal value of 0.95, except the SGD batch mean estimator – it has a much lower coverage probability than the

nominal probability. The plug-in estimator and the random-scaling estimator from ROOT-SGD have better coverage than the SGD counterparts while having the same length of the confidence interval. For $d = 20$, the ROOT-SGD random-scaling estimator has a coverage probability that converges faster to the nominal rate compared to the estimators from SGD. Furthermore, the ROOT-SGD random-scaling estimator has a higher coverage probability than the SGD random-scaling estimator, although their CI lengths are comparable after $200,000$ updates. The observations above all show the advantage of statistical inference by ROOT-SGD.



(a) d = 5 (b) d = 20

Figure 5.3: Comparison of the confidence intervals estimators in logistics regression. The first row is the coverage probability against the number of updates, and the red dashed line denotes the nominal coverage rate of 0.95. The second row is the average length of the confidence interval.

5.4.2 Hand-Written Digit Analysis

This experiment aims to visualize the confidence interval estimator that is induced by our plug-in covariance estimator. In particular, we consider the parameter estimation problem of mean estimation for the MNIST hand-written digit image set.

The data-set description is as follows. There are 60000 training images of dimension $28 * 28$, each labeled as a digit between 0 and 9. For each label, we assume that it has a mean image and that the image instances are samples from a normal mean model.

We apply the ROOT-SGD algorithm to estimate the mean image and use our plug-in covariance estimator to estimate the covariance for each digit. The implementation details are as follows. Recall that the risk function for normal mean estimation is $f(\boldsymbol{\theta}; \boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{x}\|_2^2$. Under this risk function, a stochastic gradient in ROOT-SGD is calculated by randomly sampling data from the training set of that digit. When implementing the ROOT-SGD algorithm, we set the step size $\eta = 0.05$, the burn-in period $B = 10,000$, and the number of samples $t = 100,000$. We calculate the plug-in estimator for the covariance of the mean. Note that for this risk function, we have $\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) \equiv \nabla^2 F(\boldsymbol{\theta}^*) = I$. Thus, the plug-in estimator for the asymptotic covariance of ROOT-SGD reduces to $\widehat{A}_t^{-1}\widehat{S}_t\widehat{A}_t^{-1}$. This allows a fast computation of the covariance estimator.

Using the mean image estimate and plug-in covariance estimate, we further compute the confidence interval for the mean image. We show examples of mean image estimation and the 95% confidence bound of mean estimation of digits 0,1,2 in Figure 5.4. In Figure 5.4, the confidence intervals all look reasonable.

5.5 Technique Details for Chapter 5

In this section, we provide the technique details for Chapter 5.

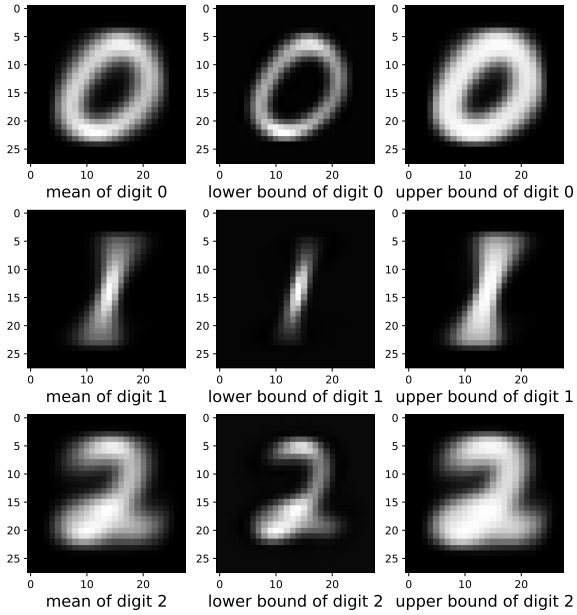


Figure 5.4: MNIST example: the mean and confidence intervals are reasonably estimated in an online fashion.

5.5.1 Proof for Proposition 5.1

In this section, we prove Proposition 5.1.

Proof. We start the proof by writing the equivalent condition for inequality (5.16).

$$
\begin{aligned}
(5.16) &\iff \|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}_t^*\|_2^2 \leq (1-\eta\alpha)^2 \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2^2 \\
&\iff \left\| \widehat{\boldsymbol{\theta}}_{t-1} - \eta \left(\nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \nabla B(\boldsymbol{\theta}_t^*) \right) - \boldsymbol{\theta}_t^* \right\|_2^2 \leq (1-\eta\alpha)^2 \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2^2 \\
&\iff \left\| \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \nabla B(\boldsymbol{\theta}_t^*) \right\|_2^2 - \frac{2}{\eta} \left\langle \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \nabla B(\boldsymbol{\theta}_t^*), \widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^* \right\rangle \\
&\qquad + \alpha(\frac{2}{\eta} - \alpha) \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2^2 \leq 0
\end{aligned}
$$

Since $B(\cdot)$ is smooth and strongly convex, by some standard equivalent conditions of smoothness and strongly convexity (for example, Lemma 2 and Lemma 4 in [77]), we

have

$$\begin{aligned}&\left\langle \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \nabla B(\boldsymbol{\theta}_t^*), \widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^* \right\rangle \\ \geq & \mu \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2^2 \geq \frac{\mu}{l} \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2 \left\| \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \nabla B(\boldsymbol{\theta}_t^*) \right\|_2 .\end{aligned}$$

For the convenience of proof, we define a new quantity $\kappa = \frac{\eta(\mu+l)}{2}$. Then by our range of η, we immediately have $\kappa \in (0, \frac{\mu}{l}]$. Using this defined quantity, a sufficient condition for (5.16) to hold is:

$$\begin{aligned}&\left\| \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \nabla B(\boldsymbol{\theta}_t^*) \right\|_2^2 - \frac{2\kappa}{\eta} \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2 \left\| \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \nabla B(\boldsymbol{\theta}_t^*) \right\|_2 \\ &\quad + \alpha(\frac{2}{\eta} - \alpha) \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2^2 \leq 0. \end{aligned} \tag{5.28}$$

Take $\alpha = \frac{\mu+l-\sqrt{(\mu+l)^2-4\mu l\kappa^2}}{2\kappa} \in (0, \frac{1}{\eta})$, then we have

$$\begin{aligned}&\left\| \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \nabla B(\boldsymbol{\theta}_t^*) \right\|_2^2 - \frac{2\kappa}{\eta} \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2 \left\| \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \nabla B(\boldsymbol{\theta}_t^*) \right\|_2 \\ &\quad + \alpha(\frac{2}{\eta} - \alpha) \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2^2 \\ =& \left(\left\| \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \nabla B(\boldsymbol{\theta}_t^*) \right\|_2 - \mu \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2 \right) \\ &\quad * \left(\left\| \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \nabla B(\boldsymbol{\theta}_t^*) \right\|_2 - l \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2 \right) \\ \leq & 0, \end{aligned} \tag{5.29}$$

where (5.29) by $\mu \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2 \leq \left\| \nabla B(\widehat{\boldsymbol{\theta}}_{t-1}) - \nabla B(\boldsymbol{\theta}_t^*) \right\|_2 \leq l \|\widehat{\boldsymbol{\theta}}_{t-1} - \boldsymbol{\theta}_t^*\|_2$. This conclude the proof for Proposition 5.1. □

5.5.2 Connections Between Assumptions

In Section 5.2.2, we list several assumptions. Among those assumptions, Assumptions 5.5 and 5.6 are consequences of other assumptions. In this section, we prove such statement.

Proof. We check that Assumption 5.5 is implied by Assumptions 5.1 and 5.3 as follows. Assume Assumptions 5.1 and 5.3 hold, then we can bound

$$\begin{aligned}
&\mathbb{E}\|\nabla f(\boldsymbol{\theta}_1;\boldsymbol{x}) - \nabla f(\boldsymbol{\theta}_2;\boldsymbol{x})\|_2^2 \\
=&\|\nabla F(\boldsymbol{\theta}_1) - \nabla F(\boldsymbol{\theta}_2)\|_2^2 + \mathbb{E}\|\delta(\boldsymbol{\theta}_1;\boldsymbol{x}) - \delta(\boldsymbol{\theta}_2;\boldsymbol{x})\|_2^2 \\
\leq&(L_1^2 + L_2^2)\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2.
\end{aligned}$$

Thus, Assumption 5.5 must hold with $L_5^2 = L_1^2 + L_2^2$.

We check that Assumption 5.6 is implied by Assumption 5.4 as follows. Assume Assumption 5.4 holds, then the first inequality in Assumption 5.6 can be derived by:

$$\begin{aligned}
&\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2 \\
\leq&\mathbb{E}\|(\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})) \otimes (\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}))\|_2 \\
&+ \mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) \otimes (\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}))\|_2 \\
&+ \mathbb{E}\|(\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})) \otimes \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2 \\
=&\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^2 + 2\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2 \\
\leq&L_3^2\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2 + 2\sqrt{\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^2\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^2} \\
\leq&L_3^2\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2 + 2L_3L_4^2\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2.
\end{aligned} \tag{5.30}$$

And the second inequality of Assumption 5.6 can be obtained as:

$$\begin{aligned}
&\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) - \mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})]\|_2^2 \\
\leq&\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^2 = \mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^4 \leq L_4^4.
\end{aligned} \tag{5.31}$$

In this way, Assumption 5.6 must hold with $L_6 = 2L_3L_4^2$, $L_6' = L_3^2$, and $L_7 = L_4^4$. □

5.5.3 Check Assumptions for Common Examples

In this section, we check that all assumptions in Section 5.2.2 hold for the common parameter estimation problems of exponential family model, linear regression and logistic regression.

5.5.3.1 Exponential Family Model

We consider the exponential family model with natural parameters, where the likelihood function is

$$L(\boldsymbol{\theta}; \boldsymbol{x}) = h(\boldsymbol{x}) \exp[\langle \boldsymbol{\theta}, \boldsymbol{T}(\boldsymbol{x}) \rangle - B(\boldsymbol{\theta})].$$

Let the risk function be the negative log-likelihood

$$f(\boldsymbol{\theta}; \boldsymbol{x}) = -\langle \boldsymbol{\theta}, \boldsymbol{T}(\boldsymbol{x}) \rangle + B(\boldsymbol{\theta}).$$

Then

$$\boldsymbol{\theta}^* = \arg\min F(\boldsymbol{\theta}) = -\langle \boldsymbol{\theta}, \mathbb{E}\boldsymbol{T}(\boldsymbol{x}) \rangle + B(\boldsymbol{\theta}),$$

thus $\mathbb{E}T(\boldsymbol{x}) = \nabla B(\boldsymbol{\theta}^*)$. Compute that

$$\nabla f(\boldsymbol{\theta}; \boldsymbol{x}) = -\boldsymbol{T}(\boldsymbol{x}) + \nabla B(\boldsymbol{\theta});$$

$$\nabla^2 f(\boldsymbol{\theta}; \boldsymbol{x}) = \nabla^2 B(\boldsymbol{\theta}).$$

$$\nabla F(\boldsymbol{\theta}) = -\mathbb{E}\boldsymbol{T}(\boldsymbol{x}) + \nabla B(\boldsymbol{\theta});$$

$$\nabla^2 F(\boldsymbol{\theta}) = \nabla^2 B(\boldsymbol{\theta}).$$

Suppose that $B(\cdot)$ is μ-strongly convex and l-smooth, $\nabla^2 B(\cdot)$ is l'-Lipchitz continuous, and the true parameter $\boldsymbol{\theta}^*$ is taken such that the second and fourth moment of $\boldsymbol{T}(\boldsymbol{x})$ are bounded. Then we can check that all assumptions in Section 5.2.2 hold as follows.

- **For Assumption 5.1:** $\nabla^2 F(\boldsymbol{\theta}) = \nabla^2 B(\boldsymbol{\theta})$, so $F(\cdot)$ is γ-strongly convex and L_1-smooth for $\gamma = \mu$, $L_1 = l$.

- **For Assumption 5.2:** It is clear that $\nabla f(\boldsymbol{\theta}; \boldsymbol{x})$ is unbiased, and we check its covariance at $\boldsymbol{\theta}^*$

$$\mathbb{E}[\|\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x}) - \nabla F(\boldsymbol{\theta}^*)\|_2^2] = \mathbb{E}\|\boldsymbol{T}(\boldsymbol{x}) - \mathbb{E}\boldsymbol{T}(\boldsymbol{x})\|_2^2 \leq \sigma_1^2$$

for a $\sigma_1^2 < \infty$ since the second moment of $\boldsymbol{T}(\boldsymbol{x})$ is bounded.

- **For Assumption 5.3:** We have $\delta(\boldsymbol{\theta}; \boldsymbol{x}) = \mathbb{E}\boldsymbol{T}(\boldsymbol{x}) - \boldsymbol{T}(\boldsymbol{x})$, then

$$\mathbb{E}\|\delta(\boldsymbol{\theta}_1; \boldsymbol{x}) - \delta(\boldsymbol{\theta}_2; \boldsymbol{x})\|_2^2 = 0,$$

so Assumption 5.3 holds with $L_2^2 = 0$.

- **For Assumption 5.4:**

$$\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}; \boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2^2 = \|\nabla^2 B(\boldsymbol{\theta}) - \nabla^2 B(\boldsymbol{\theta}^*)\|_2^2 \leq (l')^2 \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2,$$

$$\mathbb{E}\|\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2^4 = \mathbb{E}\|\nabla B(\boldsymbol{\theta}^*) - \boldsymbol{T}(\boldsymbol{x})\|_2^4 = \mathbb{E}\|\mathbb{E}[\boldsymbol{T}(\boldsymbol{x})] - \boldsymbol{T}(\boldsymbol{x})\|_2^4 \leq l_4^4,$$

for a $l_4^4 < \infty$ since the fourth moment of $\boldsymbol{T}(\boldsymbol{x})$ is bounded.

$$\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2^4 = \|\nabla^2 B(\boldsymbol{\theta}^*)\|_2^4 \leq l^4.$$

So Assumption 5.4 holds with $L_3^2 = (l')^2$, $L_4^4 = l^4$.

- **For Assumption 5.5:**

$$\mathbb{E}\|\nabla f(\boldsymbol{\theta}_1; \boldsymbol{x}) - \nabla f(\boldsymbol{\theta}_2; \boldsymbol{x})\|_2^2 = \|\nabla B(\boldsymbol{\theta}_1) - \nabla B(\boldsymbol{\theta}_2)\|_2^2 \leq l^2 \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2.$$

So Assumption 5.5 holds with $L_5^2 = l^2$.

- **For Assumption 5.6:** By Remark 5.2, we have

$$\begin{aligned}&\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x})\otimes\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2\\ \leq& L_3^2\|\boldsymbol{\theta}-\boldsymbol{\theta}^*\|_2^2 + 2L_3L_4^2\|\boldsymbol{\theta}-\boldsymbol{\theta}^*\|_2.\end{aligned}$$

And we check that

$$\begin{aligned}&\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) - \mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})]\|_2^2\\ \leq&\mathbb{E}\|\nabla^2 B(\boldsymbol{\theta}^*)\otimes\nabla^2 B(\boldsymbol{\theta}^*) - \mathbb{E}\nabla^2 B(\boldsymbol{\theta}^*)\otimes\nabla^2 B(\boldsymbol{\theta}^*)\|_2^2 = 0.\end{aligned}$$

So Assumption 5.6 holds with $L_6' = L_3^2 = (l')^2, L_6 = 2L_3L_4^2 = 2l'l^2$ and $L_7 = 0$.

5.5.3.2 Linear Regression Model

Consider the standard linear regression framework, where data $\boldsymbol{x}_i^T = (\boldsymbol{a}_i^T, b_i)$. The input $\boldsymbol{a}_i \in R^p$ are random samples from the same multivariate distribution, and the response $b_i = \boldsymbol{a}_i^T\boldsymbol{\theta}^* + \epsilon_i$, where $\boldsymbol{\theta}^*$ is the true parameter and ϵ_i's are i.i.d. centered random variables independent of $\boldsymbol{a}_i$'s. In particular, to track the dependence of Theorem 5.1 on the problem dimension p, we further assume $\boldsymbol{a} \sim N(0, I_p), \epsilon \sim N(0,1)$.

Use the squared loss

$$f(\boldsymbol{\theta};\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{a}^T\boldsymbol{\theta} - b)^2.$$

Then

$$\boldsymbol{\theta^*} = \arg\min F(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{x}}\left[\frac{1}{2}(\boldsymbol{a}^T\boldsymbol{\theta} - b)^2\right].$$

Compute that

$$\nabla f(\boldsymbol{\theta};\boldsymbol{x}) = \boldsymbol{a}(\boldsymbol{a}^T\boldsymbol{\theta} - b) = \boldsymbol{a}\boldsymbol{a}^T(\boldsymbol{\theta}-\boldsymbol{\theta}^*) - \boldsymbol{a}\epsilon;$$

$$\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) = \boldsymbol{a}\boldsymbol{a}^T;$$

$$\nabla F(\boldsymbol{\theta}) = \mathbb{E}[\nabla f(\boldsymbol{\theta}; \boldsymbol{x})] = \mathbb{E}[\boldsymbol{a}\boldsymbol{a}^T](\boldsymbol{\theta} - \boldsymbol{\theta}^*);$$

$$\nabla^2 F(\boldsymbol{\theta}) = \mathbb{E}[\nabla^2 f(\boldsymbol{\theta}; \boldsymbol{x})] = \mathbb{E}[\boldsymbol{a}\boldsymbol{a}^T] = I_p.$$

We check assumptions in Section 5.2.2:

- **For Assumption 5.1:** Since $\nabla^2 F(\boldsymbol{\theta}) = I_p$, $F(\boldsymbol{\theta})$ is γ-strongly convex and L_1-smooth with $\gamma = L_1 = 1$.

- **For Assumption 5.2:** It is clear that $\nabla f(\boldsymbol{\theta}; \boldsymbol{x})$ is unbiased, and we check its covariance at $\boldsymbol{\theta}^*$

  $$\mathbb{E}[\|\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x}) - \nabla F(\boldsymbol{\theta}^*)\|_2^2] = \mathbb{E}\|\boldsymbol{a}\epsilon\|_2^2 = p,$$

  so Assumption 5.2 holds with $\sigma_1^2 = p$.

- **For Assumption 5.3:** We have $\delta(\boldsymbol{\theta}; \boldsymbol{x}) = (\boldsymbol{a}\boldsymbol{a}^T - I_p)(\boldsymbol{\theta} - \boldsymbol{\theta}^*) - \boldsymbol{a}\epsilon$, then

  $$\mathbb{E}\|\delta(\boldsymbol{\theta}_1; \boldsymbol{x}) - \delta(\boldsymbol{\theta}_2; \boldsymbol{x})\|_2^2 = \mathbb{E}\|(\boldsymbol{a}\boldsymbol{a}^T - I_p)(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)\|_2^2 \leq \mathbb{E}\|\boldsymbol{a}\boldsymbol{a}^T - I_p\|_2^2 \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2,$$

  where $\mathbb{E}\|\boldsymbol{a}\boldsymbol{a}^T - I_p\|_2^2 \leq 2\mathbb{E}\|\boldsymbol{a}\boldsymbol{a}^T\|_2^2 + 2\|I_p\|_2^2 = 2p + 2$, so Assumption 5.3 holds with $L_2^2 = 2p + 2$.

- **For Assumption 5.4:**

  $$\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}; \boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2^2 = \mathbb{E}\|\boldsymbol{a}\boldsymbol{a}^T - \boldsymbol{a}\boldsymbol{a}^T\|_2^2 = 0,$$

  $$\mathbb{E}\|\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2^4 = \mathbb{E}\|\boldsymbol{a}\epsilon\|_2^4 \leq \mathbb{E}\|\boldsymbol{a}\|_2^4 \|\epsilon\|_2^4 = \mathbb{E}\|\boldsymbol{a}\|_2^4 \mathbb{E}\|\epsilon\|_2^4 = 3(2p + p^2),$$

  $$\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2^4 = \mathbb{E}\|\boldsymbol{a}\boldsymbol{a}^T\|_2^4 = \mathbb{E}\|\boldsymbol{a}\|_2^8 = p(p+2)(p+4)(p+6).$$

  Assumption 5.4 holds with $L_3 = 0, l_4^4 = 3p(p+2), L_4^4 = p(p+2)(p+4)(p+6)$.

- **For Assumption 5.5:**

$$\mathbb{E}\|\nabla f(\boldsymbol{\theta}_1;\boldsymbol{x}) - \nabla f(\boldsymbol{\theta}_2;\boldsymbol{x})\|_2^2 = \mathbb{E}\|\boldsymbol{a}\boldsymbol{a}^T(\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2)\|_2^2 \leq \mathbb{E}\|\boldsymbol{a}\boldsymbol{a}^T\|_2^2\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2.$$

So Assumption 5.5 holds with $L_5^2 = \mathbb{E}\|\boldsymbol{a}\boldsymbol{a}^T\|_2^2 = \mathbb{E}\|\boldsymbol{a}\|_2^4 = 2p + p^2$.

- **For Assumption 5.6:**

$$\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta};\boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2 = 0,$$

$$\begin{aligned}&\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) - \mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})]\|_2^2 \\ \leq&\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^2 \leq p(p+2)(p+4)(p+6).\end{aligned}$$

Assumption 5.6 holds with $L_6 = L_6' = 0, L_7 = p(p+2)(p+4)(p+6)$.

For linear regression, we have the constant in Theorem 5.1 is

$$C_p \lesssim \max\{\sigma_1/\gamma, L_3, \sqrt{L_4}, l_4, \sqrt{L_5}, L_6\} \asymp \sqrt{p}.$$

5.5.3.3 Logistic Regression Model

Consider the logistic regression model as follows. The data sample $\boldsymbol{x}_i^T = (\boldsymbol{a}_i^T, b_i) \in R^p \times \{-1, 1\}$. Suppose that $\boldsymbol{a} \sim N(0, I_p)$, and the data pair $(\boldsymbol{a}_i^T, b_i)$ is related by a $\boldsymbol{\theta}^* \in R^p$ such that $P(b_i = 1) = \frac{1}{1+\exp(-\langle \boldsymbol{a}_i, \boldsymbol{\theta}^* \rangle)}$. Then the negative log-likelihood as objective function

$$f(\boldsymbol{\theta};\boldsymbol{x}) = \log(1 + \exp(-b\langle \boldsymbol{a}, \boldsymbol{\theta} \rangle)),$$

then

$$\nabla f(\boldsymbol{\theta};\boldsymbol{x}) = -\frac{b\boldsymbol{a}}{1 + \exp(b\langle \boldsymbol{a}, \boldsymbol{\theta} \rangle)},$$

$$\nabla^2 f(\boldsymbol{\theta}; \boldsymbol{x}) = \frac{\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle)\boldsymbol{a}\boldsymbol{a}^T}{(1+\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))^2} = \frac{\boldsymbol{a}\boldsymbol{a}^T}{(1+\exp(\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))(1+\exp(-\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))}.$$

We check that all assumptions hold for logistic regression example.

- **For Assumption 5.1:** Calculate that

$$\nabla^2 F(\boldsymbol{\theta}) = \mathbb{E}\left[\frac{\boldsymbol{a}\boldsymbol{a}^T}{(1+\exp(\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))(1+\exp(-\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))}\right].$$

  Then by Lemma A.3. in [66], $\nabla^2 F(\boldsymbol{\theta})$ is positive definite; thus $F(\boldsymbol{\theta})$ is strongly convex.

  Furthermore, we have $\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle)/(1+\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))^2 \leq 1/4$. Thus

$$\nabla^2 F(\boldsymbol{\theta}) = \mathbb{E}\left[\frac{\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle)\boldsymbol{a}\boldsymbol{a}^T}{(1+\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))^2}\right] \preceq \frac{1}{4}\mathbb{E}[\boldsymbol{a}\boldsymbol{a}^T] = \frac{1}{4}I_p.$$

  $F(\boldsymbol{\theta})$ is $1/4$-smooth.

- **For Assumption 5.2:** It is clear that $\nabla f(\boldsymbol{\theta}; \boldsymbol{x})$ is unbiased, and we check its covariance at $\boldsymbol{\theta}^*$

$$\begin{aligned}
&\mathbb{E}[\|\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x}) - \nabla F(\boldsymbol{\theta}^*)\|_2^2] \\
=&\mathbb{E}[\|\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x}) - \mathbb{E}\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2^2] \\
\leq&\mathbb{E}[\|\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2^2] \\
=&\mathbb{E}\left[\left|-\frac{b}{1+\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}^*\rangle)}\right| * \|\boldsymbol{a}\|_2^2\right] \\
\leq&\mathbb{E}\|\boldsymbol{a}\|_2^2 = p
\end{aligned}$$

  so Assumption 5.2 holds with $\sigma_1^2 = p$.

- **For Assumption 5.3:** We have $\delta(\boldsymbol{\theta}; \boldsymbol{x}) = \nabla f(\boldsymbol{\theta}; \boldsymbol{x}) - \nabla F(\boldsymbol{\theta})$, then

$$\mathbb{E}\|\delta(\boldsymbol{\theta}_1; \boldsymbol{x}) - \delta(\boldsymbol{\theta}_2; \boldsymbol{x})\|_2^2 = 2\mathbb{E}\|\nabla f(\boldsymbol{\theta}_1; \boldsymbol{x}) - \nabla f(\boldsymbol{\theta}_2; \boldsymbol{x})\|_2^2 + 2\|\nabla F(\boldsymbol{\theta}_1) - \nabla F(\boldsymbol{\theta}_2)\|_2^2.$$

As we have checked in Assumption 5.1,

$$\|\nabla F(\boldsymbol{\theta}_1) - \nabla F(\boldsymbol{\theta}_2)\|_2^2 \leq \frac{1}{16}\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2.$$

Since $\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle)/(1+\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))^2 \leq 1/4$, we have

$$\nabla^2 f(\boldsymbol{\theta}; \boldsymbol{x}) = \frac{\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle)\boldsymbol{a}\boldsymbol{a}^T}{(1+\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))^2} \preceq \frac{1}{4}\boldsymbol{a}\boldsymbol{a}^T.$$

Thus

$$\mathbb{E}\|\nabla f(\boldsymbol{\theta}_1; \boldsymbol{x}) - \nabla f(\boldsymbol{\theta}_2; \boldsymbol{x})\|_2^2 \leq \mathbb{E}\|\boldsymbol{a}\boldsymbol{a}^T/4\|_2^2\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2 = (2p+p^2)/16\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2.$$

Assumption 5.3 holds with $L_2^2 = (1+2p+p^2)/8$, thus $L_2 \sim \mathcal{O}(p)$.

- **For Assumption 5.4:**

$$\begin{aligned}
&\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}; \boldsymbol{x}) - \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2^2 \\
=&\mathbb{E}\left\|\frac{\boldsymbol{a}\boldsymbol{a}^T}{(1+\exp(\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))(1+\exp(-\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))} - \frac{\boldsymbol{a}\boldsymbol{a}^T}{(1+\exp(\langle \boldsymbol{a}, \boldsymbol{\theta}^*\rangle))(1+\exp(-\langle \boldsymbol{a}, \boldsymbol{\theta}^*\rangle))}\right\|_2^2 \\
=&\mathbb{E}\Bigg\{\left(\frac{1}{(1+\exp(\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))(1+\exp(-\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))} - \frac{1}{(1+\exp(\langle \boldsymbol{a}, \boldsymbol{\theta}^*\rangle))(1+\exp(-\langle \boldsymbol{a}, \boldsymbol{\theta}^*\rangle))}\right)^2 \\
&* \|\boldsymbol{a}\boldsymbol{a}^T\|_2^2\Bigg\} \\
\leq&\frac{1}{16}\mathbb{E}(\langle \boldsymbol{a}, \boldsymbol{\theta}^*\rangle - \langle \boldsymbol{a}, \boldsymbol{\theta}\rangle)^2 * \|\boldsymbol{a}\|_2^4 \\
\leq&\frac{1}{16}\mathbb{E}\|\boldsymbol{a}\|_2^6\|\boldsymbol{\theta}^* - \boldsymbol{\theta}\|_2^2 = [p(p+2)(p+4)/16]\|\boldsymbol{\theta}^* - \boldsymbol{\theta}\|_2^2
\end{aligned}$$

$$\mathbb{E}\|\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2^4 \leq \mathbb{E}\|\boldsymbol{a}\|_2^4 = 2p + p^2,$$

$$\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2^4 \leq \mathbb{E}\|\boldsymbol{a}\boldsymbol{a}^T/4\|_2^4 = \mathbb{E}\|\boldsymbol{a}\|_2^8/256 = p(p+2)(p+4)(p+6)/256.$$

Assumption 5.4 holds with $L_3^2 = p(p+2)(p+4)/16, l_4^4 = p(p+2), L_4^4 = p(p+$

$2)(p+4)(p+6)/256$. So $L_3 \sim \mathcal{O}(p^{1.5})$, $l_4 \sim \mathcal{O}(p^{.5})$, $L_4 \sim \mathcal{O}(p)$.

- **For Assumption 5.5:** Since $\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle)/(1+\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))^2 \leq 1/4$, we have

$$\nabla^2 f(\boldsymbol{\theta}; \boldsymbol{x}) = \frac{\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle)\boldsymbol{a}\boldsymbol{a}^T}{(1+\exp(b\langle \boldsymbol{a}, \boldsymbol{\theta}\rangle))^2} \preceq \frac{1}{4}\boldsymbol{a}\boldsymbol{a}^T.$$

  Thus

$$\mathbb{E}\|\nabla f(\boldsymbol{\theta}_1; \boldsymbol{x}) - \nabla f(\boldsymbol{\theta}_2; \boldsymbol{x})\|_2^2 \leq \mathbb{E}\|\boldsymbol{a}\boldsymbol{a}^T/4\|_2^2 \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2^2.$$

  So Assumption 5.5 holds with $L_5^2 = \mathbb{E}\|\boldsymbol{a}\boldsymbol{a}^T\|_2^2/16 = \mathbb{E}\|\boldsymbol{a}\|_2^4/16 = (2p+p^2)/16$, thus $L_5 \sim \mathcal{O}(p)$.

- **For Assumption 5.6:** By Remark 5.2, Assumption 5.6 holds with $L_6' = L_3^2$, $L_6 = 2L_3L_4^2$ and $L_7 = L_4^4$. So $L_6' \sim \mathcal{O}(p^3)$, $L_6 \sim \mathcal{O}(p^{3.5})$, $L_7 \sim \mathcal{O}(p^4)$.

5.5.4 Ancillary Lemmas and Proof for Theorem 5.1

In this section, we prove Theorem 5.1. We first provide some ancillary lemmas that will be used to prove Theorem 5.1 in Section 5.5.4.1. Then we give the proof of Theorem 5.1 in Section 5.5.4.2.

5.5.4.1 Lemmas

We first review some ancillary lemmas that will be useful in proving Theorem 5.1.

Lemma 5.1 (Implication of strong convexity)**.** *The following condition is implied by $\gamma-$strong convexity for a differentiable function F:*

$$\|\nabla F(\boldsymbol{\theta}_1) - \nabla F(\boldsymbol{\theta}_2)\|_2 \geq \gamma\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2, \forall \boldsymbol{\theta}_1, \boldsymbol{\theta}_2.$$

Lemma 5.2 (Kronecker product rule)**.** *We have*

$$\mathrm{Vec}(ABC) = (C^T \otimes A)\,\mathrm{Vec}(B).$$

Suppose that A and B are square matrices of size m and n, respectively. Let $\lambda_1, \ldots, \lambda_m$ be the eigenvalues of A and $\mu_1, \ldots, \mu_n$ be the eigenvalues of B, then the eigenvalues of $A \otimes B$ are

$$\lambda_i \mu_j, i = 1, \ldots, m, j = 1, \ldots, n.$$

Thus

$$\|A \otimes B\|_2 = \|A\|_2 \|B\|_2,$$

and

$$\|AB\|_F = \|\operatorname{Vec}(AB)\|_2 = \|(I \otimes A)\operatorname{Vec}(B)\|_2 \leq \|I \otimes A\|_2 \|\operatorname{Vec}(B)\|_2 = \|A\|_2 \|B\|_F,$$

similarly,

$$\|AB\|_F \leq \|B\|_2 \|A\|_F.$$

Lemma 5.3 (Matrix perturbation for inverse, Lemma C.1 in [66])**.** *Let $B = A + E$ and assume that A, B are invertible. If $\|A^{-1}E\|_2 < 1/2$, we have*

$$\|B^{-1} - A^{-1}\|_2 \leq 2\|E\|_2 \|A^{-1}\|_2^2. \tag{5.32}$$

Lemma 5.4 (Asymptotic convergence of ROOT-SGD, Proposition 2 in [65])**.** *Suppose that Assumptions 5.1 to 5.4 hold. Then there exists constants c_1, c_2, given the step size $\eta \in \left(0, c_1(\frac{\gamma}{L_2^2} \wedge \frac{1}{L_1} \wedge \frac{\gamma^{1/3}}{L_4^{4/3}})\right)$ and burn-in period $B = \left\lceil \frac{c_2}{\gamma\eta} \right\rceil$ we have*

$$\sqrt{t}(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*) \xrightarrow{d} \mathcal{N}(0, \Sigma)$$

for $\Sigma = A^{-1}(S + \mathbb{E}[\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)])A^{-1}$, where $\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}) = \nabla^2 f(\boldsymbol{\theta}; \boldsymbol{x}) - \nabla^2 F(\boldsymbol{\theta})$ and Λ is by solving the following equation in Λ:

$$\Lambda A + A\Lambda - \eta\mathbb{E}[\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)] - \eta A\Lambda A = \eta S. \tag{5.33}$$

Lemma 5.5 (Convergence of ROOT-SGD, Theorem 5 in [65]). *Under Assumptions 1,2,3, take step size* $\eta < \eta_{\max} := \frac{\gamma}{8L_2^2} \wedge \frac{1}{4L_1}$ *and choose the burn-in time*

$$B := \left\lceil \frac{24}{\gamma\eta} \right\rceil,$$

then for any iteration $t \geq B$*, the iterate* $\widehat{\boldsymbol{\theta}}_t$ *from ROOT-SGD satisfies the bound*

$$\mathbb{E}\|\nabla F(\widehat{\boldsymbol{\theta}}_t)\|_2^2 \leq \frac{2700\|\nabla F(\widehat{\boldsymbol{\theta}}_0)\|_2^2}{\eta^2\gamma^2(t+1)^2} + \frac{28\sigma_1^2}{t+1}.$$

Thus

$$\mathbb{E}\|\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\|_2^2 \leq \frac{1}{\gamma^2}\mathbb{E}\|\nabla F(\widehat{\boldsymbol{\theta}}_t)\|_2^2 \leq \frac{2700\|\nabla F(\widehat{\boldsymbol{\theta}}_0)\|_2^2}{\eta^2\gamma^4(t+1)^2} + \frac{28\sigma_1^2}{\gamma^2(t+1)}.$$

5.5.4.2 Proof for Theorem 5.1

In this section, we prove Theorem 5.1, the convergence of the plug-in estimator, given the lemmas in the previous section. Recall that the plug-in estimator has each term in the asymptotic covariance replaced by their empirical counterparts. We first bound the error between each term and its empirical estimator in Lemma 5.6. With those bounds, we finally prove Theorem 5.1. For notation simplicity, we denote $\widetilde{A}_t$, $\widetilde{P}_t$, $\widetilde{\Lambda}_t$, $\widetilde{\Sigma}_t$, $\widehat{A}_t$, $\widehat{P}_t$, $\widehat{\Lambda}_t$, $\widehat{\Sigma}_t$, $\widehat{S}_t$ as $\widetilde{A}$, $\widetilde{P}$, $\widetilde{\Lambda}$, $\widetilde{\Sigma}$, $\widehat{A}$, $\widehat{P}$, $\widehat{\Lambda}$, $\widehat{\Sigma}$, $\widehat{S}$, respectively, (i.e., omit all ts in the notation) throughout the proof.

Lemma 5.6. *Under Assumptions 5.1-5.6, denote* $C_p \lesssim \max\{\sigma_1/\gamma, L_3, \sqrt{L_4}, l_4, \sqrt{L_5}, L_6\}$*, for step size* $\eta < \min(\eta_{\max}, 2\delta/L_4^2)$ *we have*

$$\mathbb{E}\|\widetilde{A} - A\|_2 \lesssim C_p^2/\sqrt{t},$$

$$\mathbb{E}\|\widetilde{A}^{-1} - A^{-1}\|_2 \lesssim C_p^2/\sqrt{t},$$

$$\mathbb{E}\|\widehat{S} - S\|_2 \lesssim C_p^4/\sqrt{t},$$

$$\mathbb{E}\|\widetilde{\Lambda} - \Lambda\|_F \lesssim [\eta\|\Lambda\|_F(L_4^2 + C_p^2) + \eta\sqrt{p}C_p^4 + \|\Lambda\|_2 C_p^2]/\sqrt{t}.$$

Proof. First, by Proposition 5.5, we have the following.

$$\begin{aligned}
\sum_{i=B+1}^{t} \mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1} - \boldsymbol{\theta}^*\|_2^2 &\leq \sum_{i=B+1}^{t} \frac{2700\|\nabla F(\widehat{\boldsymbol{\theta}}_0)\|_2^2}{\eta^2\gamma^4 i^2} + \sum_{i=B+1}^{t} \frac{28\sigma_1^2}{\gamma^2 i} \\
&= \frac{2700\|\nabla F(\widehat{\boldsymbol{\theta}}_0)\|_2^2}{\eta^2\gamma^4} \sum_{i=B+1}^{t} \frac{1}{i^2} + \frac{28\sigma_1^2}{\gamma^2} \sum_{i=B+1}^{t} \frac{1}{i} \\
&\leq \frac{2700\|\nabla F(\widehat{\boldsymbol{\theta}}_0)\|_2^2}{\eta^2\gamma^4} \int_{i=B}^{t} \frac{1}{i^2} di + \frac{28\sigma_1^2}{\gamma^2} \int_{i=B}^{t} \frac{1}{i} di \\
&\leq \frac{2700\|\nabla F(\widehat{\boldsymbol{\theta}}_0)\|_2^2}{\eta^2\gamma^4 B} + \frac{28\sigma_1^2 \log t}{\gamma^2},
\end{aligned}$$

and

$$\begin{aligned}
\sum_{i=B+1}^{t} \mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1} - \boldsymbol{\theta}^*\|_2 &\leq \sum_{i=B+1}^{t} \sqrt{\mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1} - \boldsymbol{\theta}^*\|_2^2} \\
&\leq \sum_{i=B+1}^{t} \sqrt{\frac{2700\|\nabla F(\widehat{\boldsymbol{\theta}}_0)\|_2^2}{\eta^2\gamma^4 i^2} + \frac{28\sigma_1^2}{\gamma^2 i}} \\
&\leq \frac{\sqrt{2700}\|\nabla F(\widehat{\boldsymbol{\theta}}_0)\|_2}{\eta\gamma^2} \sum_{i=B+1}^{t} \frac{1}{i} + \frac{\sqrt{28}\sigma_1}{\gamma} \sum_{i=B+1}^{t} \frac{1}{\sqrt{i}} \\
&\leq \frac{\sqrt{2700}\|\nabla F(\widehat{\boldsymbol{\theta}}_0)\|_2}{\eta\gamma^2} \int_{i=B}^{t} \frac{1}{i} + \frac{\sqrt{28}\sigma_1}{\gamma} \int_{i=B}^{t} \frac{1}{\sqrt{i}} \\
&\leq \frac{\sqrt{2700}\|\nabla F(\widehat{\boldsymbol{\theta}}_0)\|_2 \log t}{\eta\gamma^2} + \frac{2\sqrt{28}\sigma_1(\sqrt{t} - \sqrt{B})}{\gamma}.
\end{aligned}$$

Thus

$$\frac{1}{t-B} \sum_{i=B+1}^{t} \mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1} - \boldsymbol{\theta}^*\|_2^2 \lesssim \frac{C_p^2 \log t}{t}, \tag{5.34}$$

$$\frac{1}{t-B} \sum_{i=B+1}^{t} \mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1} - \boldsymbol{\theta}^*\|_2 \lesssim \frac{C_p}{\sqrt{t}}. \tag{5.35}$$

For $\mathbb{E}\|\widetilde{A}-A\|_2$:

$$
\begin{aligned}
\mathbb{E}\|\widehat{A}-A\|_2 &= \mathbb{E}\left\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)-\nabla^2 F(\boldsymbol{\theta}^*)\right\|_2 \\
&\le \mathbb{E}\left\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)-\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\right\|_2 \\
&\quad + \mathbb{E}\left\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)-\nabla^2 F(\boldsymbol{\theta}^*)\right\|_2 \\
&\le \frac{1}{t-B}\sum_{i=B+1}^{t}\mathbb{E}\|\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x})-\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2 \\
&\quad + \mathbb{E}\left\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)-\nabla^2 F(\boldsymbol{\theta}^*)\right\|_2 \\
&\le \frac{1}{t-B}\sum_{i=B+1}^{t}L_3\mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1}-\boldsymbol{\theta}^*\|_2+\mathbb{E}\left\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)-\nabla^2 F(\boldsymbol{\theta}^*)\right\|_2 \\
&\le \frac{L_3}{t-B}\sum_{i=B+1}^{t}\mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1}-\boldsymbol{\theta}^*\|_2+\sqrt{\mathbb{E}\left\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)-\nabla^2 F(\boldsymbol{\theta}^*)\right\|_2^2} \\
&= \frac{L_3}{t-B}\sum_{i=B+1}^{t}\mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1}-\boldsymbol{\theta}^*\|_2+\frac{1}{t-B}\sqrt{\sum_{i=B+1}^{t}\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})-\nabla^2 F(\boldsymbol{\theta}^*)\|_2^2} \\
&\le \frac{L_3}{t-B}\sum_{i=B+1}^{t}\mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1}-\boldsymbol{\theta}^*\|_2+\frac{1}{\sqrt{t-B}}\sqrt{\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^2} \\
&\le \frac{L_3}{t-B}\sum_{i=B+1}^{t}\mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1}-\boldsymbol{\theta}^*\|_2+\frac{1}{\sqrt{t-B}}(\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^4)^{1/4} \\
&\le \frac{L_3}{t-B}\sum_{i=B+1}^{t}\mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1}-\boldsymbol{\theta}^*\|_2+\frac{L_4}{\sqrt{t-B}} \\
&\overset{(5.35)}{\lesssim} C_p^2/\sqrt{t},
\end{aligned}
$$

and

$$
\begin{aligned}
\mathbb{E}\|\widehat{A}-\widetilde{A}\|_2 &= \mathbb{E}[\mathbb{1}_{\widehat{A}\succeq\delta I}*0+(1-\mathbb{1}_{\widehat{A}\succeq\delta I})*(\delta-\lambda_{\min}(\widehat{A}))] \\
&\lesssim \mathbb{E}[1-\mathbb{1}_{\widehat{A}\succeq\delta I}]
\end{aligned}
$$

$$
\begin{aligned}
&= 1 - P(\widehat{A} \succeq \delta I) \\
&= 1 - P(\lambda_{\min}(A + (\widehat{A} - A)) \geq \delta) \\
&\overset{(i)}{\leq} 1 - P(\|\widehat{A} - A\|_2 \leq \lambda_{\min}(A) - \delta) \\
&\overset{(ii)}{\leq} \frac{1}{\lambda_{\min}(A) - \delta} \mathbb{E}\|\widehat{A} - A\|_2 \asymp \mathbb{E}\|\widehat{A} - A\|_2,
\end{aligned}
$$

where (i) by Weyl's inequality that $\lambda_{\min}(A + B) \geq \lambda_{\min}(A) - \|B\|_2$, and (ii) by Markov inequality. Thus,

$$
\mathbb{E}\|\widetilde{A} - A\|_2 \leq \mathbb{E}\|\widehat{A} - A\|_2 + \mathbb{E}\|\widehat{A} - \widetilde{A}\|_2 \lesssim C_p^2/\sqrt{t}.
$$

For $\mathbb{E}\|\widetilde{A}^{-1} - A^{-1}\|_2$:

Denote $dA = \widetilde{A} - A$, then

$$
\begin{aligned}
\|\widetilde{A}^{-1} - A^{-1}\|_2 &\overset{(5.32)}{\leq} \mathbb{1}_{\|A^{-1}dA\|_2 \leq 1/2} 2\|dA\|_2 \|A^{-1}\|_2^2 + \mathbb{1}_{\|A^{-1}dA\|_2 > 1/2}(\|A^{-1}\|_2 + \|\widetilde{A}^{-1}\|_2) \\
&\leq 2\|dA\|_2 \|A^{-1}\|_2^2 + \mathbb{1}_{\|A^{-1}dA\|_2 > 1/2}(\lambda_{\min}(A)^{-1} + \delta^{-1}),
\end{aligned}
$$

thus

$$
\begin{aligned}
\mathbb{E}\|\widetilde{A}^{-1} - A^{-1}\|_2 &\leq 2\|A^{-1}\|_2^2 \mathbb{E}\|dA\|_2 + P(\|A^{-1}dA\|_2 > 1/2)(\lambda_{\min}(A)^{-1} + \delta^{-1}) \\
&\leq 2\|A^{-1}\|_2^2 \mathbb{E}\|dA\|_2 + 2(\lambda_{\min}(A)^{-1} + \delta^{-1})\mathbb{E}\|A^{-1}dA\|_2 \\
&\lesssim \mathbb{E}\|dA\|_2 \lesssim C_p^2/\sqrt{t}.
\end{aligned}
$$

For $\mathbb{E}\|\widehat{S} - S\|_2$:

$$
\begin{aligned}
&\mathbb{E}\|\widehat{S} - S\|_2 = \mathbb{E}\left\| \frac{1}{t-B} \sum_{i=B+1}^{t} \nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i) \nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i)^T - \mathbb{E}[\nabla f(\boldsymbol{\theta}^*; \boldsymbol{x}) \nabla f(\boldsymbol{\theta}^*; \boldsymbol{x})^T] \right\|_2 \\
\leq &\mathbb{E}\left\| \frac{1}{t-B} \sum_{i=B+1}^{t} \nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i) \nabla f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i)^T - \frac{1}{t-B} \sum_{i=B+1}^{t} \nabla f(\boldsymbol{\theta}^*; \boldsymbol{x}_i) \nabla f(\boldsymbol{\theta}^*; \boldsymbol{x}_i)^T \right\|_2
\end{aligned}
$$

$$
\begin{aligned}
&+\mathbb{E}\left\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)^T-\mathbb{E}[\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})^T]\right\|_2\\
\leq&\frac{1}{t-B}\sum_{i=B+1}^{t}\mathbb{E}\|\nabla f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)\nabla f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)^T-\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)^T\|_2\\
&+\mathbb{E}\left\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)^T-\mathbb{E}[\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})^T]\right\|_2\\
\leq&\mathbb{E}\left\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)^T-\mathbb{E}[\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})^T]\right\|_2\\
&+\frac{1}{t-B}\sum_{i=B+1}^{t}\Bigg[\mathbb{E}\|(\nabla f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)-\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i))(\nabla f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)-\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i))^T\|_2\\
&\quad+\mathbb{E}\|\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)(\nabla f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)-\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i))^T\|_2\\
&\quad+\mathbb{E}\|(\nabla f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)-\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i))\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)^T\|_2\Bigg]\\
\leq&\mathbb{E}\left\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)^T-\mathbb{E}[\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})^T]\right\|_2\\
&+\frac{1}{t-B}\sum_{i=B+1}^{t}\Bigg[\mathbb{E}\|\nabla f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)-\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\|_2^2\\
&\quad+2\sqrt{\mathbb{E}\|\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\|_2^2\mathbb{E}\|\nabla f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)-\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\|_2^2}\Bigg]\\
\leq&\underbrace{\mathbb{E}\left\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)^T-\mathbb{E}[\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})^T]\right\|_2}_{(1)}\\
&+\underbrace{\frac{1}{t-B}\sum_{i=B+1}^{t}[L_5^2\mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1}-\boldsymbol{\theta}^*\|_2^2+2L_5\sqrt{\mathbb{E}\|\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\|_2^2\mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1}-\boldsymbol{\theta}^*\|_2^2}]}_{(2)}.
\end{aligned}
$$

Where term (1):

$$
\begin{aligned}
&\mathbb{E}\left\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)^T-\mathbb{E}[\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})^T]\right\|_2\\
\leq&\sqrt{\mathbb{E}\left\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)^T-\mathbb{E}[\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})^T]\right\|_2^2}
\end{aligned}
$$

$$
\begin{aligned}
=&\sqrt{\frac{1}{t-B}\mathbb{E}\|\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})^T - \mathbb{E}[\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})^T]\|_2^2} \\
\leq&\sqrt{\frac{4}{t-B}\mathbb{E}\|\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})^T\|_2^2} \\
=&\sqrt{\frac{4}{t-B}\mathbb{E}\|\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^4} \\
\leq&\sqrt{\frac{4}{t-B}l_4^4} \lesssim C_p^2/\sqrt{t},
\end{aligned}
$$

and term (2):

$$
\begin{aligned}
&\frac{1}{t-B}\sum_{i=B+1}^{t}\left[L_5^2\mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1}-\boldsymbol{\theta}^*\|_2^2 + 2L_5\sqrt{\mathbb{E}\|\nabla f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\|_2^2\mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1}-\boldsymbol{\theta}^*\|_2^2}\right] \\
\leq&\frac{L_5^2}{t-B}\sum_{i=B+1}^{t}\mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1}-\boldsymbol{\theta}^*\|_2^2 + \frac{2L_5l_4}{t-B}\sum_{i=B+1}^{t}\sqrt{\mathbb{E}\|\widehat{\boldsymbol{\theta}}_{i-1}-\boldsymbol{\theta}^*\|_2^2} \\
\lesssim&L_5^2C_p^2\log T/T + L_5C_p^2/\sqrt{t} \lesssim C_p^4/\sqrt{t}.
\end{aligned}
$$

Thus

$$
\mathbb{E}\|\widehat{S}-S\|_2 \lesssim C_p^4/\sqrt{t}.
$$

For $\mathbb{E}\|\widetilde{\Lambda}-\Lambda\|_F$:

To bound $\mathbb{E}\|\widetilde{\Lambda}-\Lambda\|_F$, we first claim the following hold, which we will check at the end of this proof:

$$
\|P-\widetilde{P}\|_2 \lesssim L_4^2, \tag{5.36}
$$

$$
\mathbb{E}\|\widehat{P}-P\|_2 \lesssim (\sqrt{L_7}+C_p^2)/\sqrt{t}, \tag{5.37}
$$

$$
\mathbb{E}\|\widetilde{P}-P\|_2 \lesssim (\sqrt{L_7}+C_p^2)/\sqrt{t}. \tag{5.38}
$$

Let us denote

$$
dS = \widehat{S}-S, d\Lambda = \widetilde{\Lambda}-\Lambda.
$$

Define a linear operator $L_{\widetilde{P}}$ such that

$$\mathrm{Vec}(L_{\widetilde{P}}(\Lambda)) = \widetilde{P}\,\mathrm{Vec}(\Lambda),$$

then $\widetilde{\Lambda}$ satisfies the following matrix equality:

$$\widetilde{\Lambda}\widetilde{A} + \widetilde{A}\widetilde{\Lambda} - \eta L_{\widetilde{P}}(\widetilde{\Lambda}) = \eta\widehat{S}. \tag{5.39}$$

That is,

$$(\Lambda + d\Lambda)(A + dA) + (A + dA)(\Lambda + d\Lambda) - \eta L_{\widetilde{P}}(\Lambda + d\Lambda) = \eta(S + dS).$$

Subtracting (5.12) we have

$$\begin{aligned}
&d\Lambda A + \Lambda dA + d\Lambda dA + dA\Lambda + Ad\Lambda + dAd\Lambda \\
=&\eta L_{\widetilde{P}}(\Lambda + d\Lambda) - \eta\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\Lambda\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})] + \eta dS.
\end{aligned}$$

That is,

$$\begin{aligned}
&d\Lambda\widetilde{A} + \widetilde{A}d\Lambda - \eta L_{\widetilde{P}}(d\Lambda) \\
=&\eta L_{\widetilde{P}}(\Lambda) - \eta\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\Lambda\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})] + \eta dS - \Lambda dA - dA\Lambda \\
:=&\epsilon.
\end{aligned}$$

Consider the eigen-decomposition of $\widetilde{A} = UDU^T$, we then have

$$\begin{aligned}
\epsilon =& d\Lambda UDU^T + UDU^T d\Lambda - \eta\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})d\Lambda\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})] \\
&+ \eta(\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})d\Lambda\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})] - L_{\widetilde{P}}(d\Lambda)) \\
\Longrightarrow\quad U^T\epsilon U =& U^T d\Lambda UD + DU^T d\Lambda U - \eta\mathbb{E}[U^T\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})d\Lambda\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})U]
\end{aligned}$$

$$+ \eta U^T(\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) d\Lambda \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})] - L_{\widetilde{P}}(d\Lambda))U.$$

Thus

$$\begin{aligned}
&\|\epsilon\|_F = \|U^T \epsilon U\|_F \\
\geq &\|U^T d\Lambda U D + D U^T d\Lambda U\|_F - \eta \|\mathbb{E}[U^T \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) d\Lambda \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) U]\|_F \\
&- \eta \|U^T(\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) d\Lambda \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})] - L_{\widetilde{P}}(d\Lambda))U\|_F \\
\geq &2 d_{\min} \|U^T d\Lambda U\|_F - \eta \| \operatorname{Vec}(\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) d\Lambda \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})])\|_2 \\
&- \eta \| \operatorname{Vec}(\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) d\Lambda \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})] - L_{\widetilde{P}}(d\Lambda))\|_2 \\
\geq &2\delta \|d\Lambda\|_F - \eta \|\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})] \operatorname{Vec}(d\Lambda)\|_2 \\
&- \eta \|(\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})] - \widetilde{P}) \operatorname{Vec}(d\Lambda)\|_2, \qquad (5.40)
\end{aligned}$$

where $d_{\min}$ is the smallest diagonal element of the diagonal matrix D, i.e., the smallest eigenvalue of $\widetilde{A}$, so we have $d_{\min} \geq \delta$.

We now analyze the terms in (5.40). We have that

$$\begin{aligned}
&\|\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})] \operatorname{Vec}(d\Lambda)\|_2 \\
\leq &\|\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})]\|_2 \| \operatorname{Vec}(d\Lambda)\|_2 \\
\leq &\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2 \|d\Lambda\|_F \\
= &\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2^2 \|d\Lambda\|_F \leq L_4^2 \|d\Lambda\|_F, \qquad (5.41)
\end{aligned}$$

where the last inequality is by Assumption 5.4. And we have

$$\begin{aligned}
&\|(\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})] - \widetilde{P}) \operatorname{Vec}(d\Lambda)\|_2 \\
\leq &\|P - \widetilde{P}\|_2 \| \operatorname{Vec}(d\Lambda)\|_2 \lesssim L_4^2 \|d\Lambda\|_F. \qquad (5.42)
\end{aligned}$$

Thus

$$\|\epsilon\|_F \gtrsim 2\delta\|d\Lambda\|_F - \eta L_4^2\|d\Lambda\|_F. \tag{5.43}$$

On the other hand,

$$\epsilon = \eta L_{\widetilde{P}}(\Lambda) - \eta\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\Lambda\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})] + \eta dS - \Lambda dA - dA\Lambda.$$

So

$$\|\epsilon\|_F \le \eta\|L_{\widetilde{P}}(\Lambda) - \mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\Lambda\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})]\|_F + \eta\|dS\|_F + \|\Lambda dA + dA\Lambda\|_F,$$

where

$$\begin{aligned}
&\mathbb{E}\|L_{\widetilde{P}}(\Lambda) - \mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\Lambda\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})]\|_F \\
=&\mathbb{E}\|\operatorname{Vec}(L_{\widetilde{P}}(\Lambda) - \mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\Lambda\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})])\|_2 \\
=&\mathbb{E}\|(\widetilde{P} - P)\operatorname{Vec}(\Lambda)\|_2 \\
\le&(\mathbb{E}\|\widetilde{P} - P\|_2)\|\operatorname{Vec}(\Lambda)\|_2 \overset{(5.38)}{\lesssim} \|\Lambda\|_F(\sqrt{L_7} + C_p^2)/\sqrt{t},
\end{aligned}$$

$$\mathbb{E}\|dS\|_F \le \sqrt{p}\mathbb{E}\|dS\|_2 \lesssim \sqrt{p}C_p^4/\sqrt{t},$$

and the last term:

$$\begin{aligned}
\mathbb{E}\|\Lambda dA + dA\Lambda\|_F &= \mathbb{E}\|\operatorname{Vec}(\Lambda dA) + \operatorname{Vec}(dA\Lambda)\|_2 = \mathbb{E}\|(I\otimes\Lambda + \Lambda^T\otimes I)\operatorname{Vec}(dA)\|_2 \\
&\le (\|I\otimes\Lambda\|_2 + \|\Lambda^T\otimes I\|_2)\mathbb{E}\|\operatorname{Vec}(dA)\|_2 \\
&= 2\|\Lambda\|_2\mathbb{E}\|\operatorname{Vec}(dA)\|_2 \\
&= 2\|\Lambda\|_2\mathbb{E}\|dA\|_F \le 2\|\Lambda\|_2\mathbb{E}\|dA\|_2 \lesssim \|\Lambda\|_2 C_p^2/\sqrt{t}.
\end{aligned}$$

Thus

$$\begin{aligned}\mathbb{E}\|\epsilon\|_F &\lesssim [\eta\|\Lambda\|_F(\sqrt{L_7}+C_p^2)+\eta\sqrt{p}C_p^4+\|\Lambda\|_2 C_p^2]/\sqrt{t}\\ &\lesssim [\eta\|\Lambda\|_F(L_4^2+C_p^2)+\eta\sqrt{p}C_p^4+\|\Lambda\|_2 C_p^2]/\sqrt{t}. \end{aligned} \tag{5.44}$$

Combining (5.43) and (5.44), we have:

$$(2\delta-\eta L_4^2)\mathbb{E}\|d\Lambda\|_F \lesssim [\eta\|\Lambda\|_F(L_4^2+C_p^2)+\eta\sqrt{p}C_p^4+\|\Lambda\|_2 C_p^2]/\sqrt{t}. \tag{5.45}$$

That is,

$$\mathbb{E}\|\widetilde{\Lambda}-\Lambda\|_F \lesssim [\eta\|\Lambda\|_F(L_4^2+C_p^2)+\eta\sqrt{p}C_p^4+\|\Lambda\|_2 C_p^2]/\sqrt{t}.$$

Check (5.36)**,** (5.37)**,** (5.38)**:**

• For $\|\widetilde{P}-P\|_2 \lesssim L_4^2$:

$$\begin{aligned}\|\widetilde{P}-P\|_2 &\le \|\widetilde{P}\|_2+\|P\|_2\\ &\le L_4^2+\|\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})]\|_2\\ &\le L_4^2+\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2\\ &= L_4^2+\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^2\\ &\le L_4^2+\sqrt{\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\|_2^4}\le 2L_4^2.\end{aligned}$$

• For $\mathbb{E}\|\widehat{P}-P\|_2 \lesssim (\sqrt{L_7}+C_p^2)/\sqrt{t}$:

$$\begin{aligned}&\mathbb{E}\|\widehat{P}-P\|_2\\ =&\mathbb{E}\left\|\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})]-\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)\otimes\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)\right\|_2\\ \le&\mathbb{E}\left\|\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})]-\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\right\|_2\end{aligned}$$

$$+\mathbb{E}\Bigg\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i) - \frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)\otimes\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)\Bigg\|_2, \tag{5.46}$$

for which

$$\begin{aligned}
&\mathbb{E}\left\|\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})] - \frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\right\|_2 \\
\leq&\sqrt{\mathbb{E}\left\|\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x})] - \frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\right\|_2^2} \\
\leq&\frac{\sqrt{L_7}}{\sqrt{t-B}},
\end{aligned} \tag{5.47}$$

where the bound (5.47) comes from Assumption 5.6. And

$$\begin{aligned}
&\mathbb{E}\Bigg\|\frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i) - \frac{1}{t-B}\sum_{i=B+1}^{t}\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)\otimes\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)\Bigg\|_2 \\
\leq&\frac{1}{t-B}\sum_{i=B+1}^{t}\mathbb{E}[\|\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i)\otimes\nabla^2 f(\boldsymbol{\theta}^*;\boldsymbol{x}_i) - \nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)\otimes\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1};\boldsymbol{x}_i)\|_2] \\
=&\frac{L_6}{t-B}\sum_{i=B+1}^{t}\mathbb{E}[\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_{i-1}\|_2] + \frac{L_6'}{t-B}\sum_{i=B+1}^{t}\mathbb{E}[\|\boldsymbol{\theta}^* - \widehat{\boldsymbol{\theta}}_{i-1}\|_2^2] \lesssim C_p^2/\sqrt{t}.
\end{aligned} \tag{5.48}$$

Thus, $\mathbb{E}\|\widehat{P} - P\|_2 \lesssim (\sqrt{L_7} + C_p^2)/\sqrt{t}$.

• For $\mathbb{E}\|\widetilde{P} - P\|_2 \lesssim (\sqrt{L_7} + C_p^2)/\sqrt{t}$:

First, we have

$$\begin{aligned}
\|\widehat{P} - \widetilde{P}\|_2 &= \mathbb{1}_{\widehat{P}\preceq L_4^2 I} * 0 + (1 - \mathbb{1}_{\widehat{P}\preceq L_4^2 I}) * (\|\widehat{P}\|_2 - L_4^2) \\
&\leq \mathbb{1}_{\widehat{P}\preceq L_4^2 I} * 0 + (1 - \mathbb{1}_{\widehat{P}\preceq L_4^2 I}) * (\|\widehat{P} - P\|_2 + \|P\|_2 - L_4^2)
\end{aligned}$$

$$\leq (1 - \mathbb{1}_{\widehat{P} \preceq L_4^2 I}) * (\|\widehat{P} - P\|_2 + L_4^2 - L_4^2) \leq \|\widehat{P} - P\|_2.$$

Thus

$$\begin{aligned}\mathbb{E}\|\widetilde{P} - P\|_2 &\leq \mathbb{E}\|\widetilde{P} - \widehat{P}\|_2 + \mathbb{E}\|\widehat{P} - P\|_2 \\ &\leq 2\mathbb{E}\|\widehat{P} - P\|_2 \lesssim (\sqrt{L_7} + C_p^2)/\sqrt{t}.\end{aligned}$$

□

With Lemma 5.6, we now prove Theorem 5.1.

Proof. We have

$$\widetilde{\Sigma} = \widetilde{A}^{-1}(\frac{1}{\eta}\widetilde{\Lambda}\widetilde{A} + \frac{1}{\eta}\widetilde{A}\widetilde{\Lambda} - \widetilde{A}\widetilde{\Lambda}\widetilde{A})\widetilde{A}^{-1} = \frac{1}{\eta}\widetilde{A}^{-1}\widetilde{\Lambda} + \frac{1}{\eta}\widetilde{\Lambda}\widetilde{A}^{-1} - \widetilde{\Lambda},$$

and

$$\Sigma = \frac{1}{\eta}A^{-1}\Lambda + \frac{1}{\eta}\Lambda A^{-1} - \Lambda.$$

Since

$$\begin{aligned}&\mathbb{E}\|\widetilde{A}^{-1}\widetilde{\Lambda} - A^{-1}\Lambda\|_F \\ &= \mathbb{E}\|(\widetilde{A}^{-1} - A^{-1} + A^{-1})(\widetilde{\Lambda} - \Lambda + \Lambda) - A^{-1}\Lambda\|_F \\ &\leq \mathbb{E}\|(\widetilde{A}^{-1} - A^{-1})(\widetilde{\Lambda} - \Lambda)\|_F + \mathbb{E}\|(\widetilde{A}^{-1} - A^{-1})\Lambda\|_F + \mathbb{E}\|A^{-1}(\widetilde{\Lambda} - \Lambda)\|_F \\ &\leq \|\widetilde{A}^{-1} - A^{-1}\|_2\mathbb{E}\|\widetilde{\Lambda} - \Lambda\|_F + \|\Lambda\|_F\mathbb{E}\|\widetilde{A}^{-1} - A^{-1}\|_2 + \|A^{-1}\|_2\mathbb{E}\|\widetilde{\Lambda} - \Lambda\|_F \\ &\leq (\delta^{-1} + \lambda_{\min}(A))\mathbb{E}\|\widetilde{\Lambda} - \Lambda\|_F + \|\Lambda\|_F\mathbb{E}\|\widetilde{A}^{-1} - A^{-1}\|_2 + \delta^{-1}\mathbb{E}\|\widetilde{\Lambda} - \Lambda\|_F \\ &\lesssim [\eta\|\Lambda\|_F(L_4^2 + C_p^2) + \eta\sqrt{p}C_p^4 + \|\Lambda\|_2 C_p^2]/\sqrt{t} + \|\Lambda\|_F C_p^2/\sqrt{t},\end{aligned}$$

and similarly

$$\mathbb{E}\|\widetilde{\Lambda}\widetilde{A}^{-1} - \Lambda A^{-1}\|_F \lesssim [\eta\|\Lambda\|_F(L_4^2 + C_p^2) + \eta\sqrt{p}C_p^4 + \|\Lambda\|_2 C_p^2]/\sqrt{t} + \|\Lambda\|_F C_p^2/\sqrt{t},$$

we have that

$$\begin{aligned}
\mathbb{E}\|\Sigma - \widetilde{\Sigma}\|_F &\leq \frac{1}{\eta}\|\widetilde{A}^{-1}\widetilde{\Lambda} - A^{-1}\Lambda\|_F + \frac{1}{\eta}\|\widetilde{\Lambda}\widetilde{A}^{-1} - \Lambda A^{-1}\|_F + \mathbb{E}\|\widetilde{\Lambda} - \Lambda\|_F \\
&\lesssim (1 + \frac{1}{\eta})[\eta\|\Lambda\|_F(L_4^2 + C_p^2) + \eta\sqrt{p}C_p^4 + \|\Lambda\|_2 C_p^2]/\sqrt{t} + \frac{1}{\eta}\|\Lambda\|_F C_p^2/\sqrt{t} \\
&\lesssim [\|\Lambda\|_F(L_4^2 + C_p^2) + \sqrt{p}C_p^4]/\sqrt{t} + \frac{1}{\eta}\|\Lambda\|_F C_p^2/\sqrt{t}.
\end{aligned}$$

Recall that Λ is by solving following equation:

$$\Lambda A + A\Lambda - \eta\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\Lambda\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})] = \eta S.$$

Thus,

$$\begin{aligned}
\eta\|S\|_F &= \|\Lambda A + A\Lambda - \eta\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\Lambda\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})]\|_F \\
&\geq \|\Lambda A + A\Lambda\|_F - \eta\|\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\Lambda\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})]\|_F \\
&\geq 2\lambda_{\min}(A)\|\Lambda\|_F - \eta\|\mathbb{E}[\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) \otimes \nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})]\|_2\|\operatorname{Vec}(\Lambda)\|_2 \\
&\geq 2\lambda_{\min}(A)\|\Lambda\|_F - \eta\mathbb{E}\|\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\|_2^2\|\operatorname{Vec}(\Lambda)\|_2 \\
&\geq (2\lambda_{\min}(A) - \eta L_4^2)\|\Lambda\|_F.
\end{aligned}$$

That is,

$$\|\Lambda\|_F \lesssim \eta\|S\|_F.$$

To sum up, we have

$$\mathbb{E}\|\Sigma - \widetilde{\Sigma}\|_F \lesssim [\eta\|S\|_F(L_4^2 + C_p^2) + \sqrt{p}C_p^4 + \|S\|_F C_p^2]/\sqrt{t}$$

$$\lesssim [\sqrt{p}C_p^4 + \|S\|_F C_p^2]/\sqrt{t}.$$

□

5.5.5 A Preliminary Result for Proving Theorem 5.2

In this section, we prove a preliminary result for proving Theorem 5.2. The result is given in Proposition 5.2. It generalizes Lemma 5.4 to a functional form for the random function $C_t(r) := r\sqrt{t}\boldsymbol{w}^T(\widehat{\boldsymbol{\theta}}_{[rt]} - \boldsymbol{\theta}^*)$, $r \in [0,1]$.

The preliminary result uses the invariance principle of martingale CLT. So we first review the invariance principle of martingale CLT in the following lemma:

Lemma 5.7 (Theorem 4.4. in [78], rewritten in conditional Lindeberg condition)**.** *For $\{S_t, \mathcal{F}_t\}$ a zero mean martingale, define $X_t := S_t - S_{t-1}$, $V_t^2 := \sum_{i=1}^t \mathbb{E}(X_i^2|\mathcal{F}_{i-1})$, $s_t^2 := \mathbb{E}(S_t^2) = \sum_{i=1}^t \mathbb{E}(X_i^2)$. Assume the conditional Lindeberg condition holds, that is*

$$\forall \epsilon > 0, s_t^{-2} \sum_{i=1}^t \mathbb{E}\left[X_i^2 I(|X_i| > \epsilon s_t)|\mathcal{F}_{i-1}\right] \overset{t\to\infty}{\to} 0.$$

And assume that

$$s_t^{-2} V_t^2 \overset{p}{\to} \eta^2 > 0 \quad a.s..$$

Then for the random function defined on $r \in [0,1]$: $\zeta_t(r) := V_t^{-1}\{S_i + (V_{i+1}^2 - V_i^2)^{-1}(rV_t^2 - V_i^2)X_{i+1}\}$ where i is such that $V_i^2 \le rV_t^2 < V_{i+1}^2$, one have

$$\zeta_t(r) \overset{d}{\to} W_r.$$

With Lemma 5.7, we can develop the functional CLT extension of Lemma 5.4 in the following proposition:

Proposition 5.2 (Functional CLT extension of Lemma 5.4)**.** *Under assumptions of Lemma*

5.4, we have that

$$r\sqrt{t}\boldsymbol{w}^T(\widehat{\boldsymbol{\theta}}_{[rt]} - \boldsymbol{\theta}^*) \xrightarrow{d} (\boldsymbol{w}^T\Sigma\boldsymbol{w})^{1/2}W_r, \tag{5.49}$$

Proof. We make use of the martingale decomposition of ROOT-SGD updates. By equation (42) in [79], the ROOT-SGD has the difference $\boldsymbol{z}_i := \boldsymbol{v}_i - \nabla F(\widehat{\boldsymbol{\theta}}_{i-1})$ decomposed as follows:

$$\boldsymbol{z}_i = \underbrace{\frac{1}{i}\sum_{s=B}^{i}\delta_s(\widehat{\boldsymbol{\theta}}_{s-1})}_{:=M_i} + \frac{B}{i}\boldsymbol{z}_B + \underbrace{\frac{1}{i}\sum_{s=B}^{i}(s-1)(\delta_s(\widehat{\boldsymbol{\theta}}_{s-1}) - \delta_s(\widehat{\boldsymbol{\theta}}_{s-2}))}_{:=\Psi_i},$$

where $\delta_s(\boldsymbol{\theta}) := \delta(\boldsymbol{\theta}; \boldsymbol{x}_s) = \nabla f(\boldsymbol{\theta}; \boldsymbol{x}_s) - \nabla F(\boldsymbol{\theta})$. Then M_i and Ψ_i are two martingale sequences. However, it is hard to directly analyze these two sequences. To understand the behavior of M_i and Ψ_i, [79] defines two auxiliary processes:

$$N_i := \sum_{s=B}^{i}\delta_s(\boldsymbol{\theta}^*), \quad \Upsilon_i := \eta\sum_{s=B}^{i}\Xi_s(\boldsymbol{\theta}^*)\boldsymbol{y}_{s-1},$$

where $\Xi_s(\boldsymbol{\theta}) := \nabla^2 f(\boldsymbol{\theta}; \boldsymbol{x}_s) - \nabla^2 F(\boldsymbol{\theta})$, and $\boldsymbol{y}_s$ is a zero-mean Markov process defined as: $\boldsymbol{y}_{B-1} = \mathbf{0}$, $\boldsymbol{y}_s = \boldsymbol{y}_{s-1} - \eta\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}_s)\boldsymbol{y}_{s-1} + \delta_s(\boldsymbol{\theta}^*)$ for $s \geq B$.

For the auxiliary processes N_i and Υ_i, [79] shows that $\|N_t - M_t\|_2/\sqrt{t} \xrightarrow{p} 0$ and $\|\Upsilon_t - \Psi_t\|_2/\sqrt{t} \xrightarrow{p} 0$. Then for a $r \in (0, 1]$, we have $\|N_{[rt]} - M_{[rt]}\|_2/\sqrt{rt} \xrightarrow{p} 0$ and $\|\Upsilon_{[rt]} - \Psi_{[rt]}\|_2/\sqrt{rt} \xrightarrow{p} 0$. Thus, we must have

$$sup_{r\in[0,1]}\|N_{[rt]} - M_{[rt]}\|_2/\sqrt{t} \xrightarrow{p} 0, \tag{5.50}$$

$$sup_{r\in[0,1]}\|\Upsilon_{[rt]} - \Psi_{[rt]}\|_2/\sqrt{t} \xrightarrow{p} 0. \tag{5.51}$$

We claim the weak convergence of $N_{[rt]} + \Upsilon_{[rt]}$ is as follows, which we check later:

$$(N_{[rt]} + \Upsilon_{[rt]})/\sqrt{t} \xrightarrow{d} (S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)))^{1/2}\boldsymbol{W}_r, \tag{5.52}$$

where $\boldsymbol{W}_r$ is a p - dimensional random variable with each coordinate independently distributed as W_r the standard wiener process.

By (5.50) to (5.52), we have

$$\begin{aligned}
&\frac{[rt]}{\sqrt{t}}\boldsymbol{z}_{[rt]} \\
=&\frac{1}{\sqrt{t}}M_{[rt]} + \frac{B}{\sqrt{t}}\boldsymbol{z}_B + \frac{1}{\sqrt{t}}\Psi_{[rt]} \\
\xrightarrow{d}&\frac{1}{\sqrt{t}}(N_{[rt]} + \Upsilon_{[rt]}) \\
\xrightarrow{d}&(S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)))^{1/2}\boldsymbol{W}_r.
\end{aligned}$$

By [79] $\sqrt{t}\boldsymbol{v}_t \xrightarrow{p} \mathbf{0}$ and $\sqrt{t}\|\nabla F(\widehat{\boldsymbol{\theta}}_t) - A(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)\|_2 \xrightarrow{p} 0$, we have $sup_{r\in[0,1]} r\sqrt{t}\boldsymbol{v}_t \xrightarrow{p} \mathbf{0}$ and

$\sup_{r\in[0,1]} r\sqrt{t}\|\nabla F(\widehat{\boldsymbol{\theta}}_t) - A(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)\|_2 \xrightarrow{p} 0$. So

$$\frac{[rt]}{\sqrt{t}}\nabla F(\widehat{\boldsymbol{\theta}}_{[rt]-1}) = \frac{[rt]}{\sqrt{t}}\boldsymbol{v}_{[rt]} - \frac{[rt]}{\sqrt{t}}\boldsymbol{z}_{[rt]} \xrightarrow{d} (S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)))^{1/2}\boldsymbol{W}_r;$$

$$\frac{[rt]}{\sqrt{t}}A(\widehat{\boldsymbol{\theta}}_{[rt]-1} - \boldsymbol{\theta}^*) \xrightarrow{d} (S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)))^{1/2}\boldsymbol{W}_r;$$

$$\frac{[rt]}{\sqrt{t}}(\widehat{\boldsymbol{\theta}}_{[rt]} - \boldsymbol{\theta}^*) \xrightarrow{d} (A^{-1}(S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)))A^{-1})^{1/2}\boldsymbol{W}_r.$$

That is,

$$\begin{aligned}
r\sqrt{t}\boldsymbol{w}^T(\widehat{\boldsymbol{\theta}}_{[rt]} - \boldsymbol{\theta}^*) &\xrightarrow{d} (\boldsymbol{w}^T A^{-1}(S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)))A^{-1}\boldsymbol{w})^{1/2}W_r \\
&= (\boldsymbol{w}^T\Sigma\boldsymbol{w})^{1/2}W_r.
\end{aligned}$$

As the proposition claim.

It remains to prove that $(N_{[rt]} + \Upsilon_{[rt]})/\sqrt{t} \xrightarrow{d} (S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)))^{1/2}\boldsymbol{W}_r$. We use Lemma 5.7 to prove such statement. To apply Lemma 5.7, we define the quantity X_i as follows: for a fixed $\boldsymbol{w} \neq \mathbf{0}$, let $X_i = \boldsymbol{w}^T\boldsymbol{\nu}_i$ where $\boldsymbol{\nu}_i = \delta_i(\boldsymbol{\theta}^*) + \eta\Xi_i(\boldsymbol{\theta}^*)\boldsymbol{y}_{i-1}$. Then

$\boldsymbol{w}^T(N_{[rt]} + \Upsilon_{[rt]}) = \sum_{i=B}^{[rt]} X_i$.

We now prove the limiting distribution of $(N_{[rt]} + \Upsilon_{[rt]})/\sqrt{t}$ in following steps.

We first check $s_t^{-2}V_t^2 \xrightarrow{p} \eta^2 > 0$ $a.s.$. By [79], $\frac{1}{t}\sum_{i=B}^{t} \mathbb{E}(\boldsymbol{\nu}_i\boldsymbol{\nu}_i^T|\mathcal{F}_{i-1}) \xrightarrow{p} S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*))$, then

$$\begin{aligned}
\frac{1}{t}V_t^2 = \frac{1}{t}\sum_{i=1}^{t}\mathbb{E}(X_i^2|\mathcal{F}_{i-1}) &= \boldsymbol{w}^T\left[\frac{1}{t}\sum_{i=B}^{t}\mathbb{E}(\boldsymbol{\nu}_i\boldsymbol{\nu}_i^T|\mathcal{F}_{i-1})\right]\boldsymbol{w} \\
&\xrightarrow{d} \boldsymbol{w}^T(S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)))\boldsymbol{w}, \\
\frac{1}{t}s_t^2 = \mathbb{E}\left[\frac{1}{t}V_t^2\right] &\to \boldsymbol{w}^T(S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)))\boldsymbol{w}.
\end{aligned}$$

Since $C := \boldsymbol{w}^T(S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)))\boldsymbol{w} > 0$ is a constant, we also have $\frac{1}{t}V_t^2$ converges in probability implied by comvergence in distribution

$$\frac{1}{t}V_t^2 \xrightarrow{p} \boldsymbol{w}^T(S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)))\boldsymbol{w}.$$

Thus

$$s_t^{-2}V_t^2 \xrightarrow{p} 1.$$

We then check the conditional Lindeberg condition. By [79], one have that as $t \to \infty$,

$$\forall \epsilon' > 0, t^{-1}\sum_{i=1}^{t}\mathbb{E}\left[\|\nu_i\|_2^2 I(\|\nu_i\|_2 > \epsilon'\sqrt{t})|\mathcal{F}_{i-1}\right] \xrightarrow{p} 0. \tag{5.53}$$

To check the Lindeberg condition for X_i sequence for any $\epsilon > 0$, we note that

$$s_t^{-2}\sum_{i=1}^{t}\mathbb{E}\left[X_i^2 I(|X_i| > \epsilon s_t)|\mathcal{F}_{i-1}\right] \to \frac{1}{tC}\sum_{i=1}^{t}\mathbb{E}\left[X_i^2 I(|X_i| > \epsilon\sqrt{Ct})|\mathcal{F}_{i-1}\right],$$

where

$$\frac{1}{t}\sum_{i=1}^{t}\mathbb{E}\left[X_i^2 I(|X_i| > \epsilon\sqrt{Ct})|\mathcal{F}_{i-1}\right]$$

$$\leq \frac{1}{t}\sum_{i=1}^{t}\mathbb{E}\left[\|\boldsymbol{\nu}_i\|_2^2\|\boldsymbol{w}\|_2^2 I(\|\boldsymbol{\nu}_i\|_2 > \epsilon\sqrt{Ct}/\|\boldsymbol{w}\|_2)|\mathcal{F}_{i-1}\right]$$
$$\to 0 \left(\text{By taking } \epsilon' = \frac{\epsilon C}{\|\boldsymbol{w}\|_2} \text{ in (5.53)}\right).$$

Thus the Lindeberg condition holds.

We now have all the conditions in Lemma 5.7 hold, thus we have $\zeta_t(r)$ satisfies that

$$\zeta_t(r) \xrightarrow{d} W_r,$$

where $\zeta_t(r) = V_t^{-1}\{S_i + (V_{i+1}^2 - V_i^2)^{-1}(rV_t^2 - V_i^2)X_{i+1}\}$ and i is such that $V_i^2 \leq rV_t^2 < V_{i+1}^2$.

On the other hand, $V_{[rt]}^2/V_t^2 \to r$. Thus $\zeta_t\left(V_{[rt]}^2/V_t^2\right) \to \zeta_t(r)$. That is, $V_t^{-1}S_{[rt]} = \zeta_t\left(V_{[rt]}^2/V_t^2\right) \xrightarrow{d} W_r$, then by Slutsky's theorem $S_{[rt]}/\sqrt{t} \xrightarrow{d} \sqrt{C}W_r$. In this way,

$$\boldsymbol{w}^T(N_{[rt]} + \Upsilon_{[rt]})/\sqrt{t} \to S_{[rt]}/\sqrt{t} \xrightarrow{d} \sqrt{\boldsymbol{w}^T(S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)))\boldsymbol{w}}W_r.$$

Recall that the above holds for any $\boldsymbol{w} \neq \mathbf{0}$, so we have

$$(N_{[rt]} + \Upsilon_{[rt]})/\sqrt{t} \xrightarrow{d} (S + \mathbb{E}(\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)\Lambda\Xi_{\boldsymbol{x}}(\boldsymbol{\theta}^*)))^{1/2}\boldsymbol{W}_r,$$

which is the limiting distribution as claimed. □

5.5.6 Proof for Theorem 5.2

In this section, we prove Theorem 5.2 using Proposition 5.2.

Proof. By Proposition 5.2, we have that the random function $C_t(r) = r\sqrt{t}\boldsymbol{w}^T(\widehat{\boldsymbol{\theta}}_{[rt]} - \boldsymbol{\theta}^*)$ satisfies

$$C_t(r) \xrightarrow{d} (\boldsymbol{w}^T\Sigma\boldsymbol{w})^{1/2}W_r. \tag{5.54}$$

Our statistic is

$$\begin{aligned}
&\frac{\sqrt{t}\boldsymbol{w}^T(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*)}{\sqrt{\boldsymbol{w}^T V_t \boldsymbol{w}}} \\
=&\frac{C_t(1)}{\sqrt{\frac{1}{t}\sum_{i=1}^t \frac{i^2}{t}(\boldsymbol{w}^T(\widehat{\boldsymbol{\theta}}_i - \widehat{\boldsymbol{\theta}}_t))^2}} \\
\overset{t\to\infty}{\to}&\frac{C_t(1)}{\sqrt{\int_0^1 r^2 t(\boldsymbol{w}^T(\widehat{\boldsymbol{\theta}}_{[rt]} - \widehat{\boldsymbol{\theta}}_t))^2 dr}} \\
=&\frac{C_t(1)}{\sqrt{\int_0^1 (C_t(r) - rC_t(1))^2 dr}}.
\end{aligned}$$

Now $\frac{C_t(1)}{\sqrt{\int_0^1 (C_t(r)-rC_t(1))^2 dr}}$ is a continuous function of $C_t(\cdot)$, so by continuous mapping theorem (Theorem 18.11 of [80]) we have that

$$\frac{C_t(1)}{\sqrt{\int_0^1 (C_t(r) - rC_t(1))^2 dr}} \overset{d}{\to} \frac{(\boldsymbol{w}^T\Sigma\boldsymbol{w})^{1/2} W_1}{(\boldsymbol{w}^T\Sigma\boldsymbol{w})^{1/2}\sqrt{\int_0^1 (W_r - rW_1)^2 dr}} = \frac{W_1}{\sqrt{\int_0^1 (W_r - rW_1)^2 dr}}.$$

This gives us the theorem claim. □

5.5.7 Comparison of Our Covariance Estimators with Those for the ASGD Algorithm

In this section, we compare our plug-in covariance estimator and the random-scaling estimator with their counterparts for the ASDG algorithm.

5.5.7.1 Plug-in Estimator Comparison

We compare our plug-in estimator with the plug-in estimator for ASGD in [66] from two aspects: the convergence rate and the computational burden.

For the convergence rate comparison, our plug-in estimator converges faster to the true asymptotic covariance of ROOT-SGD compared to the ASGD counterpart. Recall that for ASGD, the plug-in covariance estimator $\widehat{\Sigma}_{t,ASGD}$ converges to the true asymptotic covariance Σ_{ASGD} as $\mathbb{E}\|\Sigma_{ASGD} - \widehat{\Sigma}_{t,ASGD}\|_2 \lesssim t^{-\alpha/2}$. Since $\alpha \in (1/2, 1)$, the convergence rate

is strictly slower than $\mathcal{O}(t^{-1/2})$. Our plug-in covariance estimator $\widehat{\Sigma}_t$ converges to the true asymptotic covariance Σ of ROOT-SGD as $\mathbb{E}\|\Sigma - \widehat{\Sigma}_t\|_2 \lesssim t^{-1/2}$. So the convergence speed of our estimator is strictly faster than that of the plug-in estimator in [66]. Moreover, this $\mathcal{O}(t^{-1/2})$ rate matches the optimal statistical rate in such a random sampling scheme.

For the computational aspect of the plug-in estimator, both our estimator and that in [66] can be computed in a fully online fashion. Suppose the algorithm is updated for t steps, and the parameter is of dimension p. Our plug-in estimator takes $\mathcal{O}(p^6 + tp^4)$ total arithmetic computation, while their estimator takes $\mathcal{O}(p^3 + tp^2)$ total arithmetic computation. Our plug-in estimator takes more computation. This is due to the intrinsic structure of the asymptotic covariance of the ROOT-SGD estimator: the asymptotic covariance depends on the quantity

$$\mathbb{E}(\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})\Lambda\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x})); \tag{5.55}$$

moreover, one needs to solve an equation that contains (5.55) in Λ. Thus, in our plug-in estimator, we need to evaluate the empirical counter-part of (5.55) for an unknown Λ, which is $\widehat{P}(\Lambda) = \frac{1}{t}\sum_{i=1}^{t} \nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i)\Lambda\nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i)$. To achieve this, we keep an online update of $\widehat{P} = \frac{1}{t}\sum_{i=1}^{t} \nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i) \otimes \nabla^2 f(\widehat{\boldsymbol{\theta}}_{i-1}; \boldsymbol{x}_i)$. We further invert $\widehat{P}$ when solving the empirical counterpart of Λ. Computing $\widehat{P} \in \mathcal{R}^{p^2 \times p^2}$ and its inverse then needs $\mathcal{O}(tp^4 + p^6)$ operations, which dominates the computation. As a comparison, the asymptotic covariance of the ASGD estimator does not contain any Λ term; thus, its plug-in estimator is computed with fewer computations.

The computation burden of our plug-in estimator for ROOT-SGD is the same as that for ASGD for the special case of $\nabla^2 f(\boldsymbol{\theta}^*; \boldsymbol{x}) \equiv \nabla^2 F(\boldsymbol{\theta}^*)$. Such a special case holds for the exponential family model. In this case, the asymptotic covariance of ROOT-SGD reduces to $A^{-1}SA^{-1}$, and the plug-in covariance of ROOT-SGD is then $\widehat{A}^{-1}\widehat{S}\widehat{A}^{-1}$. Then the computational complexity of our plug-in estimator becomes $\mathcal{O}(p^3 + tp^2)$.

5.5.7.2 *Hessian-free Estimator Comparison*

We compare the computation of our random-scaling estimator with the random-scaling estimator for ASGD in [70].

Both estimators can be computed fully online. For t steps of the algorithm on a problem of dimension p, our random-scaling estimator takes $\mathcal{O}(tp^2)$ arithmetic computations, which is the same as the random-scaling estimator for ASGD and is less than our plug-in estimator.

As for the asymptotic convergence of the random-scaling estimators, both our estimator and that in [70] are asymptotically consistent. Unfortunately, there is no convergence rate result in either our work or [70]. Though the convergence rates of the random-scaling estimators are not guaranteed, considering that ROOT-SGD converges faster than SGD, there is still an advantage to use ROOT-SGD and our random-scaling estimator as compared to the ASGD counterpart. And we do see such an advantage in our experiment.

CHAPTER 6

IMPROVED RATE OF FIRST ORDER ALGORITHMS FOR ENTROPIC OPTIMAL TRANSPORT

This chapter improves the state-of-the-art rate of a first-order algorithm for solving entropy regularized optimal transport. The resulting rate for approximating the optimal transport (OT) has been improved from $\widetilde{\mathcal{O}}(n^{2.5}/\epsilon)$ to $\widetilde{\mathcal{O}}(n^2/\epsilon)$, where n is the problem size and ϵ is the accuracy level. In particular, we propose an accelerated primal-dual stochastic mirror descent algorithm with variance reduction. Such special designs help us improve the rate compared to other accelerated primal-dual algorithms. We further propose a batch version of our stochastic algorithm, which improves the computational performance through parallel computing. To compare, we prove that the computational complexity of the Stochastic Sinkhorn algorithm is $\widetilde{\mathcal{O}}(n^2/\epsilon^2)$, which is slower than our accelerated primal-dual stochastic mirror algorithm. Experiments are done using synthetic and real data, and the results match our theoretical rates. Our algorithm may inspire more research to develop accelerated primal-dual algorithms that have rate $\widetilde{\mathcal{O}}(n^2/\epsilon)$ for solving OT.

6.1 Introduction

The *Optimal Transport* (OT) [81, 82, 83] is an optimization problem that has been actively studied. In this section, we review the OT problem. In Section 6.1.1, we review the OT formulation and its related concepts. In Section 6.1.2, we survey the existing algorithms for solving OT and summarize our contribution given the literature background.

6.1.1 Optimal Transport

We review the definition of OT. Given a cost matrix $C \in \mathcal{R}_+^{n\times n}$ and two vectors $\boldsymbol{p}, \boldsymbol{q} \in \Delta_n$, where $\Delta_n := \{\boldsymbol{a} \in \mathcal{R}_+^n : \boldsymbol{a}^T\mathbf{1} = 1\}$ is the standard simplex, OT is defined as follows:

$$\min_{X\in\mathcal{U}(\boldsymbol{p},\boldsymbol{q})} \langle C, X\rangle, \tag{6.1}$$

where $\mathcal{U}(\boldsymbol{p}, \boldsymbol{q}) := \left\{X \in \mathcal{R}_+^{n\times n} \,\middle|\, X\mathbf{1} = \boldsymbol{p}, X^T\mathbf{1} = \boldsymbol{q}\right\}$, and $\langle C, X\rangle := \sum_{i,j=1}^n C_{i,j}X_{i,j}$.

The ϵ-*solution* is always used when evaluating algorithm efficiency for solving OT, so we review its definition as follows. Denote the optimal solution of problem (6.1) as X^*, an $\epsilon-$solution $\widehat{X}$ is such that:

$$\widehat{X} \in \mathcal{U}(\boldsymbol{p}, \boldsymbol{q});$$
$$\langle C, \widehat{X}\rangle \leq \langle C, X^*\rangle + \epsilon.$$

Note that for a stochastic algorithm, the second condition is replaced by $\mathbb{E}\langle C, \widehat{X}\rangle \leq \langle C, X^*\rangle + \epsilon$.

Our chapter adopts a two-step approach [84] for finding an ϵ-solution to problem (6.1). In the first step, one finds an approximate solution $\widetilde{X}$ to the *entropic OT* problem (6.2).

$$\min_{X\in\mathcal{U}(\boldsymbol{p}',\boldsymbol{q}')} \langle C, X\rangle - \eta H(X), \tag{6.2}$$

where $H(X) = -\sum_{i,j} X_{i,j}\log(X_{i,j})$ is the entropy. In the second step, one rounds $\widetilde{X}$ to the original feasible region $\mathcal{U}(\boldsymbol{p}, \boldsymbol{q})$. By taking proper parameters $\eta, \boldsymbol{p}', \boldsymbol{q}'$ and requiring a suitable accuracy level when approximating problem (6.2), the work [84] guarantees the final solution to be an ϵ-solution to problem (6.1).

Table 6.1: Order of complexity of OT algorithms. We list the year of the relevant publication, the names of the methods, the simplified version of its computational complexity, and whether (a $\surd$ sign) or not (an $\times$ sign) the method solves entropic OT as an intermediate step for approximating OT in columns. The bold font indicates a rate that we derive. It is clear that among the methods that solve entropic OT, our method achieves the lowest rate.

YEAR	ALGORITHM	COMPLEXITY	SOLVES ENTROPIC OT
2013	SINKHORN [85]	n^2/ϵ^2 [86]	$\surd$
2017	GREENKHORN [84]	n^2/ϵ^2 [87]	$\surd$
2018	STOCHASTIC SINKHORN [88]	n^2/ϵ^3; $\boldsymbol{n^2/\epsilon^2}$	$\surd$
2018	APDAGD [86]	$n^{2.5}/\epsilon$	$\surd$
2018	PACKING LP [89, 90]	n^2/ϵ	$\times$
2018	BOX CONSTRAINED NEWTON [89]	n^2/ϵ	$\surd$
2019	APDAMD [87]	$n^{2.5}/\epsilon$	$\surd$
2019	DUAL EXTRAPOLATION [91]	n^2/ϵ	$\times$
2019	ACCELERATED SINKHORN [92]	$n^{7/3}/\epsilon^{4/3}$	$\surd$
2019	DIJKSTRA'S SEARCH + DFS [93]	$n^2/\epsilon + n/\epsilon^2$	$\times$
2020	APDRCD [94]	$n^{2.5}/\epsilon$	$\surd$
2021	AAM [95]	$n^{2.5}/\epsilon$	$\surd$
2022	HYBRID PRIMAL-DUAL [96]	$n^{2.5}/\epsilon$	$\surd$
2022	PDASGD [97]	$n^{2.5}/\epsilon$	$\surd$
2022	PDASMD	$\boldsymbol{n^2/\epsilon}$	$\surd$

6.1.2 Literature Review

We review the state-of-the-art algorithms that solve OT by the two-step approach and summarize their computational complexity (measured by the number of numerical operations) for giving an ϵ-solution to OT in Table 6.1. The computational complexities in Table 6.1 are shown in their order of n and ϵ, where the $\log(n)$ term is omitted.

There are four main techniques to solve problem (6.2) in current literature:

- The first technique solves the dual problem of problem (6.2) by the Bregman projection technique. Specifically, this technique partitions the dual variables into blocks and iteratively updates each block. Algorithms that use this technique include Sinkhorn algorithm [85], Greenkhorn algorithm [84] and Stochastic Sinkhorn algorithm [88].
- The second technique also solves the dual problem of problem (6.2) but uses accelerated first-order methods. Algorithms that use this technique include accelerated gradient de-

scent (APDAGD) [86], accelerated mirror descent (APDAMD) [87], accelerated alternating minimization (AAM) [95], accelerated randomized coordinate descent (APDRCD) [94] and accelerated stochastic gradient descent (PDASGD) [97]. This technique can also be combined with the first technique. See, for example, the accelerated Sinkhorn algorithm in reference [92].

- The third technique solves the dual problem of problem (6.2) by second-order algorithms. An instance that uses this technique is the box-constrained Newton algorithm [89].
- The fourth technique minimizes the primal-dual gap of problem (6.2). An instance that uses this technique is the hybrid primal-dual algorithm [96].

Besides works that use the two-step approach to solve the entropic OT first, some works directly solve the unpenalized OT problem (6.1) by linear programming [89, 90], dual-extrapolation [91], or graph-based search algorithm [93].

We compare the computational complexity in Table 6.1 of our algorithm with other state-of-the-art algorithms as follows.

First, our PDASMD algorithm belongs to the second class of algorithms to solve the entropic OT problem (6.2). All other algorithms in this class reported a rate of $\widetilde{\mathcal{O}}(n^{2.5}/\epsilon)$ for approximating OT, while our algorithm has a better rate of $\widetilde{\mathcal{O}}(n^2/\epsilon)$. Thus our algorithm improves the rate for this class. The advantage of our algorithm mainly comes from the special technique that we use: though all the algorithms in this class use the acceleration technique, no accelerated variance reduction version of stochastic mirror descent has been tried in the previous algorithms. We apply those techniques to entropic OT and find that they lead to a better theoretical rate.

Second, among all algorithms for solving entropic OT, our PDASMD algorithm still reports the best rate. There is only one algorithm on entropic OT that achieved the same rate: the box-constrained Newton algorithm. However, we note that the Newton algorithm is a second-order algorithm, which requires computing the Hessian of the objective function. By its second-order nature, each step of the Newton algorithm will be expensive in

terms of computation and memory. On the other hand, our PDASMD algorithm is based on mirror descent, which is a first-order algorithm. Our PDASMD algorithm is thus easier to implement.

Finally, the algorithms that directly solve the original OT problem also report the same optimal rate as our PDASMD algorithm, including the packing LP algorithm, the dual extrapolation algorithm, and the graph-based Dijkstra DFS algorithm (when $\epsilon \gtrsim 1/n$). Compared with those algorithms, we have the extra advantage that our algorithm can not only approximate the OT problem but also solve the entropic OT. Thus, when one wants to solve the entropic OT, our algorithm is still preferred.

Our Contribution We summarize two main contributions in this chapter as follows.

- We propose an accelerated primal-dual stochastic algorithm that has computational complexity $\widetilde{\mathcal{O}}(n^2/\epsilon)$ for solving OT. Every step of our algorithm is defined by simple arithmetic operations and is counted in the complexity calculation. Thus our algorithm is practical. Moreover, compared with other algorithms that achieve the same rate for solving OT: our algorithm has the extra advantage that it can also be applied to entropic OT; it is a first-order algorithm, so it can be easily implemented without computing the Hessian. We also propose a batch version of our algorithm to increase the computational power.
- We prove that the computational complexity of the Stochastic Sinkhorn algorithm is $\widetilde{\mathcal{O}}(n^2/\epsilon^2)$, instead of the $\widetilde{\mathcal{O}}(n^2/\epsilon^3)$ rate in the literature. Our proved rate for Stochastic Sinkhorn matches the state-of-the-art rate of Sinkhorn and Greenkhorn. Moreover, the provable rate by our accelerated primal-dual stochastic algorithm is better than that of the Stochastic Sinkhorn, which again illustrates the advantage of our algorithm.

Chapter Organization The rest of this chapter is organized as follows. In Section 6.2, we present our main algorithm of Primal-Dual Accelerated Stochastic Proximal Mirror Descent (PDASMD), show its convergence, and analyze its complexity for solving OT;

as a comparison, we also prove the rate of Stochastic Sinkhorn, which is improved over the existing result. In Section 6.3, we develop a batch version of PDASMD and show its convergence and computational complexity. In Section 6.4, we run numerical examples to support our theorems.

6.2 Primal-Dual Accelerated Stochastic Proximal Mirror Descent (PDASMD)

In this section, we present our PDASMD algorithm for solving a linear constrained convex problem, which includes the entropic OT as a special case. We analyze the convergence rate of the PDASMD algorithm, then apply it to OT and derive the computational complexity. As a comparison, we also analyze the computational complexity of the Stochastic Sinkhorn. Since our algorithm uses the Proximal Mirror Descent technique, we review the background of such a technique in Section 6.5.1 and briefly explain why it is suitable for entropic OT.

6.2.1 Definition and Notation

We first introduce some notations that we will use throughout the rest of this chapter.

Notations: For a vector $\boldsymbol{a}$: let $\text{sign}(\boldsymbol{a})$ be such that $(\text{sign}(\boldsymbol{a}))_i = 1$ if $a_i > 0$ and -1 otherwise. For matrices $X \in \mathcal{R}^{n\times o}, Y \in \mathcal{R}^{p\times q}$: let $X \otimes Y$ denote the standard Kronecker product; let $\exp(X)$ and $\log(X)$ be the element-wise exponential and logarithm of X; let $\|X\|_2$ be the operator norm of X and $\|X\|_\infty$ be $\max_{i,j} |X_{i,j}|$; denote the matrix norm induced by two arbitrary vector norms $\|\cdot\|_H$ and $\|\cdot\|_E$ as $\|X\|_{E\to H} := \max_{\boldsymbol{a}:\|\boldsymbol{a}\|_E\leq 1} \|X\boldsymbol{a}\|_H$; denote the vectorization of X as $\text{Vec}(X) = (X_{11}, ..., X_{n1}, X_{12}, ..., X_{n2}, ..., X_{1o}, ..., X_{no})^T$. For two non-negative real values $s(\kappa)$ and $t(\kappa)$, denote $s(\kappa) = \Theta(t(\kappa))$ if $\exists k > 0$ and $K > 0$ such that $kt(\kappa) \leq s(\kappa) \leq Kt(\kappa)$.

Next, we review some key definitions that will be useful. [1]

Definition 6.1 (Strong convexity)**.** *$f : \mathcal{Q} \to \mathcal{R}$ is α-strongly convex w.r.t. $\|\cdot\|_H$ if $\forall \boldsymbol{x}, \boldsymbol{y} \in$*

[1] Our definitions follow those in [33].

$\mathcal{Q}$:

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\alpha}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_H^2.$$

Definition 6.2 (Smoothness)**.** *A convex function $f : \mathcal{Q} \to \mathcal{R}$ is β-smooth w.r.t. $\|\cdot\|_H$ if $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{Q}$:*

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_{H,*} \leq \beta \|\boldsymbol{x} - \boldsymbol{y}\|_H,$$

where $\|\boldsymbol{u}\|_{H,*} := \max_{\boldsymbol{v}}\{\langle \boldsymbol{u}, \boldsymbol{v} \rangle : \|\boldsymbol{v}\|_H \leq 1\}$ is the dual norm of $\|\cdot\|_H$. Or equivalently,

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{\beta}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_H^2.$$

Definition 6.3 (Bregman divergence)**.** *For a mirror function $w(\cdot)$ that is 1-strongly convex w.r.t. $\|\cdot\|_H$, we denote by $V_{\boldsymbol{x}}(\boldsymbol{y})$ the Bregman divergence w.r.t. $\|\cdot\|_H$ generated by $w(\cdot)$, where*

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

One can conclude from the definition that

$$V_{\boldsymbol{x}}(\boldsymbol{y}) \geq \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_H^2.$$

If we further assume that the mirror function $w(\cdot)$ is γ-smooth w.r.t. $\|\cdot\|_H$, we then have

$$V_{\boldsymbol{x}}(\boldsymbol{y}) \leq \frac{\gamma}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_H^2.$$

6.2.2 General Formulation and PDASMD Algorithm

In this section, we first state a general linear constrained problem and explain how it includes entropic OT as a special case. We then propose our algorithm to solve this general problem. Finally, we show the convergence rate of our algorithm.

We consider a linear constrained problem as follows:

$$\min_{\boldsymbol{x}\in\mathcal{R}^m} f(\boldsymbol{x}) \qquad s.t. A\boldsymbol{x} = \boldsymbol{b} \in \mathcal{R}^l, \tag{6.3}$$

where f is strongly convex. One observe that the entropic OT (6.2) is a special case of problem (6.3) with $\boldsymbol{x} = \mathrm{Vec}(X)$, $f(\boldsymbol{x}) = \langle \mathrm{Vec}(C), \boldsymbol{x}\rangle + \eta \sum_{i,j=1}^n x_{in+j} \log(x_{in+j})$, $\boldsymbol{b} = (\boldsymbol{p}^T, \boldsymbol{q}^T)^T$, $A = \begin{bmatrix} \mathbf{1}^T \otimes I_n \\ I_n \otimes \mathbf{1}^T \end{bmatrix}$.

A standard approach for solving the constrained problem (6.3) is to optimize its Lagrange dual problem (6.4):

$$\begin{aligned} \min_{\boldsymbol{\lambda}}\{\phi(\boldsymbol{\lambda}) :=& \langle \boldsymbol{\lambda}, \boldsymbol{b}\rangle + \max_{\boldsymbol{x}}(-f(\boldsymbol{x}) - \langle A^T\boldsymbol{\lambda}, \boldsymbol{x}\rangle) \\ =& \langle \boldsymbol{\lambda}, \boldsymbol{b}\rangle - f(\boldsymbol{x}(\boldsymbol{\lambda})) - \langle A^T\boldsymbol{\lambda}, \boldsymbol{x}(\boldsymbol{\lambda})\rangle\}, \end{aligned} \tag{6.4}$$

where by F.O.C. $\boldsymbol{x}(\boldsymbol{\lambda})$ is such that

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}(\boldsymbol{\lambda})) = -A^T\boldsymbol{\lambda}. \tag{6.5}$$

Since problem (6.3) is a linear constrained convex problem, the strong duality holds. Thus solving problem (6.3) is equivalent to solving its dual problem (6.4). In particular, we develop a stochastic algorithm for the case that the dual is of finite sum form. We further assume that all terms in the finite sum are smooth for convergence analysis. The conditions on the dual are formalized as follows:

Assumption 6.1 (Finite-sum dual)**.** *Assume that the dual can be written as*

$$\phi(\boldsymbol{\lambda}) = \frac{1}{m}\sum_{i=1}^m \phi_i(\boldsymbol{\lambda}),$$

where ϕ_i is convex and L_i-Lipchitz smooth w.r.t. an arbitrary $\|\cdot\|_H$ norm.

Note that the assumption on the dual is reasonable and can be satisfied by some problems, including entropic OT. We now give a concrete example that the assumption holds. Consider a primal objective $f(\boldsymbol{x}) = \sum_{i=1}^{m} f_i(x_i)$ where each f_i is $\nu-$strongly convex w.r.t. another arbitrary norm $\|\cdot\|_E$ (note that it can be different from the $\|\cdot\|_H$ norm). In this case, we can solve the primal-dual relationship in equation (6.5) to get:

$$x_i(\boldsymbol{\lambda}) = (\nabla f_i)^{-1}(-\boldsymbol{a}_i^T \boldsymbol{\lambda}), i = 1, \ldots, m,$$

where $\boldsymbol{a}_i$ is the ith column of A. As a consequence, the dual problem (6.4) can be written as a finite sum:

$$\begin{aligned}\phi(\boldsymbol{\lambda}) &= \frac{1}{m}\sum_{i=1}^{m}(\langle \boldsymbol{\lambda}, \boldsymbol{b}_i\rangle - m f_i(x_i(\boldsymbol{\lambda})) - m\boldsymbol{a}_i^T\boldsymbol{\lambda} x_i(\boldsymbol{\lambda})) \\ &:= \frac{1}{m}\sum_{i=1}^{m}\phi_i(\boldsymbol{\lambda}),\end{aligned}$$

where $\boldsymbol{b}_i$'s are arbitrarily chosen vectors satisfying the constraint $\sum_{i=1}^{m}\boldsymbol{b}_i = m\boldsymbol{b}$. One can check that $\nabla\phi_i(\boldsymbol{\lambda}) = \boldsymbol{b}_i - m x_i(\boldsymbol{\lambda})\boldsymbol{a}_i$. By [98], ϕ_i is convex and L_i-Lipchitz smooth w.r.t. $\|\cdot\|_H$ norm, where $L_i \leq \frac{m}{\nu}\|\boldsymbol{a}_i\|_{E\to H,*}$.

With the finite sum representation of ϕ, we propose a PDASMD algorithm (Algorithm 6.1) to solve problem (6.3). We add a few remarks to explain the algorithm as follows.

Remark 6.1. *One should choose a specific $\|\cdot\|_H$ norm and a mirror function $w(\cdot)$ to run the algorithm. Those choices have a direct impact on the mirror descent step 10 and proximal gradient descent step 11: if we let $\|\cdot\|_H = \|\cdot\|_2$ and $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, both steps reduce to stochastic gradient descent, then the algorithm essentially reduces to the PDASGD algorithm in [97].*

Remark 6.2. *The primal variables $\boldsymbol{x}$'s in Algorithm 6.1 are updated by Steps 14 through 16, and we explain those steps as follows: The iterates in Steps 14 through 16 essentially leads to $\boldsymbol{x}^{S-1} = \left(\sum_{s=0}^{S-1}\boldsymbol{x}(\widetilde{\boldsymbol{y}}_s)/\tau_{1,s}\right) \Big/ \left(\sum_{s=0}^{S-1}(1/\tau_{1,s})\right)$. We express such updates in $\boldsymbol{x}^s$ in*

Algorithm 6.1: Primal-Dual Accelerated Stochastic Proximal Mirror Descent (PDASMD)

1: Initialize l the number of inner iterations; $\tau_2 = \frac{1}{2}$, $\boldsymbol{y}_0 = \boldsymbol{z}_0 = \widetilde{\boldsymbol{v}}_0 = \boldsymbol{v}_0 = \mathbf{0}$, $C_0 = D_0 = 0$; choose a mirror function $w(\cdot)$ that is 1-strongly convex and γ-smooth w.r.t. $\|\cdot\|_H$, and denote by $V_{\boldsymbol{x}}(\boldsymbol{y})$ the Bregman divergence generated by $w(\cdot)$; take $\bar{L} = (\sum_{i=1}^m L_i)/m$, where L_i is the smoothness (w.r.t. $\|\cdot\|_H$) for each component ϕ_i of the dual function $\phi(\cdot)$ in Assumption 6.1.
2: **for** s = 0,. . . ,S-1 **do**
3: $\tau_{1,s} \leftarrow 2/(s+4)$; $\alpha_s \leftarrow 1/(9\tau_{1,s}\bar{L})$;
4: $\boldsymbol{\mu}^s \leftarrow \nabla\phi(\widetilde{\boldsymbol{v}}^s)$.
5: **for** $j = 0$ to $l-1$ **do**
6: $k \leftarrow (sl) + j$;
7: $\boldsymbol{v}_{k+1} \leftarrow \tau_{1,s}\boldsymbol{z}_k + \tau_2\widetilde{\boldsymbol{v}}^s + (1-\tau_{1,s}-\tau_2)\boldsymbol{y}_k$;
8: Pick i randomly from $\{1, 2, \ldots, m\}$, each with probability $p_i := L_i/m\bar{L}$;
9: $\widetilde{\boldsymbol{\nabla}}_{k+1} \leftarrow \boldsymbol{\mu}^s + \frac{1}{mp_i}(\nabla\phi_i(\boldsymbol{v}_{k+1}) - \nabla\phi_i(\widetilde{\boldsymbol{v}}^s))$;
10: $\boldsymbol{z}_{k+1} = \arg\min_{\boldsymbol{z}}\{\frac{1}{\alpha_s}V_{\boldsymbol{z}_k}(\boldsymbol{z}) + \langle\widetilde{\boldsymbol{\nabla}}_{k+1}, \boldsymbol{z}\rangle\}$;
11: $\boldsymbol{y}_{k+1} = \arg\min_{\boldsymbol{y}}\{\frac{9\bar{L}}{2}\|\boldsymbol{y} - \boldsymbol{v}_{k+1}\|_H^2 + \langle\widetilde{\boldsymbol{\nabla}}_{k+1}, \boldsymbol{y}\rangle\}$.
12: **end for**
13: $\widetilde{\boldsymbol{v}}^{s+1} \leftarrow \frac{1}{l}\sum_{j=1}^{l}\boldsymbol{y}_{sl+j}$;
14: $C_s \leftarrow C_s + \frac{1}{\tau_{1,s}}$;
15: Pick $\widetilde{\boldsymbol{y}}_s$ uniform randomly from $\{\boldsymbol{y}_{sl+j}\}_{j=1}^l$, update $D_s \leftarrow D_s + \frac{1}{\tau_{1,s}}\boldsymbol{x}(\widetilde{\boldsymbol{y}}_s)$, where $\boldsymbol{x}(\cdot)$ is given by equation (6.5);
16: $\boldsymbol{x}^s = D_s/C_s$.
17: **end for**
18: **Output:** $\widetilde{\boldsymbol{x}} = \boldsymbol{x}^{S-1}$.

an iterative way to avoid storing all updates of $\widetilde{\boldsymbol{y}}_s$'s. In this way, our algorithm is memory efficient.

Remark 6.3. *The dual variables $\boldsymbol{v}, \boldsymbol{z}, \boldsymbol{y}$'s are updated by Steps 2 through 13. The update consists of outer loops indexed by s and inner loops indexed by j, which follow Algorithm 5 in [33]. One may refer to that paper for more details on how the outer and inner loops are designed for the variance reduction and acceleration to work.*

We prove the convergence rate of the PDASMD algorithm as follows:

Theorem 6.1. *In Algorithm 6.1, we choose a mirror function $w(\cdot)$ that is 1-strongly convex and γ-smooth w.r.t. $\|\cdot\|_H$ norm. Denote the primal and dual optimal solution as $\boldsymbol{x}^*$ and*

$\boldsymbol{\lambda}^*$, *respectively. Assume that* $\|\lambda^*\|_H \leq R$. *We have the convergence of the algorithm as:*

$$\|\mathbb{E}[\boldsymbol{b} - A\boldsymbol{x}^{S-1}]\|_{H,*} \leq \frac{2}{S^2 l}\left[l\bar{L}R + 18\bar{L}R\gamma\right], \tag{6.6}$$

$$f(\mathbb{E}(\boldsymbol{x}^{S-1})) - f(\boldsymbol{x}^*) \leq \frac{2}{S^2 l}\left[l\bar{L}R^2 + 18\bar{L}R^2\gamma\right]. \tag{6.7}$$

The proof of the theorem is deferred to Section 6.5.2.

6.2.3 Applying to Optimal Transport

In this section, we give the detailed procedure of applying PDASMD to get an approximation solution to the OT. Especially, we consider two cases: in the first case, we use $\|\cdot\|_H = \|\cdot\|_2$ and PDASMD reduce to PDASGD; in the second case, we use $\|\cdot\|_H = \|\cdot\|_\infty$ and prove an improved computational complexity over the first case. For the latter case, our algorithm achieves the best possible rate in the current literature. Our algorithm improves the rate of the first-order algorithms for solving entropic OT.

We apply the PDASMD algorithm to solve the entropic OT (6.2) as follows. Since problem (6.2) a special case of problem (6.3), we plug $A, \boldsymbol{b}, f(\cdot)$ into the general dual formula (6.4) to get the dual problem of problem (6.2). With a little abuse of notation, we split the dual variables as $(\boldsymbol{\tau}^T, \boldsymbol{\lambda}^T)^T$ for $\boldsymbol{\tau}, \boldsymbol{\lambda} \in \mathcal{R}^n$. The dual problem of problem (6.2) is:

$$\phi(\boldsymbol{\tau}, \boldsymbol{\lambda}) = \eta\langle \mathbf{1}_{n^2}, \boldsymbol{x}(\boldsymbol{\tau}, \boldsymbol{\lambda})\rangle - \langle \boldsymbol{p}', \boldsymbol{\tau}\rangle - \langle \boldsymbol{q}', \boldsymbol{\lambda}\rangle, \tag{6.8}$$

where the relationship between primal-dual variables is

$$\boldsymbol{x}(\boldsymbol{\tau}, \boldsymbol{\lambda}) = \exp\left(\frac{A^T(\boldsymbol{\tau}^T, \boldsymbol{\lambda}^T)^T - \mathrm{Vec}(C) - \eta\mathbf{1}_{n^2}}{\eta}\right). \tag{6.9}$$

Moreover, to get a dual with the finite-sum structure, we follow [99] to transfer the dual objective to semi-dual by fixing $\boldsymbol{\lambda}$ and solving the first order condition w.r.t. $\boldsymbol{\tau}$ in problem (6.8). With a little abuse of notation, we denote the semi-dual objective function as $\phi(\boldsymbol{\lambda})$.

Then we have

$$\begin{aligned}\phi(\boldsymbol{\lambda}) &= -\langle \boldsymbol{q}', \boldsymbol{\lambda}\rangle - \eta \sum_{i=1}^{n} p_i' \log p_i' \\ &\quad + \eta \sum_{i=1}^{n} \log\left(\sum_{j=1}^{n} \exp((\lambda_j - C_{i,j} - \eta)/\eta)\right) + \eta \\ &= \frac{1}{n}\sum_{i=1}^{n} np_i' \Bigg[- \langle \boldsymbol{q}', \boldsymbol{\lambda}\rangle - \eta \log p_i' \\ &\quad + \eta \log\left(\sum_{j=1}^{n} \exp((\lambda_j - C_{i,j} - \eta)/\eta)\right) + \eta \Bigg] \\ &:= \frac{1}{n}\sum_{i=1}^{n} \phi_i(\boldsymbol{\lambda}). \end{aligned} \tag{6.10}$$

We have the relationship between the dual variables:

$$\tau_i(\boldsymbol{\lambda}) = \eta \log p_i' - \eta \log\left(\sum_{j=1}^{n} \exp((\lambda_j - C_{i,j} - \eta)/\eta)\right).$$

It is easy to check that each $\phi_i(\boldsymbol{\lambda})$ is convex. To apply our algorithm, we further check the smoothness of $\phi_i(\boldsymbol{\lambda})$ in the following lemma:

Lemma 6.1. *$\phi_i(\cdot)$ in the semi-dual problem* (6.10) *is $\frac{np_i'}{\eta}$ smooth w.r.t. $\|\cdot\|_2$ norm, and is $\frac{5np_i'}{\eta}$ smooth w.r.t. $\|\cdot\|_\infty$ norm.*

Lemma 6.1 is proved in Section 6.5.3. By Lemma 6.1, we can calculate the parameter in PDASMD Algorithm 6.1 as $\bar{L} = 1/\eta$ for $\|\cdot\|_H = \|\cdot\|_2$, and $\bar{L} = 5/\eta$ for $\|\cdot\|_H = \|\cdot\|_\infty$. For these two cases, we can apply Algorithm 6.1 to approximate problem (6.2). We further round the approximating solution of problem (6.2) to feasible region of problem (6.1). In this way, we get an $\epsilon-$solution to problem (6.1). The full procedure is included in Section 6.5.4. We state the computational complexity of the full procedure in the following theorem:

Theorem 6.2. *Set $l = \Theta(n)$ in the PDASMD algorithm, the overall number of arithmetic*

operations for finding a solution $\widehat{X}$ *such that* $\mathbb{E}\langle C, \widehat{X}\rangle \leq \langle C, X^*\rangle + \epsilon$ *is*

- $\widetilde{\mathcal{O}}\left(\frac{n^{2.5}\|C\|_\infty(1+\sqrt{\gamma/n})}{\epsilon}\right)$ *for* $\|\cdot\|_H = \|\cdot\|_2$*;*
- $\widetilde{\mathcal{O}}\left(\frac{n^{2}\|C\|_\infty(1+\sqrt{\gamma/n})}{\epsilon}\right)$ *for* $\|\cdot\|_H = \|\cdot\|_\infty$*.*

The proof of Theorem 6.2 is in Section 6.5.4.

Remark 6.4. *The complexities still depend on* γ*, the smoothness of* $w(\cdot)$ *w.r.t.* $\|\cdot\|_H$*. For example, when taking* $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$*, we have* $\gamma = 1$ *for* $\|\cdot\|_H = \|\cdot\|_2$*, and* $\gamma = n$ *for* $\|\cdot\|_H = \|\cdot\|_\infty$*. The corresponding computational complexity is then* $\widetilde{\mathcal{O}}\left(\frac{n^{2.5}\|C\|_\infty}{\epsilon}\right)$ *and* $\widetilde{\mathcal{O}}\left(\frac{n^{2}\|C\|_\infty}{\epsilon}\right)$*. Now for* $\|\cdot\|_H = \|\cdot\|_\infty$*, as long as we choose a proper* $w(\cdot)$ *such that* $\gamma = \mathcal{O}(n)$*, the rate* $\widetilde{\mathcal{O}}\left(\frac{n^{2}\|C\|_\infty}{\epsilon}\right)$ *is achieved. One may further improve the rate by a constant by improving the dependency of* γ *on* n*. Such improvement is an open question in optimization; though we make no effort to do it in this chapter, we still note this opportunity.*

Remark 6.5. *If we choose* $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$*, we have closed-form solutions for each step of PDASMD.*

- *For both settings, step 10 of PDASMD algorithm becomes* $\boldsymbol{z}_{k+1} = \boldsymbol{z}_k - \alpha_s \widetilde{\boldsymbol{\nabla}}_{k+1}$*;*
- *For* $\|\cdot\|_H = \|\cdot\|_2$*, step 11 of PDASMD is* $\boldsymbol{y}_{k+1} = \boldsymbol{v}_{k+1} - \frac{1}{9\widetilde{L}}\widetilde{\boldsymbol{\nabla}}_{k+1}$*;*
- *For* $\|\cdot\|_H = \|\cdot\|_\infty$*, step 11 of PDASMD becomes* $\boldsymbol{y}_{k+1} = \boldsymbol{v}_{k+1} - \frac{\|\widetilde{\boldsymbol{\nabla}}_{k+1}\|_1}{9\widetilde{L}} sign(\widetilde{\boldsymbol{\nabla}}_{k+1})$*.*

It is clear that in both settings, each step of PDASMD is defined by simple arithmetic operations and thus is easy to implement. There is no gap between our theory and practice.

6.2.4 Computational Complexity of the Stochastic Sinkhorn

In this section, we prove that the computational complexity of the Stochastic Sinkhorn for finding an ϵ-solution to OT is $\widetilde{\mathcal{O}}(\frac{n^2}{\epsilon^2})$, which is improved over the known rate of $\widetilde{\mathcal{O}}(\frac{n^2}{\epsilon^3})$ [88] and matches the state-of-the-art rate of Sinkhorn and Greenkhorn [86, 87]. Moreover, our PDASMD algorithm beats the provable rate of Stochastic Sinkhorn. This illustrates the advantage of our PDASMD algorithm.

The Stochastic Sinkhorn algorithm is proposed by [88]. One can check Section 6.5.5 for a full algorithm description. We show the computational complexity of Stochastic Sinkhorn as follows:

Theorem 6.3. *Stochastic Sinkhorn finds a solution $\widehat{X}$ such that $\mathbb{E}\langle C, \widehat{X}\rangle \leq \langle C, X^*\rangle + \epsilon$ in*

$$\mathcal{O}\left(\frac{n^2\|C\|_\infty^2 \log n}{\epsilon^2}\right)$$

arithmetic operations.

The proof of Theorem 6.3 is in Section 6.5.5.

6.3 PDASMD with Batch Implementation (PDASMD-B)

In this section, we propose a batch version of PDASMD, namely the PDASMD-B algorithm. The batch implementation of the stochastic step in PDASMD-B allows parallel computing. This further improves the computational power of our algorithm.

We give PDASMD-B in Algorithm 6.2 and briefly explain it as follows. As compared to the non-batch version PDASMD in Algorithm 6.1, Step 8 of PDASMD-B now samples a small batch of samples and calculates $\widetilde{\boldsymbol{\nabla}}_{k+1}$ based on the gradient of this small batch. Other hyper-parameters in the algorithm are changed accordingly to ensure convergence.

We apply PDASMD-B to solve OT. The main steps are the same as those in Section 6.2.3; thus, we omit the details. To compute the computational complexity for giving an ϵ-solution to OT, one needs the convergence result of PDASMD-B, which we include in Section 6.5.6. And the computational complexity for solving OT is stated in the following corollary.

Corollary 6.1. *Run PDASMD-B with batch size B, $\|\cdot\|_H = \|\cdot\|_\infty$ and inner loop size $l = n/B$ (assume w.l.o.g. that l is an integer), the overall number of arithmetic operations*

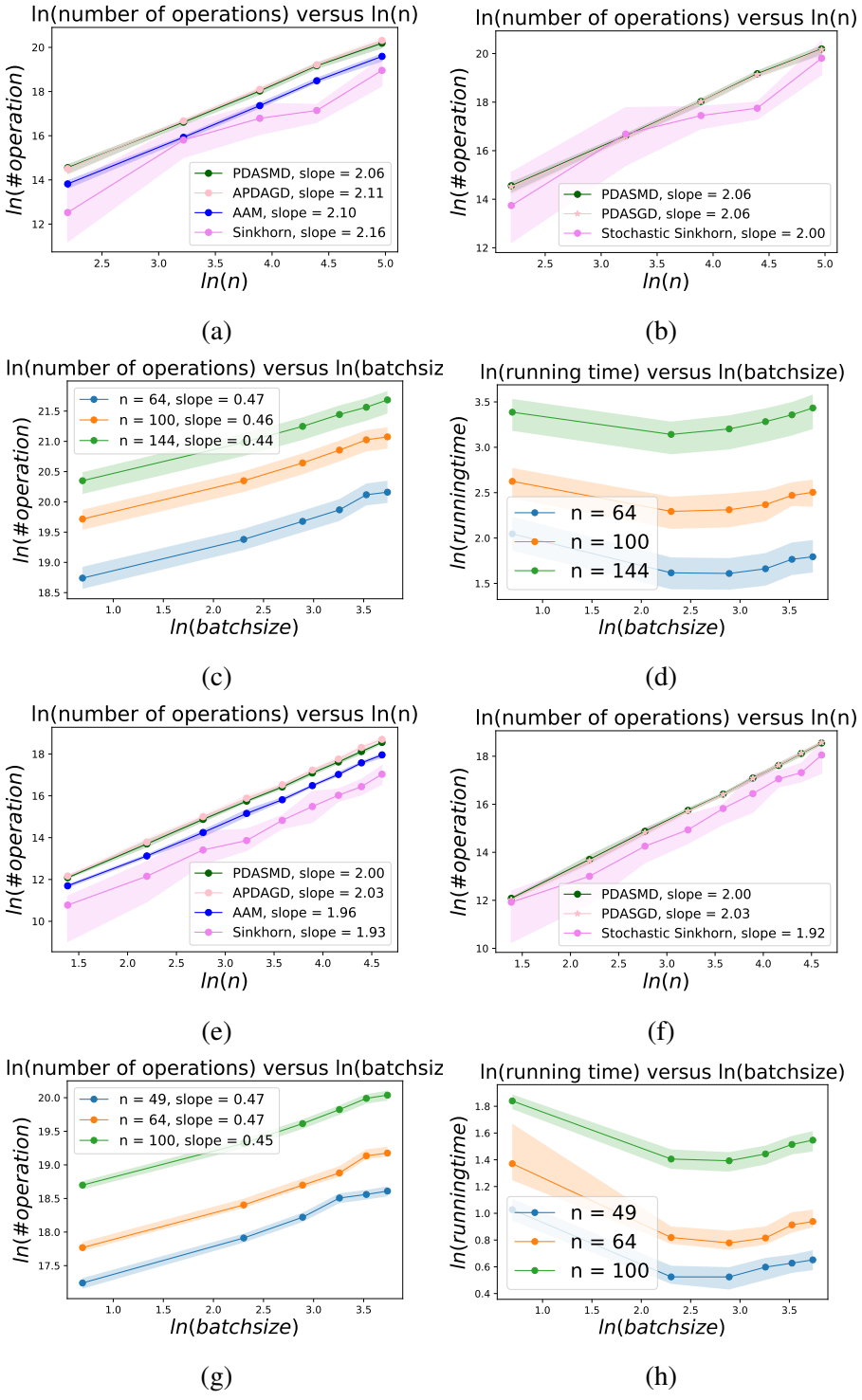


Figure 6.1: Computational complexity comparison of different algorithms for finding an ϵ-solution of OT. The logarithmic of the total number of numerical operations to achieve a given ϵ approximation error is plotted against either the logarithmic transform of the sample size n in the PDASMD algorithm (rows 1 and 3) or the batch size in the PDASMD-B algorithm (rows 2 and 4). The first two rows use synthetic data, and the last two are for the MNIST data. The relevant discussion can be seen in Section 6.4 Numerical Studies. The error bars in all the plots come from repeating the experiment using 5 pairs of randomly generated/chosen marginals.

Algorithm 6.2: Batch PDASMD (PDVRASMD-B)

Initialize l the number of inner iterations, B the batch size; set $\tau_2 \leftarrow \frac{1}{2B}$, $C_0 = D_0 = 0$, $\boldsymbol{y}_0 = \boldsymbol{z}_0 = \widetilde{\boldsymbol{v}}_0 = \boldsymbol{v}_0 = \mathbf{0}$; choose a mirror function $w(\cdot)$ that is 1-strongly convex and γ-smooth w.r.t. $\|\cdot\|_H$, and denote by $V_{\boldsymbol{x}}(\boldsymbol{y})$ the Bregman divergence generated by $w(\cdot)$; take $\bar{L} = (\sum_{i=1}^m L_i)/m$, where L_i is the smoothness (w.r.t. $\|\cdot\|_H$) for each component ϕ_i of the dual function $\phi(\cdot)$ in Assumption 6.1.
for s = 0,. . . ,S-1 **do**
  $\tau_{1,s} \leftarrow 2/(s+4)$; $\alpha_s \leftarrow 1/(9\tau_{1,s}\bar{L})$;
  $\boldsymbol{\mu}^s \leftarrow \nabla\phi(\widetilde{\boldsymbol{v}}^s)$.
  for $j = 0$ to $l-1$ **do**
    $k \leftarrow (sl) + j$;
    $\boldsymbol{v}_{k+1} \leftarrow \tau_{1,s}\boldsymbol{z}_k + \tau_2\widetilde{\boldsymbol{v}}^s + (1-\tau_{1,s}-\tau_2)\boldsymbol{y}_k$;
    Pick B samples independently from $\{1, 2, \ldots, m\}$ with replacement, where sample i is picked with probability $p_i = L_i/m\bar{L}$; denote the sampled index set as I;
    $\widetilde{\boldsymbol{\nabla}}_{k+1} \leftarrow \boldsymbol{\mu}^s + \frac{1}{B}\sum_{i\in I}\frac{1}{mp_i}(\nabla\phi_i(\boldsymbol{v}_{k+1}) - \nabla\phi_i(\widetilde{\boldsymbol{v}}^s))$;
    $\boldsymbol{z}_{k+1} = \arg\min_{\boldsymbol{z}}\{\frac{1}{\alpha_s}V_{\boldsymbol{z}_k}(\boldsymbol{z}) + \langle\widetilde{\boldsymbol{\nabla}}_{k+1}, \boldsymbol{z}\rangle\}$;
    $\boldsymbol{y}_{k+1} = \arg\min_{\boldsymbol{y}}\{\frac{9\bar{L}}{2}\|\boldsymbol{y} - \boldsymbol{v}_{k+1}\|_H^2 + \langle\widetilde{\boldsymbol{\nabla}}_{k+1}, \boldsymbol{y}\rangle\}$.
  end for
  $\widetilde{\boldsymbol{v}}^{s+1} \leftarrow \frac{1}{l}\sum_{j=1}^{l}\boldsymbol{y}_{sl+j}$;
  $C_s \leftarrow C_s + \frac{1}{\tau_{1,s}}$;
  Pick $\widetilde{\boldsymbol{y}}_s$ uniform randomly from $\{\boldsymbol{y}_{sl+j}\}_{j=1}^l$, update $D_s \leftarrow D_s + \frac{1}{\tau_{1,s}}\boldsymbol{x}(\widetilde{\boldsymbol{y}}_s)$, where $\boldsymbol{x}(\cdot)$ is given by equation (6.5);
  $\boldsymbol{x}^s = D_s/C_s$.
end for
Output: $\widetilde{\boldsymbol{x}} = \boldsymbol{x}^{S-1}$.

to find a solution $\widehat{X}$ such that $\mathbb{E}\langle C, \widehat{X}\rangle \le \langle C, X^*\rangle + \epsilon$ is

$$\widetilde{\mathcal{O}}\left(\frac{n^2\|C\|_\infty\sqrt{1/B + B\gamma/n}}{\epsilon}\right).$$

Remark 6.6. *Corollary 6.1 shows the speed-up of PDASMD-B from parallel computing. We analyzed the speed-up for two cases of γ as follows. The first case is similar to the one in Remark 6.4: taking $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$, then we have $\gamma = n$. This gives us the total computation of $\widetilde{\mathcal{O}}\left(\frac{n^2\|C\|_\infty\sqrt{B}}{\epsilon}\right)$, which is $\sqrt{B}$ times that of non-batch version. There are B batches of parallel computation, so if we ignore the communication time, our batch algorithm enjoys*

a sublinear speed-up of $\mathcal{O}(\sqrt{B})$. The second case assumes one can further improve the rate $\gamma \sim \mathcal{O}(n)$ to $\gamma \sim \mathcal{O}(\sqrt{n})$. Then for $B \leq \sqrt{n}$, the number of total computations does not increase with B, which indicates a linear speed-up of $\mathcal{O}(B)$ using parallel computing. Though such an improvement in γ is still an open question in optimization, this implies a potentially huge advantage of the batch algorithm.

6.4 Numerical Studies

In this section, we discuss the result of our numerical studies. The goals of our experiment are to check our theoretical computational complexity of the PDASMD algorithm w.r.t. the marginal size n in Theorem 6.2, and to check the theoretical computational complexity of the PDASMD-B algorithm w.r.t. the batch size B in Corollary 6.1. We use both synthetic and real grey-scale images [2] as the marginal distribution for our experiment. The technique details as data description and algorithm implementation are deferred to Section 6.5.7. We have more applications of our algorithm, including domain adaptation and color transfer, in Section 6.5.8.

Our experiment results are given in Figure 6.1. We now explain the plots and summarize the results from the plots as follows.

Figures 6.1a, 6.1b, 6.1e and 6.1f check the computational complexity of PDASMD on the marginal size n. In our experiment, we run PDASMD with $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ and $\|\cdot\|_H = \|\cdot\|_\infty$. By Theorem 6.2, for this case, when fixing the accuracy level ϵ, we should have the computational complexity $\sim \mathcal{O}(n^2)$. That is, fixing a ϵ and plotting the logarithm of computation count versus the logarithm of n, we expect to see a line with slope 2. Figures 6.1a, 6.1b (using synthetic data as marginals) and Figures 6.1e and 6.1f (using real data as marginals) have the lines corresponding to the PDASMD algorithm have slopes that are close to 2, which supports our theoretical rate.

In Figures 6.1a, 6.1b, 6.1e and 6.1f, we also include lines that correspond to other state-

[2] The MNIST dataset [100].

of-the-art algorithms. The goal is to compare the practical performance of the PDASMD algorithm with deterministic algorithms (Figures 6.1a and 6.1e) and other stochastic algorithms (Figure 6.1b and 6.1f). We conclude from the plots that the total computation numbers of the AAM, Sinkhorn and Stochastic Sinkhorn are less than that of the PDASMD, which illustrates the practical advantage of those algorithms. However, such an observation does not disqualify our PDASMD algorithm since we still have a provable complexity that is better than those algorithms. Inspired by such an observation, one may further improve the PDASMD in practice. One possible way is to combine the PDASMD algorithm with the Sinkhorn to take advantage of the better theoretical rate of PDASMD and the good empirical performance of the Sinkhorn.

Figures 6.1c and 6.1g check the computational complexity of PDASMD-B on the batch size B. We fix the accuracy level ϵ and run PDASMD-B with $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$. By Corollary 6.1, for a given marginal size n, we have the number of total computation $\sim \mathcal{O}(\sqrt{B})$. Thus, when plotting the logarithm of computation count versus the logarithm of B, we should get a line with slope 0.5. In Figures 6.1c (using synthetic data as marginals) and 6.1g (using real data as marginals), we see that for different marginal size n, the slopes are all close to $.5$. Such an observation matches our theory.

With such computational complexity of PDASMD-B on the batch size B, if we can fully parallelize, the running time of PDASMD-B should be $\sim \mathcal{O}(B^{-0.5})$. To check this, we plot the logarithm of running time versus the logarithm of B in Figures 6.1d and 6.1h. The lines fail to have slope -0.5. This is not surprising to see in practice because of the commutation time and limit in the computational resource. But from the plots, we can still benefit from the batch algorithm: when the batch size is not too large ($<= \exp(2.5)$), the running time decreases as the batch size increases. This illustrates the usefulness of the batch version algorithm in practice.

To summarize, our computational complexity of PDASMD on n and PDASMD-B on B are supported by numerical studies.

6.5 Technique Details for Chapter 6

In this section, we provide technique details for Chapter 6.

6.5.1 Proximal Mirror Descent

In this section, we review the technique of stochastic proximal mirror descent.

Let us start with the objective function:

$$\min_{\boldsymbol{x}} F(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}). \tag{6.11}$$

A popular way to minimize problem (6.11) is the Stochastic Gradient Descent (SGD). At time t, the SGD algorithm randomly samples i_t from $\{1, \ldots, n\}$ and updates as:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - b_t \nabla f_{i_t}(\boldsymbol{x}_t), \tag{6.12}$$

where b_t is the step size. Note that formula (6.12) is essentially the solution to the following ℓ_2 penalized problem:

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}} \left\{ \langle \boldsymbol{x}, \nabla f_{i_t}(\boldsymbol{x}_t) \rangle + \frac{1}{2b_t} \|\boldsymbol{x} - \boldsymbol{x}_t\|_2^2 \right\}. \tag{6.13}$$

The proximal/mirror descent is proposed by Nemirovski and Yudin [54], where they generalize the SGD by replacing the $\|\cdot\|_2^2$ term in problem (6.13) by some proximity function. There are two popular choices of proximity functions, and they lead to stochastic proximal and mirror descent, respectively. In this chapter, we use stochastic proximal mirror descent to represent both cases.

The choice of proximity function that leads to stochastic proximal gradient descent is the square of an arbitrary norm $\|\cdot\|_H$ (as compared to the $\|\cdot\|_2$ norm in problem (6.13)).

This results in the update

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}} \left\{ \langle \boldsymbol{x}, \nabla f_{i_t}(\boldsymbol{x}_t) \rangle + \frac{1}{2b_t} \|\boldsymbol{x} - \boldsymbol{x}_t\|_H^2 \right\}. \tag{6.14}$$

The choice of proximity function that gives stochastic mirror descent is the Bregman divergence. Recall that for a mirror map $w(\cdot)$, the Bregman divergence is

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

The stochastic mirror descent then updates as:

$$\boldsymbol{x}_{t+1} = \arg\min_{\boldsymbol{x}} \left\{ \langle \boldsymbol{x}, \nabla f_{i_t}(\boldsymbol{x}_t) \rangle + \frac{1}{2b_t} V_{\boldsymbol{x}_t}(\boldsymbol{x}) \right\}. \tag{6.15}$$

Note that the popular KL-divergence $\mathcal{KL}(\boldsymbol{x}||\boldsymbol{x}') := \sum_i x_i \log(x_i/x_i') - \sum_i x_i + \sum_i x_i'$ is a special case of Bregman divergence by choosing w to be the negative entropy $w(\boldsymbol{x}) = \sum_i x_i \log x_i$.

Recall that the objective is to solve the (entropic) OT, so we explain why the proximal mirror algorithm might be suitable for optimizing the entropic OT compared with the SGD.

First, the objective function of entropic OT coincides with the proximal mirror descent formulation in that each step of proximal mirror descent minimizes an inner product term plus a divergence term other than the ℓ_2 norm. In this way, the proximal mirror descent may help to prove a faster convergence when solving OT.

Second, it is pointed out that the popular Sinkhorn algorithm to solve entropic OT can be interpreted as a special case of the stochastic proximal mirror descent algorithm [101]. We briefly summarize their statement as follows. The Sinkhorn algorithm iteratively updates the dual variables $\boldsymbol{u}, \boldsymbol{v}$ of problem (6.2) by:

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k + log\boldsymbol{p}' - log(X(\boldsymbol{u}^k, \boldsymbol{v}^k)\mathbf{1}), \boldsymbol{v}^{k+1} = \boldsymbol{v}^k, \tag{6.16}$$

and

$$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k, \boldsymbol{v}^{k+1} = \boldsymbol{v}^k + log\boldsymbol{q}' - log(X(\boldsymbol{u}^k, \boldsymbol{v}^k)^T \mathbf{1}), \tag{6.17}$$

where the relationship between the primal-dual variables is

$$X(\boldsymbol{u}, \boldsymbol{v}) = \text{diag}(\exp(\boldsymbol{u}))\exp(-C/\eta)\text{diag}(\exp(\boldsymbol{v})).$$

Notice that the dual variables $\boldsymbol{u}, \boldsymbol{v}$ are equivalent to the dual variables $\boldsymbol{\lambda}, \boldsymbol{\tau}$ we use in formulation (6.9) plus constants.

To interpret Sinkhorn as a Stochastic Mirror Descent, one considers the objective function:

$$\min_{X \in \mathcal{R}^{n \times n}} f(X) := \frac{1}{2}(f_1(X) + f_2(X)) \tag{6.18}$$

$$f_1(X) = \mathcal{KL}(X\mathbf{1}||\boldsymbol{p}'), f_2(X) = \mathcal{KL}(X^T\mathbf{1}||\boldsymbol{q}'). \tag{6.19}$$

Now the objective function is a finite-sum of two functions: $f_1(\cdot)$ and $f_2(\cdot)$, then we can run SMD on it. Suppose that the SMD is initialized at $X_0 = \exp(-C/\eta)$, choose the step size $\eta = 1$ and mirror map $w(X) = \sum_{i,j} X_{i,j}(\log X_{i,j} - 1)$. When the first sample is used (i.e. the sub-gradient of f_1 is used), SMD updates as

$$\nabla w(X^{k+1}) = \nabla w(X^k) - \nabla f_1(X^k).$$

One can check that it is exactly equivalent to one step Sinkhorn update in $\boldsymbol{u}$ as step (6.16). Similarly, SMD using f_2 is equivalent to one step Sinkhorn update in v as step (6.17).

From the above, the Sinkhorn is a special case of SMD, which suggests that mirror-based algorithms may be proper for solving the entropic OT. Given the success of the Sinkhorn algorithm, it would be interesting to discover more general stochastic proximal mirror descent algorithms and study their performance for solving OT.

6.5.2 Proof for Theorem 6.1

To prove Theorem 6.1, the following lemmas are established:

Lemma 6.2 (Coupling step 1)**.** *Consider one inner loop of Algorithm 6.1, where the randomness only comes from the choice of i. It satisfies that for $\forall u$:*

$$\begin{aligned}
&\alpha_s\langle\nabla\phi(v_{k+1}), z_k - u\rangle \\
\leq&\frac{\alpha_s}{\tau_{1,s}}\left\{\phi(v_{k+1}) - \mathbb{E}[\phi(y_{k+1})] + \tau_2\phi(\widetilde{v}^s) - \tau_2\phi(v_{k+1}) - \tau_2\langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k-1}\rangle\right\} \\
&+ V_{z_k}(u) - \mathbb{E}[V_{z_{k+1}}(u)].
\end{aligned}$$

Proof. Note that the inner loop of Algorithm 6.1 is the same as Algorithm 5 in [33]. We can take $F(\cdot) = f(\cdot) = \phi(\cdot), \psi(\cdot) = 0$ and $\sigma = 0$ in Lemma E.4. of [33] to get:

$$\begin{aligned}
&\alpha_s\langle\nabla\phi(v_{k+1}), z_k - u\rangle \\
\leq&\frac{\alpha_s}{\tau_{1,s}}\left\{\phi(v_{k+1}) - \mathbb{E}[\phi(y_{k+1})] + \tau_2\phi(\widetilde{v}^s) - \tau_2\phi(v_{k+1}) - \tau_2\langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k-1}\rangle\right\} \\
&+ V_{z_k}(u) - \mathbb{E}[V_{z_{k+1}}(u)].
\end{aligned}$$

□

Lemma 6.3 (Coupling step 2)**.** *Using the Lemma 6.2, we further have*

$$\begin{aligned}
&\alpha_s\langle\nabla\phi(v_{k+1}), v_{k+1} - u\rangle \\
\leq&\alpha_s\phi(v_{k+1}) + \frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}}\phi(y_k) + \frac{\alpha_s}{\tau_{1,s}}\left(\tau_2\phi(\widetilde{v}^s) - \mathbb{E}[\phi(y_{k+1})]\right) \\
&+ V_{z_k}(u) - \mathbb{E}[V_{z_{k+1}}(u)].
\end{aligned}$$

Proof. First compute that

$$\alpha_s\langle\nabla\phi(v_{k+1}), v_{k+1} - u\rangle = \alpha_s\langle\nabla\phi(v_{k+1}), v_{k+1} - z_k\rangle + \alpha_s\langle\nabla\phi(v_{k+1}), z_k - u\rangle$$

$$
\begin{aligned}
=&\frac{\alpha_s\tau_2}{\tau_{1,s}}\langle\nabla\phi(v_{k+1}),\widetilde{v}^s-v_{k+1}\rangle+\frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}}\langle\nabla\phi(v_{k+1}),y_k-v_{k+1}\rangle\\
&+\alpha_s\langle\nabla\phi(v_{k+1}),z_k-u\rangle\\
\leq&\frac{\alpha_s\tau_2}{\tau_{1,s}}\langle\nabla\phi(v_{k+1}),\widetilde{v}^s-v_{k+1}\rangle+\frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}}(\phi(y_k)-\phi(v_{k+1}))\\
&+\alpha_s\langle\nabla\phi(v_{k+1}),z_k-u\rangle,
\end{aligned}
$$

where the second equality by the updating rule $v_{k+1}=\tau_{1,s}z_k+\tau_2\widetilde{v}^s+(1-\tau_{1,s}-\tau_2)y_k$, and the inequality by convexity of ϕ. Next, we apply Lemma 6.2 to get

$$
\begin{aligned}
&\alpha_s\langle\nabla\phi(v_{k+1}),v_{k+1}-u\rangle\\
\leq&\frac{\alpha_s\tau_2}{\tau_{1,s}}\langle\nabla\phi(v_{k+1}),\widetilde{v}^s-v_{k+1}\rangle+\frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}}(\phi(y_k)-\phi(v_{k+1}))\\
&+\frac{\alpha_s}{\tau_{1,s}}\left(\phi(v_{k+1})-\mathbb{E}[\phi(y_{k+1})]+\tau_2\phi(\widetilde{v}^s)-\tau_2\phi(v_{k+1})-\tau_2\langle\nabla\phi(v_{k+1}),\widetilde{v}^s-v_{k+1}\rangle\right)\\
&+V_{z_k}(u)-\mathbb{E}[V_{z_{k+1}}(u)]\\
=&\alpha_s\phi(v_{k+1})+\frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}}\phi(y_k)+\frac{\alpha_s}{\tau_{1,s}}\left(\tau_2\phi(\widetilde{v}^s)-\mathbb{E}[\phi(y_{k+1})]\right)\\
&+V_{z_k}(u)-\mathbb{E}[V_{z_{k+1}}(u)].
\end{aligned}
$$

□

Lemma 6.4 (One outer loop)**.** *Consider the sth epoch, assume that all randomness in the first $s-1$ epochs are fixed, we have*

$$
\begin{aligned}
&\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}\langle\nabla\phi(v_{k+1}),v_{k+1}-u\rangle+\frac{\tau_2}{\tau_{1,s+1}^2}\sum_{k=sl}^{sl+l-2}\mathbb{E}(\phi(y_{k+1})-\phi^*)\\
&+\frac{1-\tau_{1,s+1}}{\tau_{1,s+1}^2}\mathbb{E}(\phi(y_{(s+1)l})-\phi^*)\\
\leq&\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}(\phi(v_{k+1})-\phi^*)+\frac{1-\tau_{1,s}}{\tau_{1,s}^2}\mathbb{E}(\phi(y_{sl})-\phi^*)+\frac{\tau_2}{\tau_{1,s}^2}\sum_{k=sl-l}^{sl-2}(\phi(y_{k+1})-\phi^*)\\
&+9\bar{L}(\mathbb{E}V_{z_{sl}}(u)-\mathbb{E}V_{z_{(s+1)l}}(u)),
\end{aligned}
$$

where $\phi^* = \min \phi(\cdot)$.

Proof. Sum up the inequality in Lemma 6.3 for $k = sl + j, j = 0, \ldots, l-1$, we have:

$$\begin{aligned}
&\alpha_s \sum_{k=sl}^{sl+l-1} \mathbb{E}\langle \nabla\phi(v_{k+1}), v_{k+1} - u\rangle \\
\leq &\sum_{k=sl}^{sl+l-1} \Bigg\{ \alpha_s \mathbb{E}\phi(v_{k+1}) + \frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}} \mathbb{E}\phi(y_k) + \frac{\alpha_s}{\tau_{1,s}} \left(\tau_2 \phi(\widetilde{v}^s) - \mathbb{E}[\phi(y_{k+1})]\right) \\
&\quad + \mathbb{E}V_{z_k}(u) - \mathbb{E}[V_{z_{k+1}}(u)] \Bigg\} \\
= &\sum_{k=sl}^{sl+l-1} \left\{ \alpha_s \mathbb{E}\phi(v_{k+1}) - \frac{\alpha_s(\tau_{1,s}+\tau_2)}{\tau_{1,s}} \mathbb{E}\phi(y_{k+1}) \right\} \\
&+ \frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}} [\mathbb{E}\phi(y_{sl}) - \mathbb{E}\phi(y_{(s+1)l})] + \frac{\alpha_s \tau_2 l}{\tau_{1,s}} \phi(\widetilde{v}^s) + \mathbb{E}V_{z_{sl}}(u) - \mathbb{E}[V_{z_{(s+1)l}}(u)].
\end{aligned} \tag{6.20}$$

By convexity of ϕ, using Jensen's inequality, we have $\frac{1}{l}\sum_{k=sl-l}^{sl-1} \phi(y_{k+1}) \geq \phi(\frac{1}{l}\sum_{k=sl-l}^{sl-1} y_{k+1})$ $= \phi(\widetilde{v}^s)$. Thus

$$\begin{aligned}
&\alpha_s \sum_{k=sl}^{sl+l-1} \mathbb{E}\langle \nabla\phi(v_{k+1}), v_{k+1} - u\rangle + \frac{\alpha_s(\tau_{1,s}+\tau_2)}{\tau_{1,s}} \sum_{k=sl}^{sl+l-1} \mathbb{E}\phi(y_{k+1}) \\
\leq &\alpha_s \sum_{k=sl}^{sl+l-1} \mathbb{E}\phi(v_{k+1}) + \frac{\alpha_s(1-\tau_{1,s}-\tau_2)}{\tau_{1,s}} [\mathbb{E}\phi(y_{sl}) - \mathbb{E}\phi(y_{(s+1)l})] + \frac{\alpha_s\tau_2}{\tau_{1,s}} \sum_{k=sl-l}^{sl-1} \phi(y_{k+1}) \\
&+ \mathbb{E}V_{z_{sl}}(u) - \mathbb{E}V_{z_{(s+1)l}}(u).
\end{aligned}$$

Recall that $\alpha_s = 1/(9\bar{L}\tau_{1,s})$, we have

$$\begin{aligned}
&\frac{1}{\tau_{1,s}} \sum_{k=sl}^{sl+l-1} \mathbb{E}\langle \nabla\phi(v_{k+1}), v_{k+1} - u\rangle + \frac{\tau_{1,s}+\tau_2}{\tau_{1,s}^2} \sum_{k=sl}^{sl+l-1} \mathbb{E}\phi(y_{k+1}) \\
\leq &\frac{1}{\tau_{1,s}} \sum_{k=sl}^{sl+l-1} \mathbb{E}\phi(v_{k+1}) + \frac{1-\tau_{1,s}-\tau_2}{\tau_{1,s}^2} [\mathbb{E}\phi(y_{sl}) - \mathbb{E}\phi(y_{(s+1)l})] + \frac{\tau_2}{\tau_{1,s}^2} \sum_{k=sl-l}^{sl-1} \phi(y_{k+1}) \\
&+ 9\bar{L}(\mathbb{E}V_{z_{sl}}(u) - \mathbb{E}V_{z_{(s+1)l}}(u)).
\end{aligned}$$

Deducting $\phi^* = \min \phi(\cdot)$ from both sides and rearranging terms, we get

$$\begin{aligned}
&\frac{1}{\tau_{1,s}} \sum_{k=sl}^{sl+l-1} \mathbb{E}\langle \nabla \phi(v_{k+1}), v_{k+1} - u \rangle + \frac{\tau_{1,s} + \tau_2}{\tau_{1,s}^2} \sum_{k=sl}^{sl+l-2} \mathbb{E}(\phi(y_{k+1}) - \phi^*) \\
&\quad + \frac{1}{\tau_{1,s}^2} \mathbb{E}(\phi(y_{(s+1)l}) - \phi^*) \\
\leq &\frac{1}{\tau_{1,s}} \sum_{k=sl}^{sl+l-1} \mathbb{E}(\phi(v_{k+1}) - \phi^*) + \frac{1 - \tau_{1,s}}{\tau_{1,s}^2} \mathbb{E}(\phi(y_{sl}) - \phi^*) + \frac{\tau_2}{\tau_{1,s}^2} \sum_{k=sl-l}^{sl-2} (\phi(y_{k+1}) - \phi^*) \\
&\quad + 9\bar{L}(\mathbb{E}V_{z_{sl}}(u) - \mathbb{E}V_{z_{(s+1)l}}(u)).
\end{aligned}$$

By our choice of $\tau_{1,s}$ and τ_2, one can check

$$\frac{1 - \tau_{1,s+1}}{\tau_{1,s+1}^2} \leq \frac{1}{\tau_{1,s}^2}, \quad \frac{\tau_2}{\tau_{1,s+1}^2} \leq \frac{\tau_{1,s} + \tau_2}{\tau_{1,s}^2}.$$

So we further have

$$\begin{aligned}
&\frac{1}{\tau_{1,s}} \sum_{k=sl}^{sl+l-1} \mathbb{E}\langle \nabla \phi(v_{k+1}), v_{k+1} - u \rangle + \frac{\tau_2}{\tau_{1,s+1}^2} \sum_{k=sl}^{sl+l-2} \mathbb{E}(\phi(y_{k+1}) - \phi^*) \\
&\quad + \frac{1 - \tau_{1,s+1}}{\tau_{1,s+1}^2} \mathbb{E}(\phi(y_{(s+1)l}) - \phi^*) \\
\leq &\frac{1}{\tau_{1,s}} \sum_{k=sl}^{sl+l-1} \mathbb{E}(\phi(v_{k+1}) - \phi^*) + \frac{1 - \tau_{1,s}}{\tau_{1,s}^2} \mathbb{E}(\phi(y_{sl}) - \phi^*) + \frac{\tau_2}{\tau_{1,s}^2} \sum_{k=sl-l}^{sl-2} (\phi(y_{k+1}) - \phi^*) \\
&\quad + 9\bar{L}(\mathbb{E}V_{z_{sl}}(u) - \mathbb{E}V_{z_{(s+1)l}}(u)).
\end{aligned}$$

□

Finally, we can prove our main Theorem 6.1 as follows.

Proof. By Lemma 6.4, for $s = 1, \ldots, S-1$, denote $\delta(\cdot) := \phi(\cdot) - \phi^*$ we have:

$$\begin{aligned}
&\frac{1}{\tau_{1,s}} \sum_{k=sl}^{sl+l-1} \mathbb{E}\langle \nabla\phi(v_{k+1}), v_{k+1} - u\rangle + \frac{\tau_2}{\tau_{1,s+1}^2} \sum_{k=sl}^{sl+l-2} \mathbb{E}\delta(y_{k+1}) + \frac{1-\tau_{1,s+1}}{\tau_{1,s+1}^2}\mathbb{E}\delta(y_{(s+1)l}) \\
\leq &\frac{1}{\tau_{1,s}} \sum_{k=sl}^{sl+l-1} \mathbb{E}\delta(v_{k+1}) + \frac{1-\tau_{1,s}}{\tau_{1,s}^2}\mathbb{E}\delta(y_{sl}) + \frac{\tau_2}{\tau_{1,s}^2} \sum_{k=sl-l}^{sl-2} \mathbb{E}\delta(y_{k+1}) \\
&+ 9\bar{L}(\mathbb{E}V_{z_{sl}}(u) - \mathbb{E}V_{z_{(s+1)l}}(u)).
\end{aligned} \tag{6.21}$$

For $s = 0$, apply similar proof as Lemma 6.4 on inequality (6.20), we have:

$$\begin{aligned}
&\frac{1}{\tau_{1,0}} \sum_{k=0}^{l-1} \mathbb{E}\langle \nabla\phi(v_{k+1}), v_{k+1} - u\rangle + \frac{\tau_2}{\tau_{1,1}^2} \sum_{k=0}^{l-2} \mathbb{E}\delta(y_{k+1}) + \frac{1-\tau_{1,1}}{\tau_{1,1}^2}\mathbb{E}\delta(y_l) \\
\leq &\frac{1}{\tau_{1,0}} \sum_{k=0}^{l-1} \mathbb{E}\delta(v_{k+1}) + \frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2}\delta(y_0) + \frac{\tau_2 l}{\tau_{1,0}^2}\delta(\widetilde{v}^0) + 9\bar{L}(V_{z_0}(u) - \mathbb{E}V_{z_l}(u)).
\end{aligned} \tag{6.22}$$

Telescope inequality (6.21) for $s = 1, \ldots, S-1$ and add inequality (6.22), we have following bound:

$$\begin{aligned}
&\sum_{s=0}^{S-1} \frac{1}{\tau_{1,s}} \sum_{k=sl}^{sl+l-1} \mathbb{E}(\langle \nabla\phi(v_{k+1}), v_{k+1} - u\rangle - \delta(v_{k+1})) + \frac{\tau_2}{\tau_{1,S}^2} \sum_{k=(S-1)l}^{Sl-2} \mathbb{E}\delta(y_{k+1}) \\
\leq &\frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2}\delta(y_0) + \frac{\tau_2 l}{\tau_{1,0}^2}\delta(\widetilde{v}_0) + 9\bar{L}(V_{z_0}(u) - \mathbb{E}V_{z_{Sl}}(u)) - \frac{1-\tau_{1,S}}{\tau_{1,S}^2}\mathbb{E}\delta(y_{Sl}).
\end{aligned} \tag{6.23}$$

Now for the term $\langle \nabla\phi(v), v - u\rangle - \phi(v)$, we note that

$$\begin{aligned}
\langle \nabla\phi(v), v - u\rangle - \phi(v) &= \langle \nabla\phi(v), v - u\rangle - (\langle v, b\rangle - f(x(v)) - \langle A^T v, x(v)\rangle) \\
&= \langle b - Ax(v), v - u\rangle - (\langle v, b - Ax(v)\rangle - f(x(v))) \\
&= \langle b - Ax(v), -u\rangle + f(x(v)).
\end{aligned} \tag{6.24}$$

Thus

$$
\begin{aligned}
&\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}(\langle\nabla\phi(v_{k+1}),v_{k+1}-u\rangle-\delta(v_{k+1}))\\
=&\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}\left[\langle b-Ax(v_{k+1}),-u\rangle+f(x(v_{k+1}))-f(x(\lambda^*))\right]\\
=&\left\langle\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}b-A\mathbb{E}\left[\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}x(v_{k+1})\right],-u\right\rangle\\
&+\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}\mathbb{E}f(x(v_{k+1}))-\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}f(x(\lambda^*))\qquad(6.25)\\
\geq&\left\langle\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}b-A\mathbb{E}\left[\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}x(v_{k+1})\right],-u\right\rangle\\
&+\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}f\left(\mathbb{E}\left[\sum_{s=0}^{S-1}\frac{1}{\tau_{1,s}}\sum_{k=sl}^{sl+l-1}x(v_{k+1})\right]/\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\right)-\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}f(x(\lambda^*))\\
=&\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\left[f(\mathbb{E}(x^{S-1}))-f(x(\lambda^*))+\langle b-A\mathbb{E}(x^{S-1}),-u\rangle\right],
\end{aligned}
$$

where the inequality applies Jensen's inequality on convex function f. Plugging inequality (6.25) into inequality (6.23) and using the fact that $0<\frac{\tau_2}{\tau_{1,S}^2}\leq\frac{1-\tau_{1,S}}{\tau_{1,S}^2}$, we have

$$
\begin{aligned}
&\left(\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\right)(f(\mathbb{E}(x^{S-1}))-f(x(\lambda^*)))\\
\leq&\frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2}\delta(y_0)+\frac{\tau_2 l}{\tau_{1,0}^2}\delta(\widetilde{v}_0)+9\bar{L}(V_{z_0}(u)-\mathbb{E}V_{z_{Sl}}(u))-\frac{\tau_2}{\tau_{1,S}^2}\sum_{k=(S-1)l}^{Sl-1}\mathbb{E}\delta(y_{k+1})\\
&+\left(\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\right)(\langle b-A\mathbb{E}(x^{S-1}),u\rangle)\\
\leq&\frac{1-\tau_{1,0}-\tau_2}{\tau_{1,0}^2}\delta(y_0)+\frac{\tau_2 l}{\tau_{1,0}^2}\delta(\widetilde{v}_0)+9\bar{L}V_{z_0}(u)-\frac{\tau_2 l}{\tau_{1,S}^2}\mathbb{E}\delta(\widetilde{v}^S)\\
&+\left(\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\right)(\langle b-A\mathbb{E}(x^{S-1}),u\rangle),
\end{aligned}
\qquad(6.26)
$$

where the second inequality comes from the definition of $\widetilde{v}^S$ and Jensen's inequality. Recall

that inequality (6.26) holds for any u, including the one that minimizes the R.H.S.. We can further upper bound $\min_u R.H.S.$ by restricting $u \in B_H(2R) := \{u : \|u\|_H \leq 2R\}$:

$$\begin{aligned}
&\min_{u\in B_H(2R)} 9\bar{L}V_0(u) + \sum_{s=0}^{S-1} \frac{l}{\tau_{1,s}} \langle b - A\mathbb{E}(x^{S-1}), u\rangle \\
\leq &\min_{u\in B_H(2R)} 9\bar{L}\gamma\|u\|_H^2/2 + \sum_{s=0}^{S-1} \frac{l}{\tau_{1,s}} \langle b - A\mathbb{E}(x^{S-1}), u\rangle \\
\leq &\min_{u\in B_H(2R)} \langle \sum_{s=0}^{S-1} \frac{l}{\tau_{1,s}} (b - A\mathbb{E}(x^{S-1})), u\rangle + 18\bar{L}R^2\gamma \\
= &-2R \left(\sum_{s=0}^{S-1} \frac{l}{\tau_{1,s}}\right) \|b - A\mathbb{E}(x^{S-1})\|_{H,*} + 18\bar{L}R^2\gamma.
\end{aligned} \tag{6.27}$$

Plugging the bound (6.27) into inequality (6.26), we have

$$\begin{aligned}
&\left(\sum_{s=0}^{S-1} \frac{l}{\tau_{1,s}}\right) (f(\mathbb{E}(x^{S-1})) - f(x(\lambda^*))) + \frac{\tau_2 l}{\tau_{1,S}^2} \mathbb{E}\delta(\widetilde{v}^S) \\
&\quad + 2R \left(\sum_{s=0}^{S-1} \frac{l}{\tau_{1,s}}\right) \|b - A\mathbb{E}(x^{S-1})\|_{H,*} \\
\leq &\frac{1 - \tau_{1,0} - \tau_2}{\tau_{1,0}^2} \delta(y_0) + \frac{\tau_2 l}{\tau_{1,0}^2} \delta(\widetilde{v}_0) + 18\bar{L}R^2\gamma \\
= &2l\delta(0) + 18\bar{L}R^2\gamma.
\end{aligned} \tag{6.28}$$

Calculate $\sum_{s=0}^{S-1} \frac{1}{\tau_{1,s}} = \sum_{s=0}^{S-1} (s+4)/2 = (2S+3)S/4 \geq S^2/2$, then

$$f(\mathbb{E}(x^{S-1})) - f(x^*) \leq \frac{4}{S^2 l} \left[l\delta(0) + 9\bar{L}R^2\gamma\right]. \tag{6.29}$$

On the other hand, notice that $f(x(\lambda^*)) = -\phi(\lambda^*) := \phi^*$ and

$$\begin{aligned}
& f(\mathbb{E}(x^{S-1})) - f(x(\lambda^*)) \\
=& f(\mathbb{E}(x^{S-1})) + \phi^* \\
=& f(\mathbb{E}(x^{S-1})) + \langle \lambda^*, b \rangle + \max_x(-f(x) - \langle A^T\lambda^*, x\rangle) \\
\geq& f(\mathbb{E}(x^{S-1})) + \langle \lambda^*, b \rangle - f(\mathbb{E}(x^{S-1})) - \langle A^T\lambda^*, \mathbb{E}(x^{S-1})\rangle \\
=& \langle \lambda^*, b - A\mathbb{E}(x^{S-1})\rangle \geq -R\|\mathbb{E}[b - Ax^{S-1}]\|_{H,*}.
\end{aligned} \tag{6.30}$$

Plugging inequality (6.30) into inequality (6.28), we have

$$R\left(\sum_{s=0}^{S-1}\frac{l}{\tau_{1,s}}\right)\|\mathbb{E}[b - Ax^{S-1}]\|_{H,*} \leq 2l\delta(0) + 18\bar{L}R^2\gamma. \tag{6.31}$$

Thus

$$\|\mathbb{E}[b - Ax^{S-1}]\|_{H,*} \leq \frac{4\left[l\delta(0) + 9\bar{L}R^2\gamma\right]}{S^2lR}. \tag{6.32}$$

Further check that

$$\delta(0) = \phi(0) - \phi^* \leq \langle \nabla\phi(\lambda^*), 0 - \lambda^*\rangle + \frac{\bar{L}}{2}\|0 - \lambda^*\|_H^2 = \frac{\bar{L}}{2}\|\lambda^*\|_H^2 \leq \frac{\bar{L}}{2}R^2. \tag{6.33}$$

Plugging the bound (6.33) into inequalities (6.29) and (6.32), we get the theorem claim. □

6.5.3 Proof for Lemma 6.1

Proof. By Proposition 2 of [97], $\phi_i(\cdot)$ is $\frac{np_i'}{\eta}$ smooth w.r.t. $\|\cdot\|_2$. So here we only show the second part of the statement. That is, prove the smoothness w.r.t. $\|\cdot\|_\infty$.

By

$$\phi_i(\lambda) = np_i'\left(-\langle q', \lambda\rangle - \eta\log p_i' + \eta\log\left(\sum_{j=1}^{n}\exp((\lambda_j - c_{i,j} - \eta)/\eta)\right) + \eta\right),$$

we calculate that

$$\nabla\phi_i(\lambda) = np_i'\left(-q' + \frac{(\exp((\lambda_k - c_{i,k})/\eta))_{k=1,\dots,n}}{\left(\sum_{j=1}^n \exp((\lambda_j - c_{i,j})/\eta)\right)}\right).$$

The goal is $\forall\lambda, \lambda'$, bound the $\|\cdot\|_{\infty,*} = \|\cdot\|_1$ of following difference in the gradient:

$$\nabla\phi_i(\lambda) - \nabla\phi_i(\lambda') = np_i'\left(\frac{(\exp((\lambda_k - c_{i,k})/\eta))_{k=1,\dots,n}}{\left(\sum_{j=1}^n \exp((\lambda_j - c_{i,j})/\eta)\right)} - \frac{(\exp((\lambda_k' - c_{i,k})/\eta))_{k=1,\dots,n}}{\left(\sum_{j=1}^n \exp((\lambda_j' - c_{i,j})/\eta)\right)}\right).$$

Further denote $\Delta\lambda = \lambda' - \lambda$, then

$$\begin{aligned}
&\|\nabla\phi_i(\lambda) - \nabla\phi_i(\lambda')\|_1 \\
=&np_i'\left\|\frac{(\exp((\lambda_k - c_{i,k})/\eta))_{k=1,\dots,n}}{\left(\sum_{j=1}^n \exp((\lambda_j - c_{i,j})/\eta)\right)} - \frac{(\exp((\lambda_k + (\Delta\lambda)_k - c_{i,k})/\eta))_{k=1,\dots,n}}{\left(\sum_{j=1}^n \exp((\lambda_j + (\Delta\lambda)_j - c_{i,j})/\eta)\right)}\right\|_1 \\
=&np_i'\left\|\frac{(\exp((\lambda_k - c_{i,k})/\eta))_{k=1,\dots,n}}{\left(\sum_{j=1}^n \exp((\lambda_j - c_{i,j})/\eta)\right)} - \frac{(\exp((\lambda_k - c_{i,k})/\eta) * \exp((\Delta\lambda/\eta)_k))_{k=1,\dots,n}}{\left(\sum_{j=1}^n \exp((\lambda_j - c_{i,j})/\eta)\exp((\Delta\lambda/\eta)_j)\right)}\right\|_1.
\end{aligned}$$

Taking $\boldsymbol{a} = (\exp((\lambda_k - c_{i,k})/\eta))_{k=1,\dots,n}$ and $\boldsymbol{b} = \Delta\lambda/\eta$ in Lemma 6.5, we immediately have

$$\|\nabla\phi_i(\lambda) - \nabla\phi_i(\lambda')\|_1 \le np_i' 5\|\Delta\lambda/\eta\|_\infty = \frac{5np_i'}{\eta}\|\lambda - \lambda'\|_\infty. \tag{6.34}$$

Thus, $\phi_i(\cdot)$ is $\frac{5np_i'}{\eta}$ smooth w.r.t. $\|\cdot\|_\infty$ norm. □

Lemma 6.5. *Consider two vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathcal{R}^d$, and let $\exp(\boldsymbol{b})$ be the element-wise exponential of $\boldsymbol{b}$. When $\boldsymbol{a} > \boldsymbol{0}$, we have*

$$\left\|\frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_1} - \frac{\boldsymbol{a}\circ\exp(\boldsymbol{b})}{\|\boldsymbol{a}\circ\exp(\boldsymbol{b})\|_1}\right\|_1 \le 5\|\boldsymbol{b}\|_\infty.$$

Proof. Consider two cases:

First, when $\|\boldsymbol{b}\|_\infty > 0.5$:

$$\left\| \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_1} - \frac{\boldsymbol{a} \circ \exp(\boldsymbol{b})}{\|\boldsymbol{a} \circ \exp(\boldsymbol{b})\|_1} \right\|_1 \leq \left\| \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_1} \right\|_1 + \left\| \frac{\boldsymbol{a} \circ \exp(\boldsymbol{b})}{\|\boldsymbol{a} \circ \exp(\boldsymbol{b})\|_1} \right\|_1 = 2 < 5 * 0.5 < 5\|\boldsymbol{b}\|_\infty.$$

Second, when $\|\boldsymbol{b}\|_\infty \leq 0.5$:

$$\begin{aligned}
&\left\| \frac{\boldsymbol{a}}{\|\boldsymbol{a}\|_1} - \frac{\boldsymbol{a} \circ \exp(\boldsymbol{b})}{\|\boldsymbol{a} \circ \exp(\boldsymbol{b})\|_1} \right\|_1 \\
=&\sum_{i=1}^{d} \frac{|\sum_{j=1}^{d} a_i a_j \exp(b_j) - \sum_{j=1}^{d} a_i \exp(b_i) a_j|}{\|\boldsymbol{a}\|_1 \|\boldsymbol{a} \circ \exp(\boldsymbol{b})\|_1} \\
\leq&\sum_{i=1}^{d} \frac{\sum_{j=1}^{d} a_i a_j |\exp(b_j) - \exp(b_i)|}{\|\boldsymbol{a}\|_1 \|\boldsymbol{a} \circ \exp(\boldsymbol{b})\|_1} \\
\leq&\sum_{i=1}^{d} \frac{\sum_{j=1}^{d} a_i a_j (|\exp(b_j) - 1| + |\exp(b_i) - 1|)}{\|\boldsymbol{a}\|_1 \|\boldsymbol{a}\|_1 \exp(-0.5)} \\
\leq&2\exp(0.5)(\exp(0.5) - 1) \sum_{i=1}^{d} \frac{\sum_{j=1}^{d} a_i a_j (|b_j| + |b_i|)}{\|\boldsymbol{a}\|_1 \|\boldsymbol{a}\|_1} \\
\leq&4\exp(0.5)(\exp(0.5) - 1) \sum_{i=1}^{d} \frac{\sum_{j=1}^{d} a_i a_j \|\boldsymbol{b}\|_\infty}{\|\boldsymbol{a}\|_1 \|\boldsymbol{a}\|_1} \\
=&4\exp(0.5)(\exp(0.5) - 1)\|\boldsymbol{b}\|_\infty < 5\|\boldsymbol{b}\|_\infty.
\end{aligned}$$

Combine two cases we have the lemma holds. □

6.5.4 Proof for Theorem 6.2

The full procedure for finding an ϵ-solution to OT using PDASMD is given in Algorithm 6.3.

The total computational cost of Algorithm 6.3 is given in Theorem 6.2, and we prove it as follows:

Proof. We have the convergence result in Theorem 6.1 holds for the dual formulation. To extend the proof of Theorem 6.1 to the semi-dual formulation for the OT problem, we just

Algorithm 6.3: Approximating OT by PDASMD

Input: Accuracy $\epsilon > 0$, $\eta = \frac{\epsilon}{4\log(n)}$ and $\epsilon' = \frac{\epsilon}{8\|C\|_\infty}$.

Step 1: Let $\boldsymbol{p}' \in \Delta_n$ and $\boldsymbol{q}' \in \Delta_n$ be

$$\begin{pmatrix} \boldsymbol{p}' \\ \boldsymbol{q}' \end{pmatrix} = \left(1 - \frac{\epsilon'}{8}\right) \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{q} \end{pmatrix} + \frac{\epsilon'}{8n} \begin{pmatrix} \mathbf{1}_n \\ \mathbf{1}_n \end{pmatrix}.$$

Step 2: Compute $\widetilde{\boldsymbol{X}}$ by PDASMD on objective (6.2) long enough such that $f(\mathbb{E}\widetilde{\boldsymbol{x}}) - f(\boldsymbol{x}^*) \leq \frac{\epsilon}{4}$ and $\|A\mathbb{E}\widetilde{\boldsymbol{x}} - b\|_1 \leq \frac{\epsilon'}{2}$.

Step 3: Round $\widetilde{X}$ to $\widehat{X}$ by Algorithm 2 in [84] such that $\widehat{X}\mathbf{1}_n = \boldsymbol{p}$, $\widehat{X}^T\mathbf{1}_n = \boldsymbol{q}$.

Output: $\widehat{X}$

need the following equality to hold:

$$[Ax(v) - b] = \begin{bmatrix} \mathbf{0}_n \\ \nabla\phi(v) \end{bmatrix}.$$

One can easily check it is true for the semi-dual of OT. Moreover, chapter [97] shows that the stopping criteria in step 2 of Algorithm 6.3 guarantees

$$\mathbb{E}\langle C, \widehat{X}\rangle \leq \langle C, X^*\rangle + \epsilon.$$

That is, the output of Algorithm 6.3 is an $\epsilon-$solution. We now focus on the computational complexity of Algorithm 6.3.

Case 1: $\|\cdot\|_H = \|\cdot\|_2$. By Theorem 6.1 we have

$$\|\mathbb{E}[b - A\widetilde{x}]\|_2 \leq \frac{2\left[l\bar{L}R + 18\bar{L}R\gamma\right]}{S^2 l}, \tag{6.35}$$

$$f(\mathbb{E}(\widetilde{x})) - f(x^*) \leq \frac{2}{S^2 l}\left[l\bar{L}R^2 + 18\bar{L}R^2\gamma\right], \tag{6.36}$$

where $\bar{L} = \frac{1}{n}\sum_{i=1}^n \frac{np_i'}{\eta} = \frac{1}{\eta}$, and R is an upper bound for $\|\lambda^*\|_2$. By Lemma 3.2 in [87], $R = \eta\sqrt{n}(R' + .5)$ for $R' = \|C\|_\infty/\eta + \log(n) - 2\log(\min_{1\leq i,j\leq n}\{p_i', q_j'\}) \leq 4\|C\|_\infty\log(n)/\epsilon + \log(n) - 2\log(\epsilon) + 2\log(64n\|C\|_\infty) = \mathcal{O}(\|C\|_\infty\log(n)/\epsilon)$.

Using the bound $\|A\mathbb{E}\widetilde{x} - b\|_1 \leq \sqrt{2n}\|A\mathbb{E}\widetilde{x} - b\|_2$ we have the stopping criteria in step 2 satisfied for

$$\begin{aligned}
S =& \max\left\{\mathcal{O}\left(\sqrt{\frac{\bar{L}R\sqrt{n}}{\epsilon'}}\right), \mathcal{O}\left(\sqrt{\frac{\bar{L}\gamma R\sqrt{n}}{l\epsilon'}}\right), \mathcal{O}\left(\sqrt{\frac{\bar{L}R^2}{\epsilon}}\right), \mathcal{O}\left(\sqrt{\frac{\bar{L}R^2\gamma}{l\epsilon}}\right)\right\} \\
=& \max\left\{\mathcal{O}\left(\sqrt{\frac{n\log(n)\|C\|_\infty^2}{\epsilon^2}}\right), \mathcal{O}\left(\sqrt{\frac{\log(n)\gamma n\|C\|_\infty^2}{l\epsilon^2}}\right),\right. \\
& \qquad \left.\mathcal{O}\left(\frac{n^{.5}\|C\|_\infty\sqrt{\log n}}{\epsilon}\right), \mathcal{O}\left(\sqrt{\frac{n\log(n)\|C\|_\infty^2\gamma}{l\epsilon^2}}\right)\right\} \\
=& \mathcal{O}\left(\frac{n^{.5}}{\epsilon}\max\left(\|C\|_\infty\sqrt{\log n}, \sqrt{\log(n)\|C\|_\infty^2\gamma/l}\right)\right) \\
=& \mathcal{O}\left(\frac{n^{.5}\|C\|_\infty}{\epsilon}\left(\sqrt{\log n} + \sqrt{\log(n)\gamma/l}\right)\right).
\end{aligned}$$

In Algorithm 6.3, step 1 and step 3 has total number of $\mathcal{O}(n^2)$ operations, and the algorithm complexity is dominated by step 2. Now each outer loop of PDASMD has $\mathcal{O}(n^2 + nl)$ operations, thus the total number of arithmetic operations of Algorithm 6.3 is

$$\mathcal{O}\left(\frac{n^{1.5}\|C\|_\infty}{\epsilon}(n+l)\left(\sqrt{\log n} + \sqrt{\log(n)\gamma/l}\right)\right) = \widetilde{\mathcal{O}}\left(\frac{n^{2.5}\|C\|_\infty(1+\sqrt{\gamma/n})}{\epsilon}\right).$$

Case 2: $\|\cdot\|_H = \|\cdot\|_\infty$. Then $\|\cdot\|_{H,*} = \|\cdot\|_1$, and Theorem 6.1 implies that

$$\|\mathbb{E}[b - A\widetilde{x}]\|_1 \leq \frac{2\left[l\bar{L}R + 18\bar{L}R\gamma\right]}{S^2 l}, \tag{6.37}$$

$$f(\mathbb{E}(\widetilde{x})) - f(x^*) \leq \frac{2}{S^2 l}\left[l\bar{L}R^2 + 18\bar{L}R^2\gamma\right], \tag{6.38}$$

where $\bar{L} = \frac{5}{\eta}$. Now R is an upper bound for $\|\lambda^*\|_\infty$, and again by Lemma 3.2 in [87], $R = \eta(R' + .5)$ for $R' = \mathcal{O}(\|C\|_\infty \log(n)/\epsilon)$.

Thus the stopping criteria in step 2 of Algorithm 6.3 is satisfied for

$$\begin{aligned}
S =& \max\left\{\mathcal{O}\left(\sqrt{\frac{\bar{L}R}{\epsilon'}}\right), \mathcal{O}\left(\sqrt{\frac{\bar{L}R\gamma}{l\epsilon'}}\right), \mathcal{O}\left(\sqrt{\frac{\bar{L}R^2}{\epsilon}}\right), \mathcal{O}\left(\sqrt{\frac{\bar{L}R^2\gamma}{l\epsilon}}\right)\right\} \\
=& \max\left\{\mathcal{O}\left(\sqrt{\frac{\log(n)\|C\|_\infty^2}{\epsilon^2}}\right), \mathcal{O}\left(\sqrt{\frac{\log(n)\gamma\|C\|_\infty^2}{l\epsilon^2}}\right), \right. \\
& \qquad \left. \mathcal{O}\left(\sqrt{\frac{\|C\|_\infty^2\log n}{\epsilon^2}}\right), \mathcal{O}\left(\sqrt{\frac{\log(n)\|C\|_\infty^2\gamma}{l\epsilon^2}}\right)\right\} \\
=& \mathcal{O}\left(\frac{\|C\|_\infty}{\epsilon}\max\left(\sqrt{\log n}, \sqrt{\log(n)\gamma/n}\right)\right) \\
=& \mathcal{O}\left(\frac{\|C\|_\infty}{\epsilon}\left(\sqrt{\log n} + \sqrt{\log(n)\gamma/n}\right)\right).
\end{aligned}$$

Now each outer loop of PDASMD has $\mathcal{O}(n^2 + nl) = \mathcal{O}(n^2)$ operations, thus the total number of arithmetic operations of Algorithm 6.3 is

$$\mathcal{O}\left(\frac{n^2\|C\|_\infty}{\epsilon}\left(\sqrt{\log n} + \sqrt{\log(n)\gamma/n}\right)\right) = \widetilde{\mathcal{O}}\left(\frac{n^2\|C\|_\infty(1+\sqrt{\gamma/n})}{\epsilon}\right).$$

□

6.5.5 Stochastic Sinkhorn Algorithm and Proof of Computational Complexity

We first describe the Stochastic Sinkhorn algorithm. In the Stochastic Sinkhorn algorithm, the following definitions are used:

Definition 6.4 (Increasing probability function)**.** *An increasing probability function* $\Psi : \mathcal{R}_+^p \to \Delta_p$ *is such that*

$$\Psi(\boldsymbol{h}) = \left(\frac{g(h_i)}{\sum_i g(h_i)}\right)_i,$$

where $g : \mathcal{R}_+ \to \mathcal{R}_+$ *is an increasing positive function.*

Definition 6.5 (KL violation)**.** *For a matrix* $M \in \mathcal{R}_+^{p\times p}$ *and two vectors* $\boldsymbol{p}, \boldsymbol{q} \in \Delta_p$*, define*

the KL violation

$$\rho(M;\boldsymbol{p},\boldsymbol{q}) = \begin{bmatrix} (\mathcal{KL}(p_i\|(M\mathbf{1})_i))_{i=1,\ldots,p} \\ (\mathcal{KL}(q_j\|(M^T\mathbf{1})_j))_{j=1,\ldots,p} \end{bmatrix}. \tag{6.39}$$

The Stochastic Sinkhorn algorithm for solving problem (6.2) is in Algorithm 6.4.

Algorithm 6.4: Stochastic Sinkhorn

Input: $C, \boldsymbol{p}', \boldsymbol{q}', \Psi, \eta$
Calculate $A = \exp(-C/\eta)$ where all the operations are element-wise;
Initialize: $\boldsymbol{u}^0 = \boldsymbol{v}^0 = \mathbf{1}$;
for k=0,...,K-1 **do**
Calculate $\boldsymbol{h} = \Psi(\rho(X(\boldsymbol{u}^k, \boldsymbol{v}^k); \boldsymbol{p}', \boldsymbol{q}'))$, where $X(\boldsymbol{u}^k, \boldsymbol{v}^k) = \text{diag}(\boldsymbol{u}^k)A\text{diag}(\boldsymbol{v}^k)$;
Sample index I with

$$P(I = i) = h_i, \forall i \in \{1, 2, ..., 2n\}.$$

if $I \leq n$ **then**
$\boldsymbol{u}^{k+1} = (u_1^k, \ldots, u_{I-1}^k, p_I'/(A\boldsymbol{v}^k)_I, u_{I+1}^k, \ldots, u_n^k)^T$, $\boldsymbol{v}^{k+1} = \boldsymbol{v}^k$;
else
$\boldsymbol{u}^{k+1} = \boldsymbol{u}^k$, $\boldsymbol{v}^{k+1} = (v_1^k, \ldots, v_{I-n-1}^k, q_{I-n}'/(A^T\boldsymbol{u}^k)_{I-n}, v_{I-n+1}^k, \ldots, v_n^k)^T$.
end if
end for
Output: $\widetilde{X} = \text{diag}(\boldsymbol{u}^K)A\text{diag}(\boldsymbol{v}^K)$.

To find a ϵ-solution to OT, an extra rounding step is required. The full procedure is given in Algorithm 6.5.

Algorithm 6.5: Approximating OT by Stochastic Sinkhorn

Input: Accuracy $\epsilon > 0$, $\eta = \frac{\epsilon}{4\log(n)}$ and $\epsilon' = \frac{\epsilon}{8\|C\|_\infty}$.
Step 1: Let $\boldsymbol{p}' \in \Delta_n$ and $\boldsymbol{q}' \in \Delta_n$ be

$$\begin{pmatrix} \boldsymbol{p}' \\ \boldsymbol{q}' \end{pmatrix} = \left(1 - \frac{\epsilon'}{8}\right)\begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{q} \end{pmatrix} + \frac{\epsilon'}{8n}\begin{pmatrix} \mathbf{1}_n \\ \mathbf{1}_n \end{pmatrix}.$$

Step 2: Compute $\widetilde{X}$ by Stochastic Sinkhorn until $\|\widetilde{X}\mathbf{1} - \boldsymbol{p}'\|_1 + \|\widetilde{X}^T\mathbf{1} - \boldsymbol{q}'\|_1 \leq \frac{\epsilon'}{2}$.
Step 3: Round $\widetilde{X}$ to $\widehat{X}$ by Algorithm 2 in [84] such that $\widehat{X}\mathbf{1}_n = \boldsymbol{p}$, $\widehat{X}^T\mathbf{1}_n = \boldsymbol{q}$.
Output: $\widehat{X}$.

We now prove the computational complexity of Stochastic Sinkhorn in Theorem 6.3.

To prove it, we first need the convergence of Algorithm 6.4, which we show in following Lemma.

Lemma 6.6. *For a given $\epsilon > 0$, we have that Algorithm 6.4 returns a matrix $\widetilde{X}$ such that*

$$\mathbb{E}[\|\widetilde{X}\mathbf{1} - p'\|_1 + \|\widetilde{X}^T\mathbf{1} - q'\|_1] \leq \epsilon$$

in the number of iterations

$$k \leq 2 + 112nR/\epsilon.$$

Proof. Denote $(x^k, y^k) := (\log u^k, \log v^k)$ and the dual function $f(x, y) = \sum_{i,j} A_{i,j} \exp(x_i + y_j) - \langle p', x\rangle - \langle q', y\rangle$. Denote $E_k = \mathbb{E}[\|X(u^k, v^k)\mathbf{1} - p'\|_1 + \|X(u^k, v^k)^T\mathbf{1} - q'\|_1]$. By (21) in [88], Algorithm 6.4 has

$$E[f(x^k, y^k) - f(x^{k+1}, y^{k+1})] > \frac{E_k^2}{28n}. \tag{6.40}$$

Since Algorithm 6.4 only updates one element in u or v, and the updating rule for that element is the same as Greenkhorn, we have that Corollary 3.3 in [87] holds. Adding expectations to both sides, we get:

$$E[f(x^k, y^k) - f(x^*, y^*)] \leq 4RE_k. \tag{6.41}$$

Let $\delta_k = E[f(x^k, y^k) - f(x^*, y^*)]$, then by inequalities (6.40) and (6.41) we have

$$\delta_k - \delta_{k+1} \overset{(6.40)}{\geq} \frac{E_k^2}{28n} \overset{(6.41)}{\geq} \frac{\delta_k^2}{448nR^2}. \tag{6.42}$$

That is,

$$\delta_k - \delta_{k+1} \geq \max\left\{\frac{\epsilon^2}{28n}, \frac{\delta_k^2}{448nR^2}\right\}. \tag{6.43}$$

We adopt the strategy in [86] to split the process of $\{\delta_k\}$ into two halves:

First, consider the process from δ_1 to δ_t:

$$\frac{\delta_t}{448nR^2} \leq \frac{\delta_{t-1}}{448nR^2} - \left(\frac{\delta_{t-1}}{448nR^2}\right)^2 \leq \frac{1}{t-1+448nR^2/\delta_1} \Rightarrow t \leq 1 + \frac{448nR^2}{\delta_t} - \frac{448nR^2}{\delta_1}.$$

Second, consider the process from δ_t to δ_{t+m}:

$$\delta_{t+m} \leq \delta_t - \frac{\epsilon^2 m}{28n} \Rightarrow m \leq \frac{28n(\delta_t - \delta_{t+m})}{\epsilon^2}.$$

So the total number of iterations $k = t + m$ can be optimized over δ_t, i.e.

$$k \leq \min_{\delta_t \in (0,\delta_1]} \left(2 + \frac{448nR^2}{\delta_t} - \frac{448nR^2}{\delta_1} + \frac{28n\delta_t}{\epsilon^2}\right) \leq 2 + \frac{112nR}{\epsilon}.$$

□

Then we can prove Theorem 6.3 as follows:

Proof. By Lemma 6.6, we have $\mathbb{E}[\|\widetilde{X}\mathbf{1} - p'\|_1 + \|\widetilde{X}^T\mathbf{1} - q'\|_1] \leq \epsilon'/2$ for the number of iterations $k = 2 + 224nR/\epsilon'$. Thus for this k, we also have

$$\mathbb{E}[\|\widetilde{X}\mathbf{1} - p\|_1 + \|\widetilde{X}^T\mathbf{1} - q\|_1] \leq \mathbb{E}[\|\widetilde{X}\mathbf{1} - p'\|_1 + \|\widetilde{X}^T\mathbf{1} - q'\|_1] + \|p - p'\|_1 + \|q - q'\|_2 \leq \epsilon'.$$

By Theorem 1 in [84],

$$\mathbb{E}\langle C, \widehat{X}\rangle - \langle C, X^*\rangle \leq \epsilon/2 + 4(\mathbb{E}[\|\widetilde{X}\mathbf{1} - p\|_1 + \|\widetilde{X}^T\mathbf{1} - q\|_1])\|C\|_\infty \leq \epsilon,$$

which is an $\epsilon-$solution.

Now calculate the number of arithmetic operations, step 1 requires $\mathcal{O}(n)$; step 2 requires k iterations of Algorithm 6.4, each iteration requires $\mathcal{O}(n)$ operation [88]; step 3

requires $\mathcal{O}(n^2)$ operations [84]. Thus the total number of operations is

$$\mathcal{O}(n^2 R/\epsilon') = \mathcal{O}(n^2 \|C\|_\infty^2 \log n/\epsilon^2).$$

□

6.5.6 PDASMD-B Algorithm and the Convergence Rate

In this Section, we prove the convergence of PDASMD-B. The convergence rate of PDASMD-B is in the following theorem:

Theorem 6.4 (Convergence of PDASMD-B)**.** *In Algorithm 6.2, assume that the dual optimal solution has $\|\lambda^*\|_H \leq R$. Then we have the convergence of Algorithm 6.2 as:*

$$\|\mathbb{E}[\boldsymbol{b} - A\boldsymbol{x}^{S-1}]\|_{H,*} \leq \frac{2\left[(1+(l-1)/B)\bar{L}R + 18\bar{L}R\gamma\right]}{S^2 l}, \tag{6.44}$$

$$f(\mathbb{E}(\boldsymbol{x}^{S-1})) - f(\boldsymbol{x}^*) \leq \frac{2}{S^2 l}\left[(1+(l-1)/B)\bar{L}R^2 + 18\bar{L}R^2\gamma\right]. \tag{6.45}$$

The key to prove Theorem 6.4 is to find an analogue to Lemma 6.2, which we do in following steps.

Lemma 6.7 (Variance upper bound)**.**

$$\mathbb{E}[\|\widetilde{\nabla}_{k+1} - \nabla\phi(v_{k+1})\|_{H,*}^2] \leq \frac{8\bar{L}}{B}(\phi(\widetilde{v}^s) - \phi(v_{k+1}) - \langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k+1}\rangle). \tag{6.46}$$

Proof. Each ϕ_i is convex and L_i-smooth, then by Theorem 2.1.5. in [102] we have

$$\|\nabla\phi_i(v_{k+1}) - \nabla\phi_i(\widetilde{v}^s)\|_{H,*}^2 \leq 2L_i(\phi_i(\widetilde{v}^s) - \phi_i(v_{k+1}) - \langle\nabla\phi_i(v_{k+1}), \widetilde{v}^s - v_{k+1}\rangle). \tag{6.47}$$

Take expectation with respect to the randomness of index set I, note that all indexes in I

are independently selected, we have

$$
\begin{aligned}
&\mathbb{E}[\|\widetilde{\nabla}_{k+1}-\nabla\phi(v_{k+1})\|_{H,*}^2]\\
=&\frac{1}{B}\mathbb{E}\left[\left\|\nabla\phi(\widetilde{v}^s)+\frac{1}{mp_i}(\nabla\phi_i(v_{k+1})-\nabla\phi_i(\widetilde{v}^s))-\nabla\phi(v_{k+1})\right\|_{H,*}^2\right]\\
\leq&\frac{1}{B}\mathbb{E}\left[2\left\|\frac{1}{mp_i}(\nabla\phi_i(\widetilde{v}^s)-\nabla\phi_i(v_{k+1}))\right\|_{H,*}^2+2\|\nabla\phi(\widetilde{v}^s)-\nabla\phi(v_{k+1})\|_{H,*}^2\right]\\
\overset{(6.47)}{\leq}&\frac{1}{B}\mathbb{E}\left[4\frac{L_i}{m^2p_i^2}(\phi_i(\widetilde{v}^s)-\phi_i(v_{k+1})-\langle\nabla\phi_i(v_{k+1}),\widetilde{v}^s-v_{k+1}\rangle)+2\|\nabla\phi(\widetilde{v}^s)-\nabla\phi(v_{k+1})\|_{H,*}^2\right]\\
=&\frac{1}{B}[4\bar{L}(\phi(\widetilde{v}^s)-\phi(v_{k+1})-\langle\nabla\phi(v_{k+1}),\widetilde{v}^s-v_{k+1}\rangle)+2\|\nabla\phi(\widetilde{v}^s)-\nabla\phi(v_{k+1})\|_{H,*}^2]\\
\overset{(6.47)}{\leq}&\frac{8\bar{L}}{B}(\phi(\widetilde{v}^s)-\phi(v_{k+1})-\langle\nabla\phi(v_{k+1}),\widetilde{v}^s-v_{k+1}\rangle)
\end{aligned}
$$

□

Lemma 6.8 (Coupling step 1, batch version)**.** *Consider one inner loop of Algorithm 6.2, where the randomness only comes from the choice of I. It satisfies that for $\forall u$:*

$$
\begin{aligned}
&\alpha_s\langle\nabla\phi(v_{k+1}),z_k-u\rangle\\
\leq&\frac{\alpha_s}{\tau_{1,s}}\{\phi(v_{k+1})-\mathbb{E}[\phi(y_{k+1})]+\tau_2\phi(\widetilde{v}^s)-\tau_2\phi(v_{k+1})-\tau_2\langle\nabla\phi(v_{k+1}),\widetilde{v}^s-v_{k-1}\rangle\}\\
&+V_{z_k}(u)-\mathbb{E}[V_{z_{k+1}}(u)].
\end{aligned}
$$

Proof. One can easily check that the Lemma E.1. and Lemma E.3. in [33] holds for the batch version of PDASMD, where $\psi(\cdot)=0$ in these two lemmas for our case. Then we have

$$
\begin{aligned}
\phi(v_{k+1})-\mathbb{E}[\phi(y_{k+1})]\geq&\mathbb{E}\left[-\min_y\left\{\frac{9\bar{L}}{2}\|y-v_{k+1}\|_H^2+\langle\widetilde{\nabla}_{k+1},y-v_{k+1}\rangle\right\}\right]\\
&-\frac{1}{16\bar{L}}\mathbb{E}[\|\widetilde{\nabla}_{k+1}-\nabla\phi(v_{k+1})\|_{H,*}^2].
\end{aligned}
\tag{6.48}
$$

$$
\alpha_s\langle\widetilde{\nabla}_{k+1},z_{k+1}-u\rangle\leq-\frac{1}{2}\|z_k-z_{k+1}\|_H^2+V_{z_k}(u)-V_{z_{k+1}}(u).
\tag{6.49}
$$

Then

$$\begin{aligned}
&\alpha_s\langle\widetilde{\nabla}_{k+1}, z_k - u\rangle = \alpha_s\langle\widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle + \alpha_s\langle\widetilde{\nabla}_{k+1}, z_{k+1} - u\rangle \\
\overset{(6.49)}{\leq}\ &\alpha_s\langle\widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|_H^2 + V_{z_k}(u) - V_{z_{k+1}}(u). \qquad (6.50)
\end{aligned}$$

To bound $\alpha_s\langle\widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|_H^2$, consider the variable $v := \tau_{1,s}z_{k+1} + \tau_2\widetilde{v}^s + (1 - \tau_{1,s} - \tau_2)y_k$, then $v_{k+1} - v = \tau_{1,s}(z_k - z_{k+1})$. We have that

$$\begin{aligned}
&\mathbb{E}\left[\alpha_s\langle\widetilde{\nabla}_{k+1}, z_k - z_{k+1}\rangle - \frac{1}{2}\|z_k - z_{k+1}\|_H^2\right] \\
=&\mathbb{E}\left[\frac{\alpha_s}{\tau_{1,s}}\langle\widetilde{\nabla}_{k+1}, v_{k+1} - v\rangle - \frac{1}{2\tau_{1,s}^2}\|v_{k+1} - v\|_H^2\right] \\
=&\mathbb{E}\left[\frac{\alpha_s}{\tau_{1,s}}\left(\langle\widetilde{\nabla}_{k+1}, v_{k+1} - v\rangle - \frac{9\bar{L}}{2}\|v_{k+1} - v\|_H^2\right)\right] \\
\overset{(6.48)}{\leq}\ &\frac{\alpha_s}{\tau_{1,s}}\left(\phi(v_{k+1}) - \mathbb{E}[\phi(y_{k+1})] + \frac{1}{16\bar{L}}\mathbb{E}[\|\widetilde{\nabla}_{k+1} - \nabla\phi(v_{k+1})\|_{H,*}^2]\right) \\
\overset{(6.46)}{\leq}\ &\frac{\alpha_s}{\tau_{1,s}}\left(\phi(v_{k+1}) - \mathbb{E}[\phi(y_{k+1})] + \frac{1}{2B}(\phi(\widetilde{v}^s) - \phi(v_{k+1}) - \langle\nabla\phi(v_{k+1}), \widetilde{v}^s - v_{k+1}\rangle)\right).
\end{aligned} \qquad (6.51)$$

Take expectation on both sides of inequality (6.50), plug in inequality (6.51) and notice that $\mathbb{E}[\langle\widetilde{\nabla}_{k+1}, z_k - u\rangle] = \langle\nabla\phi(v_{k+1}), z_k - u\rangle$ and $\tau_2 = \frac{1}{2B}$, we get the desired bound. □

The rest of the proof for Theorem 6.4 is simply repeating the steps in Section 6.5.2, except that we replace Lemma 6.2 with Lemma 6.8. So we omit the details of the proof.

6.5.7 Details of Numerical Study

Data description. We use both synthetic and real grey-scale images as the marginal distribution. For the simulated data, we follow the data generation mechanism in [84, 97]. The images are generated by randomly positioning a square foreground on a background, with the foreground occupying about 20% of the space. The foreground has each pixel value randomly drawn from uniform $[0, 3]$, and the background has each pixel value randomly

drawn from uniform $[0, 1]$. Figure 6.2 shows some examples of the generated images.

For the real data, we randomly sample from the hand-written MNIST data set. Then we downscale the images to adjust the size of the marginal distribution. We also add a background with relatively small intensity to the down-scaled images to avoid numerical issue. With the marginal distribution determined, the cost matrix has each element calculated as the l_1 distance between the pixel locations on the image.

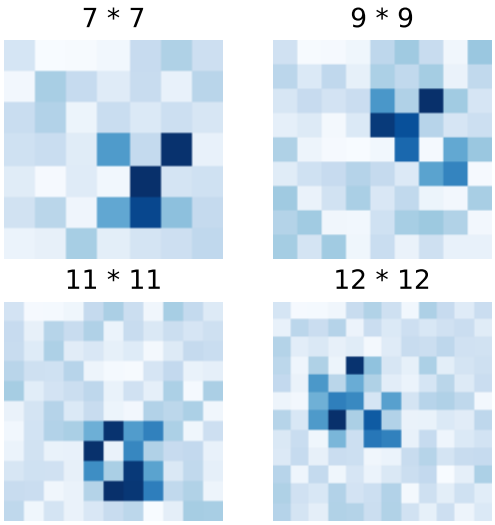


Figure 6.2: Synthetic image example.

Algorithm implementation. We compare the computational efficiency of the algorithms by measuring the number of arithmetic operations they use for finding an ϵ-solution of the OT between two marginal distributions for a fixed ϵ. To achieve this, all the algorithms are run with a rounding step. Thus for PDASMD, we run Algorithm 6.3. In particular, for step 2 of Algorithm 6.3, the PDASMD algorithm is run with the number of inner loops set to the problem size, $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ and $\|\cdot\|_H = \|\cdot\|_\infty$. We run the PDASGD algorithm by changing $\|\cdot\|_H$ to $\|\cdot\|_2$ compared to the PDASMD. Note that the PDASGD algorithm is essentially equivalent to that of [97]. We also run APDAGD [86], AAM [95], Sinkhorn [86], APDRCD [94] and Stochastic Sinkhorn (Algorithm 6.5) for comparison. The implementation of all the algorithms above follows their standard definitions; there is no hyper-parameter to tune.

We also implement experiment for PDASMD-B on both synthetic and real data. For a fixed pair of marginals, PDASMD-B is implemented for a sequence of batch sizes. The number of inner loops is set to be the problem size divided by the batch size, which matches the setting in Corollary 6.1, and we take $w(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ and $\|\cdot\|_H = \|\cdot\|_\infty$, which are the

same as the experiment of PDASMD. For each batch size, the total number of computations and the running time are recorded.

All our experiments are run on Google Colab using **NO** GPU or TPU accelerator. We attach the full code and data to reproduce our results in the Supplemental Material.

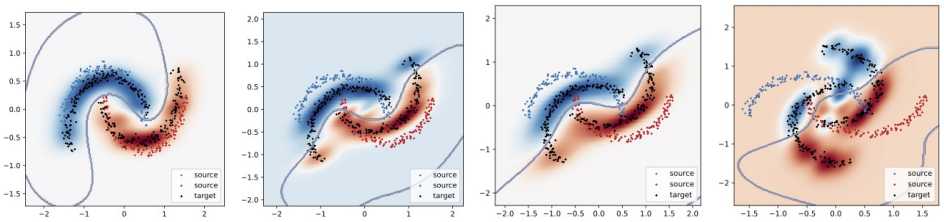
6.5.8 Application of the PDASMD Algorithm to Machine Learning Tasks

The Optimal Transport can be applied to some modern machine learning tasks such as domain adaptation and color transfer. In this section, we illustrate that our PDASMD algorithm, when applied to OT, can solve those problems.

Domain Adaptation. This experiment aims to show that our PDASMD algorithm, when applied to OT, can successfully perform domain adaptation. In short, domain adaptation means transferring knowledge from a source domain to a target domain for which data have different probability density functions. For more details on the domain adaptation problem description and its OT formulation, see [103].

We use the two-moons example to illustrate the application of our PDASMD algorithm on domain adaptation. The two moons example uses simulated data. The source domain consists of two entangled moons, where each moon represents one class. The target domain is built by applying a rotation to the two moons. We sample 150 labeled data points from each moon as our source domain. The target domain consists of the same number of samples, where the samples are independent of the source domain and are unlabeled. We use the labeled source domain data, transfer them to the target domain by OT using our PDASMD algorithm, and learn an SVM classifier with the Gaussian kernel using the transferred source data on the target domain. We test the generalization performance on 2,000 samples that follow the same distribution as the target domain.

Figure 6.3 shows the domain adaptation result. In Figure 6.3, we plot the source domain, target domain (for different rotation angles), the transformed density, and decision



(a) rotation = 20° (b) rotation = 40° (c) rotation = 50° (d) rotation = 90°

Figure 6.3: Two Moons Example

boundaries. From the plots, we see that the transformed density reasonably fits the major parts of the target domain when the rotation angle is not too large ($\leq 50°$). This shows that the PDASMD algorithm successfully performs the domain adaptation.

We report the generalization performance of the domain adaptation in Table 6.2. We have three columns: the rotation degree of the target domain in the two moons example, the mean classification error when the domain adaptation is performed using our PDASMD algorithm, and the mean classification error of OT-IT in [103] (where they solve entropic OT by the Sinkhorn algorithm). From Table 6.2, we see that our PDASMD algorithm performs better than the Sinkhorn when the rotation degree is large ($> 50°$).

Table 6.2: Mean Classification Error over 10 Repetitions of the Two Moons Example.

ROTATION DEGREE	ERROR BY PDASMD	ERROR BY OT-IT [103]
10°	0.022	**0**
20°	0.054	**0.007**
30°	**0.043**	0.054
40°	0.169	**0.102**
50°	**0.221**	**0.221**
70°	**0.317**	0.398
90°	**0.488**	0.508

Color Transfer. This experiment shows that our PDASMD algorithm successfully performs the color transfer task. The color transfer takes two input images and imposes the color palette of the first image onto a second one. Color transfer can be formulated as an OT problem. For more details see references [104, 105].

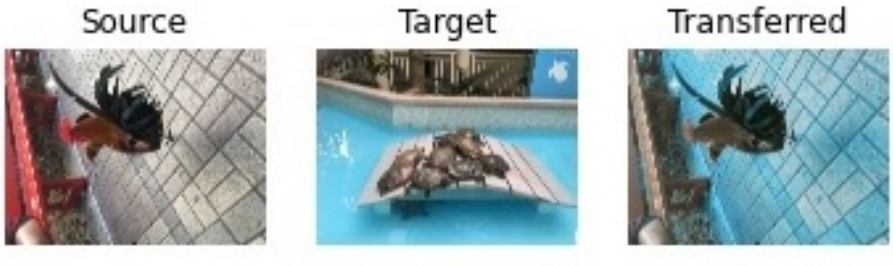


Figure 6.4: Color Transfer Example

For an example of the color transfer problem, we apply our PDASMD algorithm to solve the corresponding OT problem. We show the color transfer result in Figure 6.4. Though we cannot evaluate the color transfer result quantitatively, one can tell from Figure 6.4 that the color of the target image has been successfully transferred to the source image. This shows that our PDASMD algorithm successfully performs the color transfer task.

CHAPTER 7
CONCLUSION AND FUTURE WORK

This chapter discusses the findings of this thesis and proposes some future directions.

In Chapter 2, we show that the directional bias holds to stochastic gradient descent (SGD) with annealing step size in kernel learning. Specifically, the first stage of SGD with moderate step size should run long enough; then, in the second stage, by decreasing step size, we have the directional bias towards the largest eigenvalue of the Hessian, which helps in obtaining a better generalization error bound. Our result matches a technique for tuning the step size that people adopt in practice: starting with a large step size, running long enough until the error plateaus, then decreasing the step size [27]. Although this technique is always used for speed convergence, we show that it also helps in predictive power, which becomes even better. Our result applies to kernel learning, and one can consider extending it to deep learning, since some deep learning tasks can be approximated by kernel learning [15].

In Chapter 3, we analyze the implicit regularization property of variance reduced stochastic mirror descents (VRSMD), which covers both underfitting and overfitting cases in linear regression. In particular, our theorem shows that the implicit regularization property can help VRSMD find a sparse ground truth with a small error. One may further investigate an early stopped VRSMD estimator to see if it is minimax optimal. Our experiment shows that VRSMD with early stopping has comparable prediction performance to the Hadamard GD that is minimax optimal for sparse regression [13, 12]. This leads to the open question of how to select a good mirror map and an optimal stopping time in VRSMD (or variants of VRSMD) such that the resulting estimator is minimax optimal and thus generalizes well.

In Chapter 4, we show that the VRSMD and the variance reduced stochastic natural gradient descent (VRSNGD) are equivalent. Using such equivalence, we can easily extend

provable results from one algorithm to the other. In particular, we prove the convergence rate of VRSNGD using the convergence result of VRSMD, and we show the parametric invariance of VRSMD using such a property of VRSNGD. One may further study both algorithms in detail for the case where the parameter lies in a Riemannian manifold to promote further understanding.

In Chapter 5, we provide two online covariance estimators for the Recursive One-Over-T SGD (ROOT-SGD) algorithm in online learning. Our plug-in covariance estimator converges to the true asymptotic covariance at the optimal rate $\mathcal{O}(1/\sqrt{t})$, where t is the sample size. Our random-scaling covariance estimator is asymptotically consistent. Comparing the two estimators, the random-scaling covariance estimator has less computation but does not have a guaranteed convergence rate. Our covariance estimators help to develop asymptotically consistent statistical inference. One may apply our statistical inference method to a broad range of stochastic optimization problems. One can consider further improving the statistical inference in a finite learning case. For example, one may debias the point estimator, then use the debiased point estimator combined with our covariance estimators for statistical inference.

In Chapter 6, we propose a new first-order algorithm – Primal-Dual Accelerated Stochastic Proximal Mirror Descent (PDASMD) – for solving the entropic OT (Optimal Transport). We achieve a better computational complexity compared to other first-order algorithms when solving entropic OT. One may apply the acceleration and variance reduction techniques that we use in our algorithm to other first-order algorithms to improve their provable computational complexity. And one may further investigate the batch version of our PDASMD algorithm to improve the computational power of the PDASMD algorithm by parallel computing.

REFERENCES

[1] S. Ma, R. Bassily, and M. Belkin, "The power of interpolation: Understanding the effectiveness of sgd in modern over-parametrized learning," *arXiv:1712.06559*, 2018.

[2] P. L. Bartlett, A. Montanari, and A. Rakhlin, "Deep learning: A statistical viewpoint," *arXiv preprint arXiv:2103.09177*, 2021.

[3] G. Wahba, *Spline models for observational data*. SIAM, 1990.

[4] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on image processing*, vol. 16, no. 2, pp. 349–366, 2007.

[5] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *International Conference on Machine Learning*, 2006, pp. 377–384.

[6] M. Derezinski, F. T. Liang, and M. W. Mahoney, "Exact expressions for double descent and implicit regularization via surrogate random design," in *Advances in Neural Information Processing Systems*, vol. 33, Curran Associates, Inc., 2020, pp. 5152–5164.

[7] J. Wu, D. Zou, V. Braverman, and Q. Gu, "Direction matters: On the implicit bias of stochastic gradient descent with moderate learning rate," in *International Conference on Learning Representations*, 2021.

[8] T. Liu, Z. Chen, E. Zhou, and T. Zhao, "A diffusion approximation theory of momentum sgd in nonconvex optimization," *arXiv:1802.05155*, 2018.

[9] A. Ali, E. Dobriban, and R. Tibshirani, "The implicit regularization of stochastic gradient flow for least squares," in *International Conference on Machine Learning*, PMLR, 2020, pp. 233–244.

[10] G. Blanc, N. Gupta, G. Valiant, and P. Valiant, "Implicit regularization for deep neural networks driven by an Ornstein-Uhlenbeck like process," in *Proceedings of Thirty Third Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, vol. 125, PMLR, 2020, pp. 483–513.

[11] Y. Cao, Z. Fang, Y. Wu, D.-X. Zhou, and Q. Gu, "Towards understanding the spectral bias of deep learning," *arXiv:1912.01198*, 2020.

[12] T. Vaskevicius, V. Kanade, and P. Rebeschini, "Implicit regularization for optimal sparse recovery," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.

[13] P. Zhao, Y. Yang, and Q.-C. He, "Implicit regularization via hadamard product over-parametrization in high-dimensional linear regression," *arXiv preprint arXiv:1903.09367*, 2019.

[14] M. Belkin, S. Ma, and S. Mandal, "To understand deep learning we need to understand kernel learning," in *International Conference on Machine Learning*, PMLR, 2018, pp. 541–549.

[15] A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," 2018.

[16] R. Parhi and R. D. Nowak, "What kinds of functions do deep neural networks learn? insights from variational spline theory," *arXiv:2105.03361*, 2021.

[17] T. Liang and A. Rakhlin, "Just interpolate: Kernel 'Ridgeless' regression can generalize," *The Annals of Statistics*, vol. 48, no. 3, pp. 1329–1347, 2020.

[18] B. Schölkopf, J. Weston, E. Eskin, C. Leslie, and W. S. Noble, "A kernel approach for learning from almost orthogonal patterns," in *European Conference on Machine Learning*, Springer, 2002, pp. 511–528.

[19] J. Weston, B. Schölkopf, E. Eskin, C. Leslie, and W. S. Noble, "Dealing with large diagonals in kernel matrices," *Annals of the institute of statistical mathematics*, vol. 55, no. 2, pp. 391–408, 2003.

[20] Q. Tao, S. Scott, N. Vinodchandran, T. T. Osugi, and B. Mueller, "An extended kernel for generalized multiple-instance learning," in *16th IEEE International Conference on Tools with Artificial Intelligence*, IEEE, 2004, pp. 272–277.

[21] H. Saigo, J.-P. Vert, N. Ueda, and T. Akutsu, "Protein homology detection using string alignment kernels," *Bioinformatics*, vol. 20, no. 11, pp. 1682–1689, 2004.

[22] M. Cuturi, J.-P. Vert, O. Birkenes, and T. Matsui, "A kernel for time series based on global alignments," in *ICASSP'07*, IEEE, 2007.

[23] M. Cuturi, "Fast global alignment kernels," in *International Conference on Machine Learning*, 2011, pp. 929–936.

[24] J. Kandola, T. Graepel, and J. Shawe-Taylor, "Reducing kernel matrix diagonal dominance using semi-definite programming," in *Learning Theory and Kernel Machines*, Springer, 2003, pp. 288–302.

[25] E. Candes and T. Tao, "The dantzig selector: Statistical estimation when p is much larger than n," *The annals of Statistics*, vol. 35, no. 6, pp. 2313–2351, 2007.

[26] S. Chen and D. Donoho, "Basis pursuit," in *Proceedings of 1994 28th Asilomar Conference on Signals, Systems and Computers*, IEEE, vol. 1, 1994, pp. 41–44.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[28] G. Raskutti, M. J Wainwright, and B. Yu, "Minimax-optimal rates for sparse additive models over kernel classes via convex programming.," *Journal of Machine Learning Research*, vol. 13, no. 2, 2012.

[29] H. Hazimeh, R. Mazumder, and P. Radchenko, "Grouped variable selection with discrete optimization: Computational and statistical perspectives," *arXiv:2104.07084*, 2021.

[30] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[31] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," *arXiv:2106.11342*, 2021.

[32] W. Li, Z. Wang, Y. Zhang, and G. Cheng, "Variance reduction on adaptive stochastic mirror descent," *arXiv preprint arXiv:2012.13760*, 2021.

[33] Z. Allen-Zhu, "Katyusha: The first direct acceleration of stochastic gradient methods," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 8194–8244, 2017.

[34] J. Fan, Z. Yang, and M. Yu, "Understanding implicit regularization in over parameterized single index model," *Journal of the American Statistical Association*, pp. 1–14, 2022.

[35] S. Arora, N. Cohen, W. Hu, and Y. Luo, "Implicit regularization in deep matrix factorization," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.

[36] S. Azulay *et al.*, "On the implicit bias of initialization shape: Beyond infinitesimal mirror descent," *arXiv preprint arXiv:2102.09769*, 2021.

[37] S. L. Smith, B. Dherin, D. Barrett, and S. De, "On the origin of implicit regularization in stochastic gradient descent," in *International Conference on Learning Representations*, 2021.

[38] N. Azizan and B. Hassibi, "Stochastic Gradient/Mirror Descent: Minimax optimality and implicit regularization," in *International Conference on Learning Representations*, 2019.

[39] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," *Advances in Neural Information Processing Systems*, vol. 26, pp. 315–323, 2013.

[40] L. M. Nguyen, J. Liu, K. Scheinberg, and M. Takáč, "SARAH: A novel method for machine learning problems using stochastic recursive gradient," in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., ser. Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 2613–2621.

[41] C. Fang, C. J. Li, Z. Lin, and T. Zhang, "Spider: Near-optimal non-convex optimization via stochastic path-integrated differential estimator," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31, Curran Associates, Inc., 2018.

[42] S. J. Reddi, A. Hefny, S. Sra, B. Poczos, and A. Smola, "Stochastic variance reduction for nonconvex optimization," in *International Conference on Machine Learning*, 2016, pp. 314–323.

[43] H. Karimi, J. Nutini, and M. Schmidt, "Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2016, pp. 795–811.

[44] A. Sidford, "Introduction to optimization theory lecture notes Chapter 5," 2019.

[45] Y. Nesterov, "Introductory lectures on convex programming volume i: Basic course," *Lecture notes*, vol. 3, no. 4, p. 5, 1998.

[46] A. Ben-Tal and A. Nemirovski, "Lectures on modern convex optimization (2012)," *SIAM, Philadelphia, PA. Google Scholar Google Scholar Digital Library Digital Library*, 2011.

[47] G. Raskutti and S. Mukherjee, "The information geometry of mirror descent," *IEEE Transactions on Information Theory*, vol. 61, no. 3, pp. 1451–1457, 2015.

[48] G. Casella and R. L. Berger, *Statistical inference*. Cengage Learning, 2021.

[49] O. Barndorff-Nielsen, *Information and exponential families: in statistical theory*. John Wiley & Sons, 2014.

[50] P. McCullagh and J. A. Nelder, *Generalized linear models*. Routledge, 2019.

[51] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural computation*, vol. 10, no. 2, pp. 251–276, 1998.

[52] S.-i. Amari, "Differential-geometrical methods in statistics," *Lecture Notes on Statistics*, vol. 28, p. 1, 1985.

[53] C. Rao, *Information and accuracy attainable in the estimation of statistical parameters. kotz s & johnson nl (eds.), breakthroughs in statistics volume i: Foundations and basic theory, 235–248*, 1945.

[54] A. Nemirovski, D. Yudin, and E. Dawson, *Problem Complexity and Method Efficiency in Optimization* (A Wiley-Interscience publication). Wiley, 1983, ISBN: 9780471103455.

[55] F. Nielsen and R. Nock, "Entropies and cross-entropies of exponential families," in *2010 IEEE International Conference on Image Processing*, IEEE, 2010, pp. 3621–3624.

[56] K. S. Azoury and M. K. Warmuth, "Relative loss bounds for on-line density estimation with the exponential family of distributions," *Machine Learning*, vol. 43, no. 3, pp. 211–246, 2001.

[57] A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh, and J. Lafferty, "Clustering with bregman divergences.," *Journal of machine learning research*, vol. 6, no. 10, 2005.

[58] S.-i. Amari, *Information geometry and its applications*. Springer, 2016, vol. 194.

[59] J. Hiriart-Urruty and C. Lemarechal, *Convex Analysis and Minimization Algorithms II: Advanced Theory and Bundle Methods* (Grundlehren der mathematischen Wissenschaften). Springer Berlin Heidelberg, 1996, ISBN: 9783540568520.

[60] J. Martens, "New insights and perspectives on the natural gradient method," *Journal of Machine Learning Research*, vol. 21, no. 146, pp. 1–76, 2020.

[61] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[62] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM journal on control and optimization*, vol. 30, no. 4, pp. 838–855, 1992.

[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[64] T. Dozat, "Incorporating nesterov momentum into adam," 2016.

[65] C. J. Li, W. Mou, M. Wainwright, and M. Jordan, "Root-sgd: Sharp nonasymptotics and asymptotic efficiency in a single algorithm," in *Conference on Learning Theory*, PMLR, 2022, pp. 909–981.

[66] X. Chen, J. D. Lee, X. T. Tong, and Y. Zhang, "Statistical inference for model parameters in stochastic gradient descent," *The Annals of Statistics*, vol. 48, no. 1, pp. 251–273, 2020.

[67] X. Chen, Z. Lai, H. Li, and Y. Zhang, "Online statistical inference for gradient-free stochastic optimization," *arXiv preprint arXiv:2102.03389*, 2021.

[68] W. Zhu, X. Chen, and W. B. Wu, "Online covariance matrix estimation in stochastic gradient descent," *Journal of the American Statistical Association*, pp. 1–12, 2021.

[69] W. J. Su and Y. Zhu, "Uncertainty quantification for online learning and stochastic approximation via hierarchical incremental gradient descent," *arXiv preprint arXiv:1802.04876*, 2018.

[70] S. Lee, Y. Liao, M. H. Seo, and Y. Shin, "Fast and robust online inference with stochastic gradient descent via random scaling," *arXiv preprint arXiv:2106.03156*, 2021.

[71] N. M. Kiefer, T. J. Vogelsang, and H. Bunzel, "Simple robust testing of regression hypotheses," *Econometrica*, vol. 68, no. 3, pp. 695–714, 2000.

[72] D. N. Politis, J. P. Romano, and M. Wolf, *Subsampling*. Springer Science & Business Media, 1999.

[73] S. Lahiri and S. Lahiri, *Resampling methods for dependent data*. Springer Science & Business Media, 2003.

[74] K. M. Abadir and P. Paruolo, "Simple robust testing of regression hypotheses: A comment," *Econometrica*, vol. 70, no. 5, pp. 2097–2099, 2002.

[75] A. Defazio, F. Bach, and S. Lacoste-Julien, "Saga: A fast incremental gradient method with support for non-strongly convex composite objectives," in *Advances in Neural Information Processing Systems*, vol. 27, Curran Associates, Inc., 2014.

[76] K. M. Abadir and P. Paruolo, "Two mixed normal densities from cointegration analysis," *Econometrica: Journal of the Econometric Society*, pp. 671–680, 1997.

[77] X. Zhou, "On the fenchel duality between strong convexity and lipschitz continuous gradient," *arXiv preprint arXiv:1803.06573*, 2018.

[78] P. Hall and C. C. Heyde, *Martingale limit theory and its application*. Academic press, 2014.

[79] C. J. Li, W. Mou, M. J. Wainwright, and M. I. Jordan, *Root-sgd: Sharp nonasymptotics and asymptotic efficiency in a single algorithm*, 2020. arXiv: 2008.12690 [math.OC].

[80] A. W. Van der Vaart, *Asymptotic statistics*. Cambridge university press, 2000, vol. 3.

[81] G. Monge, "Mémoire sur la théorie des déblais et des remblais," *Histoire de l'Académie Royale des Sciences de Paris*, 1781.

[82] L. V. Kantorovich, "On the translocation of masses," in *Dokl. Akad. Nauk. USSR (NS)*, vol. 37, 1942, pp. 199–201.

[83] C. Villani, *Optimal transport: old and new*. Springer, 2009, vol. 338.

[84] J. Altschuler, J. Weed, and P. Rigollet, "Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 1961–1971.

[85] M. Cuturi, "Sinkhorn Distances: Lightspeed Computation of Optimal Transport," in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., vol. 26, Curran Associates, Inc., 2013.

[86] P. Dvurechensky, A. Gasnikov, and A. Kroshnin, "Computational optimal transport: Complexity by accelerated gradient descent is better than by sinkhorn's algorithm," in *International conference on machine learning*, PMLR, 2018, pp. 1367–1376.

[87] T. Lin, N. Ho, and M. Jordan, "On efficient optimal transport: An analysis of greedy and accelerated mirror descent algorithms," in *International Conference on Machine Learning*, PMLR, 2019, pp. 3982–3991.

[88] B. K. Abid and R. Gower, "Stochastic algorithms for entropy-regularized optimal transport problems," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2018, pp. 1505–1512.

[89] J. Blanchet, A. Jambulapati, C. Kent, and A. Sidford, "Towards optimal running times for optimal transport," *arXiv preprint arXiv:1810.07717*, 2018.

[90] K. Quanrud, "Approximating Optimal Transport With Linear Programs," in *2nd Symposium on Simplicity in Algorithms (SOSA 2019)*, J. T. Fineman and M. Mitzenmacher, Eds., ser. OpenAccess Series in Informatics (OASIcs), vol. 69, Dagstuhl,

Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, 6:1–6:9, ISBN: 978-3-95977-099-6.

[91] A. Jambulapati, A. Sidford, and K. Tian, "A Direct tilde{O} (1/epsilon) Iteration Parallel Algorithm for Optimal Transport," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019.

[92] T. Lin, N. Ho, and M. I. Jordan, "On the efficiency of sinkhorn and greenkhorn and their acceleration for optimal transport," *arXiv preprint arXiv:1906.01437*, 2019.

[93] N. Lahn, D. Mulchandani, and S. Raghvendra, "A graph theoretic additive approximation of optimal transport," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[94] W. Guo, N. Ho, and M. Jordan, "Fast Algorithms for Computational Optimal Transport and Wasserstein Barycenter," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, S. Chiappa and R. Calandra, Eds., ser. Proceedings of Machine Learning Research, vol. 108, PMLR, 2020, pp. 2088–2097.

[95] S. Guminov, P. Dvurechensky, N. Tupitsa, and A. Gasnikov, "On a Combination of Alternating Minimization and Nesterov's Momentum," in *Proceedings of the 38th International Conference on Machine Learning*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 3886–3898.

[96] A. Chambolle and J. P. Contreras, "Accelerated bregman primal-dual methods applied to optimal transport and wasserstein barycenter problems," *arXiv preprint arXiv:2203.00802*, 2022.

[97] Y. Xie, Y. Luo, and X. Huo, "An accelerated stochastic algorithm for solving the optimal transport problem," *arXiv e-prints*, 2022.

[98] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical programming*, vol. 103, no. 1, pp. 127–152, 2005.

[99] A. Genevay, M. Cuturi, G. Peyré, and F. Bach, "Stochastic optimization for large-scale optimal transport," in *NIPS 2016-Thirtieth Annual Conference on Neural Information Processing System*, 2016.

[100] Y. LeCun, "The mnist database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[101] K. Mishchenko, "Sinkhorn algorithm as a special case of stochastic mirror descent," *CoRR*, vol. abs/1909.06918, 2019. arXiv: 1909.06918.

[102] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Science & Business Media, 2003, vol. 87.

[103] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *arXiv preprint arXiv:1507.00504*, 2015.

[104] S. Ferradans, N. Papadakis, G. Peyré, and J.-F. Aujol, "Regularized discrete optimal transport," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1853–1882, 2014.

[105] J. Rabin, S. Ferradans, and N. Papadakis, "Adaptive color transfer with relaxed optimal transport," in *2014 IEEE international conference on image processing (ICIP)*, IEEE, 2014, pp. 4852–4856.