First-order methods for structured optimization

by

Huang Fang

B.Sc, The Central University of Finance and Economics, China, 2011

M.Sc, The University of California, Davis, US, 2015

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Computer Science)

The University of British Columbia

(Vancouver)

October 2021

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

First-order methods for structured optimization

submitted by **Huang Fang** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Computer Science**.

Examining Committee:

Michael P. Friedlander, Computer Science
Supervisor

Nicolas J.A. Harvey, Computer Science
Supervisory Committee Member

Mark Schmidt, Computer Science
Supervisory Committee Member

Brian Wetton, Mathematics
University Examiner

Bruce Shepherd, Computer Science
University Examiner

Chih-Jen Lin, Computer Science, National Taiwan University
External Examiner

Abstract

First-order methods are gaining substantial interest in the past two decades because of their superior performance in solving today's large-scale problems. In this thesis, we study some widely used first-order methods for problems that satisfy certain structures. Specifically, in the first part, we contribute to coordinate optimization and show that greedy coordinate descent (GCD) has an implicit screening ability that usually selects coordinates that are nonzero at the solution, which explains why GCD works exceptionally well for problems that admit sparse solutions. We also extend the elegant safe-screening rule that depends on duality gap to atomic-norm regularized problems. In the second part, we study online mirror descent (OMD) with unknown time horizon and unbounded domain, which is known to suffer from linear regret. We provide a stabilization technique and show that the stabilized-OMD can achieve sublinear regret. We also build the connection between stabilized-OMD and dual averaging. In the third part, we derive improved iteration complexity of the stochastic subgradient method for over-parameterized models that satisfy an interpolation condition. The obtained iteration complexity matches the rate of the stochastic gradient method applied to smooth problems that also satisfy an interpolation condition. Our analysis partially explains the empirical observation that nonsmoothness in modern machine learning models sometimes does not slow down the training process.

Lay Summary

Optimization algorithms are prevalent in modern data-driven applications. The recent immense growth of data volumes has spurred the rapid development of new varieties of algorithms efficient for large problems. Despite their success in practice, the theory behind these algorithms is incomplete and often fails to explain empirical performance. This thesis aims to fill some of these gaps, contribute to a better understanding of some widely used optimization algorithms, and shed light on the design of new algorithms.

Preface

The main body of this thesis is based several collaborative papers that are either published or currently under review.

- The work presented in the first part of Chapter 2 (Section 2.3) was published in AISTATS, 2020 (Fang et al., 2020a). I am the primary contributor to this paper. The convergence analysis and experiments are developed by me. Zhenan Fan participated in some discussions and was involved in polishing the proofs. Yifan Sun improved the writing and presentation of the paper and provided some useful suggestions. Michael Friedlander provided overall supervision, technical guidance, writing assistance, and financial support.

- The material in the second part of Chapter 2 (Section 2.4) was submitted and currently under review. Zhenan Fan and me contributed equally to this paper, the main theorem of this paper is based on our discussion and the case study on nuclear-norm regularized problem is developed by me. Michael Friedlander provided overall supervision, technical guidance, writing assistance, and financial support.

- The material in Chapter 3 was published in ICML, 2020 (Fang et al., 2020b). This paper originates from a conjecture in Nick Harvey's graduate course CPSC531H. I disproved the conjecture when taking Nick Harvey's course and extended this result to a research paper. The stabilization technique owns to Nick Harvey. I proposed to apply the stabilization technique to online mirror descent (OMD) and developed the sublinear regret bound. The first-order regret bound is based on my discussion with Nick Harvey. Victor Portella provided the analysis to compare stabilized-OMD and dual averaging. Nick

Harvey, Victor Portella and me were heavily involved in writing up the proofs. Michael Friedlander provided overall supervision, technical guidance, writing assistance, and financial support.

- The material in Chapter 4 was published in ICLR, 2021 (Fang et al., 2021). I am the primary contributor to this paper. The improved iteration complexities, lower bound analysis, experiments and initial draft are from me. Zhenan Fan participated in some discussions and checked the proofs. Michael Friedlander provided overall supervision, technical guidance, writing assistance, and financial support.

Table of Contents

List of Tables

List of Figures

Acknowledgments

This thesis would not have been possible without the help and kindness of many people throughout the last several years. First, I would like to thank my supervisor Michael Friedlander for providing constant support and giving me the freedom to explore a diverse set of projects. I have learned very much from Michael's broad knowledge in optimization and computation. His rigorous research attitude and cheerful spirit have given me an example to pursue.

I would also like to thank Cho-Jui Hsieh, who led me on the path of research and helped me building my research skills. He has been a role model for me to pursue throughout my Ph.D. study. I am also grateful to Nick Harvey, with whom I have had the privilege to collaborative. I learned how to think as a theoretician from him. Next, I would like to thank the university examiners, Bruce Shepherd and Brian Wetton, and the external examiner, Chih-Jen Lin for reading my thesis and providing valuable feedback.

I am thankful to my other collaborators Zhenan Fan, Victor Portella, Yifan Sun for helping me to be more productive than I would have been able to on my own. I would like to thank Naomi Graham and Emma Hansen for proofreading some materials of the thesis and providing helpful feedback to improve the presentation. I would like to thank my friends Zhenan Fan, Wen Xiao, Liran Li, Xin Ding, Qiuyan Liu and Anyi Bao for the company during my Ph.D. study, they make my life in Vancouver more joyful.

Last but not least, my academic journey is not possible without the understanding and encouragement from my parents. I would like to thank them as they always support me to pursue what I want. I would like to dedicate this thesis to them.

Chapter 1

Introduction

Optimization is a fundamental aspect of many fields, including machine learning, data mining, signal processing, and bioinformatics. In modern applications such as recommender systems, computer vision, and natural language processing, both the amount of data and the model's size can be very large. For example, some of today's social networks could have hundreds of millions of users, and some recent neural network architectures could have billions of parameters. As a result, efficient optimization algorithms that can handle both big data and complex models are of great interest in recent years. Many optimization algorithms were proposed or re-discovered in the last two decades to solve today's large-scale optimization problems. Due to the low numerical accuracy required by most machine learning and data mining tasks, first-order optimization methods with cheap per-iteration computational cost dominate certain fields in machine learning. In particular, stochastic gradient descent (SGD) and coordinate descent (CD) are the two most important representatives. SGD and its variants are dominant for the big-N problems — i.e., problems with a large number of samples, while CD and its variants are highly efficient in handling the structured big-p problems — i.e., models with a large number of parameters.

The empirical success of first-order methods has driven extensive research in recent years on understanding the effectiveness of these methods as well as developing new variants of first-order methods for emerging applications. However, despite some notable progress made by some pioneering researchers in recent years,

there are still many open problems left unsolved in this area. Given the importance and popularity of first-order optimization algorithms, there is a great need to solve these open problems. This thesis aims to solve some of these open problems and contribute to a better understanding of first-order optimization methods in different scenarios. In this chapter, we summarize some classical results of some important first-order algorithms and describe the contributions of this thesis.

1.1 Notation and preliminaries

Throughout this thesis, unless otherwise specified, we use capital letters $A, B, \ldots$ to denote matrices, lowercase letters $x, y, w, \ldots$ to denote vectors, except L which usually denote the smoothness parameter, and m, n, d which are commonly used as the dimension of matrices or vectors. We use Greek letters $\alpha, \beta, \gamma, \ldots$ to denote scalars. The ith entry of a given vector x is denoted as x_i, and we use $x^{(t)}$ to represent the t-th iterate in an algorithm. The norm $\|\cdot\|$ stands for Euclidean-norm. We define $[n]$ to be the set $\{1, 2, \ldots, n\}$. We denote a solution of an optimization problem to be x^* and the optimal function value $f(x^*)$ to be f^*, we assume that x^* exist and $f^* > -\infty$ throughout this thesis.

Before proceeding to the classical results of first-order methods, we review some standard definitions that are commonly used in the literature of convex optimization.

Definition 1.1.1. A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y)$$

for any $x, y \in \mathbb{R}^d$, and any $\lambda \in [0, 1]$.

Definition 1.1.2. For a closed, convex function $f : \mathbb{R}^d \to \mathbb{R}$, the subdifferential of f at x is defined as

$$\partial f(x) \coloneqq \left\{ v \in \mathbb{R}^d \;\middle|\; f(y) \geq f(x) + \langle v, y - x \rangle,\ \forall y \in \mathbb{R}^d \right\}.$$

Definition 1.1.3. The dual norm of a given norm $\|\cdot\|$ is defined as

$$\|z\|_* \coloneqq \sup\{\langle z, x \rangle \mid \|x\| \leq 1, x \in \mathbb{R}^d\}.$$

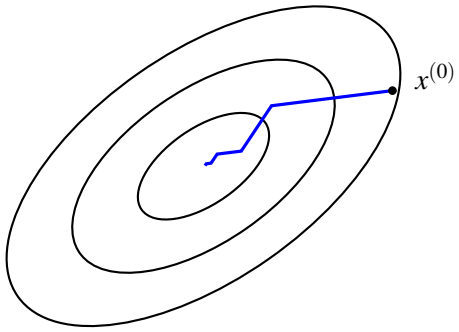


Figure 1.1: Illustration of gradient descent

Definition 1.1.4. A function $f : \mathbb{R}^d \to \mathbb{R}$ is γ-Lipschitz continuous with respect to a norm $\|\cdot\|$ for some $\gamma > 0$ if $\forall x, y \in \mathbb{R}^d$,

$$|f(x) - f(y)| \leq \gamma \|x - y\|.$$

Definition 1.1.5. A function $f : \mathbb{R}^d \to \mathbb{R}$ is L-smooth with respect to a norm $\|\cdot\|$ for some $L > 0$ if it is differentiable and $\forall x, y \in \mathbb{R}^d$,

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|_*,$$

where $\|\cdot\|_*$ is the dual norm paired with $\|\cdot\|$.

Definition 1.1.6. A function $f : \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex with respect to a norm $\|\cdot\|$ for some $\mu \geq 0$ if $\forall x, y \in \mathbb{R}^d$,

$$f(x) \geq f(y) + \langle g, x - y \rangle + \frac{\mu}{2} \|x - y\|^2, \quad \forall g \in \partial f(y).$$

1.2 Gradient and subgradient descent

Gradient and subgradient descent (GD, subGD) are perhaps the oldest first-order optimization methods that can be traced back to 1847 (Cauchy, 1847). Consider the unconstrained optimization problem

$$\min_{x \in \mathbb{R}^d} f(x), \tag{1.1}$$

where f is a continuous function. In the t-th iteration, (sub)GD updates the iterate by moving it to the negative (sub)gradient direction

$$x^{(t+1)} = x^{(t)} - \eta_t g, \quad g \in \partial f(x^{(t)}),$$

where η_t is the stepsize used in the t-th iteration. An illustration of GD is shown in Figure 1.1. It is known that GD has the following convergence properties:

- When f is convex and L-smooth, GD with constant stepsize $\eta_t = 1/L$ is guaranteed to achieve an ε-accurate solution[1] in $\mathcal{O}(L\varepsilon^{-1})$ iterations (Nesterov, 2004). This rate can be improved to $\mathcal{O}(L\varepsilon^{-1/2})$ by adopting Nesterov's acceleration technique (Nesterov, 1983) and was proven to be optimal for first-order methods in the convex and smooth setting;

- When f is μ-strongly convex and L-smooth, GD with constant stepsize $\eta_t = 1/L$ achieves a linear convergence rate $\mathcal{O}\left((L/\mu)\log(\varepsilon^{-1})\right)$ (Nesterov, 2004);

- When f is convex but not necessarily smooth, GD with decaying stepsize $\eta_t \propto t^{-1/2}$ converges at the rate $\mathcal{O}(L\varepsilon^{-2})$ (Nemirovski and Yudin, 1983; Shor, 1984). If we treat L as a constant, then the rate $\mathcal{O}(\varepsilon^{-2})$ has been proven to be optimal for first-order methods (Nemirovski and Yudin, 1983) in the convex and nonsmooth setting;

- When f is L-smooth but not necessarily convex, GD may not converge to a global solution. Instead, GD with constant stepsize $\eta_t = 1/L$ can converge to an ε-stationary point[2] in $\mathcal{O}(L\varepsilon^{-1})$ iterations. This is a well-known result that can be found in optimization textbooks, but its origin is not known to the author's knowledge.

Note that many results in the literature treat the smoothness or strongly convex parameters as constants and omit them in the big-O notation. However, as we will show in Section 1.3, these parameters play an important role in understanding the convergence of coordinate descent. In order to compare the convergence rates

[1] A point x is an ε-accurate solution if $f(x) - f^* \leq \varepsilon$.
[2] A point x is an ε-stationary point if $\|\nabla f(x)\|^2 \leq \varepsilon$.

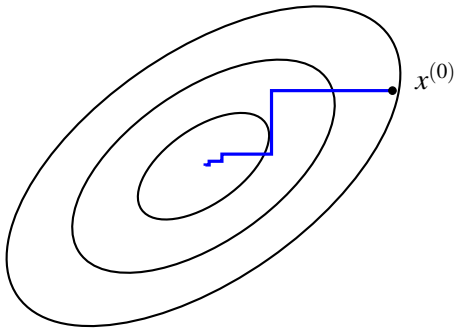


Figure 1.2: Illustration of cooridnate descent

between different first-order algorithms, we will not treat them as constants in this chapter.

1.3 Coordinate descent

Coordinate descent (CD) is a simple extension of GD. Instead of updating the whole vector x in each iteration, CD selects a single coordinate (or a small set of coordinates) according to some coordinate selection rules and apply standard gradient step to them. Consider the single coordinate selection case with the update rule

$$x^{(t+1)} = x^{(t)} - \eta_t \nabla_{i_t} f(x^{(t)}) \boldsymbol{e}_{i_t},$$

where i_t is the coordinate selected by some selection rules in the t-th iteration and $\boldsymbol{e}_{i_t}$ is a zero vector with a one in the i_t-th entry. An illustration of CD is shown in Figure 1.2.

CD is an old algorithm. Its history can be traced back to 1874 as an iterative method due to Seidel to solve linear systems, also known as the Gauss-Seidel method. Empirically, CD is simple and effective. Its superior performance against other classical algorithms on some machine learning problems makes it a popular optimizer for a wide range of applications, including clustering (Lloyd, 1982), support vector machines (Chang and Lin, 2011; Joachims, 1999; Platt, 1998) and LASSO (Hastie et al., 2008; Sylvain Sardy and Tseng, 2000). On the theory side, pioneering researchers made important contributions to understand the convergence

behavour of CD in late 20th century (Bertsekas and Tsitsiklis, 1989; Luo and Tseng, 1992, 1993; Tseng, 2001). However, the global convergence rate of CD was not clear until Nesterov's seminal work (Nesterov, 2012), in which Nesterov explicitly answered why CD could be significantly faster than GD.

Consider problem 1.1, where f is L-smooth. Let L_i denote the coordinate-wise smoothness parameter of f, namely for any $i \in [d]$, i.e.,

$$|\nabla_i f(x + \alpha \boldsymbol{e}_i) - \nabla_i f(x)| \leq L_i |\alpha|, \quad \forall x \in \mathbb{R}^d, \alpha \in \mathbb{R}.$$

Let $L_{\max} \coloneqq \max_{i \in [d]} L_i$. It is easy to verify that $L_{\max} \leq L \leq dL_{\max}$. With a simple random coordinate selection rule, Nesterov demonstrated that one could adopt a longer stepsize $1/L_{\max}$ compared with the stepsize $1/L$ used in GD when updating a single coordinate. Based on this observation, Nesterov established the following convergence rates for randomized CD (RCD):

- When f is convex, RCD is guaranteed to obtain an ε-accurate solution within $\mathcal{O}(dL_{\max}\varepsilon^{-1})$ iterations in expectation;
- When f is μ-strongly convex, RCD has the rate $\mathcal{O}\left((dL_{\max}/\mu)\log(\varepsilon^{-1})\right)$ in expectation.

Recall that $L_{\max} \leq L \leq dL_{\max}$, the above rates indicate that RCD is slower than GD (see the rates in Section 1.2). However, Nesterov pointed out that for a wide range of problems, RCD can be implemented cleverly such that running d RCD steps cost roughly the same time as one GD iteration. Note that the convergence rate of RCD in terms of epoch (d RCD iterations) is $\mathcal{O}(L_{\max}\varepsilon^{-1})$ for convex objective and $\mathcal{O}\left((L_{\max}/\mu)\log(\varepsilon^{-1})\right)$ for strongly convex objective. Therefore RCD is faster than GD in terms of runtime. We refer interested readers to Nesterov (2012) for more details.

Extensive research has been carried on since Nesterov's 2012 work. Here we just list some notable results: convergence analysis for proximal RCD (Richtárik and Takác, 2014), different coordinate selection rules (Beck and Tetruashvili, 2013; Lee and Wright, 2018; Nutini et al., 2015; Recht and Ré, 2012; Saha and Tewari, 2013; Sun and Ye, 2021), accelerated RCD (Allen Zhu et al., 2016; Lee and Sidford, 2013; Lin et al., 2015), and parallel CD with multiple computing cores (Bradley

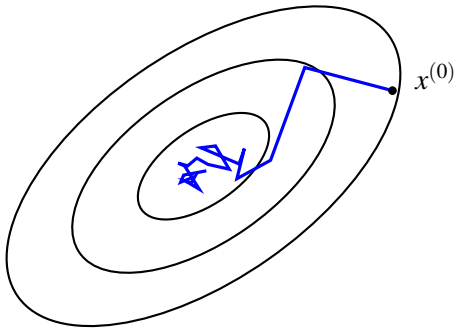


Figure 1.3: Illustration of stochastic gradient descent

et al., 2011; Hannah et al., 2019; Jaggi et al., 2014; Liu et al., 2014a; Richtárik and Takác, 2011; You et al., 2016).

1.4 Stochastic gradient descent

For many machine learning and data mining applications, we are interested in solving an *empirical-risk minimization* problem that can be written as a finite sum minimization problem

$$\min_{x\in\mathbb{R}^d} f(x) \coloneqq \frac{1}{n}\sum_{i=1}^{n} f_i(x),$$

where n is the number of training samples that could be very large in modern applications, for example 10^7 for the imagenet dataset[3].

GD or subGD are expensive in this setting since they need to call n gradient oracles for a single iteration. Instead of calculating the exact gradient in each iteration, stochastic gradient descent (SGD) (Robbins and Monro, 1951) method calculates an unbiased estimator of the true gradient from a small subset of the whole training samples. Consider sampling one training sample in each iteration, the update rule of SGD is

$$x^{(t+1)} = x^{(t)} - \eta_t g, \quad g \in \partial f_{i_t}(x^{(t)}),$$

for some i_t uniform randomly chosen from $\{1,2,\ldots,n\}$ in each iteration. An

[3] http://www.image-net.org/

illustration of SGD is shown in Figure 1.3. Obviously, the per iteration cost of SGD is n time cheaper than GD. At the cost of using cheap and noisy gradient for update, SGD converges slower than GD in terms of the number of iterations. SGD has the following convergence guarantee under the bounded variance assumption ($\sup_{x\in\mathbb{R}^d} \mathbb{E}[\|\nabla f_i(x) - \nabla f(x)\|] \leq \sigma$ for some $\sigma > 0$):

- When f is convex and γ-Lipschitz continuous, SGD with decaying stepsize $\eta_t \propto t^{-1/2}$ could obtain an ε-solution within $\mathcal{O}((\gamma^2+\sigma^2)\varepsilon^{-2})$ iterations (Nemirovski et al., 2009) in expectation;

- When f is μ-strongly convex and γ-Lipschitz continuous, SGD converges to global minimum with a rate $\mathcal{O}((\gamma^2+\sigma^2)\mu^{-1}\varepsilon^{-1})$ (Lacoste-Julien et al., 2012; Nemirovski et al., 2009) in expectation;

- When f is L-smooth but not necessarily convex, SGD with decaying stepsize $\eta_t \propto t^{-1/2}$ could approach an ε-stationary point within $\mathcal{O}(L\varepsilon^{-1} + L\sigma^2\varepsilon^{-2})$ iterations (Ghadimi and Lan, 2013) in expectation.

Compared with the rates of GD stated in Section 1.2, the convergence rates of SGD described above are worse. However, an extraordinary fact of SGD is that its convergence rates are independent of the total number of samples n. Therefore, SGD fits perfectly to applications with a large number of training samples and require low numerical accuracy. The superior empirical performance of SGD makes it and its variants such as ADAM (Kingma and Ba, 2015) the default choice for training modern machine learning models. Note that modern machine learning models often yield nonconvex and nonsmooth objective, but SGD still usually performs unreasonably well in these tasks.

An important advance of SGD in the last decade is the variance reduction technique proposed by Schmidt et al. (2017), who proposed the first linear convergent SGD (under strong convexity) based algorithm called SAG (Schmidt et al., 2017). Following this work, there is a line of research that tries to improve SGD with variance reduction under different scenarios. Examples include the well-known SVRG (Johnson and Zhang, 2013), SAGA (Defazio et al., 2014), and SARAH (Nguyen et al., 2017) algorithms.

1.5 Mirror descent

The mirror descent (MD) algorithm (Beck and Teboulle, 2003; Nemirovski and Yudin, 1983) is a generalization of the classical projected subgradient descent (PGD) method. The major difference between MD and PGD is that MD explicitly distinguishes the "primal" and "dual" spaces and applies the gradient update in the dual space. Consider problem 1.1 with the constraint $x \in \mathcal{X}$, the update rule of MD can be written as

$$x^{(t+1)} = \arg\min_{x \in \mathcal{X}} \left\{ \langle x, g \rangle + \frac{1}{\eta_t} B_\Phi(x, x^{(t)}) \right\}, \quad g \in \partial f(x^{(t)}),$$

where B_Φ is the Bregman divergence induced by the mirror map $\Phi : \mathbb{R}^d \to \mathbb{R}$, detailed explanation of these technical terms will be described in Chapter 3.

We can recover the update rule of PGD by choosing the mirror map Φ to be 2-norm squared. Therefore PGD is a special case of MD. MD's main advantage over PGD is that MD, for certain applications, can enjoy a better Lipschitz constant and simpler projection than PGD by choosing an appropriate mirror map. A typical example that favours MD than PGD is when the constraint set is a probability simplex, i.e., $\mathcal{X} = \{\sum_{i=1}^d x_i = 1, x_i \geq 0\}$. In this case, MD has the following advantages:

- When f is γ-Lipschitz continuous with respect to $\|\cdot\|_\infty$. It follows that f is $\sqrt{d}\gamma$-Lipschitz continuous with respect to $\|\cdot\|_2$ and the number of iterations required to obtain an ε-solution would be $\mathcal{O}(\sqrt{d}\gamma\varepsilon^{-2})$. By using negative entropy as the mirror map, MD can improve the rate to $\mathcal{O}(L\varepsilon^{-2})$, which could be significant when d is large.

- The projection in Euclidean space requires us to do a sorting while the projection used in MD is a simple scaling operation by using negative entropy as the mirror map.

Given the above observations, MD with the negative entropy mirror map (also known as the exponentiated gradient descent) perfectly fits problems with Lipschitz constant related to infinity norm and probability simplex constraints. A classic example of this type is the *prediction with expert advice* problem in the online learning community. Therefore MD is an useful algorithmic template for online

learning problems and has been heavily studied in recent years.

1.6 Summary of contributions

As we described in previous sections, the theory of first-order optimization algorithms is heavily studied. However, there are some deficiencies with the current analysis:

- For some algorithms, the current analyses work for quite general problems. At the cost of generality, the analyses are sometimes too pessimistic and fail to explain some common empirical observations for problems that satisfy certain structures.

- For some other algorithms, the analyses work only for a restricted class of problems and cannot be applied to problems that satisfy some other important structures, such as problems with atomic-norm regularization instead of the classic 1-norm regularization.

This thesis aims to improve and extend the classical analysis of some first-order optimization algorithms for problems that satisfy certain structures.

In Chapter 2, we focus on coordinate descent and sparse optimization. First, we consider applying coordinate descent with greedy selection rule (GCD) for composite objectives in form $f + g$. When the composite problem admits sparse solutions, empirical evidence in the literature suggests that GCD, when initialized as zero vector, has an implicit variable selection ability. Therefore, GCD can usually select coordinates that are nonzero at the solution and converge significantly faster than randomized CD. By leveraging the composite problem structure and sparse solution, we present an improved analysis of GCD for sparse optimization and theoretically explained why GCD has the implicit variable selection property. Second, we consider problems with general atomic-sparsity and extend the analysis of the gap safe screening rule (Ndiaye et al., 2017) to general atomic sets. We also study the effectiveness of gap safe screening rule for problems with low-rank solutions.

In Chapter 3, we try to improve mirror descent in the context of online learning. In the literature of online convex optimization, online mirror descent (OMD) and

dual averaging (DA) are two fundamental algorithmic templates. They are known to have a very similar (or even identical) performance guarantee in most scenarios when a fixed stepsize is used. However, for dynamic stepsize, OMD is provably inferior to dual averaging. It is known that OMD with a dynamic stepsize scheduling can suffer from linear regret. We modify the OMD algorithm through a simple technique called *stabilization* and give essentially the same abstract regret bound for stabilized-OMD and DA by modifying the classical OMD convergence analysis in a careful and modular way. Simple corollaries of these bounds show that OMD with stabilization and DA enjoy the same performance guarantees in many applications even under dynamic stepsize scheduling.

In Chapter 4, we study the convergence behaviour of the stochastic subgradient descent (SSGD) method applied to over-parameterized nonsmooth optimization problems that satisfy an interpolation condition. By leveraging the composite structure of the empirical risk minimization problems, we prove that SSGD converges, respectively, with rates $\mathcal{O}(\varepsilon^{-1})$ and $\mathcal{O}(\log(\varepsilon^{-1}))$ for convex and strongly convex objectives when interpolation holds. These rates coincide with established rates for the stochastic gradient descent (SGD) method applied to smooth problems that also satisfy an interpolation condition. Our analysis provides a partial explanation for the empirical observation that sometimes SGD and SSGD behave similarly for training smooth and nonsmooth machine learning models. We also prove that the rate $\mathcal{O}(\varepsilon^{-1})$ is optimal for the subgradient method in the convex and interpolation setting.

Chapter 2

Coordinate descent and sparse optimization

Coordinate descent (CD) is gaining increasing interest due to its simplicity and effectiveness in the last two decades. It is the state-of-the-art optimization algorithm for problems that satisfy the *coordinate-friendly structure* (Nutini et al., 2015; Peng et al., 2016), that is the computation time required to update all coordinates is roughly the same as the time of one gradient descent step. Problems of this type including the ℓ_1 regularized least squares (LASSO) (Sylvain Sardy and Tseng, 2000) and the support vector machines (SVMs) (Platt, 1998).

2.1 Different selection rules

The coordinate selection rule used in the CD algorithm plays an important role in the convergence of CD. Five selection rules are commonly used in practice — uniform random, non-uniform random, cyclic, random permuted cyclic and greedy selection rules.

- **Randomized CD (RCD):** RCD uniform randomly selects a coordinate from $\{1,2,\ldots,d\}$ to update in each iteration. It is the version that Nesterov analyzed in his seminal work (Nesterov, 2012). The convergence analysis of randomized CD is simple since the coordinates selected in different iterations are independent from each other. RCD has good theoretical properties, it is

provably faster (in terms of run time) than GD for problems that satisfy the coordinate-friendly structure. However, the implementation of RCD is cache unfriendly due to random access in memory.

- **Non-uniform random CD (Non-uniform RCD):** Non-uniform RCD is similar to RCD except that it adopts non-uniform sampling to select coordinate to update. A typical example of non-uniform sampling is the Lipschitz sampling, this sampling strategy sample a coordinate to update with probability that proportional to the coordinate's Lipschitz constant. This non-uniform sampling strategy was first studied by Nesterov (2012). Nesterov showed that the Lipschitz sampling could improve the convergence rate of RCD when the Lipschitz constants among coordinates are highly imbalanced.

- **Cyclic CD (CCD):** CCD is the oldest version of CD and can be traced back to the Gauss-Seidel algorithm for solving linear systems. CCD predefine a permutation π of $\{1,2,\ldots,d\}$ and selects coordinate $\pi_{((i-1) \bmod d+1)}$ in the ith iteration. CCD is easy to implement and more cache friendly than RCD because of the nature of sequential access. Though CCD usually converges at a similar rate as (or sometimes even faster than) RCD in practice, its convergence analysis is hard due to the inherent dependency among the coordinates selected. In fact, Nesterov stated that "it is almost impossible to estimate the rate of convergence" for CCD (Nesterov, 2012). The non-asymptotic convergence rate of CCD was later investigated by Saha and Tewari (2013), who obtained a non-asymptotic rate by imposing the *isotonicity* assumption, which is a strong assumption that may not hold in practice. Later Beck and Tetruashvili (2013) obtained the first global convergence rate of CCD without any further assumptions (and actually with simple and clean proofs). However, the rate that Beck and Tetruashvili obtained for CCD is much worse than RCD and this result seems to be inconsistent with empirical observations, Beck and Tetruashvili stated that the rate they derived may be improvable. This gap between theory and practice was filled by Sun and Ye (2021), who formally proved that there exist instances such that CCD is provably $\Omega(d^2)$ slower than RCD (Sun and Ye, 2021). However, their counterexample does not apply to random permuted cyclic CD (Lee and Wright, 2018), which we describe

next.

- **Random permuted cyclic CD (RPCD):** RPCD can be viewed as a midway point between RCD and CCD. It produces a random permutation at the beginning of each epoch and runs CCD with the permutation generated. The coordinates being selected are independent between epochs but dependent within each epoch. In terms of cache-friendliness, RPCD is better than RCD but worse than CCD. RPCD is easy to implement and performs similarly tp (or even better than) RCD empirically in many cases. Many efficient solvers for machine learning problems such as LIBLINEAR (Fan et al., 2008) are actually built on RPCD. The convergence rate of RPCD can be directly obtained from Beck and Tetruashvili's analyses for CCD (Beck and Tetruashvili, 2013). But again, this rate cannot explain the empirical performance of RPCD. Provably showing that RPCD outperforms RCD for least square problems turns out to be a hard mathematical problem: Recht and Ré (2012) abstracted the RPCD versus RCD problem into a formal conjecture called the "matrix AMGM" inequality (Recht and Ré, 2012)[1], which is the matrix version of the arithmetic-geometric mean inequality. The matrix AMGM inequality was recently proven to be false (Lai and Lim, 2020; Sa, 2020) and there exist instances such that RPCD is provably inferior to RCD.

- **Greedy CD (GCD):** GCD is also known as the Gauss-Southwell (GS) rule for solving linear system. Different from RCD, CCD and RPCD, GCD selects the coordinate from $\arg\max_{i\in[d]} |\nabla_i f(x^{(t)})|$, which has the largest marginal change at each iteration. It is worth noting that one may not be able to implement the greedy rule efficiently even if the problem satisfy the coordinate-friendly structure. That is, the run time of one epoch of GCD is usually much more expensive than one gradient descent step. However, exceptions do exist, for example the dual problem of kernel SVM (a quadratic programming problem with box constraints) whose kernel matrix has similar number of nonzeros among its columns, in these cases GCD can be imple-

[1] The original matrix AMGM inequality from Recht and Ré (2012) was originally motivated by the problem of random permuted SGD versus SGD, but its argument can be easily incorporated to CD.

mented efficiently[2] and one epoch of GCD has roughly the same computation cost as one gradient descent iteration. The state-of-the-art solvers for kernel SVM — LIBSVM (Chang and Lin, 2011) and SVMLight (Joachims, 1999) both are built on GCD.

The computational costs of the above selection rules are different. Random, cyclic and random permuted cyclic selection rule shares similar property. They can all benefit from the coordinate-friendly structure, that is, the runtime of d iterations with these three selection rules cost roughly the same time as one gradient descent step. For GCD, as mentioned above, the coordinate-friendly structure alone is not sufficient for it to have efficient implementation. To implement the greedy selection rule efficiently, stronger problem structures (such as the Hessian matrix need to be both row- and column-wise sparse, see a more comprehensive discussion in Nutini et al. (2015)) are required. The computational cost of non-uniform RCD is similar to GCD, and it depends on the sparsity structure of problems. For problems with a balanced sparsity structure (for example, the least-square problem with a balanced number of non-zeros for different columns), the time required to update different coordinate costs similar time. Therefore, d steps of non-uniform RCD cost similar time as one gradient descent step in this case. For problems with a highly unbalanced sparsity structure (for example, least-square problem with one column completely dense and other columns very sparse), the computational cost of RCD with the Lipschitz sampling could be very high; one coordinate update could be as expensive as one gradient descent step in the worst case.

2.2 Greedy coordinate descent (GCD)

GCD and GD are closely related. Both of these two methods can be interpreted as steepest descent method on the first-order Taylor expansion of the objective function, where GD is the steepest descent with respect to 2-norm and GCD is the steepest descent with respect to 1-norm (Boyd and Vandenberghe, 2004, §9.4). Different from the convergence rate of RCD, the convergence of GCD is more "similar" to GD and both of them enjoy dimension free convergence rate. For convex

[2] But require to store the kernel matrix in memory.

and smooth (but not necessarily strongly convex) objective function f, GCD has the convergence rate (see, for example, in work from Dhillon et al. (2011)[3])

$$f(x^{(t)}) - f^* \leq \frac{L_{\max} R_1(x^{(0)})}{t},$$

where $R_1(x^{(0)}) = \sup_{x \in \mathcal{X}^*} \|x - x^{(0)}\|_1^2$, $\mathcal{X}^*$ is the solution set, $L_{\max}$ is the maximum coordinate-wise smoothness parameter, see Chapter 1.3 for its definition. Note that the convergence rate of GD (Nesterov, 2004) is

$$f(x^{(t)}) - f^* \leq \frac{LR(x^{(0)})}{2t}.$$

where L is the smoothness parameter with respect to 2-norm and $R(x^{(0)})$ is the initial condition under 2-norm. Therefore convergence rates of both GCD and GD are independent from the problem size d, their only difference is how the smoothness parameter and distance to solution is measured, where the parameters of GCD depend on 1-norm and the parameters of GD depend on 2-norm.

For strongly convex objective, GCD has the convergence

$$f(x^{(t)}) - f^* \leq \left(1 - \frac{\mu_1}{L_{\max}}\right)^t \left(f(x^{(0)}) - f^*\right).$$

This rate is from the refined analysis of GCD from Nutini et al. (2015), where μ_1 is the strongly convex parameter with respect to 1-norm. Compared to the convergence of GD for strongly convex objective

$$f(x^{(t)}) - f^* \leq \left(1 - \frac{\mu_2}{L}\right)^t \left(f(x^{(0)}) - f^*\right),$$

which is built on the smoothness and strongly convex parameters (L and μ_2) with respect to 2-norm. By norm inequality, we know that $\mu_2/d \leq \mu_1 \leq \mu_2$ and $L_{\max} \leq L \leq dL_{\max}$. The above rates implies that GCD converges faster than RCD (in terms of iterations) since $\mu_1 \leq \mu_2$. Moreover, these rates demonstrate that GCD could be faster than GD when $\mu_1/L_{\max} > \mu_2/L$ and slower than GD otherwise.

[3]The original proof from Dhillon et al. (2011) used the smoothness parameter with respect to 1-norm. However, their proof can be modified to use the coordinate-wise smoothness parameter $L_{\max}$.

2.3 GCD for sparse optimization

We consider the composite problem

$$\min_{x\in\mathbb{R}^d} F(x) \coloneqq f(x) + g(x), \tag{2.1}$$

where f is a strongly convex and L-smooth function, and the regularizer g is a function that is separable and convex, but not necessarily smooth. Note that the dual formulation of the SVM problem with a bias term has an additional linear constraint and does not satisfy (2.1) because in this case g is not separable, the formulation in (2.1) only applies to dual SVM without the bias term. For sparsity inducing regularizer including 1-norm regularization and non-negative constraints, it is observed in practice that GCD could converge significantly faster than RCD. However, the refined analysis from Nutini et al. (2015) does not apply to composite problems and cannot explain why GCD is faster than RCD in this scenario. This gap was filled by Karimireddy et al. (2019), who proved that the bound

$$F(x^{(t)}) - F^* \leq \left(1 - \frac{\mu_1}{L_{\max}}\right)^{\lceil t/2 \rceil} (F(0) - F^*) \tag{2.2}$$

holds for 1-norm regularization and box-constraints by adding a post-processing step in each iteration. This result extends Nutini et al.'s refined analysis for composite problems and illustrates the theoretical advantage of GCD over RCD.

A weakness of the above result is that the rate (2.2) suggests GCD applied to strongly convex composite problems, with either 1-norm regularization or non-negative constraints, has the same rate as it does for problems without regularizers — i.e., minimizing only $f(x)$ instead of the sum $f(x) + g(x)$. However, GCD applied to composite problems with sparsity inducing regularization is usually significantly faster than its non-regularized counterpart and empirically exhibit screening ability, that is the greedy selection rule can mostly focus on coordinates that are nonzero at solution. Therefore an improved convergence analysis is needed to explain this phenomenon. We present our work on GCD for sparse optimization in this section to fill this gap between theory and practice.

Algorithm 1 A generic template for GCD

Input: functions f and $g_i \; \forall i \in [d]$.
$W_0 = \emptyset$
$x^{(0)} = 0$
for $t = 0, 1, 2, \ldots$ **do**
    Coordinate selection: select i according to the GS-s rule
    Gradient step: $x^{(t+\frac{1}{2})} = x^{(t)} - (1/L_i)\nabla_i f(x^{(t)})\boldsymbol{e}_i$
    Prox step: $x^{(t+1)} = \text{prox}_{(1/L_i)g_i}\left(x^{(t+\frac{1}{2})}\right)$
    Post-processing; see (2.4)
    Update working set: $W_{t+1} = W_t \cup \{i\}$; see definition 2.3.1
end for

2.3.1 Problem setup

Formally, we make the following assumptions on f and g in problem (2.1):

- f is L_i coordinate-wise smooth $\forall i \in [d]$, and we let $L_{\max} = \max_{i \in [d]} L_i$.
- f is L_∞-smooth with respect to the ∞-norm.
- f is μ_p strongly convex with respect to the p-norm, $p \in \{1, 2\}$.
- $g = \lambda \|\cdot\|_1$ or $g = \delta_{\geq 0}$, where $\delta_{\geq 0}$ is the indicator function on the set $\{x_i \geq 0 \mid x \in \mathbb{R}^d\}$ that vanishes on the nonnegative orthant, and is $+\infty$ otherwise. (It is an open problem if our analysis could be extended to general separable regularizers that are nonsmooth at 0 for all $i \in [d]$).

We consider proximal CD algorithm with the GS-s selection rule (Nutini et al., 2015):

Selection rule 2.1 (GS-s rule)**.** *Select coordinate* $i \in \arg\max_{j \in [d]} Q_j(x^{(t)})$*, where*

$$Q_j(x) = \min_{s \in \partial g_j(x_j)} |\nabla_j f(x) + s|. \tag{2.3}$$

The GS-s rule that appeared in Nutini et al. (2015) is a natural extension of the vanilla greedy selection rule to composite problems, it has been widely used in the literature (Bertsekas, 1999; Li and Osher, 2009; Shevade and Keerthi, 2003; Wu and Lange, 2008). GS-r, GS-q rules are other variants that are commonly being

used, see the discussion from Nutini et al. (2015) and the references therein. Here we focus on the GS-s rule. The detailed algorithm of GCD for composite problem is shown in Algorithm 1, where the post-processing step is defined as

$$x_i^{(t+1)} := 0 \quad \text{if} \quad x_i^{(t+1)} x_i^{(t)} < 0, \tag{2.4}$$

and the proximal operator is defined as

$$\text{prox}_{\eta g_i}(x) := \underset{u \in \mathbb{R}^d}{\arg\min} \left\{ \frac{1}{2}\|u - x\|_2^2 + \eta g_i(u_i) \right\}.$$

We also make the following definitions to help our analysis.

Definition 2.3.1. The *working set* W_t is the set of indices selected up to and including iteration t. Define also $W := \bigcup_{t=0}^{\infty} W_t$ as the overall working set.

Definition 2.3.2. The *support* of a vector x is the set $\text{supp}(x) = \{ i \mid x_i \neq 0 \}$.

2.3.2 Analysis

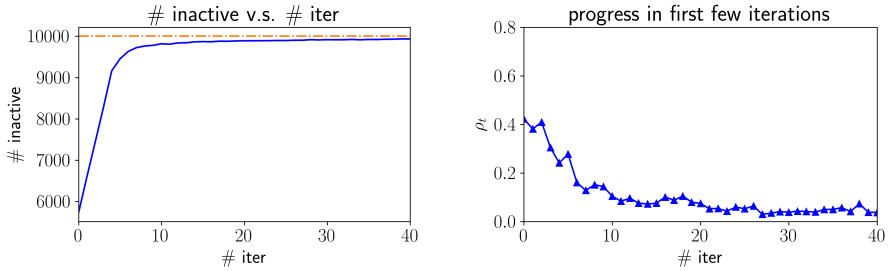
We now study the screening ability of GCD and develop a bound on the size of the working set W, i.e., focus on coordinates that are nonzero at solution. We require the following quantity, often used in sparsity pattern identification:

$$\delta_i := \min \{ -\nabla_i f(x^*) - \ell_i, u_i + \nabla_i f(x^*) \}, \tag{2.5}$$

where $\partial g_i(x_i^*) = [\ell_i, u_i]$; see Hare and Lewis (2007); Lewis and Wright (2011); Nutini et al. (2017); Sun et al. (2019). The constant δ_i is closely related to the distance to the relative interior of the sparse manifold, and is an important quantity in sparse manifold identification (Lewis and Wright, 2011). Optimality conditions for (2.1) imply that $\delta_i = 0$ if $x_i^* \neq 0$, and $\delta_i \geq 0$ if $x_i^* = 0$. Because x^* is unique (by strong convexity), these quantities are problem-specific and algorithmically invariant. The definition in (2.5) leads to the following identification result.

Lemma 2.3.1 (Nutini et al., 2017)**.** *If for some $t > 0$ and some coordinate $i \in [d]$, $x_i^* = 0$ and $x^{(t)}$ satisfies*

$$|\nabla_i f(x^{(t)}) - \nabla_i f(x^*)| \leq \delta_i \quad \text{and} \quad x_i^{(t)} = 0,$$



(a) The evolution of the number of inactive variables.

(b) Objective progress, where ρ_t is defined in (2.8).

Figure 2.1: Exploratory investigations.

then after one coordinate proximal gradient step $x_i^{(t+1)} = 0$.

This lemma suggests that if $\nabla_i f(x^{(t)})$ is close to $\nabla_i f(x^*)$ and $x_i^* = 0$, then the ith entry of $x^{(t)}$ will be correctly identified as 0.

Numerical motivation

We present some numerical observations to motivate our analysis. Consider a LASSO problem from random synthetic data. Define

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2 \qquad \text{and} \qquad g(x) = \lambda \|x\|_1, \tag{2.6}$$

where $A \in \mathbb{R}^{50\times 10^4}$ and $b := Ax^\sharp + \varepsilon$. The elements A_{ij}, ε_i, and non-zeros in the solution $x^\sharp$ are distributed as standard Gaussians. We randomly select 10 elements from $x^\sharp$ to be non-zeros and set λ to be 2.

Figures 2.1a and 2.1b show the evolution of the number of "inactive" variables and objective progress for Algorithm 1.

In Figure 2.1a, we define

$$\#\text{inactive} := \sum_{i=1}^{d} \mathbf{1}\left\{ |\nabla_i f(x^{(t)}) - \nabla_i f(x^*)| \le \delta_i \right\}. \tag{2.7}$$

According to Lemma 2.3.1, this quantity measures how many variables are staying "inactive", i.e., will not move away from 0 in the next iteration. From Figure 2.1a,

we observe that most variables are initially incorrectly labeled as "active", i.e., $|\nabla_i f(x^{(t)}) - \nabla_i f(x^*)| > \delta_i$, but a large number of them quickly switch to "inactive" within first few iterations.

In Figure 2.1b, we illustrate the objective progress at each step by plotting ρ_t, defined to satisfy

$$F(x^{(t+1)}) - F^* = (1-\rho_t)\left(F(x^{(t)}) - F^*\right). \tag{2.8}$$

These experiments illustrate the fact that the initial convergence of GCD, which, for sparse solutions, may be sufficient to quickly identify the few non-zeros. From this experiment we observe that

- GCD converges fast initially and $\nabla f(x^{(t)})$ quickly approaches $\nabla f(x^*)$ when $x^{(t)}$ is still sparse; and
- before $\text{supp}(x^{(t)})$ has grown significantly, the coordinates i where $x_i^* = 0$ have mostly become inactive, and thus future coordinates that enter W are constrained to $\text{supp}(x^{(t)})$.

We then rigorously characterize these observations. Before proceeding to our results, we first introduce some needed concepts.

Definition 2.3.3. The function f is $\mu_p^{(\tau)}$-strongly convex with respect to $\|\cdot\|_p$ and sparse vectors if $\forall x, y \in \mathbb{R}^d$ such that whenever $|\text{supp}(x) \cup \text{supp}(y)| \leq \tau$,

$$f(x) \geq f(y) + \langle \nabla f(y), x-y \rangle + \frac{\mu_p^{(\tau)}}{2}\|x-y\|_p^2,$$

where $p \in \{1,2\}$.

The concept of strongly convexity with respect to sparse vectors is not new. For example, see the idea of restricted strongly convex (Negahban and Wainwright, 2012) in the literature.

It can be easily verified that $\mu_1^{(\tau)}$ and $\mu_2^{(\tau)}$ satisfy the following conditions:

$$\begin{aligned}
&\mu_1 = \mu_1^{(d)} \leq \mu_1^{(d-1)} \leq \cdots \leq \mu_1^{(1)}, \\
&\mu_2 = \mu_2^{(d)} \leq \mu_2^{(d-1)} \leq \cdots \leq \mu_2^{(1)}, \\
&\mu_2^{(\tau)}/\tau \leq \mu_1^{(\tau)} \leq \mu_2^{(\tau)} \quad \forall \tau \in [d].
\end{aligned} \tag{2.9}$$

Next, we present a formal analysis to answer why GCD may converge fast initially, and give a bound on the size of the working set W.

Theorem 2.3.1 (Fast initial convergence)**.** *Let $\tau = |\mathrm{supp}(x^*)|$ and let $\{x^{(i)}\}_{i=1}^{\infty}$ be the iterates generated by Algorithm 1 with the GS-s rule (selection rule 2.1). Then for any $t < d - \tau$,*

$$F(x^{(t)}) - F^* \leq \prod_{i=1}^{\lceil t/2 \rceil} \left(1 - \frac{\mu_1^{(\tau+i-1)}}{L_{\max}}\right) (F(0) - F^*) \tag{2.10}$$

$$\leq \prod_{i=1}^{\lceil t/2 \rceil} \left(1 - \frac{\mu_2}{(\tau+i-1)L_{\max}}\right) (F(0) - F^*). \tag{2.11}$$

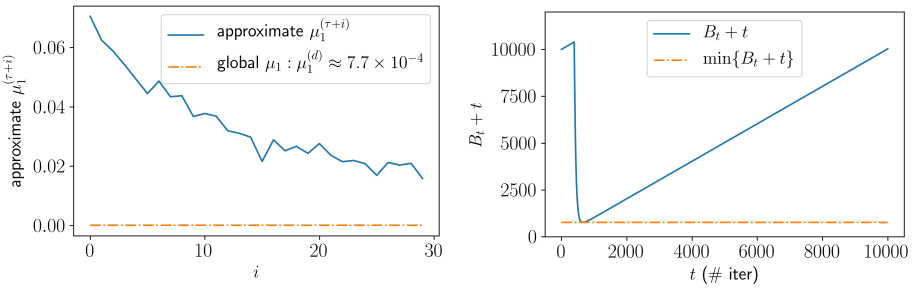
Proof is placed in Appendix. The bound in (2.11) follows from (2.9). In Theorem 2.3.1 we show two different bounds, which allow us to draw comparisons to existing results, below.

Bound (2.11). Nesterov (2012) and Richtárik and Takác (2014) established that RCD exhibits the rate

$$\mathbb{E}\left[F(x^{(t)}) - F^*\right] \leq \left(1 - \frac{\mu_2}{dL_{\max}}\right) (F(0) - F^*).$$

Compared to (2.11), we see that the dimension d is replaced by the quantity $(\tau + i - 1)$, which, if $\tau = |\mathrm{supp}(x^*)|$ is small and we are in the first few iterations, may be much smaller than d. This reflects the fast initial convergence often observed in practice; cf. Figure 2.1b.

Bound (2.10). Nutini et al. (2015) and Karimireddy et al. (2019) established for the GCD method the linear convergence rate described by (2.2). Compared to (2.10),



(a) The trend of approximate $\mu_1^{(\tau+i)}$, where $\tau = 10$ in this example.

(b) The curve illustrates the upper bound of $|W|$ in Theorem 2.3.2.

Figure 2.2: Illustrations for Theorem 2.3.1 and 2.3.2

we see that μ_1 is replaced with the quantity $\mu_1^{(\tau+i-1)}$, which is potentially much larger in the early stages (i is small), particularly if τ is small (x^* is sparse). This is confirmed by Figure 2.2a, which shows how in practice this quantity can be much larger than μ_1. In particular, when τ and i are small, $\mu_2/d \ll \mu_1^{(\tau+i-1)}$ and the convergence rate for GCD is initially significantly faster than RCD, even in the worst case.

The rate we derived here is based on two important ingredients: zero initialization and sparse solution. The screening ability of GCD does not hold without either of these properties.

To better understand the effect of the quantity $\mu_1^{(\tau+i-1)}$ on the convergence rate, we conduct a simulation using the LASSO problem described in eq. (2.6). The term $\mu_1^{(\tau+i-1)}$ is hard to compute in general, and thus here we set $\tau = 10$, and for each $i \in \{1,2,\ldots,30\}$ we generate 10^3 random $(\tau+i-1)$-sparse vectors and approximate $\mu_1^{(\tau+i-1)}$ as the minimum of $\|Ax\|_2^2/\|x\|_1^2$ over the sample vectors. The plot of approximate $\mu_1^{(\tau+i-1)}$ against i is shown in Figure 2.2a, and its pattern clearly supports our previous argument.

To develop an upper bound for the size of working set, we define the error measure

$$p_\delta(\alpha) = \sum_{i=1}^{d} \mathbf{1}\{\alpha \le \delta_i\},$$

which we use to quantify the number of inactive elements in the iterates, as in eq. (2.7).

Theorem 2.3.2 (Working set bound)**.** *Let $\tau = |\text{supp}(x^*)|$ and $\{x^{(i)}\}_{i=1}^{\infty}$ be the iterates generated by Algorithm 1 with the GS-s rule (selection rule 2.1). Then*

$$|W| \leq \min_{t \in [d]} \{B_t + t\}, \tag{2.12}$$

where

$$B_t := d + \tau - p_\delta \left(L_\infty \sup_{i \geq t} \left\{ \|x^{(i)} - x^*\|_1 \right\} \right).$$

The term L_∞ is the smoothness parameter with respect to infinity norm. The detailed proof is placed in Appendix. We give a short interpretation to better understand this bound. Note that B_t is a decreasing function of t. Thus, $|W|$ is bounded by the infimum of the sum of a decreasing and increasing function. (See Figure 2.2b.) Theorem 2.3.2 implies that if $x^{(t)}$ converges quickly to x^* (i.e., with $t \ll d$), then the bound (2.12) will be far less than d.

Again, consider the synthetic LASSO problem ($A \in \mathbb{R}^{50 \times 10^4}, \lambda = 2$) as a concrete example to illustrate this bound. In this example, the curve of $B_t + t$ is shown in Figure 2.2b and the infimum of $B_t + t$ is about 1000 in this case. This experiment demonstrates that the bound we derived in Theorem 2.3.2 is non-trivial, especially for problems whose d is large.

Furthermore, we can use Theorem 2.3.1 and 2.3.2 to derive an alternative bound that depends only on the constant $\mu_1^{(\tau+i)}, i \in [d-\tau]$, instead of the iterates $x^{(i)}$'s.

Corollary 2.3.1. *Let $\tau = |\text{supp}(x^*)|$ and let $\{x^{(i)}\}_{i=1}^{\infty}$ be the sequence of iterates generated by Algorithm 1 with the GS-s rule (selection rule 2.1). Then B_t in bound* (2.12) *can be replaced by*

$$B_t := d + \tau - p_\delta \left(\left[\frac{2L_\infty^2}{\mu_1} \prod_{i=0}^{t-1} \left(1 - \frac{\mu_1^{(\tau+i)}}{L_{\max}} \right) R \right]^{1/2} \right),$$

where $R = F(0) - F^*$ is the initial objective gap.

2.3.3 Improved selection rule

Our analysis in previous section provides an upper bound of $|W|$. In this section, we propose a variant of the GS-s rule that could favour an even smaller working set.

The resulting algorithm, which we call Δ-GCD, is Algorithm 1 with the following modified selection rule.

Selection rule 2.2 (Δ-GS-s rule)**.** *Given the fixed parameter* $\Delta \in (0,1]$*, select coordinate*

$$i \in \begin{cases} \underset{i\in[d]}{\arg\max} \quad Q_i(x^{(t)}), & \text{if} \quad \Delta \max\limits_{i\in[d]} Q_i(x^{(t)})^2 \geq \max\limits_{i\in W_t} Q_i(x^{(t)})^2 \\ \underset{i\in W_t}{\arg\max} \quad Q_i(x^{(t)}), & \text{if} \quad \Delta \max\limits_{i\in[d]} Q_i(x^{(t)})^2 < \max\limits_{i\in W_t} Q_i(x^{(t)})^2 \end{cases}$$

where W_t *denotes the set of indices accrued thus far and* Q_i *is defined by* (2.3)*.*

Note that when $\Delta = 1$, the Δ-GS-s and GS-s rules are equivalent. Intuitively, the Δ-GS-s rule, with small Δ, is more likely to focus on the current working set; on the other hand, a large Δ encourages the algorithm to include unexplored coordinates and expand the current working set. Thus Δ controls the trade-off between the size of working set and the progress we can make when staying in the current working set. This is similar to the exploration/exploitation trade-off in the context of online learning (Auer et al., 1995).

Theorem 2.3.3. *Let* $\{x^{(i)}\}_{i=1}^{\infty}$ *be the iterates generated by Algorithm 1 with the* Δ*-GS-s rule (selection rule 2.2) and let* W_Δ *be the final working set. Then for all* $t > 0$*,*

$$F(x^{(t)}) - F^* \leq \left(1 - \frac{\Delta \mu_1^{(|W_\Delta|)}}{L_{\max}}\right)^{\lceil t/2 \rceil} (F(0) - F^*) \tag{2.13}$$

$$\leq \left(1 - \frac{\Delta \mu_2}{|W_\Delta| L_{\max}}\right)^{\lceil t/2 \rceil} (F(0) - F^*)\,. \tag{2.14}$$

Theorem 2.3.3 explicitly described the trade-off between the convergence rate and the size of working set. Similar to Theorem 2.3.1, we provide two bounds for easier interpretation: eq. (2.13) can be viewed as a refinement of the strong convexity parameter in Karimireddy et al. (2019) and Nutini et al. (2015); and eq. (2.14) where the variable dimension dependency that appears in Nesterov (2012) and Richtárik and Takác (2014) is replaced by the size of the final working set. The Δ-GCD variant is expected to outperform standard GCD when the latter has

a comparatively large working set, and Δ-GCD can reduce the size of working set with an appropriate value of Δ.

To better understand the relationship between W_Δ and Δ, we present an description of W_Δ as $\Delta \to 0$. First, consider the standard GCD algorithm where at each iteration we additionally minimize the objective over the current working set, i.e., the next iterate is obtained as

$$x^{(t+1)} := \underset{\text{supp}(x) \subseteq W_{t+1}}{\arg\min} \; f(x) + g(x), \tag{2.15}$$

where all variables not in the working set are held fixed at 0. The algorithm terminates when the iterate $x^{(t+1)}$ is optimal for (2.15). The resulting method is known as the *totally corrective greedy algorithm*, which is closely related to orthogonal matching pursuit for sparse least squares; see works from Pati et al. (1993); Davis et al. (1997); and Foucart and Rauhut (2013). Note that when $\Delta \to 0$, the result selection rule is close to the selection rule used in Algorithm 1 from Shevade and Keerthi (2003). We denote the final working set from this scheme as $W^\sharp$.

Next, note that as $\Delta \to 0$, the Δ-GS-s selection rule tends to select indices from the current working set, and thus the Δ-GS algorithm converges to a solution of (2.15). However, when the Δ-GCD iterate is close to the exact minimizer, the Δ-GS-s rule must eventually expand the working set. As the following result shows, W_Δ converges to $W^\sharp$.

Theorem 2.3.4. *Let $k = |W^\sharp|$ and let $\{x^{(t)}\}_{t=0}^k$ be the iterates generated by the totally corrective greedy algorithm. Assume that* $\arg\max_{i \in [d]} Q_i(x^{(t)})$ *are singletons for $t = 0, 1, \ldots, k-1$ and $\delta_i > 0 \; \forall x_i^* = 0$. Then $\exists \; \varepsilon > 0$, such that*

$$W_\Delta = W^\sharp$$

for any $\Delta < \varepsilon$.

If the totally corrective greedy algorithm can yield a small working set, then we expect that a sufficiently small value of Δ would also yield a small working set (but could probably slow down the convergence according to Theorem 2.3.3). Hence,

Table 2.1: Properties of the experimental data. Here, d denotes the number of features and n denotes the number of samples.

Datasets	colon	leukemia	make_circle	ijcnn1
d	2,000	7,129	2	22
n	62	72	1,000	35,000

our new algorithm Δ-GCD can be viewed as a flexible greedy algorithm between the two extreme cases — standard GS-GCD and the totally corrective greedy algorithm.

2.3.4 Numerical experiments

We conduct experiments on both real world data and synthetic data to illustrate the importance of zero initialization and exam the effectiveness of our proposed Δ-GCD.

The statistics of our experimental data are shown in Table 2.1, where the datasets colon, leukemia, and ijcnn1 were obtained from the LIBSVM website[4] (Chang and Lin, 2011). The make_circle dataset were generated from the scikit-learn package (Pedregosa et al., 2011). We solve the LASSO problem over the colon and leukemia datasets, and the dual RBF kernel SVM over the make_circle and ijcnn1 datasets. For ijcnn1, we follow the parameter settings described by Hsieh et al. (2014), and thus set $\gamma = 2$ and $C = 32$, where γ is the free parameter in the RBF kernel and $1/C$ is the hinge-loss weight parameter. All experiments are conducted on a machine with 4 CPUs and 16GB memory.

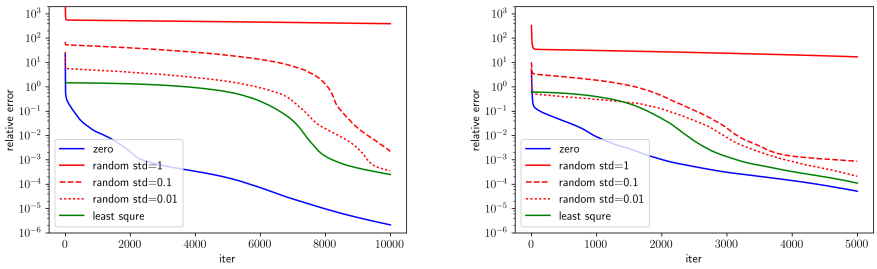
Code

The code to reproduce our experimental results is publicly available at https://github.com/fanghgit/Greed_Meets_Sparsity.

Zero v.s. other initializations

We compare the convergence of standard GCD for solving LASSO problems over different initialization strategies, we include the following initialization strategies

[4] https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/



(a) LASSO, data: leukemia, $\lambda = 0.1$. **(b)** LASSO, data: colon, $\lambda = 0.1$.

Figure 2.3: Comparison between different kinds of initialization

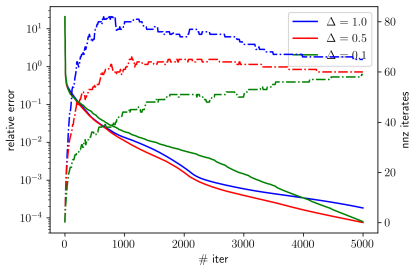
into our comparisons:

- Zero initialization: $x^{(0)} = 0$.
- Random initialization: $x^{(0)}$ is generated from Gaussian distributions $\mathcal{N}(0, \sigma I_d)$, for $\sigma \in \{1, 0.1, 0.01\}$.
- Least-squares initialization: $x^{(0)} = (A^T A + \lambda I_d)^{-1} A^T b$. This initialization starts the method at a low objective value, but is not sparse.

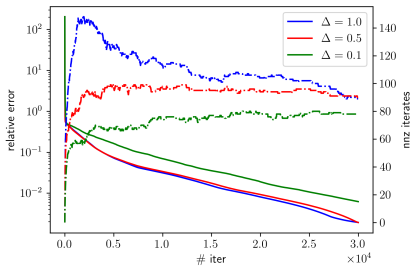
In Figure 2.3, we can see that zero initialization clearly outperforms other initialization strategies, and random initialization tends to be the worst. In particular, GCD with zero initialization is able to get close to a solution even before one pass of all coordinates. On the other hand, although GCD with least-squares initialization has a better initial objective value than zero initialization, it suffers from slow convergence and requires at least a full pass of all coordinates before reaching the same low error, which is consistent with our intuition. Random initialization with different standard deviations also vary in their performance and random initialization with smaller variance tends to converge faster; however they are still outperformed by the zero initialization for the same reasons as the least-squares initialization.

Evaluation of Δ-GCD

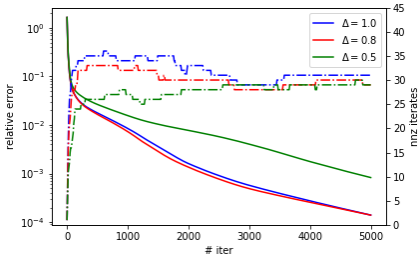
We evaluate the proposed Δ-GCD algorithm on LASSO, 1-norm regularized logistic regression, and kernel SVM problems. As shown in Figure 2.4, the value of Δ



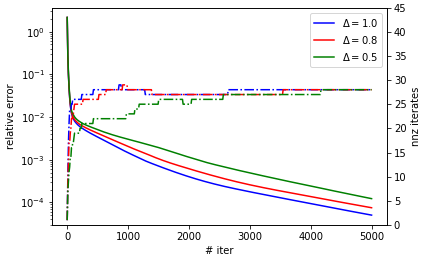
(a) LASSO, data: leukemia, $\lambda = 0.1$. Solid line is objective value, dashed line is # non-zeros in $x^{(t)}$.



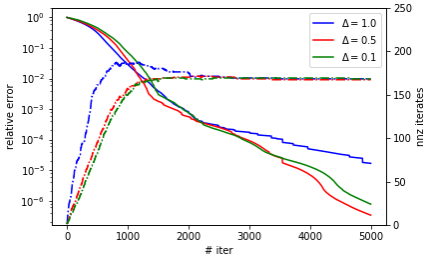
(b) LASSO, data: leukemia, $\lambda = 0.01$. Solid line is objective value, dashed line is # non-zeros in $x^{(t)}$.



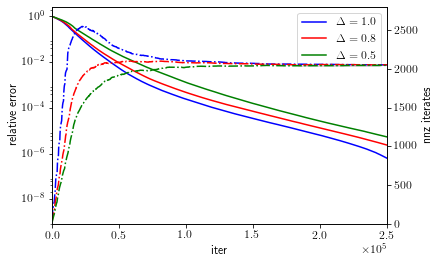
(c) L_1-regularized logistic regression, data: colon, $\lambda = 0.1$.



(d) L_1-regularized logistic regression, data: leukemia, $\lambda = 0.1$.



(e) kernel SVM, data: make_circle, $C = 10, \gamma = 0.5$.



(f) kernel SVM, data: ijcnn1, $C = 32, \gamma = 2$.

Figure 2.4: Compare Δ-GCD with different choices of Δ.

has a clear impact on the size of the working set, where smaller values of Δ tend to promote sparser iterates for all the test problems. This trend is more obvious when the underlying solution is less sparse, as shown in Figures 2.4b and 2.4f. This is because vanilla GCD produces a working set that is much larger than needed. Sometimes this stronger screening ability of a smaller Δ can lead to slightly faster

convergence compared to standard GCD (i.e., $\Delta = 1$) as shown in Figures 2.4a and 2.4b. A by-product of Δ-GCD is early identification of the final sparsity pattern, which can be leveraged in two-stage methods (Bertsekas, 1976; Daniilidis et al., 2009; Ko et al., 1994; Wright, 2012). However, the acceleration functionality of Δ-GCD is not present for all test problems, since vanilla GCD already has a strong screening ability for constraining the size of the working set.

2.3.5 Discussion

By bringing techniques from sparsity pattern identification and convergence analysis of GCD, we formally analyze the screening ability of GCD and explicitly answered why GCD is usually fast for sparse optimization. We also propose an improved selection rule with a stronger ability to encourage sparse iterates and connect to existing algorithms.

For future work, it would be interesting to generalize our analysis and relax the strong-convex assumption on the function f. In particular, one could consider problems where x^* may not be unique (but $\text{supp}(x^*)$ may be). The core of our analysis relies on understanding the convergence of the iterates themselves, and not just the function values. Thus, the challenge in generalizing our analysis to more general smooth objectives requires a different proof technique. Another direction is to tighten the working set bound in Theorem 2.3.2. The bound illustrated in Figure 2.2b is still about 10 times worse than the actual size of the working set, and for small value of λ, the actual working set become larger and our bound can be trivially larger than d. Our analysis also require the regularizer to be 1-norm or nonnegative constraint, it is an open problem to extend our analysis to general regularizer that are non-smooth at origin.

2.4 Gap-based safe-screening rules for atomic-norm regularized problem

The safe-screening rules, originally proposed by Ghaoui et al. (2012), generally refer to approaches that correctly identify the coordinates that can be safely discarded without changing the solution or hindering the optimization process. It has been successfully applied to coordinate optimization algorithms to reduce the overall computational effort and achieved promising result for 1-norm regularized problem. Due to the empirical success of safe-screening rule on sparse optimization, various screening strategies (Atamtūrk and Gómez, 2020; Bao et al., 2020; Bonnefoy et al., 2015; Kuang et al., 2017; Liu et al., 2014b; Ndiaye et al., 2017; Raj et al., 2015; Wang et al., 2014, 2013; Xiang et al., 2017; Zhang et al., 2017) have been proposed for difference tasks in recent years. In particular, Ndiaye et al. (2017) proposed an effective and elegant screening framework based on duality gap called the *gap safe-screening rule*. The gap-based safe-screening rule has been shown to be useful for 1- and group-norm regularized problems.

2.4.1 From coordinates to atoms

Sparse optimization problems are characterized by solutions that are sparse vectors, i.e., few nonzero entries. The idea of vector sparsity can be generalized to "atomic" sparsity. That is, the problem attains a solution that is sparse with respect to some atomic set $\mathcal{A}$, where $\mathcal{A}$ could potentially contain infinite number of elements. Formally, for a given atomic set $\mathcal{A} \subseteq \mathbb{R}^d$, an optimal solution x^* can be represented as

$$x^* = \sum_{a \in \mathcal{A}} c_a a, \quad c_a \geq 0, \tag{2.16}$$

where only a small number of coefficients c_a are nonzero. The archetypal example is a sparse vector, which is sparse with respect to the set of signed canonical unit vectors $\mathcal{A} = \{\pm \boldsymbol{e}_1, \ldots, \pm \boldsymbol{e}_d\}$. 1-norm regularization is the standard approach to produce sparse solution. The atoms that participate nontrivially in the decomposition (2.16) represent latent structure in the solution. The notion of atomic sparsity is prevalent in machine learning (Argyriou et al., 2007; Meinshausen and Bühlmann, 2006; Tibshirani, 1996; Yuan and Lin, 2006) and signal processing (Candes et al., 2015),

and has been formalized in the context of inverse problems by Chandrasekaran et al. (2012).

The gap-based safe-screening rules have been shown to be effective for 1-norm regularized problems. In the following sections, we will extend the gap-based safe-screening rule to general atomic sets and investigate whether it is possible to use the generalized gap-based safe-screening rule to save computation for nuclear-norm regularized problems.

2.4.2 Some technical tools

We introduce in this section the basic tools of convex analysis and atomic sparsity that serve as the cornerstone of our analysis, most of the material for sections 2.4.1 to 2.4.3 is a summary of the framework described by Fan et al. (2020). We make the blanket assumption that the atomic set $\mathcal{A} \subseteq \mathbb{R}^d$ is compact, and that the origin is contained in its convex hull (we do not assume that $\mathcal{A}$ is convex). The gauge function to the set $\mathcal{A}$ measure the magnitude of a function relative to that set.

Definition 2.4.1 (Gauge function)**.** The gauge function with respect to $\mathcal{A}$ is defined as

$$\gamma_{\mathcal{A}}(x) = \inf \left\{ \sum_{a \in \mathcal{A}} c_a \;\middle|\; x = \sum_{a \in \mathcal{A}} c_a a,\ c_a \geq 0, \forall a \in \mathcal{A} \right\}. \tag{2.17}$$

The gauge function is always convex, nonnegative, and positively homogeneous. However, it is not necessarily a norm because it may not be symmetric (unless $\mathcal{A}$ is centrosymmetric) and may vanish or take infinite value at points off the origin (unless the origin is in the relative interior of $\mathcal{A}$). Definition 2.4.1 makes explicit the role of a gauge function as a convex penalty for atomic sparsity. The support of a vector x describes the atoms that contribute positively in the decomposition described by (2.17).

Definition 2.4.2 (Atomic support)**.** The atomic support for a point $x \in \mathbb{R}^d$ with respect to the set $\mathcal{A}$ is defined to be the set $\mathcal{S}_{\mathcal{A}}(x)$ that satisfies

$$\gamma_{\mathcal{A}}(x) = \sum_{a \in \mathcal{S}_{\mathcal{A}}(x)} c_a, \qquad x = \sum_{a \in \mathcal{S}_{\mathcal{A}}(x)} c_a a, \quad \text{and} \quad c_a > 0 \ \ \forall a \in \mathcal{S}_{\mathcal{A}}(x).$$

The atomic set of signed 1-hot unit vectors $\mathcal{A} = \{\pm \boldsymbol{e}_i \mid i = 1, 2, \ldots, d\}$, for

Atomic sparsity	$\mathcal{A}$	$\gamma_{\mathcal{A}}(x)$	$\mathcal{S}_{\mathcal{A}}(x)$	$\sigma_{\mathcal{A}}(z)$
non-negative	$\mathrm{cone}(\{\boldsymbol{e}_1,\ldots,\boldsymbol{e}_d\})$	$\delta_{\geq 0}$	$\mathrm{cone}(\{\boldsymbol{e}_i \mid x_i > 0\})$	$\delta_{\leq 0}$
element-wise	$\{\pm\boldsymbol{e}_1,\ldots,\pm\boldsymbol{e}_d\}$	$\|\cdot\|_1$	$\{\mathrm{sign}(x_i)\boldsymbol{e}_i \mid x_i \neq 0\}$	$\|\cdot\|_\infty$
low rank	$\{uv^T \mid \|u\|_2 = \|v\|_2 = 1\}$	nuclear-norm	singular vectors of x	spectral norm
PSD & low rank	$\{uu^T \mid \|u\|_2 = 1\}$	$\mathrm{tr}+\delta_{\succeq 0}$	eigenvectors of x	$\max\{\lambda_{\max}, 0\}$

Table 2.2: Commonly used sets atom sets and the corresponding gauge and support functions. The indicator function $\delta_{\mathcal{C}}(x)$ is zero if x is in the set $\mathcal{C}$ and $+\infty$ otherwise. The commonly used group-norm is also an atomic norm; see Fan et al. (2020, Example 4.7).

example, the support $\mathcal{S}_{\mathcal{A}}(x)$ coincides with the nonzero elements of x with the corresponding sign. The support function, defined below, is dual to the gauge function, and provides a key tool for identifying atoms associated with the support of a vector.

Definition 2.4.3 (Exposed faces and ε-exposed faces)**.** The exposed face and ε-exposed face for a point $z \in \mathbb{R}^d$ with respect to the set $\mathcal{A}$ is defined by

$$\mathcal{F}_{\mathcal{A}}(z) = \{a \in \mathcal{A} \mid \langle a, z\rangle = \sigma_{\mathcal{A}}(z)\}, \quad \mathcal{F}_{\mathcal{A}}(z,\varepsilon) = \{a \in \mathcal{A} \mid \langle a, z\rangle \geq \sigma_{\mathcal{A}}(z) - \varepsilon\} \tag{2.18}$$

where $\sigma_{\mathcal{A}}(z) = \sup_{a\in\mathcal{A}}\langle a, z\rangle$ is the support function with respect to $\mathcal{A}$.

Note that when $\varepsilon = 0$, the ε-exposed face coincides with the exposed face. We list in Table 2.2 some commonly used atomic sets, their corresponding gauge and support functions, and atomic supports.

2.4.3 Problem setup

We consider the gauge regularized problem

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad f(Mx) \qquad \text{subject to} \qquad \gamma_{\mathcal{A}}(x) \leq \tau, \tag{P}$$

where f is an L-smooth and convex function, $M : \mathbb{R}^d \to \mathbb{R}^m$ is a linear operator and $\tau > 0$ is a given constant can control the sparsity level. The gauge function constraint in (P) can generate a solution that is sparse with respect to the atomic set $\mathcal{A}$, and a wide range of application including LASSO and matrix completion can be formulated as (P) (with an appropriate choice of the atomic set).

The dual problem of (P) is

$$\underset{y\in\mathbb{R}^m}{\text{minimize}} \quad d(y) := f^*(y) + \tau\sigma_{\mathcal{A}}(M^*y), \tag{D}$$

where $f^*(y) = \sup_{w\in\mathbb{R}^m} \langle y, w\rangle - f(w)$ is the convex conjugate function of f, and $M^* : \mathbb{R}^m \to \mathbb{R}^d$ is the adjoint operator of M, which satisfies $\langle Mx, y\rangle = \langle x, M^*y\rangle$ for all $x \in \mathbb{R}^d$ and $y \in \mathbb{R}^m$. The dual problem plays an important role in safe-screening rules. Most safe-screening rules including the very first one proposed by Ghaoui et al. (2012) rely on dual variables to discard redundant features.

2.4.4 The gap-based safe-screening rule

Next, we extend the gap-based safe-screening rule from Ndiaye et al. (2017) to general atomic sets with the language of atomic sparsity. Note that a similar analysis can be found in a recent work from Sun and Bach (2020).

Given a feasible solution x of (P), safe-screening rule aims to infer the atoms with nonzero coefficients at solution based on x. We know that $\mathcal{S}_{\mathcal{A}}(x)$ could be different from $\mathcal{S}_{\mathcal{A}}(x^*)$ even for feasible solution x that is arbitrarily close to x^*. Fortunately, the following support-face relationship from Fan et al. (2020) allow us to constrain $\mathcal{S}_{\mathcal{A}}(x^*)$ by constructing a superset of $\mathcal{F}_{\mathcal{A}}(M^*y^*)$ from the dual problem.

Lemma 2.4.1 (Fan et al., 2020, Proposition 4.5 and Theorem 5.1)**.** *Let x^* and y^* be optimal primal-dual solutions for problem* (P) *and* (D)*. Then*

$$\mathcal{S}_{\mathcal{A}}(x^*) \subseteq \mathcal{F}_{\mathcal{A}}(M^*y^*).$$

Define the atomic operator norm by $\|M\|_{\mathcal{A}} := \max_{a\in\mathcal{A}} \|Ma\|_2$, we can get the following result based on Lemma 2.4.1.

Theorem 2.4.1 (Generalized gap-based safe-screening rule)**.** *Given a primal feasible point x and a dual feasible point y, we have*

$$\mathcal{S}_{\mathcal{A}}(x^*) \subseteq \mathcal{F}_{\mathcal{A}}(M^*y, \varepsilon), \tag{2.19}$$

where $\varepsilon = 2\|M\|_{\mathcal{A}}\sqrt{2L(f(Mx)+d(y))}$, and $f(Mx)+d(y)$ is the duality gap.

Proof of Theorem 2.4.1. We require the following standard technical tool for the proof of Theorem 2.4.1

Lemma 2.4.2 (Kakade et al., 2009, Theorem 6)**.** *If f is L-smooth, then f^* is $1/L$-strongly convex.*

Let y^* denote the optimal dual variable for (D). We show that $\|y-y^*\|_2$ can be bounded by the duality gap. By Lemma 2.4.2, we know that f^* is $1/L$-strongly convex, and it follows that $d(y)$ is also $1/L$-strongly convex. Then by the definition of strongly convexity, we have

$$\forall s \in \partial d(y^*),\ \ d(y) \geq d(y^*) + \langle s, y-y^* \rangle + \frac{1}{2L}\|y-y^*\|_2^2.$$

By the optimality condition, we know that $0 \in \partial d(y^*)$. Therefore, by reordering the inequality, we can get

$$\begin{aligned} \|y-y^*\|_2 &\leq \sqrt{2L(d(y)-d(y^*))} \\ &\leq \sqrt{2L(f(Mx)+d(y))} \quad \forall x \quad \text{s.t.} \quad \gamma_{\mathcal{A}}(x) \leq \tau. \end{aligned} \tag{2.20}$$

Next, we show that $\mathcal{F}_{\mathcal{A}}(M^*y^*) \subseteq \mathcal{F}_{\mathcal{A}}(M^*y, \varepsilon)$. For any $a \in \mathcal{F}_{\mathcal{A}}(M^*y^*)$,

$$\begin{aligned} \langle a, M^*y \rangle &= \langle a, M^*y^* \rangle + \langle a, M^*y - M^*y^* \rangle \\ &\geq \sigma_{\mathcal{A}}(M^*y^*) - \left(\max_{a \in \mathcal{A}} \|Ma\| \right) \|y-y^*\|_2 \\ &= \sigma_{\mathcal{A}}(M^*y) - (\sigma_{\mathcal{A}}(M^*y) - \sigma_{\mathcal{A}}(M^*y^*)) - \|M\|_{\mathcal{A}}\|y-y^*\|_2 \\ &\geq \sigma_{\mathcal{A}}(M^*y) - 2\|M\|_{\mathcal{A}}\|y-y^*\|_2 \\ &\geq \sigma_{\mathcal{A}}(M^*y) - 2\|M\|_{\mathcal{A}}\sqrt{2L(f(Mx)+d(y))} \\ &\geq \sigma_{\mathcal{A}}(M^*y) - \varepsilon. \end{aligned}$$

The last line is true by the definition of ε. Combining above result with Lemma 2.4.1, the proof is finished. □

The proof of Theorem 2.4.1 mirrors the original proof of the gap-based safe-screening rule for 1-norm regularization (Ndiaye et al., 2017) — redundant atoms (or coordinates) can be identified by bounding $\|y-y^*\|$ with the duality gap. By

using the support-face relationship described in Lemma 2.4.1 instead of the first-order optimality condition with 1-norm regularization, we can generalize their proof to atomic sets.

Next we characterize the atom identification property of the generalized gap-based safe-screening rule.

Proposition 2.4.1 (Atomic identification)**.** *Let $\{x^{(t)}\}_{t=1}^{\infty}$ and $\{y^{(t)}\}_{t=1}^{\infty}$ be sequences that converge to optimal primal and dual solutions x^* and y^* respectively. Let $\{\varepsilon^{(t)}\}_{t=1}^{\infty}$ be the gaps defined in Theorem 2.4.1 evaluated at $x^{(t)}$ and $y^{(t)}$. Then the set $\mathcal{A}^{(t)} := \cap_{j=1}^{t} \mathcal{F}_{\mathcal{A}}(M^* y^{(j)}, \varepsilon^{(j)})$ has the Painleveé-Kuratowski set limit (Rockafellar and Wets, 2009, p. 111)*

$$\lim_{t\to\infty} \mathcal{A}^{(t)} = \mathcal{F}_{\mathcal{A}}(M^* y^*). \tag{2.21}$$

Proposition 2.4.1 ensures that the safe-screening rule (2.19) is guaranteed to eventually discard superfluous atoms as long as we have available an iterative solver that can generate primal iterates that converge to a solution. For polyhedral atomic set, e.g., atomic set with finite elements, it is straightforward to verify that Proposition 2.4.1 implies the following finite-time atom identification property:

$$\exists\, T > 0 \quad \text{such that} \quad \mathcal{A}^{(t)} = \mathcal{F}_{\mathcal{A}}(M^* y^*) \quad \forall t > T.$$

The implementation of the generalized gap-based safe-screening rule for polyhedral atomic sets is also straightforward. One can store all atoms in memory during computation, and the gap-based safe-screening rule offers a computable way to discard redundant atoms periodically during the optimization. When $\mathcal{F}_{\mathcal{A}}(M^* y, \varepsilon)$ is small enough, let $\widehat{\mathcal{A}} := \mathcal{F}_{\mathcal{A}}(M^* y, \varepsilon) := \{\hat{a}_i\}_{i=1}^{r}$, we can solve the reduced low-dimensional problem

$$\underset{x\in\mathbb{R}^r, x\geq 0}{\text{minimize}} \;\; f\left(M \sum_{i=1}^{r} \hat{a}_i x_i\right) \qquad \text{subject to} \qquad \sum_{i=1}^{r} x_i \leq \tau$$

instead of the original high-dimensional problem efficiently using algorithm such as accelerated projected gradient descent.

A remarkable aspect of the generalized gap-based safe-screening rule (and also

the original gap safe-screening rule) is that it depends solely on the duality gap, and hence is algorithm agnostic. As long as we have an algorithm that guarantees duality gap converges to 0, the gap-based safe-screening rule will recover $\mathcal{F}_{\mathcal{A}}(M^*y^*)$ within a finite number of iterations (for finite atomic sets). The gap-based safe-screening rule has been successfully applied to algorithms such as conditional gradient descent and projected coordinate descent to achieve promising performance.

2.4.5 Gap-based safe-screening rule for nuclear norm

We explore a key question whether the generalized safe-screening rule can provide any computational advantage for atomic sets $\mathcal{A}$ with infinite number of atoms. In particular we consider the nuclear-norm regularized problems whose atomic set is the set of rank-one matrices, i.e.,

$$\mathcal{A} = \{\, uv^T \mid u \in \mathbb{R}^n, v \in \mathbb{R}^m, \|u\|_2 = \|v\|_2 = 1 \,\}.$$

In the following proposition, we show that the ε-exposed face of M^*y contains all the singular vectors of M^*y when ε is strictly positive.

Proposition 2.4.2 (Limitation of ε-Face)**.** *Let $M^*y = U\Sigma V^T$ be the full SVD decomposition of M^*y, where $\Sigma = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_{\min\{n,m\}})$, $\sigma_i \geq 0 \;\forall\, 0 < i \leq \min\{n,m\}$. For any $\varepsilon \geq 0$, the ε-exposed face can be explicitly expressed as*

$$\mathcal{F}_{\mathcal{A}}(M^*y, \varepsilon) = \left\{ Up(Vq)^T \;\middle|\; \sum_{i=1}^{\min\{n,m\}} \sigma_i p_i q_i \geq \sigma_1 - \varepsilon, \|p\|_2 = \|q\|_2 = 1 \right\}.$$

Then for any $\varepsilon > 0$, there exist p,q with all entries being nonzero, such that $Up(Vq)^T \in \mathcal{F}_{\mathcal{A}}(M^*y, \varepsilon)$.

Proof of Proposition 2.4.2. By the definition of $\mathcal{F}_{\mathcal{A}}(M^*y, \varepsilon)$,

$$\begin{aligned} \mathcal{F}_{\mathcal{A}}(M^*y, \varepsilon) &= \{\, uv^T \mid \langle uv^T, M^*y \rangle \geq \sigma_1 - \varepsilon,\ \|u\|_2 = \|v\|_2 = 1 \,\} \\ &= \{\, uv^T \mid \langle uv^T, U\Sigma V^T \rangle \geq \sigma_1 - \varepsilon,\ \|u\|_2 = \|v\|_2 = 1 \,\}. \end{aligned}$$

We know that U, V are orthonormal matrices, by setting $u = Up$ and $v = Vp$ for

some $p, q \in \mathbb{R}^{\min\{n,m\}}$ such that $\|p\|_2 = \|q\|_2 = 1$, we obtain

$$\begin{aligned}
\mathcal{F}_{\mathcal{A}}(M^*y, \varepsilon) &= \{ Up(Vq)^T \mid \langle Up(Vq)^T, U\Sigma V^T \rangle \geq \sigma_1 - \varepsilon,\ \|p\|_2 = \|q\|_2 = 1 \} \\
&= \{ Up(Vq)^T \mid p^T \Sigma q \geq \sigma_1 - \varepsilon,\ \|p\|_2 = \|q\|_2 = 1 \} \\
&= \left\{ Up(Vq)^T \;\middle|\; \sum_{i=1}^{\min\{n,m\}} \sigma_i p_i q_i \geq \sigma_1 - \varepsilon, \|p\|_2 = \|q\|_2 = 1 \right\}.
\end{aligned}$$

The above finishes the proof. □

Proposition 2.4.2 indicates that $\mathcal{F}_{\mathcal{A}}(M^*y, \varepsilon)$ contains not only the top singular vectors of M^*y but also the bottom singular vectors—even if ε is arbitrarily close to 0. This result is unfortunate since the gap-based safe-screening rule stated in Theorem 2.4.1 does not allow us to discard any singular vectors of M^*y and thus require a full SVD decomposition of M^*y even if the duality gap is arbitrarily close to zero.

2.4.6 Approximation with partial SVD

The face of $\mathcal{A}$ exposed by the vector M^*y^* is given by

$$\mathcal{F}_{\mathcal{A}}(M^*y^*) = \left\{ uv^T \mid u^T(M^*y^*)v = \sigma_1(M^*y^*) \right\},$$

where $\sigma_1(M^*y^*)$ is the largest singular value of M^*y^*. Therefore, when there are few singular vectors associated with the largest singular value, only the top few singular vectors of M^*y^* are actual useful atoms. This property motivates us to use the partial SVD decomposition of M^*y to extract the reduced atomic set. This hard-thresholding technique has been widely used as a heuristic. Formally, given a dual feasible solution y with partial SVD decomposition

$$M^*y = U_r \Sigma_r V_r^T \quad U_r \in \mathbb{R}^{n \times r}, V_r \in \mathbb{R}^{m \times r}, r \ll \min\{n, m\},$$

we construct the corresponding reduced atomic set

$$\widehat{\mathcal{A}} = \{ U_r p q^T V_r^T \mid \|p\|_2 = \|q\|_2 = 1 \},$$

and solve the reduced problem over $\widehat{\mathcal{A}}$.

First, we give a concrete example showing that the partial SVD of M^*y is not able to give us a safe cover of $\mathcal{F}_{\mathcal{A}}(M^*y^*)$ even when $\mathcal{F}_{\mathcal{A}}(M^*y^*)$ is a singleton and y arbitrarily close to y^*.

Example 2.4.1 (Limitation of partial SVD)**.** *Consider the problem*

$$\underset{X\in\mathbb{R}^{n\times n}}{\text{minimize}} \ \frac{1}{2}\|X-Z\|_F^2 \qquad \textit{subject to} \qquad \|X\|_* \leq 1, \tag{2.22}$$

where

$$Z = U\,\mathrm{diag}(2,0.1,\ldots,0.1)V^T \quad \textit{and} \quad U = V = \begin{bmatrix} \sqrt{1-\varepsilon} & 0 & \ldots & -\sqrt{\varepsilon} \\ 0 & 1 & \ldots & \\ \vdots & & \ddots & \\ \sqrt{\varepsilon} & 0 & & \sqrt{1-\varepsilon} \end{bmatrix}_{n\times n}$$

for some $\varepsilon \in (0,1)$*. The dual problem is*

$$\underset{Y\in\mathbb{R}^{n\times n}}{\text{minimize}} \ \frac{1}{2}\|Y-Z\|_F^2 - \frac{1}{2}\|Z\|_F^2 + \|Y\|_2. \tag{2.23}$$

The solution for problem (2.22) *and problem* (2.23) *are*

$$X^* = U\,\mathrm{diag}(1,0,\ldots,0)V^T \quad \textit{and} \quad Y^* = Z - X^* = U\,\mathrm{diag}(1,0.1,\ldots,0.1)V^T.$$

Let $U \coloneqq [u_1,u_2,\ldots,u_n], V \coloneqq [v_1,v_2,\ldots,v_n]$*, then obviously* $\mathcal{S}_{\mathcal{A}}(X^*) = \mathcal{F}_{\mathcal{A}}(Y^*) = u_1v_1^T$ *is a singleton. We construct the following dual feasible solution*

$$\widehat{Y} = \mathrm{diag}(1,0.1,\ldots,0.1).$$

Let $\widehat{U},\widehat{V}$ *be the singular vectors of* $\widehat{Y}$*, then* $\widehat{U} = \widehat{V} = [\boldsymbol{e}_1,\boldsymbol{e}_2,\ldots,\boldsymbol{e}_n]$*, where* $\boldsymbol{e}_i$ *is the n-dimensional vector with 1 at the ith entry and 0 at other entries. In order to cover* $u_1 = [\sqrt{1-\varepsilon},0\ldots,\sqrt{\varepsilon}]^T$*, we need both the top singular and bottom singular vectors* $\boldsymbol{e}_1$ *and* $\boldsymbol{e}_n$*, and therefore any top-r SVD decomposition of* $\widehat{Y}$ *with* $r < n$ *will end up to be "unsafe". It is also easy to verify that* $\|\widehat{Y}-Y^*\|_F = \mathcal{O}(\sqrt{\varepsilon})$*. Note that our argument holds for any* $\varepsilon \in (0,1)$*. Therefore,* $\forall \varepsilon \in (0,1)$*,* $\exists y \in \mathbb{B}(y^*,\varepsilon)$

such that only full SVD decomposition of M^*y can guarantee a safe coverage of $\mathcal{F}_{\mathcal{A}}(M^*y^*)$. □

This result shows that the screening rule with partial SVD is not safe. Therefore, we can only resort to an approximate screening rule. We use the one-sided Hausdorff distance

$$\rho(\mathcal{A}_1, \mathcal{A}_2) \coloneqq \sup_{a_1 \in \mathcal{A}_1} \inf_{a_2 \in \mathcal{A}_2} \|a_1 - a_2\|_F$$

to measure the similarity between any two subsets $\mathcal{A}_1$ and $\mathcal{A}_2$ of the atomic set $\mathcal{A}$. The next result shows that there is a set $\widehat{\mathcal{A}}$ that is close to $\mathcal{S}_{\mathcal{A}}(x^*)$, then there must exist a point in $\widehat{\mathcal{A}}$ that is close to x^*.

Proposition 2.4.3 (Hausdorff error bound)**.** *Given $\widehat{\mathcal{A}} \subseteq \mathcal{A}$, there exist $x \in \text{cone}(\widehat{\mathcal{A}})$ such that*

$$\|x - x^*\|_F \leq \rho(\mathcal{S}_{\mathcal{A}}(x^*), \widehat{\mathcal{A}}) \sqrt{|\mathcal{S}_{\mathcal{A}}(x^*)|} \|x^*\|_F.$$

Proof. Proof of Proposition 2.4.3 Let $x^* = \sum_{a \in \mathcal{S}_{\mathcal{A}}(x^*)} c_a a, c_a > 0$. By the definition of the one-sided Hausdorff distance ρ, for any $a \in \mathcal{S}_{\mathcal{A}}(x^*)$, there exist a corresponding $\hat{a} \in \widehat{\mathcal{A}}$ such that

$$\|\hat{a} - a\|_F \leq \rho(\mathcal{S}_{\mathcal{A}}(x^*), \widehat{\mathcal{A}}).$$

Let $\hat{x} = \sum_{a \in \mathcal{S}_{\mathcal{A}}(x^*)} c_a \hat{a}$, then it is easy to verity that $\hat{x} \in \text{cone}(\widehat{\mathcal{A}})$ and

$$\|x - x^*\|_F \leq \rho(\mathcal{S}_{\mathcal{A}}(x^*), \widehat{\mathcal{A}}) \sum_{a \in \mathcal{S}_{\mathcal{A}}(x^*)} c_a \overset{\text{(i)}}{\leq} \rho(\mathcal{S}_{\mathcal{A}}(x^*), \widehat{\mathcal{A}}) \sqrt{|\mathcal{S}_{\mathcal{A}}(x^*)|} \|x^*\|_F,$$

(i) is true since the decomposition $x^* = \sum_{a \in \mathcal{S}_{\mathcal{A}}(x^*)} c_a a, c_a > 0$ is an orthonormal decomposition and $\|x^*\|_F^2 = \sum c_a^2$ when our atomic-set is the set of rank-one matrices. □

Next, we study the approximation ability of the partial SVD decomposition of a given feasible dual solution M^*y to $\mathcal{F}_{\mathcal{A}}(M^*y^*)$.

Theorem 2.4.2. *Let y be a dual feasible vector. Let $M^*y = U_r \Sigma_r V_r^T, U_r \in \mathbb{R}^{n \times r}, V_r \in \mathbb{R}^{m \times r}$ be the top-r SVD decomposition where $r < \min\{n, m\}$. Denote $\{\sigma_i\}_{i=1}^{\min\{n,m\}}$ as the singular values and $\widehat{\mathcal{A}} = \{ U_r p q^T V_r^T \mid \|p\|_2 = \|q\|_2 = 1 \}$ be the our reduced*

atomic set, assume $\sigma_1 > \sigma_{r+1}$, then

$$\rho(\mathcal{F}_{\mathcal{A}}(M^*y^*), \widehat{\mathcal{A}}) \leq \rho(\mathcal{F}_{\mathcal{A}}(M^*y, \varepsilon), \widehat{\mathcal{A}}) = \sqrt{2\min\left\{\frac{\varepsilon}{\sigma_1 - \sigma_{r+1}}, 1\right\}},$$

where ε is the same as defined in Theorem 2.4.1.

Proof. Proof for Theorem 2.4.2 By the definition of $\rho(\cdot,\cdot)$, it is straightforward that

$$\rho(A,C) \leq \rho(B,C) \quad \forall A,B,C \subseteq \mathbb{R}^{n\times m} \quad \text{such that} \quad A \subseteq B.$$

We know that $\mathcal{F}_{\mathcal{A}}(M^*y^*) \subseteq \mathcal{F}_{\mathcal{A}}(M^*y,\varepsilon)$ from Theorem 2.4.1, then obviously we have

$$\rho(\mathcal{F}_{\mathcal{A}}(M^*y^*), \widehat{\mathcal{A}}) \leq \rho(\mathcal{F}_{\mathcal{A}}(M^*y, \varepsilon), \widehat{\mathcal{A}}).$$

For any $\mathcal{A}_1, \mathcal{A}_2 \subseteq \mathcal{A}$,

$$\rho(\mathcal{A}_1, \mathcal{A}_2) = \sqrt{\sup_{a_1\in\mathcal{A}_1} \inf_{a_2\in\mathcal{A}_2} \|a_1 - a_2\|_F^2} = \sqrt{2 - 2\left(\inf_{a_1\in\mathcal{A}_1} \sup_{a_2\in\mathcal{A}_2} \langle a_1, a_2\rangle\right)}, \quad (2.24)$$

where the second equality holds since $\|a_1\|_F = \|a_2\|_F = 1$ by the definition of $\mathcal{A}$. Define $\mathcal{A}_1 = \mathcal{F}_{\mathcal{A}}(M^*y,\varepsilon)$ and $\mathcal{A}_2 = \widehat{\mathcal{A}} = \left\{ U_r pq^T V_r^T \;\middle|\; \|p\|_2 = \|q\|_2 = 1 \right\}$, where U_r, V_r are the top-r singular vectors of M^*y. Let $k := \min\{n,m\}$, $\mathcal{C}_1 = \{(p,q) \mid \sum_{i=1}^k \sigma_i p_i q_i \geq \sigma_1 - \varepsilon,\ \|p\|_2 = \|q\|_2 = 1,\ p,q \in \mathbb{R}^k\}$ and $\mathcal{C}_2 = \{(\hat{p},\hat{q}) \mid \|\hat{p}\|_2 = \|\hat{q}\|_2 = 1,\ \hat{p},\hat{q} \in \mathbb{R}^r\}$, then

$$\begin{aligned}
\rho(\mathcal{A}_1, \mathcal{A}_2) &= \sqrt{2 - 2\left(\min_{p,q\in\mathcal{C}_1} \max_{\hat{p},\hat{q}\in\mathcal{C}_2} \langle U pq^T V^T, U_r \hat{p}\hat{q}^T V_r^T\rangle\right)} \\
&= \sqrt{2 - 2\left(\min_{p,q\in\mathcal{C}_1} \max_{\hat{p},\hat{q}\in\mathcal{C}_2} \left(\sum_{i=1}^r p_i\hat{p}_i\right)\left(\sum_{i=1}^r q_i\hat{q}_i\right)\right)} \\
&= \sqrt{2 - 2\left(\min_{p,q\in\mathcal{C}_1} \|p_{1:r}\|_2 \|q_{1:r}\|_2\right)} \qquad (2.25)
\end{aligned}$$

Now we consider the subproblem in (2.25):

$$\begin{aligned} \min_{p,q} \quad & \|p_{1:r}\|_2 \|q_{1:r}\|_2 \\ \text{subject to} \quad & \sum_{i=1}^{k} \sigma_i p_i q_i \geq \sigma_1 - \varepsilon,\ \|p\|_2 = \|q\|_2 = 1,\ p,q \in \mathbb{R}^k. \end{aligned} \tag{P$_1$}$$

If p^* and q^* is a solution of the problem (P$_1$), then it is easy to verify that

$$\begin{aligned} \tilde{p} &= \left[\|p^*_{1:r}\|_2, 0, \ldots, \|p^*_{r+1:k}\|_2, 0, \ldots, 0\right] \\ \text{and} \quad \tilde{q} &= \left[\|q^*_{1:r}\|_2, 0, \ldots, \|q^*_{r+1:k}\|_2, 0, \ldots, 0\right] \end{aligned}$$

is also a valid solution. Therefore there must exist solution p^*, q^* such that $p_i = q_i = 0\ \forall i \notin \{1, r+1\}$, that is only p^*_1, q^*_1 and p^*_{r+1} and q^*_{r+1} are greater or equal than 0. This allow us to further reduce the problem to

$$\begin{aligned} \min_{p_1,q_1,p_{r+1},q_{r+1}} \quad & p_1 q_1 \\ \text{subject to} \quad & \sigma_1 p_1 q_1 + \sigma_{r+1} p_{r+1} q_{r+1} \geq \sigma_1 - \varepsilon, \\ & p_1^2 + p_{r+1}^2 = q_1^2 + q_{r+1}^2 = 1,\ p_1, q_1, p_{r+1}, q_{r+1} \geq 0. \end{aligned}$$

It is easy to verify that when $\sigma_1 - \sigma_{r+1} \geq \varepsilon$, the above problem attains solution at

$$p_1 = q_1 = \sqrt{\frac{\sigma_1 - \sigma_{r+1} - \varepsilon}{\sigma_1 - \sigma_{r+1}}} \quad \text{and} \quad p_{r+1} = q_{r+1} = \sqrt{1 - p_1^2}.$$

When $\sigma_1 - \sigma_{r+1} < \varepsilon$, the solution is simply $p_1 = q_1 = 0, p_{r+1} = q_{r+1} = 1$. Therefore the optimal value of (P$_1$) is $\max\{1 - \varepsilon/(\sigma_1 - \sigma_{r+1}), 0\}$, plug this into eq. (2.25) and the proof is finished. □

Oustry developed a related result based on the two-sided Hausdorff distance (Oustry, 2000, Theorem 2.11). Directly applying his Theorem to our context would end up with a bound $\mathcal{O}(\sqrt{\varepsilon/(\sigma_r - \sigma_{r+1})})$, which is looser than the bound shown in Theorem 2.4.2 because $\sigma_1 \geq \sigma_r \geq \sigma_{r+1}$.

2.4.7 Discussion

Our extension of gap-based safe-screening rules to the various forms of atomic-norm regularization is based on the convex calculus of sublinear functions. Our proposed screening rules can provide practical computational advantages when the atomic sets are polyhedral. As demonstrated by Example 2.4.1, however, there are limitations of the rule when used for non-polyhedral atomic sets. In that case, Theorem 2.4.2 provides an error bound based on the truncated SVD.

Further research opportunities remain, particularly for designing meaningful safe-screening rules for non-polyhedral sets. For example, it seems possible to design safe-screening rules for nuclear-norm regularized problems that are particular to the search directions generated by the conditional-gradient method.

Chapter 3

Online mirror descent with unknown time horizon

In modern big data applications such as recommendation and advertisement, data usually comes in a stream and one is required to make decisions in an online manner. From the perspective of optimization, such problems are usually formulated under the online convex optimization (OCO) framework. In OCO, a player is required to make a sequence of online decisions over discrete time steps. Each decision incurs a cost given by a convex function that is only revealed to the player before they make that decision. The goal of the player is to minimize the total cost.

Formally, let T denote the number of decisions required to make. For each time step $t \in \{1, 2, \ldots, T\}$, our algorithm proposes a point $x^{(t)}$ from a closed convex set $\mathcal{X} \subseteq \mathbb{R}^d$, and an adversary simultaneously picks a convex cost function f_t. This function penalizes the proposal $x^{(t)}$ by the amount $f_t(x^{(t)})$, and the cost of the iteration at time t is defined as $f_t(x^{(t)})$. The goal of the algorithm is to minimize the *regret* against an unknown comparison point $z \in \mathcal{X}$ at time T

$$\text{Regret}(T, z) \coloneqq \sum_{t=1}^{T} f_t(x^{(t)}) - \sum_{t=1}^{T} f_t(z).$$

The $\text{Regret}(T, z)$ measures the difference between the total cost of the algorithm and the cost of the competitor z up to time T. Our goal is to develop algorithm that ensures its regret is **sublinear** in T against **any** competitor. In this way, the average

cost incurred by the algorithm will be guaranteed to be smaller or equal to the cost from the best competitor as $T \to \infty$.

Online mirror descent (OMD) and dual averaging (DA) are two important algorithm templates for OCO from which many classical online learning algorithms can be derived as special cases; see works from Shalev-Shwartz (2012) and McMahan (2017) for examples. For Lipschitz continuous functions $\{f_t\}_{t\geq 1}$, when the number of decisions to be made T is known in advance, the algorithm could use T as an input and the performance of OMD and DA (with properly chosen stepsize) are shown to be very similar (Hazan, 2016). That is, they achieve essentially the same regret bound when using the same stepsize scheduling. However, in the more challenging setting when T is not known a priori (unknown time horizon), there is a fundamental difference in the regret rates of OMD and DA with a similar time-varying stepsize scheduling. While DA can guarantee sublinear regret bound $\mathcal{O}(\sqrt{T})$[1] for any $T > 0$ (Nesterov, 2009),there are instances such that OMD could suffer asymptotically linear regret, i.e., $\Omega(T)$ (Orabona and Pál, 2018).

In this chapter, we introduce a *stabilization* technique to fix OMD in the unknown time horizon setting and give essentially the same regret bound for stabilized-OMD and DA. We present our convergence analysis in a careful and modular way that allows for straightforward and flexible proofs. We also adapt our stabilized-OMD for composite objective setting.

3.1 Background

We review some standard technical tools used by OMD and briefly describe some known properties of OMD and DA in the literature of OCO.

3.1.1 Definitions and notations

Both OMD and DA are parameterized by a special convex function Φ, often referred as a regularizer or a mirror map (for $\mathcal{X}$), which among other properties needs[2] to be of Legendre type (Rockafellar, 1970, Chapter 26). Formally, throughout the

[1] This is rate equivalent to the iteration complexity $\mathcal{O}(\varepsilon^{-2})$. To keep our notation consistent with the literature, we describe the rate in terms of T instead of ε in this chapter.

[2] One may relax this condition in some cases. For a detailed discussion on the conditions needed on the mirror map, see (Bubeck, 2011, § 5.2).

chapter we assume that the function $\Phi\colon \bar{\mathcal{D}} \to \mathbb{R}$ is a closed convex function such that $\operatorname{int}\bar{\mathcal{D}} \cap \operatorname{ri}\mathcal{X} \neq \emptyset$ (where $\operatorname{ri}\mathcal{X}$ denotes the relative interior of $\mathcal{X}$), and whose conjugate is differentiable on $\mathbb{R}^d$. Moreover, we also assume that Φ is of Legendre type, which means that Φ is strictly convex on its domain[3] and essentially smooth, that is, for $\mathcal{D} \coloneqq \operatorname{int}\bar{\mathcal{D}}$ we have

- $\mathcal{D}$ is nonempty,
- Φ is differentiable on $\mathcal{D}$, and
- $\lim_{x\to\partial\mathcal{D}} \|\nabla\Phi(x)\| = +\infty$, where $\partial\mathcal{D}$ is the boundary of $\mathcal{D}$, i.e., $\partial\mathcal{D} \coloneqq \operatorname{cl}\mathcal{D} \setminus \mathcal{D}$.

The gradient of the mirror map $\nabla\Phi : \mathcal{D} \to \mathbb{R}^d$ and the gradient of its conjugate $\nabla\Phi^* : \mathbb{R}^d \to \mathcal{D}$ are mutually inverse bijections between the primal space $\mathcal{D}$ and the dual space $\mathbb{R}^d$. We will adopt the following notational convention. Any vector in the primal space will be written without a hat, such as $x \in \mathcal{D}$. The same letter with a hat, namely $\hat{x}$, will denote the corresponding dual vector:

$$\hat{x} \coloneqq \nabla\Phi(x) \qquad \text{and} \qquad x \coloneqq \nabla\Phi^*(\hat{x}) \qquad \text{for all letters } x \quad .$$

Essential smoothness ensures not only that Φ is differentiable on the interior of its domain, but also that the slope of Φ increases arbitrarily fast near the boundary of its domain. The latter guarantees, at least intuitively, that the function to be increasing near and in the direction of the boundary of its domain. This property is fundamental for mirror descent to be well-defined (although not essential for dual averaging) since it ensures that the Bregman Projection onto $\mathcal{X}$ is attained by a point on $\mathcal{D}$ where Φ is differentiable, and uniqueness is a consequence of the strict convexity of Φ. Some mirror maps we shall look are classical cases of the OCO literature such as the negative entropy $x \in \mathbb{R}^d_+ \mapsto \sum_{i=1}^d x_i \ln x_i$ and the squared 2-norm $\frac{1}{2}\|\cdot\|_2^2$, and details on the reasons they are mirror maps can be found in the literature (Bubeck, 2011, 2015; Shalev-Shwartz, 2012). In particular, Bubeck (2011, Section 5.2) discussed the properties of functions of Legendre type and why

[3] In fact we only need Φ to be strictly convex on some convex subsets of the domain (Rockafellar, 1970, Chapter 26), but for the sake of simplicity we assume that Φ is strictly convex on its entire domain.

Algorithm 2 Pseudocode for both online mirror descent and dual averaging with dynamic stepsize given by η_t on iteration t. These methods differ only in how the iterate $\hat{y}^{(t+1)}$ is updated.

Input: $x^{(1)} \in \mathcal{X} \cap \mathcal{D}, \eta : \mathbb{N} \to \mathbb{R}_{>0}$.
for $t = 1, 2, \ldots$ **do**
  Incur cost $f_t(x^{(t)})$ and receive $g_t \in \partial f_t(x^{(t)})$
  $\hat{x}^{(t)} = \nabla\Phi(x^{(t)})$
  [**OMD update**] $\hat{y}^{(t+1)} = \hat{x}^{(t)} - \eta_t g_t$
  [**DA update**] $\hat{y}^{(t+1)} = \hat{x}^{(1)} - \eta_t \sum_{i \le t} g_i$
  $y^{(t+1)} = \nabla\Phi^*(\hat{y}^{(t+1)})$
  $x^{(t+1)} = \Pi_{\mathcal{X}}^{\Phi}(y^{(t+1)})$
end for

requiring the conjugate of the mirror map to be differentiable on the whole space is not necessary for mirror descent to be well-defined if one restricts the gradient steps in the dual space in some ways.

Given a mirror map Φ, the Bregman divergence with respect to Φ is defined by

$$D_\Phi(x,y) \;\coloneqq\; \Phi(x) - \Phi(y) - \langle \nabla\Phi(y), x - y \rangle, \qquad \forall x \in \bar{\mathcal{D}}, \forall y \in \mathcal{D}. \tag{3.1}$$

Throughout this chapter it will be convenient to use the notation

$$D_\Phi\left(\begin{smallmatrix} a \\ b \end{smallmatrix}; c\right) \;\coloneqq\; D_\Phi(a,c) - D_\Phi(b,c) \;=\; \Phi(a) - \Phi(b) - \langle \nabla\Phi(c), a - b \rangle. \tag{3.2}$$

In the important special case where $\Phi(x) = \frac{1}{2}\|x\|_2^2$, the Bregman divergence relates to the Euclidean distance, i.e., $D_\Phi(x,y) = \frac{1}{2}\|x-y\|_2^2$. When $\mathcal{D} = \mathbb{R}_+^d$ and $\Phi(x) = \sum_{i=1}^d x_i \log x_i$, the Bregman divergence becomes the generalized Kullback-Leibler (KL) divergence. The projection operator induced by the Bregman divergence is written as $\Pi_{\mathcal{X}}^{\Phi}(y) \coloneqq \arg\min\{D_\Phi(x,y) \mid x \in \mathcal{X}\}$.

A general template for optimization in the mirror descent framework is shown in Algorithm 2. The two classical algorithms, online mirror descent and dual averaging, are incarnations of this, differing only in how the dual variable $\hat{y}^{(t)}$ is updated.

3.1.2 OMD and DA with constant stepsize

When the time horizon T is known in advance, constant stepsize that depends on T can be adopted to achieve sublinear regret. In particular, as characterized by the following Theorem, OMD and DA with the same stepsize can obtain exactly the same regret bound.

Theorem 3.1.1 (Nesterov, 2009, Theorem 1, Hazan, 2016, Theorem 5.6)**.** *Suppose that Φ is ρ-strongly convex with respect to a norm $\|\cdot\|$ and pick a constant stepsize $\eta_t := \eta > 0$ for all $t \geq 1$. Let $\{x^{(t)}\}_{t\geq 1}$ be the sequence of iterates generated by Algorithm 2. Then for any sequence of convex functions $\{f_t\}_{t\geq 1}$ with each $f_t : \mathcal{X} \to \mathbb{R}$, the following bound holds for both OMD and DA updates,*

$$\text{Regret}(T,z) \;\leq\; \sum_{t=1}^{T} \frac{\eta \|g_t\|_*^2}{2\rho} + \frac{D_\Phi(z,x^{(1)})}{\eta}. \tag{3.3}$$

When T is known as a priori, $\mathcal{O}(\sqrt{T})$ regret can be obtained by setting $\eta = 1/\sqrt{T}$ in eq. (3.7). Interestingly, though OMD and DA with constant stepsize have same regret bound, the proofs used to derive this bound tend to be quite different.

3.1.3 OMD and DA with dynamic stepsize

In the unknown time horizon scenario, a dynamic stepsize with $\eta_t \propto 1/\sqrt{t}$ is usually adopted in the literature of online learning (Beck and Teboulle, 2003; Zinkevich, 2003). Moreover, when the domain $\mathcal{X}$ is bounded, both OMD and DA with stepsize $\eta_t \propto 1/\sqrt{t}$ have $O(\sqrt{T})$ regret bounds (with differing constants). However, when we allow the domain $\mathcal{X}$ to be unbounded, OMD is shown to be to provably worse than DA:

Theorem 3.1.2 (Linear regret for OMD, Orabona and Pál, 2018, Theorem 3)**.** *Set $\eta_t = 1/\sqrt{t}$. Let $\{x^{(t)}\}_{t\geq 1}$ denote the sequence of iterates generated by Algorithm 2 with OMD update. For any $T \geq 3$ there exists a sequence of convex 1-Lipschitz functions $\{f_t\}_{t=1}^T$ and an initial point $x^{(1)} \in \mathcal{X}$ such that*

$$\sup_{z\in\mathcal{X}} D_\Phi(z,x^{(1)}) \text{ is bounded} \qquad \text{and} \qquad \text{Regret}(T,z) = \Omega(T),$$

while Algorithm 2 with DA update can always guarantee sublinear regret bound $O(\sqrt{T})$ using a similar stepsizes (which differ only by constants).

Moreover, there are examples showing that for offline 1-dimensional gradient descent (i.e., mirror descent with 2 norm square regularization), a stepsize that is either asymptotically $o(1/\sqrt{t})$ or $\omega(1/\sqrt{t})$ cannot achieve regret bound $\mathcal{O}(\sqrt{t})$ for all $t > 0$. So OMD with stepsizes of the form $t^{-\alpha}$ with $\alpha > 0$ cannot obtain optimal regret when $\mathcal{X}$ is unbounded. A natural question is if we can improve OMD to make it provably work with dynamic stepsizes. In the next section we provide a fix for OMD with dynamic stepsizes through a stabilization technique and later we show its connection with dynamic DA.

3.2 Stabilized OMD

The intuition for the idea is as follows. Suppose $\mathcal{Z} \subseteq \mathcal{X}$ is a set of comparison points with respect to which we wish our algorithm to have low regret. Usually, we assume $\sup_{z\in\mathcal{Z}} D_\Phi(z, x^{(1)})$ is bounded, that is, the initial point is not too far (with respect to the Bregman divergence) from any comparison point. Since $\sup_{z\in\mathcal{Z}} D_\Phi(z, x^{(1)})$ is bounded (but not necessarily $\sup_{z\in\mathcal{Z}, x\in\mathcal{X}} D_\Phi(z, x)$), the point $x^{(1)}$ is the only point in $\mathcal{X}$ that is known to be somewhat close (with respect to the Bregman divergence) to all the other points in $\mathcal{X}$. Thus, iterates computed by the algorithm should remain reasonably close to $x^{(1)}$ so that no other point $z \in \mathcal{Z}$ is too far from the iterates. If there were such a point z, an adversary could later chose functions so that picking z every round would incur low loss. At the same time, OMD would take many iterations to converge to z since consecutive OMD iterates tend to be close with respect to the Bregman divergence. That is, the algorithm would have high regret against z. To prevent this, the stabilization technique modifies each iterate $x^{(t)}$ to mix in a small fraction of $x^{(1)}$. This idea is not entirely new: it appears, for example, in the original Exp3 algorithm (Auer et al., 2002a), although for different reasons.

There are two ways to realize the stabilization idea.

Primal Stabilization. Replace $x^{(t)}$ with a convex combination of $x^{(t)}$ and $x^{(1)}$.

Dual Stabilization. Replace $\hat{y}^{(t)}$ with a convex combination of $\hat{y}^{(t)}$ and $\hat{x}^{(1)}$ (Recall from Algorithm 2 that $\hat{y}^{(t)}$ is the dual iterate computed by taking a gradient step). An illustration for dual stabilization is shown in Figure 3.1.

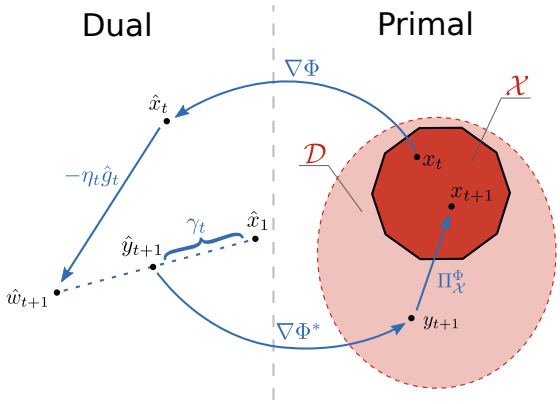


Figure 3.1: Illustration of the t-th iteration of DS-OMD.

After a draft of this chapter was made publicly available (the preliminary version released in ICML 2020), we were informed that an idea similar to primal stabilization had appeared in the Robust Optimistic Mirror Descent algorithm (Kangarshahi et al., 2018). Their setting is somewhat different since they perform optimistic steps. Furthermore, their results are somewhat weaker in terms of constant factors and since they cannot handle Bregman projections.

Additionally, after the ICML version of this paper was published we were told about ideas similar to primal stabilization were involved in the Twisted Mirror Descent(TMD) algorithm (György and Szepesvári, 2016). More specifically, TMD is a meta-algorithm adds a step at each iteration controlled by some sequence of functions that may depend on previous iterates and has, as a special case, primal stabilization. For the case of prediction with expert advice, they showed (György and Szepesvári, 2016, Example 6) how primal stabilization yields good bounds on the shifting regret. In our work, we extended the idea of primal stabilization for cases beyond the one of prediction with expert's advice.

3.2.1 Dual-stabilized OMD

Algorithm 3 gives pseudocode showing our modification of OMD to incorporate dual stabilization.

Theorem 3.2.1 (Regret bound for dual-stabilized OMD)**.** *Assume that* $\eta_t \geq \eta_{t+1} > 0$

Algorithm 3 Dual-stabilized online mirror descent, with dynamic stepsize η_t. The parameters γ_t control the amount of stabilization.

Input: $x^{(1)} \in \mathcal{X},\ \eta : \mathbb{N} \to \mathbb{R}_+,\ \gamma : \mathbb{N} \to (0,1]$
for $t = 1, 2, \ldots$ **do**
  Incur cost $f_t(x^{(t)})$ and receive $g_t \in \partial f_t(x^{(t)})$

$$\hat{x}^{(t)} = \nabla\Phi(x^{(t)}) \qquad \triangleright \text{ map primal iterate to dual space}$$

$$\hat{w}^{(t+1)} = \hat{x}^{(t)} - \eta_t g_t \qquad \triangleright \text{ gradient step in dual space} \tag{3.4}$$

$$\hat{y}^{(t+1)} = \gamma_t \hat{w}^{(t+1)} + (1-\gamma_t)\hat{x}^{(1)} \qquad \triangleright \text{ stabilization in dual space} \tag{3.5}$$

$$y^{(t+1)} = \nabla\Phi^*(\hat{y}^{(t+1)}) \qquad \triangleright \text{ map dual iterate to primal space}$$

$$x^{(t+1)} = \Pi_{\mathcal{X}}^{\Phi}(y^{(t+1)}) \qquad \triangleright \text{ project onto feasible region} \tag{3.6}$$

end for

for all $t \geq 1$. Define $\gamma_t = \eta_{t+1}/\eta_t \in (0,1]$ for all $t \geq 1$. Let $\{x^{(t)}\}_{t\geq 1}$ be the sequence of iterates generated by Algorithm 3. Then for any sequence of convex functions $\{f_t\}_{t\geq 1}$ with $f_t : \mathcal{X} \to \mathbb{R}$ for each $t \geq 1$,

$$\text{Regret}(T,z) \ \leq\ \sum_{t=1}^{T} \frac{D_\Phi\left(\begin{smallmatrix} x^{(t)} \\ x^{(t+1)} \end{smallmatrix}; w^{(t+1)}\right)}{\eta_t} + \frac{D_\Phi(z, x^{(1)})}{\eta_{T+1}} \quad \forall T > 0. \tag{3.7}$$

Note that strong convexity of Φ is **not** assumed. As we will see in Section 3.3.1, the term $D_\Phi\left(\begin{smallmatrix} x^{(t)} \\ x^{(t+1)} \end{smallmatrix}; w^{(t+1)}\right)$ can be easily bounded when the mirror map is strongly convex. This yields sublinear regret for $\eta_t \propto 1/\sqrt{t}$, which is not the case for the classical OMD when $\sup_{x,y\in\mathcal{X}} D_\Phi(x,y) = +\infty$.

Proof (of Theorem 3.2.1).

The first step is the same as in the standard OMD proof. For all $z \in \mathcal{X}$,

$$\begin{aligned} f_t(x^{(t)}) - f_t(z) &\overset{\text{(i)}}{\leq} \langle g_t, x^{(t)} - z \rangle \\ &\overset{\text{(ii)}}{=} \frac{1}{\eta_t}\langle \hat{x}^{(t)} - \hat{w}^{(t+1)}, x^{(t)} - z \rangle \\ &\overset{\text{(iii)}}{=} \frac{1}{\eta_t}\left(D_\Phi(x^{(t)}, w^{(t+1)}) - D_\Phi(z, w^{(t+1)}) + D_\Phi(z, x^{(t)})\right), \end{aligned} \tag{3.8}$$

where (i) is from the subgradient inequality, (ii) follows from eq. (3.4) and (iii) is by Proposition B.1.6.

The next step exhibits the main point of stabilization. Without stabilization we would have

$$x^{(t+1)} = \Pi_{\mathcal{X}}^{\Phi}(w^{(t+1)}) \text{ and } D_\Phi(z, w^{(t+1)}) \geq D_\Phi(z, x^{(t+1)}) + D_\Phi(x^{(t+1)}, w^{(t+1)})$$

by Proposition B.1.8, so eq. (3.8) would lead to a telescoping sum involving $D_\Phi(z,\cdot)$ *if the stepsize were fixed*. With a dynamic stepsize the analysis is trickier: we need a claim that leads to telescoping terms by relating $D_\Phi(z, w^{(t+1)})$ to $D_\Phi(z, x^{(t+1)})$.

Claim 3.2.1. *Assume that* $\gamma_t = \eta_{t+1}/\eta_t \in (0,1]$. *Then*

$$\begin{aligned} &(3.8) \\ &\leq \frac{D_\Phi(x^{(t)}_{x^{(t+1)}}; w^{(t+1)})}{\eta_t} + \underbrace{\Big(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\Big)}_{\text{telescopes}} D_\Phi(z, x^{(1)}) + \underbrace{\frac{D_\Phi(z, x^{(t)})}{\eta_t} - \frac{D_\Phi(z, x^{(t+1)})}{\eta_{t+1}}}_{\text{telescopes}}. \end{aligned}$$

Proof. First we derive the inequality

$$\begin{aligned} &\gamma_t \big(D_\Phi(z, w^{(t+1)}) - D_\Phi(x^{(t+1)}, w^{(t+1)})\big) + (1-\gamma_t) D_\Phi(z, x^{(1)}) \\ &\overset{\text{(i)}}{\geq} \gamma_t D_\Phi(z_{x^{(t+1)}}; w^{(t+1)}) + (1-\gamma_t) D_\Phi(z_{x^{(t+1)}}; x^{(1)}) \\ &\overset{\text{(ii)}}{=} D_\Phi(z_{x^{(t+1)}}; y^{(t+1)}) \\ &\overset{\text{(iii)}}{\geq} D_\Phi(z, x^{(t+1)}) \end{aligned}$$

where (i) is from the fact that $D_\Phi(x^{(t+1)}, x^{(1)}) \geq 0$ and $\gamma_t \leq 1$, (ii) follows from Proposition B.1.7 and eq. (3.5) and (iii) is by Proposition B.1.8 and eq. (3.6).

Rearranging and using $\gamma_t > 0$ yields

$$D_\Phi(z, w^{(t+1)}) \;\geq\; D_\Phi(x^{(t+1)}, w^{(t+1)}) - \Big(\frac{1}{\gamma_t} - 1\Big) D_\Phi(z, x^{(1)}) + \frac{1}{\gamma_t} D_\Phi(z, x^{(t+1)}). \tag{3.9}$$

Plugging this into eq. (3.8) yields

$$\begin{aligned}(3.8) \;&=\; \frac{1}{\eta_t}\Big(D_\Phi(x^{(t)}, w^{(t+1)}) - D_\Phi(z, w^{(t+1)}) + D_\Phi(z, x^{(t)})\Big) \\ &\leq\; \frac{1}{\eta_t}\Bigg(D_\Phi(x^{(t)}, w^{(t+1)}) - D_\Phi(x^{(t+1)}, w^{(t+1)}) + \Big(\frac{1}{\gamma_t} - 1\Big)D_\Phi(z, x^{(1)}) \\ &\qquad\qquad - \frac{1}{\gamma_t}D_\Phi(z, x^{(t+1)}) + D_\Phi(z, x^{(t)})\Bigg),\end{aligned}$$

by eq. (3.9). The claim follows by the definition of γ_t. □

The final step is very similar to the standard OMD proof. Summing eq. (3.8) over t and using Claim 3.2.1 leads to the desired telescoping sum.

$$\begin{aligned}&\sum_{t=1}^{T}\big(f_t(x^{(t)}) - f_t(z)\big) \\ \leq\; &\sum_{t=1}^{T}\Bigg(\frac{D_\Phi(\substack{x^{(t)} \\ x^{(t+1)}}; w^{(t+1)})}{\eta_t} + \Big(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\Big)D_\Phi(z, x^{(1)}) \\ &\qquad\qquad + \frac{D_\Phi(z, x^{(t)})}{\eta_t} - \frac{D_\Phi(z, x^{(t+1)})}{\eta_{t+1}}\Bigg) \\ \leq\; &\sum_{t=1}^{T}\frac{D_\Phi(\substack{x^{(t)} \\ x^{(t+1)}}; w^{(t+1)})}{\eta_t} + \left(\frac{1}{\eta_1} + \sum_{t=1}^{T}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)\right)D_\Phi(z, x^{(1)}) \\ =\; &\sum_{t=1}^{T}\frac{D_\Phi(\substack{x^{(t)} \\ x^{(t+1)}}; w^{(t+1)})}{\eta_t} + \frac{D_\Phi(z, x^{(1)})}{\eta_{T+1}}.\end{aligned}$$

□

3.2.2 Primal-stabilized OMD

Algorithm 4 gives pseudocode showing our modification of OMD to incorporate primal stabilization.

The algorithm is analyzed in the following Theorem.

Algorithm 4 Online mirror descent with primal stabilization.

Input: $x^{(1)} \in \mathbb{R}^d$, $\eta : \mathbb{N} \to \mathbb{R}$, $\gamma : \mathbb{N} \to \mathbb{R}$.
for $t = 1, 2, \ldots$ **do**
  Incur cost $f_t(x^{(t)})$ and receive $g_t \in \partial f_t(x^{(t)})$

$$\hat{x}^{(t)} = \nabla\Phi(x^{(t)}) \qquad \triangleright \text{ map primal iterate to dual space}$$

$$\hat{w}^{(t+1)} = \hat{x}^{(t)} - \eta_t g_t \qquad \triangleright \text{ gradient step in dual space} \tag{3.12}$$

$$w^{(t+1)} = \nabla\Phi^*(\hat{w}^{(t+1)}) \qquad \triangleright \text{ map dual iterate to primal space} \tag{3.13}$$

$$y^{(t+1)} = \Pi_{\mathcal{X}}^{\Phi}(w^{(t+1)}) \qquad \triangleright \text{ project onto feasible region} \tag{3.14}$$

$$x^{(t+1)} = \gamma_t y^{(t+1)} + (1-\gamma_t)x^{(1)} \qquad \triangleright \text{ stabilization in primal space} \tag{3.15}$$

end for

Theorem 3.2.2 (Regret bound for primal-stabilized OMD)**.** *Assume that $\eta_t \geq \eta_{t+1} > 0$ for all $t \geq 1$. Define $\gamma_t = \eta_{t+1}/\eta_t \in (0,1]$ for all $t \geq 1$. Let $\{x^{(t)}\}_{t\geq 1}$ be the sequence of iterates generated by Algorithm 4. Furthermore, assume that*

$$\text{for all } z \in \mathcal{X}, \text{ the map } x \mapsto D_\Phi(z,x) \text{ is convex on } \mathcal{X}. \tag{3.10}$$

Then for any sequence of convex functions $\{f_t\}_{t\geq 1}$ with each $f_t : \mathcal{X} \to \mathbb{R}$,

$$\text{Regret}(T,z) \;\leq\; \sum_{t=1}^{T} \frac{D_\Phi\left(\begin{smallmatrix} x^{(t)} \\ y^{(t+1)} \end{smallmatrix}; w^{(t+1)}\right)}{\eta_t} + \frac{D_\Phi(z,x^{(1)})}{\eta_{T+1}} \quad \forall T > 0. \tag{3.11}$$

Proof (of Theorem 3.2.2).

Let $z \in \mathcal{X}$. The first step is identical to the proof of Theorem 3.2.1 since the update rule in (3.12) is exactly the same as (3.4). Therefore, we have that (3.8) holds, that is,

$$f_t(x^{(t)}) - f_t(z) \;\leq\; \frac{1}{\eta_t}\left(D_\Phi(x^{(t)}, w^{(t+1)}) - D_\Phi(z, w^{(t+1)}) + D_\Phi(z, x^{(t)})\right).$$

Claim 3.2.2. *Assume that $\gamma_t = \eta_{t+1}/\eta_t \in (0,1]$. Then*

$$\begin{aligned}
&(3.8)\\
&\leq \frac{D_\Phi\left(\begin{smallmatrix} x^{(t)} \\ y^{(t+1)} \end{smallmatrix}; w^{(t+1)}\right)}{\eta_t} + \underbrace{\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)}_{\text{telescopes}} D_\Phi(z, x^{(1)}) + \underbrace{\frac{D_\Phi(z, x^{(t)})}{\eta_t} - \frac{D_\Phi(z, x^{(t+1)})}{\eta_{t+1}}}_{\text{telescopes}}.
\end{aligned}$$

Proof. First, we derive the inequality

$$\begin{aligned}
&\gamma_t \left(D_\Phi(z, w^{(t+1)}) - D_\Phi(y^{(t+1)}, w^{(t+1)})\right) + (1-\gamma_t) D_\Phi(z, x^{(1)})\\
&= \gamma_t D_\Phi\left(\begin{smallmatrix} z \\ y^{(t+1)} \end{smallmatrix}; w^{(t+1)}\right) + (1-\gamma_t) D_\Phi(z, x^{(1)})\\
&\geq \gamma_t D_\Phi(z, y^{(t+1)}) + (1-\gamma_t) D_\Phi(z, x^{(1)}) \qquad \text{(by Proposition B.1.8 and eq. (3.14))}\\
&\geq D_\Phi(z, x^{(t+1)}) \qquad \text{(by eq. (3.15), (3.10) and } \gamma_t \in (0,1]) \quad .
\end{aligned}$$

Rearranging and using $\gamma_t > 0$ yields

$$D_\Phi(z, w^{(t+1)}) \;\geq\; D_\Phi(y^{(t+1)}, w^{(t+1)}) - \left(\frac{1}{\gamma_t} - 1\right) D_\Phi(z, x^{(1)}) + \frac{1}{\gamma_t} D_\Phi(z, x^{(t+1)}). \tag{3.16}$$

Plugging this into eq. (3.8) yields

$$\begin{aligned}
(3.8) \;&=\; \frac{1}{\eta_t}\left(D_\Phi(x^{(t)}, w^{(t+1)}) - D_\Phi(z, w^{(t+1)}) + D_\Phi(z, x^{(t)})\right)\\
&\leq \frac{1}{\eta_t}\Bigg(D_\Phi(x^{(t)}, w^{(t+1)}) - D_\Phi(y^{(t+1)}, w^{(t+1)}) + \left(\frac{1}{\gamma_t} - 1\right) D_\Phi(z, x^{(1)})\\
&\qquad\qquad - \frac{1}{\gamma_t} D_\Phi(z, x^{(t+1)}) + D_\Phi(z, x^{(t)})\Bigg),
\end{aligned}$$

by eq. (3.16). The claim follows by the definition of γ_t. □

The final step is very similar to the proof of Theorem 3.2.1. The only difference is that we are using Claim 3.2.2 instead of Claim 3.2.1 and we replace $D_\Phi\left(\begin{smallmatrix} x^{(t)} \\ x^{(t+1)} \end{smallmatrix}; w^{(t+1)}\right)$ with $D_\Phi\left(\begin{smallmatrix} x^{(t)} \\ y^{(t+1)} \end{smallmatrix}; w^{(t+1)}\right)$. Formally, summing (3.8) over t and

using Claim 3.2.2 leads to the desired telescoping sum, that is,

$$\begin{aligned}
&\sum_{t=1}^{T}\big(f_t(x^{(t)})-f_t(z)\big)\\
&\le \sum_{t=1}^{T}\left(\frac{D_\Phi\left(\substack{x^{(t)}\\ y^{(t+1)}};w^{(t+1)}\right)}{\eta_t}+\Big(\frac{1}{\eta_{t+1}}-\frac{1}{\eta_t}\Big)D_\Phi(z,x^{(1)})+\right.\\
&\qquad\qquad\left.\frac{D_\Phi(z,x^{(t)})}{\eta_t}-\frac{D_\Phi(z,x_{t+1})}{\eta_{t+1}}\right)\\
&\le \sum_{t=1}^{T}\frac{D_\Phi\left(\substack{x^{(t)}\\ y^{(t+1)}};w^{(t+1)}\right)}{\eta_t}+\left(\frac{1}{\eta_1}+\sum_{t=1}^{T}\left(\frac{1}{\eta_{t+1}}-\frac{1}{\eta_t}\right)\right)D_\Phi(z,x^{(1)})\\
&= \sum_{t=1}^{T}\frac{D_\Phi\left(\substack{x^{(t)}\\ y^{(t+1)}};w^{(t+1)}\right)}{\eta_t}+\frac{D_\Phi(z,x^{(1)})}{\eta_{T+1}}.
\end{aligned}$$

□

3.2.3 Dual averaging

In this section, we show that Nesterov's dual averaging algorithm can be obtained from a small modification to dual-stabilized online mirror descent. Furthermore our proof of Theorem 3.2.1 can be adapted to analyze this algorithm.

The main difference between DS-OMD and dual averaging is in the gradient step, as we now explain. In iteration $t+1$ of DS-OMD, the gradient step is taken from $\hat{x}^{(t)}$, the dual counterpart of the iterate $x^{(t)}$:

$$\text{DS-OMD gradient step:} \qquad \hat{w}^{(t+1)} \;=\; \hat{x}^{(t)}-\eta_t g_t.$$

Suppose that the algorithm is modified so that the gradient step is taken from $\hat{y}^{(t)}$, the dual point from iteration t *before* projection onto the feasible region. (Here $\hat{y}^{(1)}$ is defined to be $\hat{x}^{(1)}$.) The resulting gradient step is:

$$\text{Lazy gradient step:} \qquad \hat{w}^{(t+1)} \;=\; \hat{y}^{(t)}-\eta_t g_t. \tag{3.17}$$

As before, we set

$$\hat{y}^{(t+1)} \;=\; \gamma_t\hat{w}^{(t+1)}+(1-\gamma_t)\hat{x}^{(1)} \tag{3.18}$$

Algorithm 5 Dual averaging with stepsize re-indexed as $\eta_2, \eta_3, \ldots$

Input: $x^{(1)} \in \mathcal{X},\ \eta : \mathbb{N} \to \mathbb{R}_+,\ \gamma : \mathbb{N} \to (0,1]$
$\hat{y}^{(1)} = \nabla\Phi(x^{(1)})$
for $t = 1, 2, \ldots$ **do**
  Incur cost $f_t(x^{(t)})$ and receive $g_t \in \partial f_t(x^{(t)})$
  $\hat{y}^{(t+1)} = \hat{x}^{(1)} - \eta_{t+1} \sum_{i \le t} g_i$ ▷ dual averaging update
  $y^{(t+1)} = \nabla\Phi^*(\hat{y}^{(t+1)})$ ▷ map dual iterate to primal space
  $x^{(t+1)} = \Pi_{\mathcal{X}}^{\Phi}(y^{(t+1)})$ ▷ project onto feasible region
end for

where $\gamma_t = \eta_{t+1}/\eta_t$. Then a simple inductive proof yields the following claim.

Claim 3.2.3. *$\hat{w}^{(t)} = \hat{x}^{(1)} - \eta_{t-1} \sum_{i<t} g_i$ and $\hat{y}^{(t)} = \hat{x}^{(1)} - \eta_t \sum_{i<t} g_i$ for all $t > 1$.*

Thus, the algorithm with the lazy gradient step can be written as in Algorithm 5. This is equivalent to Algorithm 2 with the DA update, except that η_t in Algorithm 2 corresponds to η_{t+1} in Algorithm 5.

Theorem 3.2.3 (Regret bound for dual averaging)**.** *Assume that $\eta_t \ge \eta_{t+1} > 0$ for all $t > 1$. Let $\{x^{(t)}\}_{t \ge 1}$ be the sequence of iterates generated by Algorithm 5. Then for any sequence of convex functions $\{f_t\}_{t \ge 1}$ with each $f_t : \mathcal{X} \to \mathbb{R}$,*

$$\text{Regret}(T, z) \ \le\ \sum_{t=1}^{T} \frac{D_\Phi\left(\begin{smallmatrix} x^{(t)} \\ x^{(t+1)} \end{smallmatrix}; \nabla\Phi^*(\hat{x}^{(t)} - \eta_t g_t)\right)}{\eta_t} + \frac{D_\Phi(z, x^{(1)})}{\eta_{T+1}} \quad \forall T > 0. \tag{3.19}$$

The proof parallels the proof of Theorem 3.2.1.

Proof (of Theorem 3.2.3).

The first step is very similar to the proof of Theorem 3.2.1. For all $z \in \mathcal{X}$,

$$\begin{aligned}
& f_t(x^{(t)}) - f_t(z) && (3.20)\\
\le\ & \langle g_t, x^{(t)} - z \rangle && \text{(subgradient inequality)}\\
=\ & \frac{1}{\eta_t} \langle \hat{y}^{(t)} - \hat{w}^{(t+1)}, x^{(t)} - z \rangle && \text{(by eq. (3.17))}\\
=\ & \frac{1}{\eta_t} \left(D_\Phi(x^{(t)}, w^{(t+1)}) - D_\Phi(z, w^{(t+1)}) + D_\Phi\left(\begin{smallmatrix} z \\ x^{(t)} \end{smallmatrix}; y^{(t)}\right) \right), && (3.21)
\end{aligned}$$

where we have used Proposition B.1.5 instead of Proposition B.1.6.

As in the proof of Theorem 3.2.1, the next step is to relate $D_\Phi(z, w^{(t+1)})$ to $D_\Phi(z, y^{(t+1)})$ so that eq. (3.21) can be bounded using a telescoping sum. The following claim is similar to Claim 3.2.1.

Claim 3.2.4. *Assume that $\gamma_t = \eta_{t+1}/\eta_t \in (0, 1]$. Then*

(3.21)
$$\leq \frac{D_\Phi\left(\begin{smallmatrix} x^{(t)} \\ x^{(t+1)} \end{smallmatrix}; w^{(t+1)}\right)}{\eta_t} + \underbrace{\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)}_{\text{telescopes}} D_\Phi(z, x^{(1)}) + \underbrace{\frac{D_\Phi\left(\begin{smallmatrix} z \\ x^{(t)} \end{smallmatrix}; y^{(t)}\right)}{\eta_t} - \frac{D_\Phi\left(\begin{smallmatrix} z \\ x^{(t+1)} \end{smallmatrix}; y^{(t+1)}\right)}{\eta_{t+1}}}_{\text{telescopes}}.$$

Proof. The first two steps are identical to the proof of Claim 3.2.1.

$$\begin{aligned} & \gamma_t \left(D_\Phi(z, w^{(t+1)}) - D_\Phi(x^{(t+1)}, w^{(t+1)})\right) + (1-\gamma_t) D_\Phi(z, x^{(1)}) \\ \overset{\text{(i)}}{\geq}\ & \gamma_t D_\Phi\left(\begin{smallmatrix} z \\ x^{(t+1)} \end{smallmatrix}; w^{(t+1)}\right) + (1-\gamma_t) D_\Phi\left(\begin{smallmatrix} z \\ x^{(t+1)} \end{smallmatrix}; x^{(1)}\right) \\ \overset{\text{(ii)}}{=}\ & D_\Phi\left(\begin{smallmatrix} z \\ x^{(t+1)} \end{smallmatrix}; y^{(t+1)}\right). \end{aligned}$$

where (i) is from the fact that $D_\Phi(x^{(t+1)}, x^{(1)}) \geq 0$ and $\gamma_t \leq 1$, (ii) is by Proposition B.1.7 and eq. (3.18). Rearranging and using $\gamma_t > 0$ yields

$$D_\Phi(z, w^{(t+1)}) \geq D_\Phi(x^{(t+1)}, w^{(t+1)}) - \left(\frac{1}{\gamma_t} - 1\right) D_\Phi(z, x^{(1)}) + \frac{D_\Phi\left(\begin{smallmatrix} z \\ x^{(t+1)} \end{smallmatrix}; y^{(t+1)}\right)}{\gamma_t}. \tag{3.22}$$

Plugging this into eq. (3.21) yields

$$\begin{aligned} (3.21) \ =\ & \frac{1}{\eta_t}\left(D_\Phi(x^{(t)}, w^{(t+1)}) - D_\Phi(z, w^{(t+1)}) + D_\Phi\left(\begin{smallmatrix} z \\ x^{(t)} \end{smallmatrix}; y^{(t)}\right)\right) \\ \leq\ & \frac{1}{\eta_t}\left(D_\Phi(x^{(t)}, w^{(t+1)}) - D_\Phi(x^{(t+1)}, w^{(t+1)}) + \left(\frac{1}{\gamma_t} - 1\right) D_\Phi(z, x^{(1)}) \right. \\ & \left. - \frac{D_\Phi\left(\begin{smallmatrix} z \\ x^{(t+1)} \end{smallmatrix}; y^{(t+1)}\right)}{\gamma_t} + D_\Phi\left(\begin{smallmatrix} z \\ x^{(t)} \end{smallmatrix}; y^{(t)}\right)\right), \end{aligned}$$

by eq. (3.22). The claim follows by the definition of γ_t. □

The final step is very similar to the proof of Theorem 3.2.1. Summing eq. (3.21) over t and using Claim 3.2.4 leads to the desired telescoping sum.

$$\begin{aligned}
&\sum_{t=1}^{T}\big(f_t(x^{(t)})-f_t(z)\big)\\
&\le \sum_{t=1}^{T}\left(\frac{D_\Phi\left(\begin{smallmatrix}x^{(t)}\\ x^{(t+1)}\end{smallmatrix};w^{(t+1)}\right)}{\eta_t}+\Big(\frac{1}{\eta_{t+1}}-\frac{1}{\eta_t}\Big)D_\Phi(z,x^{(1)})\right.\\
&\qquad\qquad\left.+\frac{D_\Phi\left(\begin{smallmatrix}z\\ x^{(t)}\end{smallmatrix};y^{(t)}\right)}{\eta_t}-\frac{D_\Phi\left(\begin{smallmatrix}z\\ x^{(t+1)}\end{smallmatrix};y^{(t+1)}\right)}{\eta_{t+1}}\right)\\
&\le \sum_{t=1}^{T}\frac{D_\Phi\left(\begin{smallmatrix}x^{(t)}\\ x^{(t+1)}\end{smallmatrix};w^{(t+1)}\right)}{\eta_t}+\left(\frac{1}{\eta_1}+\sum_{t=1}^{T}\left(\frac{1}{\eta_{t+1}}-\frac{1}{\eta_t}\right)\right)D_\Phi(z,x^{(1)})\\
&= \sum_{t=1}^{T}\frac{D_\Phi\left(\begin{smallmatrix}x^{(t)}\\ x^{(t+1)}\end{smallmatrix};w^{(t+1)}\right)}{\eta_t}+\frac{D_\Phi(z,x^{(1)})}{\eta_{T+1}}. \qquad (3.23)
\end{aligned}$$

For the second inequality we have also used that $D_\Phi\left(\begin{smallmatrix}z\\ x^{(1)}\end{smallmatrix};y^{(1)}\right)=D_\Phi(z,x^{(1)})$ since $x^{(1)}=y^{(1)}$.

Notice that eq. (3.23) is syntactically identical to eq. (3.7); the only difference is the definition of $w^{(t+1)}$ in these two settings. It turns out to be more useful for applications to provide a convenient upper bound on eq. (3.23), which is the conclusion of this theorem. Referring to eq. (3.17), we see that $w^{(t+1)}$ is closely related to $y^{(t)}$, but less closely related to $x^{(t)}$. To control $D_\Phi\left(\begin{smallmatrix}x^{(t)}\\ x^{(t+1)}\end{smallmatrix};w^{(t+1)}\right)$, it turns out to be convenient to apply Proposition B.1.9 as follows. Taking $p=y^{(t)}$, $\pi=x^{(t)}=\Pi^\Phi_{\mathcal{X}}(y^{(t)})$, $v=x^{(t+1)}$ and $\hat{q}=\eta_t g_t$, we obtain

$$\begin{aligned}
D_\Phi\left(\begin{smallmatrix}x^{(t)}\\ x^{(t+1)}\end{smallmatrix};w^{(t+1)}\right) &\overset{\text{(i)}}{=} -D_\Phi\left(\begin{smallmatrix}v\\ \pi\end{smallmatrix};\nabla\Phi^*(\hat{p}-\hat{q})\right)\\
&\overset{\text{(ii)}}{\le} -D_\Phi\left(\begin{smallmatrix}v\\ \pi\end{smallmatrix};\nabla\Phi^*(\hat{\pi}-\hat{q})\right)\\
&= D_\Phi\left(\begin{smallmatrix}x^{(t)}\\ x^{(t+1)}\end{smallmatrix};\nabla\Phi^*(\hat{x}^{(t)}-\eta_t g_t)\right),
\end{aligned}$$

where (i) is from the fact that $\hat{w}^{(t+1)}=\hat{y}^{(t)}-\eta_t g_t=\hat{p}-\hat{q}$ and (ii) is by Proposition B.1.9. Plugging this into (3.23) completes the proof. □

3.2.4 Remarks

Interestingly, the *doubling trick* (Shalev-Shwartz, 2012) on OMD can be viewed as an incarnation of stabilization. To see this, set $\eta_t := 1/\sqrt{2^{\lfloor \lg t \rfloor}}$ and $\gamma_t := \mathbf{1}_{\{t \text{ is a power of } 2\}}$. Then, for each dyadic interval of length 2^ℓ, the first iterate is $x^{(1)}$ and a fixed learning rate $1/\sqrt{2^\ell}$ is used. Thus, with these parameters, Algorithm 3 reduces to the doubling trick.

One should note that in Theorem 3.2.1 the stabilization parameter γ_t used in round $t \geq 1$ depends on the stepsizes for rounds t *and* $t+1$, need to "peek into the future". Thus, to use stabilization as in Theorem 3.2.1 the stepsize for round t can depend on information available only *up to round* $t-1$. This will come into play, for example, when we derive first-order regret bounds on Section 3.3.2 where the stepsize is based on the subgradients of the past functions (instead of simply depending on the count of rounds). Reindexing the stepsizes could fix the problem, but then the proof of Theorem 3.2.1 would look syntactically odd. Although this dependence on the future may seem unnatural, in Section 3.4 we shall see that under some mild conditions, stabilized OMD coincides exactly with DA with dynamic stepsizes after reindexing. This extends the same behavior observed between OMD and DA when the stepsizes are fixed. In this sense, stabilization may seem as a natural way to fix OMD for dynamic stepsizes.

3.3 Applications

3.3.1 Strongly-convex mirror maps

We now analyze the algorithms of the previous section in the scenario that the mirror map is strongly convex. Let η_t, γ_t, f_t be as above. The following result is a corollary of Theorems 3.2.1, 3.2.2 and 3.2.3.

Corollary 3.3.1 (Regret bound for dual-stabilized OMD)**.** *Suppose that Φ is ρ-strongly convex on $\mathcal{X}$ with respect to a norm $\|\cdot\|$. Let $\{x^{(t)}\}_{t=1}^\infty$ be the iterates produced by Algorithms 3, 4 or 5. (For Algorithm 4, the additional assumption*

(3.10) *is required.) Then*

$$\text{Regret}(T,z) \;\leq\; \sum_{t=1}^{T} \frac{\eta_t \|g_t\|_*^2}{2\rho} + \frac{D_\Phi(z,x^{(1)})}{\eta_{T+1}} \quad \forall T > 0.$$

This is identical to Nesterov's bound for dual averaging (Nesterov, 2009, eq. 2.15) (taking his $\lambda_i = 1$ and his $\beta_i = 1/\eta_i$). The proof is based on the following simple proposition, which bounds the Bregman divergence when Φ is strongly convex. See, e.g., Bubeck (Bubeck, 2015, pp. 300). A proof is given in Appendix B.2.

Proposition 3.3.1. *Suppose that Φ is ρ-strongly convex on $\mathcal{X}$ with respect to $\|\cdot\|$. Consider any $x, x' \in \mathcal{X}$ and $\hat{q} \in \mathbb{R}^d$. Then*

$$D_\Phi\left(\begin{smallmatrix} x \\ x' \end{smallmatrix}; \nabla\Phi^*(\hat{x} - \hat{q})\right) \;\leq\; \|\hat{q}\|_*^2 / 2\rho.$$

Proof (of Corollary 3.3.1). The regret bounds proven by Theorems 3.2.1, 3.2.2 and 3.2.3 all involve a summation with terms of the form

$$\begin{aligned}
&3.2.1: \; D_\Phi\left(\begin{smallmatrix} x^{(t)} \\ x^{(t+1)} \end{smallmatrix}; w^{(t+1)}\right) \\
&3.2.2: \; D_\Phi\left(\begin{smallmatrix} x^{(t)} \\ y^{(t+1)} \end{smallmatrix}; w^{(t+1)}\right) \\
&3.2.3: \; D_\Phi\left(\begin{smallmatrix} x^{(t)} \\ x^{(t+1)} \end{smallmatrix}; \nabla\Phi^*(\hat{x}^{(t)} - \eta_t g_t)\right).
\end{aligned}$$

For Theorems 3.2.1 and 3.2.3, we have $x^{(t+1)} \in \mathcal{X}$, whereas for Theorem 3.2.2 we also have $y^{(t+1)} \in \mathcal{X}$ by eq. (3.14). For Theorems 3.2.1 and 3.2.2 we have $w^{(t+1)} = \nabla\Phi^*(\hat{x}^{(t)} - \eta_t g_t)$ by eq. (3.4) and eq. (3.10). Therefore all of these terms may be bounded using Proposition 3.3.1 with $x = x^{(t)}$ and $\hat{q} = \eta_t g_t$. This yields the claimed bound. □

3.3.2 Prediction with expert advice

Next, we consider the setting of "prediction with expert advice". In this setting, $\mathcal{D}$ is $\mathbb{R}^d_{>0}$, $\mathcal{X}$ is the simplex $\Delta_d \subset \mathbb{R}^d$, and the mirror map is $\Phi(x) = \sum_{i=1}^d x_i \log x_i$. (On $\mathcal{X}$, Φ is the negative of the entropy function.) The gradient of the mirror map and

its conjugate are

$$\nabla\Phi(x)_i \;=\; \ln(x_i)+1 \qquad \text{and} \qquad \nabla\Phi^*(\hat{x})_i \;=\; \exp(\hat{x}_i-1). \quad (3.24)$$

For any two points $a \in \bar{\mathcal{D}}$, $b \in \mathcal{D}$, a small calculation shows that $D_\Phi(a,b)$ is the generalized KL-divergence

$$D_{\mathrm{KL}}(a,b) \;=\; \sum_{i=1}^{d} a_i \ln(a_i/b_i) - \|a\|_1 + \|b\|_1.$$

Note that the KL-divergence is convex on its second argument for any $a \in \bar{\mathcal{D}} = \mathbb{R}^d_{\geq 0}$ since the functions $-\ln(\cdot)$ and absolute value are both convex. This means that the general regret bounds for all algorithms discussed in Section 3.2, including primal stabilized OMD, hold in this setting.

In this section, we will use the theorems in Section 3.2 to derive regret bounds for this setting without much extra-work. As an intermediate step, we will derive bounds that use the following function:

$$\Lambda(a,b) \;\coloneqq\; D_{\mathrm{KL}}(a,b) + \|a\|_1 - \|b\|_1 + \ln\|b\|_1 \;=\; \sum_{i=1}^{d} a_i \ln(a_i/b_i) + \ln\|b\|_1,$$

which is a useful tool in the analysis of algorithms for the experts' problem. For examples, see works from de Rooij et al. (2014, §2.1) and Cesa-Bianchi et al. (2007, Lemma 4). An initial observation shows that Λ is non-negative in the experts' setting.

Proposition 3.3.2. *$\Lambda(a,b) \geq 0$ for all $a \in \mathcal{X}$, $b \in \mathcal{D}$.*

Proof. Let us write $\Lambda(a,b) = -\sum_{i=1}^{d} a_i \ln \frac{b_i}{a_i} + \ln\left(\sum_{i=1}^{d} b_i\right)$. Since a is a probability distribution, we may apply Jensen's inequality to show that this expression is non-negative. □

The following result is a corollary of Theorems 3.2.1, 3.2.2 and 3.2.3.

Corollary 3.3.2. *Assume that $\eta_t \geq \eta_{t+1} > 0$ for all $t \geq 1$. Define $\gamma_t = \eta_{t+1}/\eta_t \in (0,1]$. Let $x^{(1)}$ be the uniform distribution $\mathbf{1}/d$ and let $x^{(2)}, x^{(3)}, \ldots$ be the iterates*

produced by Algorithms 3, 4 or 5. Then

$$\text{Regret}(T,z) \;\leq\; \sum_{t=1}^{T} \frac{\Lambda(x^{(t)}, \nabla\Phi^*(\hat{x}^{(t)} - \eta_t g_t))}{\eta_t} + \frac{\ln d}{\eta_{T+1}} \qquad \forall T > 0. \quad (3.25)$$

The proof is a direct consequence of the following proposition, which is proven in Appendix B.3.

Proposition 3.3.3. *Let $a, b \in \mathcal{X}$ and $c \in \mathcal{D}$. Then $D_\Phi\binom{a}{b}; c) \leq \Lambda(a,c)$.*

Proof (of Corollary 3.3.2). First of all, recall that the D_{KL} is convex on its second argument, which allows us to use the bound from eq. (3.11) for primal stabilized OMD. As in the proof of Corollary 3.3.1, we first observe that the regret bounds (3.7), (3.11) and (3.19) all have sums with terms of the form $D_\Phi\binom{x^{(t)}}{u^{(t)}}; \nabla\Phi^*(\hat{x}^{(t)} - \eta_t g_t))$ for some $u^{(t)} \in \mathcal{X}$. These terms may be bounded using Proposition 3.3.3. Finally, the standard inequality $\sup_{z\in\mathcal{X}} D_{\text{KL}}(z, x^{(1)}) \leq \ln d$ completes the proof. □

Anytime regret

As another corollary of Corollary 3.3.2 we now derive an anytime regret bound in the case of bounded costs. This matches the best known bound appearing in the literature (Bubeck, 2011, Theorem 2.4) (Gerchinovitz, 2011, Proposition 2.1).

Corollary 3.3.3. *Suppose that $g_t \in [0,1]^d$ for all t. Define $\eta_t = 2\sqrt{\ln(d)/t}$ and $\gamma_t = \eta_{t+1}/\eta_t$. Let $x^{(1)}$ be the uniform distribution $\mathbf{1}/d$ and let $x^{(2)}, x^{(3)}, \ldots$ be the iterates produced by Algorithms 3, 4 or 5. Then*

$$\text{Regret}(T,z) \;\leq\; \sqrt{T \ln d} \qquad \forall T \geq 1,\; z \in \mathcal{X}. \quad (3.26)$$

The proof follows from Corollary 3.3.2 and Hoeffding's Lemma, as shown below.

Lemma 3.3.1 (Hoeffding's Lemma (Cesa-Bianchi and Lugosi, 2006, Lemma 2.2))**.** *Let X be a random variable with $a \leq X \leq b$. Then for any $s \in \mathbb{R}$,*

$$\ln \mathbb{E}[e^{sX}] - s\mathbb{E}X \;\leq\; \frac{s^2(b-a)^2}{8}.$$

Proof (of Corollary 3.3.3). Denote g_{ti} to be the ith entry of the vector g_t. By eq. (3.24) we have $\nabla\Phi^*(\hat{x}^{(t)} - \eta_t g_t)_i = x_i^{(t)} \exp(-\eta_t g_{ti})$ for each $i \in [d]$. This together with Lemma 3.3.1 for $s = -\eta_t$ yields

$$\Lambda(x^{(t)}, \nabla\Phi^*(\hat{x}^{(t)} - \eta_t g_t)) = \eta_t \langle x^{(t)}, g_t \rangle + \ln\left(\sum_{j=1}^{d} x_j^{(t)} e^{-\eta_t g_{tj}}\right) \;\leq\; \frac{\eta_t^2}{8}. \tag{3.27}$$

Plugging this and $\eta_t = 2\sqrt{\ln d / t}$ into eq. (3.25), we obtain the bound

$$\begin{aligned} \text{Regret}(T, z) \;&\leq\; \sqrt{\ln d}\left(\frac{1}{4}\sum_{t=1}^{T}\frac{1}{\sqrt{t}} + \frac{\sqrt{T+1}}{2}\right) && (3.28) \\ &\leq\; \sqrt{\ln d}\left(\frac{2\sqrt{T}-1}{4} + \frac{\sqrt{T}+0.5}{2}\right) \;\leq\; \sqrt{T \ln d} && (3.29) \end{aligned}$$

by Fact B.1.3 and sub-additivity of square root.

□

First-order regret bound

The regret bound presented in previous section depends on $\sqrt{T}$; this is known as the "zeroth-order" regret bound. In some scenarios the cost of the best expert up to time T can be far less than T. This makes the problem somewhat easier, and it is possible to improve the regret bound. Formally, let L_T^* denote the cost of the best expert at time T. Then $L_T^* \leq T$ due to our assumption that all costs are at most 1. A "first-order" regret bound depends on $\sqrt{L_T^*}$ instead of $\sqrt{T}$. See, for example, the book from Cesa-Bianchi and Lugosi (2006, §2.4).

The only modification to the algorithm is to change the stepsize. If the costs are "smaller than expected", then intuitively time is progressing "slower than expected". We will adopt an elegant idea from Auer et al. (2002b), which is to use the algorithm's cost itself as a measure of the progression of time, and to incorporate this into the stepsize. They call this a "self-confident" stepsize.

Corollary 3.3.4. *Assume that $g_t \in [0,1]^d$. Set $\eta_t = \sqrt{\ln(d)/(1 + \sum_{i<t}\langle g_i, x_i\rangle)}$ and $\gamma_t = \eta_{t+1}/\eta_t$. Denote the minimum total cost of any expert up to time T as $L_T^* :=$*

$\min_{j\in[d]} \sum_{t=1}^{T} g_{tj}$. *Then*

$$\text{Regret}(T,z) \;\le\; 2\sqrt{\ln(d)L_T^*} + 8\ln d \qquad \forall T \ge 1.$$

The main ingredients are the following alternative bound on Λ, which is proven in Appendix B.3, and some standard scalar inequalities.

Proposition 3.3.4. *Let $a \in \mathcal{X}$, $\hat{q} \in [0,1]^d$ and $\eta > 0$. Then $\Lambda(a, \nabla\Phi^*(\hat{a} - \eta\hat{q})) \le \eta^2\langle a, \hat{q}\rangle/2$.*

Proof (of Corollary 3.3.4). From Corollary 3.3.2 and Proposition 3.3.4, we have

$$\sum_{t=1}^{T} \left(\langle g_t, x^{(t)}\rangle - \langle g_t, z\rangle\right) \;\le\; \sum_{t=1}^{T} \frac{\eta_t\langle g_t, x^{(t)}\rangle}{2} + \frac{\ln d}{\eta_{T+1}}. \tag{3.30}$$

The algorithm's total cost at time t is denoted $A_t = \sum_{i\le t}\langle g_i, x^{(i)}\rangle$. Recall that the total cost of the best expert at time T is $L_T^* = \min_{z\in\Delta_d}\sum_{t=1}^{T}\langle g_t, z\rangle$ and the stepsize is $\eta_t = \sqrt{\ln(d)/(1+A_{t-1})}$. Substituting these into eq. (3.30),

$$\begin{aligned} A_T - L_T^* \;&\le\; \sqrt{\ln d}\left(\frac{1}{2}\sum_{t=1}^{T}\frac{\langle g_t, x^{(t)}\rangle}{\sqrt{1+A_{t-1}}} \;+\; \sqrt{1+A_T}\right) \\ &\le\; \sqrt{\ln d}\left(\sqrt{A_T} \;+\; \sqrt{A_T}+1\right) \end{aligned}$$

by Proposition B.1.1 with $a_i = \langle g_i, x^{(i)}\rangle$ and $u = 1$. Rewriting the previous inequality, we have shown that

$$A_T - L_T^* \;\le\; 2\sqrt{\ln(d)A_T} + \sqrt{\ln d}.$$

By Proposition B.1.2 we obtain

$$A_T - L_T^* \;\le\; 2\sqrt{\ln(d)L_T^*} + \sqrt{\ln d} + 2\,(\ln d)^{3/4} + 4\ln d.$$

Since the left-hand side equals $\text{Regret}(T,z)$, the result follows. □

Now we compare our bound with some existing results in the literature: our constant term of 2 obtained in Corollary 3.3.4 is better than the constant $(\sqrt{2}/(\sqrt{2}-$

1)) obtained by the doubling trick (Cesa-Bianchi and Lugosi, 2006, Exercise 2.8), and the constant ($2\sqrt{2}$) in Auer et al. (2002b)'s work but worse than the constant ($\sqrt{2}$) of the best known first-order regret bound (Yaroshinsky et al., 2004), which is obtained by a sophisticated algorithm. We also match the constant 2 of the Hedge algorithm from de Rooij et al. (2014, Theorem 8). Their result is actually more general; we could similarly generalize our analysis, but that would deviate too far from the main purpose of this chapter.

3.4 Comparing DS-OMD and DA

In this section we shall write the iterates of dual-stabilized OMD in two equivalent forms. First we shall write it in a proximal-like formulation similar to the mirror descent formulation in Beck and Teboulle (2003), shedding some light into the intuition behind dual-stabilization. We then write the iterates from DS-OMD in a form very similar to the original definition of DA in Nesterov (2009). The later will allow us to intuitively understand why OMD does not play well with dynamic step-size and to derive simple sufficient conditions under which DS-OMD and DA generate the same iterates, mimicking the relation between OMD and DA for a fixed stepsize.

Beck and Teboulle (2003) showed that the iterate $x^{(t+1)}$ for round $t+1$ from OMD is the unique minimizer over $\mathcal{X}$ of $\eta_t \langle g_t, \cdot \rangle + D_\Phi(\cdot, x^{(t)})$, where $g_t \in \partial f_t(x^{(t)})$. The next proposition extends this formulation to DS-OMD, recovering the result from Beck and Teboulle when $\gamma_t = 1$. The proof, which can be found in Appendix B.4, is a simple application of the optimality conditions of eq. (3.31).

Proposition 3.4.1. *Let $\{f_t\}_{t\geq 1}$ be a sequence of convex functions with $f_t \colon \mathcal{X} \to \mathbb{R}$ for each $t \geq 1$. Let $\eta \colon \mathbb{N} \to \mathbb{R}_{>0}$ and $\gamma \colon \mathbb{N} \to [0,1]$. Let $\{x^{(t)}\}_{t\geq 1}$ and $\{g_t\}_{t\geq 1}$ be as in Algorithm 3. Then, for any $t \geq 1$,*

$$\{x^{(t+1)}\} = \operatorname*{arg\,min}_{x\in\mathcal{X}} \Big(\gamma_t \big(\eta_t \langle g_t, x \rangle + D_\Phi(x, x^{(t)}) \big) + (1-\gamma_t) D_\Phi(x, x^{(1)}) \Big). \tag{3.31}$$

In spite of their similar descriptions, Orabona and Pál (2018) showed that OMD and DA may behave in extremely different ways even on the well-studied experts' problem with similar choices of stepsizes. This extreme difference in behavior is not

clear from the classical algorithmic description of these methods as in Algorithm 2. In the case of DA, it is well-known that DA can be seen as an instance of the FTRL algorithm; see Bubeck (2015, §4.4) or Hazan (2016, §5.3.1). More specifically, if $\{x^{(t)}\}_{t\geq 1}$ and $\{g_t\}_{t\geq 1}$ are as in Algorithm 2 with the DA update, then for every $t \geq 0$, [4]

$$\{x^{(t+1)}\} = \operatorname*{arg\,min}_{x\in\mathcal{X}} \left(\eta_{t+1} \sum_{i=1}^{t} \langle g_i, x\rangle - \langle \hat{x}^{(1)}, x\rangle + \Phi(x) \right). \tag{3.32}$$

In the next theorem, proven in Appendix B.4, we write DS-OMD in a similar form, but with vectors from the normal cone of $\mathcal{X}$ creeping into the formula due to the back and forth between the primal and dual spaces. Recall that the **normal cone** of $\mathcal{X}$ at a point $x \in \mathcal{X}$ is the set

$$N_{\mathcal{X}}(x) := \{\, p \in \mathbb{R}^d \mid \langle p, z - x\rangle \leq 0 \;\forall\, z \in \mathcal{X} \,\}.$$

The result in McMahan (2017, Theorem 11) is similar but slightly more intricate due to the use of time-varying mirror maps. Moreover, this result does not directly apply when we have stabilization.

Theorem 3.4.1. *Let $\{f_t\}_{t\geq 1}$ with $f_t : \mathcal{X} \to \mathbb{R}$ be a sequence of convex functions and let $\eta : \mathbb{N} \to \mathbb{R}_{>0}$ be non-increasing. Let $\{x^{(t)}\}_{t\geq 1}$ and $\{g_t\}_{t\geq 1}$ be as in Algorithm 3. Then, there are $\{p_t\}_{t\geq 1}$ with $p_t \in N_{\mathcal{X}}(x^{(t)})$ for all $t \geq 1$ such that, if $\gamma_i = 1$ for all $i \geq 1$, then for all $t \geq 0$*

$$\{x^{(t+1)}\} = \operatorname*{arg\,min}_{x\in\mathcal{X}} \left(\sum_{i=1}^{t} \langle \eta_i g_i + p_i, x\rangle - \langle \hat{x}^{(1)}, x\rangle + \Phi(x) \right) \tag{3.33}$$

and if $\gamma_i = \frac{\eta_{i+1}}{\eta_i}$ for all $i \geq 1$, then for all $t \geq 0$

$$\{x^{(t+1)}\} = \operatorname*{arg\,min}_{x\in\mathcal{X}} \left(\eta_{t+1} \sum_{i=1}^{t} \langle g_i + p'_i, x\rangle - \langle \hat{x}^{(1)}, x\rangle + \Phi(x) \right). \tag{3.34}$$

where $p'_t := \frac{1}{\eta_t} p_t \in N_{\mathcal{X}}(x^{(t)})$ for every $t \geq 1$.

With the above theorem, we may compare the iterates generated from DA, OMD, and DS-OMD by comparing the formulas eq. (3.32), eq. (3.33), and eq. (3.34). For

[4] The $\langle \nabla\Phi(x^{(1)}), x\rangle$ term disappears if $x^{(1)}$ minimizes Φ on $\mathcal{X}$.

the simple unconstrained case where $\mathcal{X} = \mathbb{R}^d$ we have $N_{\mathcal{X}}(x^{(t)}) = \{0\}$ for each $t \geq 1$ and DA and DS-OMD are identical. However, if the stepsize is not constant, OMD *is not* equivalent to the latter methods. In particular, if $\eta_t \propto 1/\sqrt{t}$, eq. (3.33) shows that the subgradients of the earlier-seen functions have a bigger weight on the iterates if compared to the subgradients of functions from later rounds. In other words, OMD may be sensitive to the ordering of the functions, and adversarial orderings may affect its performance.

When $\mathcal{X}$ is an arbitrary convex set, DA and DS-OMD are not necessarily equivalent anymore due to the vectors from the normal cone of $\mathcal{X}$. If we know that the iterates live in the relative interior of $\mathcal{X}$, the next lemma shows that these vectors do not affect the set of minimizers from eq. (3.34).

Lemma 3.4.1. *For any $\mathring{x} \in \operatorname{ri}\mathcal{X}$, we have $N_{\mathcal{X}}(\mathring{x}) = (-N_{\mathcal{X}}(\mathring{x})) \cap N_{\mathcal{X}}(\mathring{x})$. In particular, for any $p \in N_{\mathcal{X}}(\mathring{x})$ we have $\langle p, x \rangle = \langle p, \mathring{x} \rangle$ for every $x \in \mathcal{X}$.*

With this lemma, we can easily derive simple and intuitive conditions under which DS-OMD and DA are equivalent.

Corollary 3.4.1. *Let $\mathcal{D} \subseteq \mathbb{R}^d$ be the interior of the domain of Φ, let $\{x^{(t)}\}_{t\geq 1}$ be the DS-OMD iterates as in Algorithm 3 and let $\{x^{(t)'}\}_{t\geq 1}$ be the DA iterates as in Algorithm 2 with DA updates. If $\mathcal{D} \cap \mathcal{X} \subseteq \operatorname{ri}\mathcal{X}$, then $x^{(t)} = x^{(t)'}$ for each $t \geq 1$.*

Proof. Let $t \geq 1$. Since $x^{(t)} = \Pi^{\Phi}_{\mathcal{X}}(y^{(t)})$, where $y^{(t)}$ is as in Algorithm 3, Lemma B.4.2 implies $x^{(t)} \in \mathcal{D} \cap \mathcal{X} \subseteq \operatorname{ri}\mathcal{X}$. By Lemma 3.4.1 we have that the vectors on the normal cone in eq. (3.34) do not affect the set of minimizers, which implies that eq. (3.32) and eq. (3.34) are equivalent. □

An important special case of the above corollary is the prediction with expert advice setting as in Section 3.3.2, where $\mathcal{D} = \mathbb{R}^d_{>0}$ and $\mathcal{X}$ is the simplex Δ_d. In this setting, $\mathcal{X} \cap \mathcal{D} = \{\, x \in (0,1)^d \mid \sum_{i=1}^d x_i = 1 \,\} = \operatorname{ri}\mathcal{X}$. By the previous corollary DS-OMD and DA produce the same iterates in this case even for dynamic stepsizes. Classical OMD and DA were already known to be equivalent in the experts' setting for a *fixed* learning rate (Hazan, 2016, §5.4.2). In contrast, with a dynamic stepsize, the DA and OMD iterates are certainly different, since OMD with a dynamic learning rate may have linear regret (Orabona and Pál, 2018), whereas DA has sublinear regret.

3.5 Discussion

In this chapter we modified OMD via *stabilization* in order to guarantee sublinear regret even when using the method with a dynamic stepsize. We showed that (primal and dual) stabilized-OMD recover the regret bounds enjoyed by DA in the anytime setting, presented some applications of our results, and analyzed the similarities and differences between DS-OMD, OMD, and DA.

Our bounds for the problem of prediction with expert advice nearly match the current state-of-the-art. A distinctive feature of our proofs are their relative simplicity if compared to other results from the literature. It is our hope that the simplicity of our analysis framework allows it to be extended to other problems. Moreover, the modularity of our proofs allowed us to extend this analysis for DA, a fact interesting on its own since drastically different analysis techniques are usually used to analyze DA in the literature (such as the Follow the Leader-Be the Leader Lemma and optimality conditions of eq. (3.32), see Section 2.3 from Shalev-Shwartz (2012) for an example). This together with our analysis from Section 3.4 helps demystify the connections between DA and OMD, since in spite of having similar descriptions they had extremely different analyses and behaved wildly differently in some scenarios. We believe that a better understanding between the differences between DA and OMD will be helpful in future applications and in the design of new algorithms.

Chapter 4

Fast convergence of stochastic subgradient descent under interpolation

4.1 Background and motivation

We consider the empirical-risk minimization problem:

$$\min_{x\in\mathbb{R}^d} f(x) \coloneqq \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{P}$$

where n is the number of data points, and f_i's are continuous functions that measures how well our model x fit the data. The empirical-risk minimization formulation is prevalent in data-fitting problems—from problems that as simple as linear regression to complicated modern applications such as using deep neural networks for image classification and language modeling.

Stochastic (sub)gradient descent (SGD) is an simple first-order algorithm with the update rule[1]:

$$x^{(t+1)} = x^{(t)} - \eta_t g_t \quad g_t \in \partial f_i(x^{(t)}),$$

[1] One could also update the iterate based on a small batch of data points, which is known as the mini-batch SGD. In this thesis, we consider the single sample case for simplicity.

objective $f(x)$	smooth	nonsmooth	strongly cvx
GD	$\mathcal{O}(\varepsilon^{-1})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\log(\varepsilon^{-1}))$
SGD	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-1})$

Table 4.1: Iteration complexity of deterministic gradient and stochastic gradient methods.

where i is uniform randomly sampled from $\{1,2,\ldots,n\}$ in each iteration. The iteration complexities of SGD and GD for convex objective functions are summarized in Table 4.1. Although in general SGD converges slower than GD, a remarkable property of SGD is that both the per iteration cost and the convergence rate of SGD are **independent** from n. This property makes SGD particularly effective for (P) when n is large. In fact, SGD and its variants are indeed the most popular algorithms for modern big data applications including image classification and language modeling.

Given the empirical success and popularity of SGD, a huge line of works have been devoted into

- designing new variants of SGD to improve its convergence either theoretically or empirically;
- refining the classical analysis of SGD in various settings e.g., convex and nonconvex , and thus better understand SGD from a theoretical point of view.

This chapter belongs to the latter category. Next, we will briefly review some milestones in the recent development of SGD and describe the motivation for this chapter.

4.1.1 Practical algorithms based on SGD

SGD is the state-of-the-art algorithm for big data problems. Many practical algorithms developed for a wide range of applications are related to SGD. For example, the randomized Kaczmarz algorithm (Kaczmarz, 1937; Needell et al., 2014) for solving linear systems and the Pegasos algorithm (Shalev-Shwartz et al., 2007) for solving linear support vector machines. In the field of deep learning and reinforcement learning, specialized algorithms such as ADAM (Kingma and Ba, 2015), RMSprop (Tieleman and Hinton, 2012), TRPO (Schulman et al., 2015) and

PPO (Schulman et al., 2017), etc. are developed to adapt SGD to different problem structures. Beside efficiency, SGD and its variants are believed to exhibit *implicit* regularization (Gunasekar et al., 2017), which is another factor that makes SGD the dominate algorithm for training machine learning models.

4.1.2 Parallel and distributed SGD

The computation performance of single processor is reaching its limit due to energy dissipation. Designing scalable algorithms that obtain near-linear speed up with multiple computing cores to address real-world huge scale problems becomes an important research topic. Under such background, many parallel SGD algorithms (Goyal et al., 2017; Li et al., 2014; Lian et al., 2015; Liu et al., 2015; Recht et al., 2011; You et al., 2018) are proposed to adapt SGD to multi-CPUs, multi-GPUs or multi-machines setting. Both synchronous and asynchronous parallel SGD algorithms have been studied heavily in both theory and practice. On the theory side, Bertsekas and Tsitsiklis first formally brought optimization and parallel computation together, their seminal work (Bertsekas and Tsitsiklis, 1989) established the theoretical foundations of parallel optimization and the theoretical tools therein are still being used nowadays (the book from Bertsekas and Tsitsiklis is very forward-thinking in 1989 since parallel computing is less of interest at that time). More recently, HOGWILD! (Recht et al., 2011), an asynchronous parallel SGD with theoretical guarantees and empirical success, brought resurged interest in parallel SGD for machine learning tasks; the convergence of asynchronous parallel SGD in various settings are also studied by subsequent works (Lian et al., 2015; Liu et al., 2015). On the practical side, researchers are keeping decreasing the training time of neural networks on the imagenet dataset by scaling SGD to hundreds or even thousands of GPUs. The training time is reduced from the original 6 days (Krizhevsky et al., 2012) to 1 hour in 2017 (Goyal et al., 2017), around 10 minutes in 2018 (You et al., 2018) and less than 5 minutes by the time of the completion of this thesis[2].

[2]Benchmark on imagenet training: https://dawn.cs.stanford.edu/benchmark/ImageNet/train.html, last access time 07/02/2021.

4.1.3 Variance reduction

The variance reduction technique that originates from Schmidt et al. (2017) is certainly an exciting advancement of SGD in the last decade. The resulting variance-reduced SGD algorithms including SAG (Schmidt et al., 2017), SVRG (Johnson and Zhang, 2013), SAGA (Defazio et al., 2014) and SARAH (Nguyen et al., 2017) improved the iteration complexities of SGD from $\mathcal{O}(\varepsilon^{-2})$ to $\mathcal{O}(\varepsilon^{-1})$ for convex and smooth objectives and $\mathcal{O}(\varepsilon^{-1})$ to $\mathcal{O}(\log(\varepsilon^{-1}))$ for strongly convex and smooth objectives while maintaining the cheap per iteration cost. These rates matches the rates of deterministic GD.

4.1.4 SGD with the interpolation condition

Although variance-reduced SGD theoretically attains a faster convergence rate than vanilla SGD, however practitioners find that it cannot outperform vanilla SGD in the practice of training modern machine learning models. This discrepancy between theory and practice was recently filled by Ma et al. (2018), who showed that for over-parameterized models that satisfy the *interpolation* condition, vanilla SGD has an inherent variance reduction functionality as iterates converging to the solution. The interpolation condition means that our model has the ability to fit all training data perfectly; we will give a more formal description of this concept in Section 4.2. Therefore, in the practice of training over-parameterized models, vanilla SGD already has the fast convergence rate that similar to the deterministic GD, and further explicitly adding the variance reduction step to SGD will not give us a better convergence.

The detailed convergence rate of SGD for convex objective functions under the interpolation condition is summarized in Table 4.2. As we can see from the table, there is a gap between the convergence rate of SGD for smooth and nonsmooth objectives under interpolation. However, in the practice of training machine learning models, the nonsmoothness from the model does not cause much trouble (Glorot et al., 2011; Goodfellow et al., 2016). Neural networks with nonsmooth activation function such as ReLU activation can usually be trained as fast as (or even faster than) the one with smooth activation function such as softplus. Therefore there is a discrepancy between theory and practice in the nonsmooth world.

objective $f(x)$	smooth	nonsmooth	SC + smooth	SC + nonsmooth
SGD	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\varepsilon^{-1})$	$\mathcal{O}(\varepsilon^{-1})$
SGD + interpolation	$\mathcal{O}(\varepsilon^{-1})$	$\mathcal{O}(\varepsilon^{-2})$	$\mathcal{O}(\log(\varepsilon^{-1}))$	$\mathcal{O}(\varepsilon^{-1})$

Table 4.2: Iteration complexity of SGD with and without the interpolation condition. SC stands for strongly convex.

In this chapter, we aim to fill the gap between the convergence rates of SGD for smooth and nonsmooth objectives under the interpolation condition. We will describe some semi-smoothness properties of the empirical-risk minimization problem. These properties, together with the interpolation condition, allow us to prove that stochastic **subgradient** method has iteration complexity $\mathcal{O}(\varepsilon^{-1})$ for convex objectives, and $\mathcal{O}(\log(\varepsilon^{-1}))$ for strongly-convex objectives. These rates improved the classic bounds $\mathcal{O}(\varepsilon^{-2})$ and $\mathcal{O}(\varepsilon^{-1})$ for convex and strongly-convex objectives and match the convergence rates of SGD for convex and smooth objectives under interpolation. We also prove that the iteration bound $\mathcal{O}(\varepsilon^{-1})$ is optimal in the convex and interpolation setting. In contrast to the case with a smooth objective function, subgradient-based methods cannot be further accelerated for nonsmooth model—even with the interpolation assumption.

4.2 Preliminaries

First, we introduce a terminology used in this chapter. We use SSGD to denote the stochastic **sub**gradient descent, which is widely used when the objective function is nonsmooth.

We consider our objective to be the unconstrained empirical risk-minimization problem

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad f(x) \coloneqq \frac{1}{n}\sum_{i=1}^{n} f_i(x) \qquad \text{where each} \qquad f_i(x) \coloneqq \ell(h_i(x)) \tag{4.1}$$

and n is the number of data points. Throughout the paper, we use x^* to denote any solution of eq. (4.1), and thus $f^* \coloneqq f(x^*)$ is the optimal objective value. We assume that the loss function $\ell : \mathbb{R} \to \mathbb{R}_{\geq 0}$ is 1-dimensional function that is nonnegative

with $\inf \ell = 0$, convex and 1-smooth, i.e.,

$$|\ell'(\alpha) - \ell'(\beta)| \leq |\alpha - \beta| \quad \forall \alpha, \beta \in \mathbb{R}.$$

Without loss of generality, we also assume $\ell(0) = 0$. Common examples for the loss function include

- 2-norm loss: $\ell(x) = x^2$;
- logistic loss: $\ell(x) = \log(1 + \exp(x))$;
- 2-norm hinge loss: $\ell(x) = (\max\{0, x\})^2$.

The n functions h_i's are Lipschitz continuous with respect to a fixed parameter L:

$$|h_i(x) - h_i(y)| \leq L\|x - y\| \quad \forall x, y \in \mathbb{R}^d, \ \forall i \in [n].$$

We make no assumption on their smoothness properties. Many key machine learning tasks can be formulated as (4.1), including training deep neural networks with nonsmooth activations, such as the ReLU function. Here and throughout, the function $\|\cdot\|$ is the 2-norm of a vector, unless otherwise specified.

Our analysis relies on the Clark *generalized* gradient (Clarke, 1990) of the nonconvex and nonsmooth function h, defined as the convex hull of all valid limiting gradients:

$$\partial h(x) := \operatorname{conv}\{u \mid u = \lim_{k \to \infty} \nabla h(x_k), \ x_k \to x\},$$

This definition additionally requires h to be almost everywhere differentiable by Rademacher's theorem. We refer readers to Clarke (1990), and more recently, to Zhang et al. (2020), who use this generalized gradient in a related analysis. These properties of the generalized gradient are needed for our analysis:

(chain rule) $\qquad \partial f_i(x) = \ell'(h(x)) \cdot \partial h(x),$

(gradient bound) $\qquad \|\partial h_i(x)\| \leq L, \ \forall x \in \mathbb{R}^d, \ \forall i \in [n].$

The second property follows from the L-Lipschitz continuity of each function h_i. We define the norm of the generalized gradient at a vector x as

$$\|\partial h_i(x)\| = \sup\{\|z\| \mid z \in \partial h_i(x)\}.$$

Algorithm 6 Stochastic subgradient descent. The learning rate function $\eta_t : \mathbb{N} \to \mathbb{R}_+$ returns the learning rate at iteration t.

Initialize: $x^{(1)} \in \mathbb{R}^d$
for $t = 1, 2, \ldots$ **do**
  select $i \in \{1, 2, \ldots, n\}$ uniformly at random
  compute $g^{(t)} \in \partial f_i(x^{(t)})$
  $x^{(t+1)} = x^{(t)} - \eta_t g^{(t)}$
end for

Algorithm 6 describes the SSGD method.

The interpolation condition

By our assumption on f_i's, we can immediately conclude that $f \geq 0$. The interpolation condition means that our model has the ability to fit all training samples perfectly and therefore achieve zero training loss at the solution. Formally, the interpolation condition is defined as $f^* = 0$. It has been shown that the interpolation condition holds for overparameterized neural networks (Jacot et al., 2018), and the interpolation condition is gaining increasing interest in recent years. Note that we do not make interpolation as an assumption here and most of our analysis in the following sections holds in general without assuming interpolation. We will highlight the interpolation condition when the presented results require the interpolation condition.

4.3 Main results

We present some semi-smoothness properties and the convergence analysis in this section.

4.3.1 Bounds and Lipschitz properties of the generalized gradient

Our analysis hinges on establishing that the objective function f is "almost" differentiable at points with small objective value. This is implied by the following proposition, which holds even without the interpolation condition.

Proposition 4.3.1 (Generalized growth condition)**.** *Assume that the assumptions*

stated in Section 4.2 hold. Then for all d-vectors x,

$$\|\partial f_i(x)\|^2 \leq 2L^2 f_i(x) \qquad \forall i \in [n]. \tag{4.2}$$

Consequently,

$$\|\partial f(x)\|^2 \leq \frac{1}{n}\sum_{i=1}^{n} \|\partial f_i(x)\|^2 \leq 2L^2 f(x). \tag{4.3}$$

Eq. (4.3) implies that if the objective value $f(x) = 0$, then the generalized gradient contains only the origin: $\partial f(x) = \{0\}$. It thus follows from (Clarke, 1981, Property 10) that f is differentiable at any point with zero objective value. This means that if there is a solution with a zero value, then it must be a fixed point of the subgradient method. This property does not hold for functions that are nonsmooth at solution (for example the absolute value function $f(x) = |x|$) and makes it possible for the subgradient method to converge to solution with constant learning rate.

Proof of Proposition 4.3.1. Fix an arbitrary d-vector x. The subdifferential $\partial f_i(x) = \ell'(h_i(x)) \cdot \partial h_i(x)$. Because each function h_i is L-Lipschitz continuous, we can then deduce that

$$\begin{aligned}
\|\partial f_i(x)\|^2 &\leq L^2 \left[\ell'(h_i(x))\right]^2 \\
&\overset{\text{(i)}}{=} L^2(\ell'(h_i(x)) - \ell'(0))^2 \\
&\overset{\text{(ii)}}{\leq} 2L^2(\ell(h_i(x)) - \ell(0)) \\
&= 2L^2 f_i(x).
\end{aligned} \tag{4.4}$$

Step (i) follows from the assumption that $\ell(0) = \min_{\lambda \in \mathbb{R}} \ell(\lambda) = 0$, which implies $\ell'(0) = 0$. Step (ii) follows from the fact that any convex L-smooth function $u : \mathbb{R}^n \to \mathbb{R}$ satisfies the bound

$$u(a) - u(b) - \langle \nabla u(b), a - b \rangle \geq \frac{1}{2L}\|\nabla u(a) - \nabla u(b)\|^2;$$

see Nesterov (2014, Theorem 2.1.5). Make the identifications

$$u = \ell, \quad a = h_i(x), \quad b = 0$$

to immediately obtain eq. (4.4) and thus the proof for eq. 4.2.

Eq. (4.3) can be obtained directly from Jensen's inequality:

$$\|\partial f(x)\|^2 = \left\| \frac{1}{n} \sum_{i=1}^{n} \partial f_i(x) \right\|^2 \leq \frac{1}{n} \sum_{i=1}^{n} \|\partial f_i(x)\|^2 \leq \frac{1}{n} \sum_{i=1}^{n} 2L^2 f_i(x) = 2L^2 f(x).$$

□

The composite structure of the functions f_i allows us to develop a semi-Lipschitz bound on their generalized gradients. Moreover, we show that it is possible to obtain a global convex majorant for each function f_i, which holds without assuming convexity. Interestingly, eq. (4.6) overlaps with the "semi-smoothness" property of over-parameterized neural networks derived by Allen-Zhu et al. (2019, Theorem 4). In contrast to the result from Allen-Zhu et al. (2019), however, our result does not use any special properties of over-parameterized neural networks. Proposition 4.3.2 may thus be of more general interest.

Proposition 4.3.2 (Semi-smoothness)**.** *Assume that the assumptions stated in Section 4.2 hold. Then for all vectors x_1 and x_2, and each $i \in [n]$,*

$$\|\partial f_i(x_2) - \partial f_i(x_1)\| \leq L^2 \|x_2 - x_1\| + 2L\sqrt{2 \min\{f_i(x_1), f_i(x_2)\}}, \tag{4.5}$$

and

$$f_i(x_2) \leq f_i(x_1) + \langle \partial f_i(x_1), x_2 - x_1 \rangle + \frac{L^2}{2} \|x_2 - x_1\|^2 + 2L\|x_2 - x_1\| \sqrt{2 f_i(x_1)}. \tag{4.6}$$

The proof of Proposition 4.3.2 is rather tedious, interested readers can find it in Appendix C.

4.3.2 Convergence rate of stochastic subgradient descent

We now present a global convergence analysis for the SSGD algorithm under the additional assumption that the objective f of eq. (4.1) is convex. We develop a bound on the expected progress of the objective value that depends on the optimal value f^*, rather than on the Lipschitz bound on the function itself, which is the usual bound in the literature. Our proof is based on a simple modification of the classical proof of subgradient descent method.

Theorem 4.3.1 (Global convergence rate of SSGD)**.** *Assume f is convex. Then for any positive integer T and any learning rate function η_t that satisfies $\sum_{t=1}^{T}(\eta_t - L^2\eta_t^2) > 0$,*

$$\min_{t\in[T]} \mathbb{E}[f(x^{(t)}) - f^*] \leq \frac{\|x^{(1)} - x^*\|^2 + 2L^2 f^* \sum_{t=1}^{T} \eta_t^2}{2\sum_{t=1}^{T}(\eta_t - L^2\eta_t^2)}. \tag{4.7}$$

Proof. Let $g_i^{(t)} \in \partial f_i(x^{(t)})$. Then each iterate $x^{(t)}$ satisfies the bound

$$\begin{aligned}
&\mathbb{E}\left[\|x^{(t+1)} - x^*\|^2 \;\middle|\; x^{(t)}\right] \\
=\;& \frac{1}{n}\sum_{i=1}^{n} \|x^{(t)} - \eta_t g_i^{(t)} - x^*\|^2 \\
=\;& \|x^{(t)} - x^*\|^2 - 2\eta_t \left\langle \frac{1}{n}\sum_{i=1}^{n} g_i^{(t)}, x^{(t)} - x^* \right\rangle + \frac{1}{n}\sum_{i=1}^{n} \eta_t^2 \|g_i^{(t)}\|^2 \\
\overset{\text{(i)}}{\leq}\;& \|x^{(t)} - x^*\|^2 - 2\eta_t (f(x^{(t)}) - f(x^*)) + \frac{1}{n}\sum_{i=1}^{n} \eta_t^2 \|g_i^{(t)}\|^2 \qquad (4.8) \\
\overset{\text{(ii)}}{\leq}\;& \|x^{(t)} - x^*\|^2 - 2\eta_t (f(x^{(t)}) - f(x^*)) + 2\eta_t^2 L^2 f(x^{(t)}), \qquad (4.9)
\end{aligned}$$

where (i) follows from the convexity of f, and (ii) follows from Proposition 4.3.1. Take expectations on both sides of the inequality (4.9) and rearrange to obtain

$$(2\eta_t - 2L^2\eta_t^2) \cdot \mathbb{E}[(f(x^{(t)}) - f^*)] \leq \mathbb{E}[\|x^{(t)} - x^*\|^2] - \mathbb{E}[\|x^{(t+1)} - x^*\|^2] + 2L^2\eta_t^2 f^*. \tag{4.10}$$

Summing inequality (4.10) over $t \in \{1, 2, \ldots, T\}$ yields

$$\sum_{t=1}^{T} (2\eta_t - 2L^2\eta_t^2)\mathbb{E}[f(x^{(t)}) - f^*] \leq \|x^{(1)} - x^*\|^2 + 2L^2 f^* \sum_{t=1}^{T} \eta_t^2.$$

Divide both sides by $2\sum_{t=1}^{T}(\eta_t - L^2\eta_t^2) > 0$, we obtain the desired result. □

This proof mirrors closely the classical proof of SSGD, which assumes that the subdifferential of each f_i is bounded, i.e., $\|\partial f_i(x)\| \leq G$ for some constant G. We use Proposition 4.3.1 in eq. (4.8) to avoid the bounded subgradient assumption and to express the convergence rate using the minimal value f^*. This modification allows us to leverage the interpolation assumption that $f^* = 0$. We use Theorem 4.3.1 to immediately deduce the following convergence rate result.

Corollary 4.3.1 (Global convergence rate of SSGD with constant learning rate)**.** *Assume that f is convex and that the learning rate $\eta_t = 1/(2L^2)$ is constant for all $t > 0$. Then for any positive integer T, the SSGD iterates $x^{(t)}$ satisfy*

$$\min_{t \in [T]} \mathbb{E}[f(x^{(t)}) - f^*] \leq (2L^2/T)\|x^{(1)} - x^*\|^2 + f^*.$$

Furthermore, when $f^* = 0$ (interpolation holds),

$$\min_{t \in [T]} \mathbb{E}[f(x^{(t)})] - f^*] \leq (2L^2/T)\|x^{(1)} - x^*\|^2.$$

Theorem 4.3.1 indicates that the SSGD method converges at rate $\mathcal{O}(\varepsilon^{-1})$ when interpolation holds. This rate matches the convergence rate of SGD for smooth objective functions under interpolation (Schmidt and Le Roux, 2013). When interpolation does not hold, Theorem 4.3.1 implies that SSGD, on expectation, could obtain objective value lower than $2f^* + \varepsilon$ in $\mathcal{O}(\varepsilon^{-1})$ time, which could be close to f^* when f^* is close to 0 (interpolation nearly holds).

Next, we derive convergence rate under the stronger assumption that f is strongly convex, which means that

$$f(x_1) \geq f(x_2) + \langle g, x_1 - x_2 \rangle + \frac{\mu}{2}\|x_1 - x_2\|^2 \quad \forall x_1, x_2 \in \mathbb{R}^d,\ \forall g \in \partial f(x_2) \quad (4.11)$$

for some constant $\mu > 0$. Note that recent works indicated that in order to prove linear convergence rate, the strong convexity assumption can be relaxed to other weaker assumptions that could hold even for some nonconvex functions (Karimi et al., 2016; Qian et al., 2019). For simplicity, we assume strong convexity in our analysis.

Theorem 4.3.2 (Global convergence rate of SSGD under strong convexity)**.** *Assume that f is μ-strongly convex and that the learning rate $\eta_t = 1/L^2$ is constant for all $t > 0$. Then for any positive integer T, the SSGD iterates $x^{(t)}$ satisfy*

$$\mathbb{E}[\|x^{(T)} - x^*\|^2] \leq \left(1 - \frac{\mu}{L^2}\right)^{T-1} \|x^{(1)} - x^*\|^2 + \frac{2}{\mu} f^*. \tag{4.12}$$

Proof. Use the definition of the SSGD iteration to obtain

$$\begin{aligned}
&\mathbb{E}\left[\|x^{(t+1)} - x^*\|^2 \;\middle|\; x^{(t)}\right] \\
&\overset{\text{(i)}}{=} \|x^{(t)} - x^*\|^2 - 2\eta_t \left\langle \frac{1}{n}\sum_{i=1}^{n} g_i^{(t)}, x^{(t)} - x^* \right\rangle + \frac{1}{n}\sum_{i=1}^{n} \eta_t^2 \|g_i^{(t)}\|^2 \\
&\overset{\text{(ii)}}{\leq} \|x^{(t)} - x^*\|^2 - 2\eta_t (f(x^{(t)}) - f^*) - \mu\eta_t \|x^* - x^{(t)}\|^2 + \frac{1}{n}\sum_{i=1}^{n} \eta_t^2 \|g_i^{(t)}\|^2 \\
&\overset{\text{(iii)}}{\leq} (1 - \mu\eta_t)\|x^{(t)} - x^*\|^2 - 2\eta_t (f(x^{(t)}) - f^*) + 2\eta_t^2 L^2 f(x^{(t)}) \qquad (4.13) \\
&\overset{\text{(iv)}}{=} \left(1 - \frac{\mu}{L^2}\right)\|x^{(t)} - x^*\|^2 + \frac{2}{L^2} f^*, \qquad (4.14)
\end{aligned}$$

where (i) follows from the same argument of the proof of Theorem 4.3.1; (ii) follows from the μ-strong convexity of f (see eq. (4.12)); (iii) follows from eq. (4.2); and (iv) follows from the definition of the learning rate $\eta_t = 1/L^2$.

Taking expectation to both sides of eq. (4.14) and recursively apply it to $t \in \{1, 2, \ldots, T\}$ to deduce

$$\begin{aligned}
\mathbb{E}[\|x^{(T)} - x^*\|^2] &\leq \left(1 - \frac{\mu}{L^2}\right)^{T-1} \|x^{(1)} - x^*\|^2 + \sum_{t=0}^{T-2} \left(1 - \frac{\mu}{L^2}\right)^{t} \frac{2}{L^2} f^* \\
&\overset{\text{(i)}}{\leq} \left(1 - \frac{\mu}{L^2}\right)^{T-1} \|x^{(1)} - x^*\|^2 + \frac{2}{\mu} f^*,
\end{aligned}$$

where (i) follows from the fact that $\sum_{t=0}^{\infty}\left(1-\frac{\mu}{L^2}\right)^t = L^2/\mu$.

□

Theorem 4.3.2 indicates that the SSGD converges to the ball centered at x^* with radius $\sqrt{2f^*/\mu}$ at a linear rate. If interpolation also holds, SSGD converges to the solution linearly. Again, this rate matches the convergence rate of SGD for smooth and strongly convex objectives in the interpolation setting (Ma et al., 2018; Schmidt and Le Roux, 2013). Similar to Corollary 4.3.1, when interpolation is nearly satisfied, SSGD converges linearly to an $\sqrt{2f^*/\mu}$-approximate solution.

Corollary 4.3.1 and Theorem 4.3.2 also provide insight into the effect of learning rate schedules on the performance of SSGD. A learning rate schedule η_t that decays as $t^{-1/2}$ is optimal because it causes the algorithm to exhibit a complexity bound of $\mathcal{O}(\varepsilon^{-2})$, which is the theoretical lower bound (Nemirovski and Yudin, 1983). However, the learning rate schedule $\mathcal{O}(t^{-1/2})$ is slow in practice. Corollary 4.3.1 and Theorem 4.3.2 partially explain the discrepancy between the theory and practice: many machine learning models exhibit interpolation or near interpolation, and an aggressive constant learning-rate schedule works better than the conservative worst-case optimal learning rate $\eta_t = \mathcal{O}(t^{-1/2})$.

Other than the learning rate scheduling $\eta_t = 1/L^2$, we can adopt a more carefully tuned learning rate scheduling (Stich, 2019) to obtain a refined convergence rate. We start with eq. (4.13), by setting $\eta_t \leq 1/(2L^2)$, we obtain

$$\begin{aligned}
\mathbb{E}[\|x^{(T)}-x^*\|^2] &\leq (1-\mu\eta_t)\|x^{(t)}-x^*\|^2 - 2\eta_t(f(x^{(t)})-f^*) + 2\eta_t^2L^2f(x^{(t)}) \\
&\leq (1-\mu\eta_t)\|x^{(t)}-x^*\|^2 - 2(\eta_t-\eta_t^2L^2)(f(x^{(t)})-f^*) + 2\eta_t^2L^2f^* \\
&\leq (1-\mu\eta_t)\|x^{(t)}-x^*\|^2 - \eta_t(f(x^{(t)})-f^*) + 2\eta_t^2L^2f^*. \qquad (4.15)
\end{aligned}$$

The last line is true by the condition $\eta_t \leq 1/(2L^2)$. Eq. (4.15) coincides with Stich (2019, Lemma 1) by setting his $\gamma_t = \eta_t$, $\sigma^2 = 2L^2f^*$. Then we can directly apply Stich (2019, Theorem 5) to get the following corollary.

Corollary 4.3.2. *Assume that f is μ-strongly convex for some $\mu \geq 0$. For all positive integer T, there exist a constant learning rate scheduling $\eta_t := \alpha \leq 1/(2L^2)$,*

such that the SSGD iterates $x^{(t)}$ satisfy

$$\min_{t\in[T+1]} \mathbb{E}[f(x^{(t)}) - f^*] \leq 64L^2\|x^{(1)} - x^*\|^2 \exp\left(-\frac{\mu T}{4L^2}\right) + \frac{72L^2 f^*}{\mu T}$$

for $\mu > 0$, and

$$\min_{t\in[T]} \mathbb{E}[f(x^{(t)}) - f^*] \leq \frac{4L^2\|x^{(1)} - x^*\|^2}{T} + \frac{4L^2 f^*\|x^{(1)} - x^*\|}{\sqrt{T}}$$

for $\mu = 0$.

4.3.3 Lower bounds

We have proven that, under interpolation, SGD for smooth problems and SSGD for nonsmooth problems exhibit the same convergence rates. This causes us to consider the following questions:

- Is it possible to induce momentum-type acceleration for SSGD under interpolation? Vaswani et al. (2019) and Liu and Belkin (2020) showed recently that acceleration is possible for SGD under interpolation. It seems plausible that similar techniques could be used for nonsmooth problems.
- Can the composite structure $f_i = \ell \circ h_i$, which is central to our analysis, be used to establish an improved convergence rate for SSGD without interpolation? In other words, we know the lower bound on the iteration complexity for any subgradient method for nonsmooth functions is $\Omega(\varepsilon^{-2})$. What, then, is the lower bound for minimizing the structured nonsmooth function $\ell \circ h$?

Unfortunately, the answer to these questions is "no", as we show below. First, we disprove the first conjecture by deriving the lower iteration bound for any algorithm with access only to a subgradient oracle in the interpolation setting.

Theorem 4.3.3 (Lower bound with interpolation)**.** *Given $t < d$ and positive constants L and R, and an initial vector $x^{(1)}$. Let $\ell = \frac{1}{2}(\cdot)^2$. Then there exists an L-Lipschitz function h such that $f := l \circ h$ is convex, $f^* = \min_x f(x) = 0$, $\|x^{(1)} - x^*\| \leq$*

R such that

$$\min_{1\le s\le t} f(x^{(s)}) - f^* \ge \frac{L^2R^2}{2(t+1)}. \tag{4.16}$$

Proof. With out loss of generality, we assume the initial point $x^{(1)} = 0$. Let $h(x) = L\|x - x^*\|_\infty$, and

$$x^* = \Big[\underbrace{\frac{R}{\sqrt{t+1}} + \varepsilon, \frac{R}{\sqrt{t+1}} + \frac{\varepsilon}{2}, \ldots, \frac{R}{\sqrt{t+1}} + \frac{\varepsilon}{2^t}}_{t+1 \text{ entries}}, \quad \underbrace{0, \ldots, 0}_{(d-t-1) \text{ entries}} \Big],$$

where ε is some arbitrary constant greater than 0. It is easy to check that ℓ is 1-smooth, h is L-Lipschitz, $f = \ell \circ h$ is convex, and $f^* = 0$.

We follow the proof template from Nemirovski and Yudin (1983) and assume that any algorithm that uses only subgradient information generates iterates

$$x^{(s)} \in \operatorname{span}\{\partial f(x^{(1)}), \ldots, \partial f(x^{(t)})\}$$

for all $s \le t$. By our construction, we can further obtain $x^{(s)} \in \operatorname{span}\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_t\}$ for all $s \le t$. Therefore, for all $s \le t$,

$$f(x^{(s)}) - f^* \ge \frac{1}{2}\left(\frac{LR}{\sqrt{t+1}} + \frac{L\varepsilon}{2^{s-1}}\right)^2.$$

Note that the above holds $\forall \varepsilon > 0$. Taking $\varepsilon \to 0^+$, we completes the proof. □

Theorem 4.3.3 established $\Omega(\varepsilon^{-1})$ iteration complexity for subgradient method under interpolation. Combining this result with the $\mathcal{O}(\varepsilon^{-1})$ iteration complexity in Corollary 4.3.1, we can conclude that the rate $\mathcal{O}(\varepsilon^{-1})$ is optimal in the interpolation setting and no acceleration is possible for subgradient-based methods. Then we proceed to the lower bound without assuming interpolation.

Theorem 4.3.4 (Lower bound without interpolation)**.** *Given $t < d, L, R > 0$ and an initial point $x^{(1)}$. Let $\ell(\cdot) = \frac{1}{2}(\cdot)^2$. Then there exist an L-Lipschitz function h*

satisfying $f := l \circ h$ is convex, $\|x^{(1)} - x^*\| \leq R$ such that

$$\min_{1 \leq s \leq t} f(x^{(s)}) - f^* \geq \frac{L^2 R^2}{\sqrt{t+1}}. \tag{4.17}$$

Proof. Similar to the proof of Theorem 4.3.3, we set $x^{(1)} = 0$. But we change the construction of $h(x)$ to $h(x) = L(\|x - x^*\|_\infty + R)$, and x^* is defined in the same way as in the proof of Theorem 4.3.3:

$$x^* = \Big[\underbrace{\frac{R}{\sqrt{t+1}} + \varepsilon, \frac{R}{\sqrt{t+1}} + \frac{\varepsilon}{2}, \ldots, \frac{R}{\sqrt{t+1}} + \frac{\varepsilon}{2^t}}_{t+1 \text{ entries}}, \quad \underbrace{0, \ldots, 0}_{(d-t-1) \text{ entries}} \Big],$$

where ε is some arbitrary constant greater than 0. Following the same analysis of the proof of Theorem 4.3.3. $x^{(s)} \in \text{span}\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_t\}$ $\forall s \leq t$ and

$$f(x^{(s)}) - f^* \geq \frac{L^2}{2} \left(\frac{R}{\sqrt{t+1}} + \frac{\varepsilon}{2^{s-1}} + R \right)^2 - \frac{L^2 R^2}{2} \geq \frac{L^2 R^2}{\sqrt{t+1}} \qquad \forall s \leq t,$$

By taking $\varepsilon \to 0^+$, we have $\|x^{(1)} - x^*\| \leq R$ and completes the proof. □

Theorem 4.3.4 provides an $\Omega(\varepsilon^{-2})$ iteration complexity for subgradient-based methods for solving the structured function $\ell \circ h$. This matches the lower bound for solving general nonsmooth objective function with subgradient-based method, and implies that the structure $\ell \circ h$ itself without interpolation cannot give us an improved iteration complexity of subgradient-based methods.

4.4 Numerical experiments

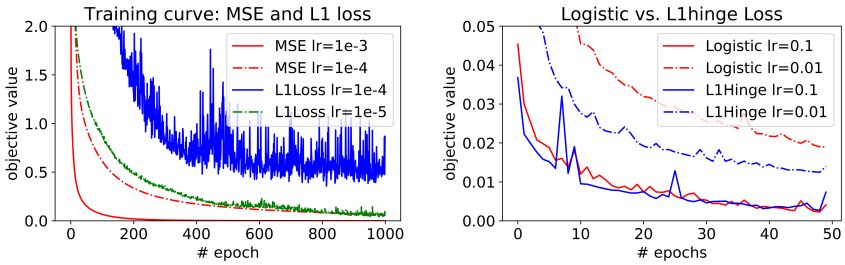
The smoothness of the loss function ℓ is crucial to our analysis. We now present some numerical experiments to compare the convergence of SSGD for training ReLU neural networks with smooth and nonsmooth loss functions. We denote $\{x_i\}_{i=1}^n$ as training samples and $\{y_i\}_{i=1}^n$ as training labels. $\hat{y}_i$ stands for the prediction of the i-th sample x_i from the trained model. Note that we use the letter x to represent the model in previous sections, to be consistent to convention, we use it to denote data points in this section.

4.4.1 Teacher-student setup

We randomly generate a small neural network with one hidden layer (16 neurons and ReLU activation) as the teacher network. The network takes 16 dimensional vectors as inputs and outputs a scalar. Then we generate 128 random vectors from the Gaussian distribution as our training data $\{x_i\}_{i=1}^{128} \subset \mathbb{R}^{16}$ and get their corresponding labels $\{y_i\}_{i=1}^{128}$ as the output of the teacher network. In order to ease the training and satisfy the interpolation assumption, we overparameterize the student neural network and set it to be a one hidden layer network with 512 neurons and ReLU activation. We train the student network with different loss functions: squared loss e.g., $(y_i - \hat{y}_i)^2$ and absolute loss e.g., $|y_i - \hat{y}_i|$ with different learning rates. The training curves are shown in Figure 4.1a. We can observe that the training curve with squared loss is smoother than the curve with absolute loss. For absolute loss, the performance of SSGD is more sensitive to the change of learning rate and we need to decrease the learning rate to obtain a lower objective value. These observations validate the importance of the smoothness of loss function under the interpolation setting.

4.4.2 Classify 4's and 9's on MNIST dataset

We train the `LeNet` (Lecun et al., 1998) on the MNIST dataset to classify 4's and 9's. To convert this task to a binary classification problem, we transform the labels $\{y_i\}_{i=1}^{n}$ to $\{-1,+1\}^n$. Then we run SSGD to train the model with difference loss functions: logistic loss e.g., $\log(1+\exp(-y_i\hat{y}_i))$ and L1-hinge loss e.g., $\max\{0, 1-y_i\hat{y}_i\}$ and with different learning rates. The training curves are presented in Figure 4.1b. Different from the teacher-student experiment, SSGD in this task perform similarly with smooth and nonsmooth loss functions. We conjecture that this is because the objective function with L1-hinge loss almost satisfies Proposition 4.3.1 locally at the solution, namely eq. 4.2 holds for most training samples in a neighbourhood of the solution. Our observation supports this conjecture. We observe that more that 95% of our final predictions $\hat{y}_i$'s satisfy the condition $|\hat{y}_i| > 2$. Since the L1-hinge loss is locally smooth when $|\hat{y}_i| > 2$, we can thus say that most training samples satisfy eq. 4.2 locally at the solution. While for the teacher-student training problem, its objective is nonsmooth at solution (when



(a) Teacher-student training **(b)** Classify 4's and 9's on MINST.

Figure 4.1: The performance of SSGD with smooth and nonsmooth loss functions.

zero residual is attained) since the absolute value function $\ell(x) := |x|$ is nonsmooth at 0. Therefore the objective of the teacher-student training problem does not satisfy Proposition 4.3.1 locally as solution and running SSGD to solve it could suffer from slow convergence.

4.5 Discussion

An empirical-risk minimization problem based on composite functions has sufficient structure to allow for a tight convergence analysis that explains the effectiveness of stochastic subgradient descent methods on nonsmooth problems with interpolation. Surprisingly, the complexity bounds $\mathcal{O}(\varepsilon^{-1})$ and $\mathcal{O}(\log(\varepsilon^{-1}))$ that we prove under interpolation match those of stochastic gradient descent for smooth functions.

Chapter 5

Conclusion and future work

This thesis contributes to a better understanding of the theory behind some widely used first-order optimization algorithms for problems that satisfy specific structures. We must admit that the progress described in this thesis is just a small step towards understanding first-order methods thoroughly. There are still a lot of interesting and important open problems on first-order optimization algorithms unsolved. Next, we summarize some of the results obtained in this thesis and discuss possible future directions.

5.1 Coordinate optimization

In Chapter 2, we provided theoretical justification for the implicit screening functionality of GCD. However, our analysis only works for composite problems with 1-norm regularization or non-negative constraints. Whether it is possible to extend our analysis to general regularizers that are nonsmooth at origin is left unanswered. In Chapter 2, we also established the fast convergence rate of GCD, but the expensive greedy selection rule is still the bottleneck for GCD's implementation in practice. Developing a reliable approximate greedy selection rule that could reduce the overhead while preserving the fast convergence rate is a possible direction to explore.

In the literature of coordinate descent, it is often assumed that the objective is unconstrained or the constraint set is separable. Specialized analysis of CD with

simple nonseparable constraint set such as the linear constraint $\{x \mid \sum_{i=1}^{d} a_i x_i = b\}$ has also appeared in the literature (Tseng and Yun, 2010). Developing a general framework to analyze CD's convergence (if possible) for a more general nonseparable constraint set is an important topic from the perspective of theory and algorithm design.

5.2 Mirror descent

In Chapter 3, we provided a careful study of OMD when using dynamic stepsize. By modifying the OMD via a stabilization technique, we obtain the $\mathcal{O}(\sqrt{T})$ regret and therefore fixed the divergence issue of OMD under unknown time horizon and unbounded domain. Through the stabilization technique, we are also able to analyze the similarities and difference between stabilized-OMD, OMD and DA.

Here we post one possible future direction. We know that the theory of OMD and DA works for any mirror map that satisfies the basic assumptions listed in Section 3.1.1. However, to the author's knowledge, the only examples that can demonstrate the advantage of OMD and DA over vanilla projected subgradient descent (PGD) are the experts' problem and the bandit problem (when negative entropy mirror map is used). It is of interest to identify new applications and the corresponding new mirror maps that could enjoy faster convergence rate with OMD and DA than PGD.

5.3 Stochastic subgradient descent

In Chapter 4, we identified a convex-composite structure and develop some semi-smoothness properties of the empirical-risk minimization problem. The semi-smoothness properties allow us to derive improved convergence rates of SSGD when the interpolation holds.

Our convergence analysis is based on the convexity properties of f. As we mention in connection with Theorem 4.3.2, the strong convexity assumption can be relaxed to weaker conditions, but even these exclude important nonconvex models that appear in neural networks. It is still an open problem as to whether a linear convergence rate for subgradient methods under a weaker assumptions, such as the restricted secant inequality (RSI). In Section 4.3.3 we proved that the rate $\mathcal{O}(\varepsilon^{-1})$

is optimal for subgradient-based methods under interpolation. However, this lower bound holds only for subgradient-based algorithms. Smoothing techniques based on Moreau envelopes (Nesterov, 2005) can sometimes lead to acceleration for nonsmooth optimization, which may be further avenue to explore for obtaining an accelerated SSGD method.

Bibliography

Allen-Zhu, Z., Li, Y., and Song, Z. (2019). A convergence theory for deep learning via over-parameterization. In *Proceedings of ICML*, pages 242–252. → page 78

Allen Zhu, Z., Qu, Z., Richtárik, P., and Yuan, Y. (2016). Even faster accelerated coordinate descent using non-uniform sampling. In *Proceedings of ICML*, pages 1110–1119. → page 6

Argyriou, A., Evgeniou, T., and Pontil, M. (2007). Multi-task feature learning. In *Proceesings of NeurIPS*. → page 31

Atamtūrk, A. and Gómez, A. (2020). Safe screening rules for ℓ_0-regression. In *Proceesings of ICML*. → page 31

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (1995). Gambling in a rigged casino: The adversarial multi-arm bandit problem. In *36th Annual Symposium on Foundations of Computer Science*, pages 322–331. → page 25

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002a). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1). → page 49

Auer, P., Cesa-Bianchi, N., and Gentile, C. (2002b). Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75. → pages 64, 66, 115

Bao, R., Gu, B., and Huang, H. (2020). Fast oscar and owl with safe screening rules. In *Proceesings of ICML*. → page 31

Beck, A. and Teboulle, M. (2003). Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175. → pages 9, 48, 66, 118

Beck, A. and Tetruashvili, L. (2013). On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060. → pages 6, 13, 14

Bertsekas, D. P. (1976). On the goldstein-levitin-polyak gradient projection method. *IEEE Transactions on automatic control*, 21(2):174–184. → page 30

Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific. → page 18

Bertsekas, D. P. and Tsitsiklis, J. N. (1989). *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Inc., USA. → pages 6, 72

Bonnefoy, A., Emiya, V., Ralaivola, L., and Gribonval, R. (2015). Dynamic screening: Accelerating first-order algorithms for the lasso and group-lasso. *IEEE Trans. Signal Process.*, 63(19):5121–5132. → page 31

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA. → pages 15, 107

Bradley, J. K., Kyrola, A., Bickson, D., and Guestrin, C. (2011). Parallel coordinate descent for l1-regularized loss minimization. In *Proceedings of ICML*, pages 321–328. → page 6

Bubeck, S. (2011). Introduction to online optimization. unpublished. → pages 45, 46, 63

Bubeck, S. (2015). Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357. → pages 46, 61, 67, 118

Candes, E. J., Eldar, Y. C., Strohmer, T., and Voroninski, V. (2015). Phase retrieval via matrix completion. *SIAM review*, 57(2):225–251. → page 31

Cauchy, A.-L. (1847). Methode générale pour la résolution des systemes d'éqations simultanées. *Comptes Rendus*, 25(2):536–538. → page 3

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, learning, and games*. Cambridge University Press. → pages 63, 64, 66

Cesa-Bianchi, N., Mansour, Y., and Stoltz, G. (2007). Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2-3):321–352. → page 62

Chandrasekaran, V., Recht, B., Parrilo, P. A., and Willsky, A. S. (2012). The convex geometry of linear inverse problems. *Foundations of Computational mathematics*, 12(6):805–849. → page 32

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. → pages 5, 15, 27

Clarke, F. H. (1981). Generalized gradients of lipschitz functionals. *Advances in Mathematics*, 40(1):52–67. → page 77

Clarke, F. H. (1990). *Optimization and nonsmooth analysis*, volume 5. SIAM. → page 75

Daniilidis, A., Sagastizábal, C., and Solodov, M. (2009). Identifying structure of nonsmooth convex functions by the bundle technique. *SIAM Journal on Optimization*, 20(2):820–840. → page 30

Davis, G., Mallat, S., and Avellaneda, M. (1997). Adaptive greedy approximations. *Constructive approximation*, 13(1):57–98. → page 26

de Rooij, S., van Erven, T., Grünwald, P. D., and Koolen, W. M. (2014). Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research (JMLR)*, 15:1281–1316. → pages 62, 66

Defazio, A., Bach, F., and Lacoste-Julien, S. (2014). Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Proceedings of NeurIPS*, volume 27, pages 1646–1654. → pages 8, 73

Dhillon, I. S., Ravikumar, P., and Tewari, A. (2011). Nearest neighbor based greedy coordinate descent. In *Proceedings of NeuraIPS*, pages 2160–2168. → page 16

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874. → page 14

Fan, Z., Jeong, H., Sun, Y., and Friedlander, M. P. (2020). Atomic decomposition via polar alignment: The geometry of structured optimization. *Foundations and Trends in Optimization*, 3(4):280–366. → pages 32, 33, 34

Fang, H., Fan, Z., and Friedlander, M. P. (2021). Fast convergence of stochastic subgradient method under interpolation. In *Proceedings of ICLR*. → page vi

Fang, H., Fan, Z., Sun, Y., and Friedlander, M. (2020a). Greed meets sparsity: Understanding and improving greedy coordinate descent for sparse optimization. In *Proceeding of AISTATS*. → page v

Fang, H., Harvey, N. J., Portella, V. S., and Friedlander, M. P. (2020b). Online mirror descent and dual averaging: keeping pace in the dynamic case. In *Proceedings of ICML*, volume 119. → page v

Foucart, S. and Rauhut, H. (2013). *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel. → page 26

Gerchinovitz, S. (2011). *Prediction of individual sequences and prediction in the statistical framework: some links around sparse regression and aggregation techniques*. PhD thesis, Université Paris-Sud. → page 63

Ghadimi, S. and Lan, G. (2013). Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368. → page 8

Ghaoui, L. E., Viallon, V., and Rabbani, T. (2012). Safe feature elimination in sparse supervised learning. *Pacific Journal of Optimization*, 8(4):667–698. → pages 31, 34

Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of AISTATS*, pages 315–323. → page 73

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. The MIT fPress. → page 73

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. (2017). Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv e-prints*. → page 72

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. (2017). Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, volume 30. → page 72

György, A. and Szepesvári, C. (2016). Shifting regret, mirror descent, and matrices. In *Proceedings of ICML'16*, pages 2943–2951. → page 50

Hannah, R., Feng, F., and Yin, W. (2019). A2BCD: asynchronous acceleration with optimal complexity. In *International Conference on Learning Representations, ICLR*. → page 7

Hare, W. L. and Lewis, A. S. (2007). Identifying active manifolds. *Algorithmic Operations Research*, 2(2):75. → page 19

Hastie, T., Friedman, J. H., and Tibshirani, R. (2008). Regularization paths and coordinate descent. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, page 3. → page 5

Hazan, E. (2016). Introduction to online convex optimization. *Foundations and Trends in Optimization*, 2(3-4):157–325. → pages 45, 48, 67, 68

Hsieh, C.-J., Si, S., and Dhillon, I. (2014). A divide-and-conquer solver for kernel support vector machines. In *Proceedings of ICML*, pages 566–574. → page 27

Jacot, A., Hongler, C., and Gabriel, F. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Proceedings of NeuraIPS*, pages 8580–8589. → page 76

Jaggi, M., Smith, V., Takác, M., Terhorst, J., Krishnan, S., Hofmann, T., and Jordan, M. I. (2014). Communication-efficient distributed dual coordinate ascent. In *Proceedings of NeuraIPS*, pages 3068–3076. → page 7

Joachims, T. (1999). Making large-scale SVM learning practical. In *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 169–184. MIT Press, Cambridge, MA. → pages 5, 15

Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. In *Proceedings of NeurIPS*, volume 26, pages 315–323. → pages 8, 73

Kaczmarz, S. (1937). Angenaherte aufl ¨ osung von systemen linearer gleichungen. *Bulletin International de l'Académie Polonaise des Sciences et des Lettres*, 35:335–357. → page 71

Kakade, S., Shalev-Shwartz, S., and Tewari, A. (2009). On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. *Unpublished Manuscript, http://ttic. uchicago. edu/shai/papers/KakadeShalevTewari09. pdf*, 2(1). → page 35

Kangarshahi, E. A., Hsieh, Y., Sahin, M. F., and Cevher, V. (2018). Let's be honest: An optimal no-regret framework for zero-sum games. In *Proceedings of ICML'18*, pages 2493–2501. → page 50

Karimi, H., Nutini, J., and Schmidt, M. (2016). Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Proceedings of ECML PKDD*, pages 795–811. → page 81

Karimireddy, S. P., Koloskova, A., Stich, S. U., and Jaggi, M. (2019). Efficient greedy coordinate descent for composite problems. In *Proceedings of AISTATS*, pages 2887–2896. → pages 17, 22, 25, 104, 105

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of ICLR*. → pages 8, 71

Ko, M., Zowe, J., et al. (1994). An iterative two-step algorithm for linear complementarity problems. *Numerische Mathematik*, 68(1):95–106. → page 30

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceeding of NeurIPS*, volume 25. → page 72

Kuang, Z., Geng, S., and Page, D. (2017). A screening rule for l1-regularized ising model estimation. In *Proceesings of NeurIPS*. → page 31

Lacoste-Julien, S., Schmidt, M., and Bach, F. R. (2012). A simpler approach to obtaining an $o(1/t)$ convergence rate for the projected stochastic subgradient method. *ArXiv*, abs/1212.2002. → page 8

Lai, Z. and Lim, L. (2020). Recht-re noncommutative arithmetic-geometric mean conjecture is false. In *Proceedings of ICML*, volume 119, pages 5608–5617. → page 14

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pages 2278–2324. → page 86

Lee, C.-P. and Wright, S. J. (2018). Random permutations fix a worst case for cyclic coordinate descent. *IMA Journal of Numerical Analysis*, 39(3):1246–1275. → pages 6, 13

Lee, Y. T. and Sidford, A. (2013). Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *Proceedings of FOCS*, pages 147–156. → page 6

Lewis, A. S. and Wright, S. J. (2011). Identifying activity. *SIAM Journal on Optimization*, 21(2):597–614. → page 19

Li, M., Andersen, D. G., Smola, A. J., and Yu, K. (2014). Communication efficient distributed machine learning with the parameter server. In *Proceeding of NeurIPS*, volume 27. → page 72

Li, Y. and Osher, S. (2009). Coordinate descent optimization for l1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Problems and Imaging*, 3(3):487—-503. → page 18

Lian, X., Huang, Y., Li, Y., and Liu, J. (2015). Asynchronous parallel stochastic gradient for nonconvex optimization. In *Proceeding of NeurIPS*, volume 28. → page 72

Lin, Q., Lu, Z., and Xiao, L. (2015). An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization*, 25(4):2244–2273. → page 6

Liu, C. and Belkin, M. (2020). Accelerating SGD with momentum for over-parameterized learning. In *Proceedings of ICLR*. → page 83

Liu, J., Wright, S. J., Ré, C., Bittorf, V., and Sridhar, S. (2014a). An asynchronous parallel stochastic coordinate descent algorithm. In *Proceedings of ICML*, pages 469–477. → page 7

Liu, J., Wright, S. J., Ré, C., Bittorf, V., and Sridhar, S. (2015). An asynchronous parallel stochastic coordinate descent algorithm. *Journal of Machine Learning Research*, 16(10):285–322. → page 72

Liu, J., Zhao, Z., Wang, J., and Ye, J. (2014b). Safe screening with variational inequalities and its application to lasso. In *Proceesings of ICML*. → page 31

Lloyd, S. P. (1982). Least squares quantization in PCM. *IEEE Trans. Information Theory*, 28(2):129–136. → page 5

Luo, Z. and Tseng, P. (1992). On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1):361–379. → page 6

Luo, Z.-Q. and Tseng, P. (1993). Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46(1):157–178. → page 6

Ma, S., Bassily, R., and Belkin, M. (2018). The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In *Proceedings of ICML*, pages 3331–3340. → pages 73, 82

McMahan, H. B. (2017). A survey of algorithms and analysis for adaptive online learning. *Journal of Machine Learning Research*, 18:90:1–90:50. → pages 45, 67, 115

Meinshausen, N. and Bühlmann, P. (2006). High dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34. → page 31

Ndiaye, E., Fercoq, O., Gramfort, A., and Salmon, J. (2017). Gap safe screening rules for sparsity enforcing penalties. *Journal of Machine Learning Research*, 18:128:1–128:33. → pages 10, 31, 34, 35

Needell, D., Ward, R., and Srebro, N. (2014). Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *Proceeding of NeurIPS*, volume 27. → page 71

Negahban, S. N. and Wainwright, M. J. (2012). Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13:1665–1697. → page 21

Nemirovski, A., Juditsky, A. B., Lan, G., and Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609. → page 8

Nemirovski, A. and Yudin, D. (1983). *Problem Complexity and Method Efficiency in Optimization*. Wiley Interscience. → pages 4, 9, 82, 84

Nesterov, Y. (2009). Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259. → pages 45, 48, 61, 66

Nesterov, Y. (2012). Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362. → pages 6, 12, 13, 22, 25

Nesterov, Y. (2014). *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Incorporated, 1 edition. → page 78

Nesterov, Y. E. (1983). A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Dokl. Akad. Nauk SSSR*, 269:543–547. → page 4

Nesterov, Y. E. (2004). *Introductory Lectures on Convex Optimization - A Basic Course*, volume 87 of *Applied Optimization*. Springer. → pages 4, 16

Nesterov, Y. E. (2005). Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152. → page 90

Nguyen, L. M., Liu, J., Scheinberg, K., and Takáč, M. (2017). SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *Proceeding of ICML*, volume 70, pages 2613–2621. → pages 8, 73

Nutini, J., Schmidt, M., and Hare, W. (2017). "active-set complexity" of proximal gradient: How long does it take to find the sparsity pattern? *Optimization Letter*. → pages 19, 109

Nutini, J., Schmidt, M. W., Laradji, I. H., Friedlander, M. P., and Koepke, H. A. (2015). Coordinate descent converges faster with the gauss-southwell rule than random selection. In *Proceedings of the International Conference on Machine Learning*, pages 1632–1641. → pages 6, 12, 15, 16, 17, 18, 19, 22, 25, 104

Orabona, F. and Pál, D. (2018). Scale-free online learning. *Theor. Comput. Sci.*, 716:50–69. → pages 45, 48, 66, 68

Oustry, F. (2000). A second-order bundle method to minimize the maximum eigenvalue function. *Mathematical Programming*, 89(1):1–33. → page 42

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S. (1993). Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE. → page 26

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830. → page 27

Peng, Z., Wu, T., Xu, Y., Yan, M., and Yin, W. (2016). Coordinate friendly structures, algorithms and applications. *Annals of Mathematical Sciences and Applications*, 1:57–119. → page 12

Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14. → pages 5, 12

Qian, X., Richtárik, P., Gower, R. M., Sailanbayev, A., Loizou, N., and Shulgin, E. (2019). SGD with arbitrary sampling: General analysis and improved rates. In *Proceedings of ICML*, pages 5200–5209. → page 81

Raj, A., Olbrich, J., Gärtner, B., Schölkopf, B., and Jaggi, M. (2015). Screening rules for convex problems. *ArXiv*, abs/1609.07478. → page 31

Recht, B. and Ré, C. (2012). Toward a noncommutative arithmetic-geometric mean inequality: Conjectures, case-studies, and consequences. In *The 25th Annual Conference on Learning Theory*, pages 11.1–11.24. → pages 6, 14

Recht, B., Re, C., Wright, S., and Niu, F. (2011). Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. In *Proceeding of NeurIPS*, volume 24. → page 72

Richtárik, P. and Takác, M. (2011). Efficient serial and parallel coordinate descent methods for huge-scale truss topology design. In *Proceedings of the International Conference on Operations Research*, pages 27–32. → page 7

Richtárik, P. and Takác, M. (2014). Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38. → pages 6, 22, 25

Robbins, H. and Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, 22(3):400–407. → page 7

Rockafellar, R. T. (1970). *Convex Analysis*. Princeton University Press, Princeton. → pages 45, 46, 121

Rockafellar, R. T. and Wets, R. J.-B. (2009). *Variational analysis*, volume 317. Springer Science & Business Media. → pages 36, 114

Sa, C. D. (2020). Random reshuffling is not always better. In *Proceedings of NeurIPS*. → page 14

Saha, A. and Tewari, A. (2013). On the nonasymptotic convergence of cyclic coordinate descent methods. *SIAM Journal on Optimization*, 23(1):576–601. → pages 6, 13

Schmidt, M. and Le Roux, N. (2013). Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv e-prints*. → pages 80, 82

Schmidt, M., Roux, N. L., and Bach, F. R. (2017). Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1-2):83–112. → pages 8, 73

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). Trust region policy optimization. In *Proceedings of ICML*, volume 37, pages 1889–1897. → page 71

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv e-prints*, page arXiv:1707.06347. → page 72

Shalev-Shwartz, S. (2012). Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194. → pages 45, 46, 60, 69

Shalev-Shwartz, S., Singer, Y., and Srebro, N. (2007). Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of ICML*, page 807–814. → page 71

Shevade, S. K. and Keerthi, S. S. (2003). A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics*, 19(17):2246–2253. → pages 18, 26

Shor, N. Z. (1984). *Minimization methods for non-differentiable functions*. Springer Series in Computational Mathematics, Springer. → page 4

Stich, S. U. (2019). Unified optimal analysis of the (stochastic) gradient method. *arXiv e-prints*, page arXiv:1907.04232. → page 82

Sun, R. and Ye, Y. (2021). Worst-case complexity of cyclic coordinate descent: $o(n^2)$ gap with randomized version. *Mathematical Programming*, 185(1):487–520. → pages 6, 13

Sun, Y. and Bach, F. R. (2020). Safe screening for the generalized conditional gradient method. *ArXiv*, abs/2002.09718. → page 34

Sun, Y., Jeong, H., Nutini, J., and Schmidt, M. W. (2019). Are we there yet? manifold identification of gradient-related proximal methods. In *Proceedings of AISTATS*, pages 1110–1119. → page 19

Sylvain Sardy, A. G. B. and Tseng, P. (2000). Block coordinate relaxation methods for nonparametric wavelet denoising. *Journal of Computational and Graphical Statistics*, 9(2):361–379. → pages 5, 12

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288. → page 31

Tieleman, T. and Hinton, G. (2012). Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning. → page 71

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475–494. → page 6

Tseng, P. and Yun, S. (2010). A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training. *Computational Optimization and Applications*, 47. → page 89

Vaswani, S., Bach, F., and Schmidt, M. (2019). Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *Proceedings of AISTATS*, pages 1195–1204. → page 83

Wang, J., Zhou, J., Liu, J., Wonka, P., and Ye, J. (2014). A safe screening rule for sparse logistic regression. In *Proceesing of NeurIPS*, pages 1053–1061. → page 31

Wang, J., Zhou, J., Wonka, P., and Ye, J. (2013). Lasso screening rules via dual polytope projection. In *Advances in Neural Information Processing Systems 26*, pages 1070–1078. → page 31

Wright, S. J. (2012). Accelerated block-coordinate relaxation for regularized optimization. *SIAM Journal on Optimization*, 22(1):159–186. → page 30

Wu, T. T. and Lange, K. (2008). Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, 2(1):224—-244. → page 18

Xiang, Z. J., Wang, Y., and Ramadge, P. J. (2017). Screening tests for lasso problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(5):1008–1027. → page 31

Yaroshinsky, R., El-Yaniv, R., and Seiden, S. S. (2004). How to better use expert advice. *Machine Learning*, 55(3):271–309. → page 66

You, Y., Lian, X., Liu, J., Yu, H., Dhillon, I. S., Demmel, J., and Hsieh, C. (2016). Asynchronous parallel greedy coordinate descent. In *Proceedings of NeuralPS*, pages 4682–4690. → page 7

You, Y., Zhang, Z., Hsieh, C.-J., Demmel, J., and Keutzer, K. (2018). Imagenet training in minutes. In *Proceedings of the International Conference on Parallel Processing*, page 1–10. → page 72

Yuan, M. and Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society (Series B)*, 68:49–67. → page 31

Zhang, J., Lin, H., Sra, S., and Jadbabaie, A. (2020). On complexity of finding stationary points of nonsmooth nonconvex functions. In *Proceedings of ICML*. → page 75

Zhang, W., Hong, B., Liu, W., Ye, J., Cai, D., He, X., and Wang, J. (2017). Scaling up sparse support vector machines by simultaneous feature and sample reduction. In *Proceesings of ICML*. → page 31

Zhang, X. (2013). Bregman divergence and mirror descent lecture notes. Available at http://users.cecs.anu.edu.au/~xzhang/teaching/bregman.pdf. Last accessed July. 10, 2021. → pages 117, 118

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of ICML*, pages 928–936. → page 48

Appendix A

Appendix for Chapter 2

A.1 Proofs for Section 2.3

Preliminaries

We introduce some notations and Lemmas that appear in works from Nutini et al. (2015) and Karimireddy et al. (2019).

We say that a coordinate gradient step is *bad* if the iterate crosses origin or end at origin, i.e., $x_i^{(t+1)} x_i^{(t)} < 0$ or $x_i^{(t+1)} = 0$, otherwise we call this step a *good* step, see more details in Karimireddy et al. (2019). We denote the set of good steps until the t-th iteration as $\mathcal{G}_t$. Because a bad step always follows by a good step, it is easy to verify that

$$|\mathcal{G}_t| \leq \left\lceil \frac{t}{2} \right\rceil. \tag{A.1}$$

Recall the selection rule in Section 2.3:

Selection rule A.1 (GS-s rule)**.** *Select* $i \in \arg\max_j Q_j(x^{(t)})$ *where*

$$Q_i(x) = \min_{s \in \partial g_i} |\nabla_i f(x) + s|. \tag{A.2}$$

Lemma A.1.1 (Karimireddy et al., 2019, Theorem 1)**.** *Assume* f *is* μ_1*-strongly convex with respect to 1-norm, then the iterates generated from Algorithm 1 with*

GS-s rule (selection rule A.2) satisfy

$$F(x^{(t)}) - F(x^*) \leq \left(1 - \frac{\mu_1}{L_{\max}}\right)^{\lceil t/2 \rceil} (F(0) - F(x^*)).$$

Lemma A.1.2 (Karimireddy et al., 2019, Lemma 2)**.** *Consider g to be 1-norm regularization or non-negative constraint. If the t-th iteration is a good step, then we have*

$$F(x^{(t+1)}) \leq F(x^{(t)}) - \frac{1}{2L_{\max}} \max_{i \in [d]} Q_i(x^{(t)})^2, \tag{A.3}$$

where Q_i is defined in the GS-s rule (selection rule A.2).

The above two Lemmas due to the work from Karimireddy et al. (2019). The second Lemma comes from the 'proof sketch' of (Karimireddy et al., 2019, Lemma 2, page 5).

Proof of Lemma 2.3.1

Proof. If i is not select by Algorithm 1 at the t-th iteration, then $x_i^{(t+1)} = 0$ trivially remains at 0.

If i is selected at the t-th iteration, by assuming $|\nabla_i f(x^{(t)}) - \nabla_i f(x^*)| \leq \delta_i$, we know that

$$\begin{aligned} &-\delta_i + \nabla_i f(x^*) \leq \nabla_i f(x^{(t)}) \leq \delta_i + \nabla_i f(x^*) \\ \overset{\text{(i)}}{\Rightarrow} &-u_i \leq \nabla_i f(x^{(t)}) \leq -l_i, \end{aligned} \tag{A.4}$$

where (i) follows directly from the definition of $\delta_i := \min\{-\nabla_i f(x^*) - l_i, u_i + \nabla_i f(x^*)\}$.

Next we show that $\operatorname{prox}_{g_i/L_i}\left(0 - \frac{1}{L_i}\nabla_i f(x^{(t)})\right) = 0$. By the definition of the proximal operator

$$\operatorname{prox}_{g_i/L_i}\left(0 - \frac{1}{L_i}\nabla_i f(x^{(t)})\right) = \operatorname*{arg\,min}_{y \in \mathbb{R}} \left\{ \frac{1}{2}\left(y - \left(-\frac{1}{L_i}\nabla f_i(x^{(t)})\right)\right)^2 + \frac{1}{L_i} g_i(y) \right\}.$$

This minimization problem is strongly convex and thus attains a unique solution

that satisfies

$$0 \in y + \frac{1}{L_i}\nabla_i f(x^{(t)}) + \frac{1}{L_i}\partial g_i(y). \tag{A.5}$$

Knowing that $-u_i \leq \nabla_i f(x^{(t)}) \leq -l_i$ from (A.4) and $\text{int}\partial g_i(0) = (l_i, u_i)$ by the definition of l_i and u_i. We can easily conclude that $y = 0$ satisfies (A.5) and therefore

$$x_i^{(t+1)} = \text{prox}_{g_i/L_i}\left(0 - \frac{1}{L_i}\nabla_i f(x^{(t)})\right) = 0.$$

□

Proof of Theorem 2.3.1

Proof. Let $t \leq d - \tau$ and recall the definition of *good* steps until the t-th iteration from section A.1: let $\mathcal{G}_t = \{i_1, i_2, \ldots, i_k\}$, where $k = |\mathcal{G}_t| \geq \lceil t/2 \rceil$.

At iteration i_m for $m \in [k]$, we know $x^{(i_m)}$ is guaranteed to be $m-1$–sparse because the number of non-zeros of the iterate at most can increase by one for one *good* step and will not increase for *bad* steps. By assuming f is $\mu_1^{(\tau+m-1)}$ strongly convex w.r.t. 1-norm and $\tau + m - 1$–sparse vectors, we know that F is also $\mu_1^{(\tau+m-1)}$ strongly convex w.r.t. 1-norm and $\tau + m - 1$–sparse vectors. Moreover $|\text{supp}(y) \cup \text{supp}(x^{(i_m)})| \leq \tau + m - 1$ is true $\forall y \in \mathbb{R}^d$ that is τ-sparse. Then by the definition of $\mu_1^{(\tau+m-1)}$, we have

$$F(y) \geq F(x^{(i_m)}) + \langle \partial F(x^{(i_m)}), y - x^{(i_m)} \rangle + \frac{\mu_1^{(\tau+m-1)}}{2}\|y - x^{(i_m)}\|_1^2 \tag{A.6}$$

for any τ-sparse vectors y (with a little bit abuse of notation, we use $\partial F(x^{(t)})$ to denote any vectors in the subdifferential of $F(x^{(t)})$). Minimize both sides of (A.6)

w.r.t. y that is τ-sparse, we get

$$\begin{aligned}
F(x^*) &\geq F(x^{(i_m)}) - \sup_{\|y\|_0 \leq \tau} \left(\langle -\partial F(x^{(i_m)}), y - x^{(i_m)} \rangle - \frac{\mu_1^{(\tau+m-1)}}{2} \|y - x^{(i_m)}\|_1^2 \right) \\
&\geq F(x^{(i_m)}) - \sup_{y \in \mathbb{R}^d} \left(\langle -\partial F(x^{(i_m)}), y - x^{(i_m)} \rangle - \frac{\mu_1^{(\tau+m-1)}}{2} \|y - x^{(i_m)}\|_1^2 \right) \\
&\overset{\text{(i)}}{=} F(x^{(i_m)}) - \left(\frac{\mu_1^{(\tau+m-1)}}{2} \|\cdot\|_1^2 \right)^* (-\partial F(x^{(i_m)})) \\
&\overset{\text{(ii)}}{=} F(x^{(i_m)}) - \frac{1}{2\mu_1^{(\tau+m-1)}} \|\partial F(x^{(i_m)})\|_\infty^2,
\end{aligned}$$

where (i) is from the definition of conjugate function, and (ii) is from the fact that $\left(\frac{1}{2}\|\cdot\|_1^2\right)^* = \frac{1}{2}\|\cdot\|_\infty^2$ (Boyd and Vandenberghe, 2004).

More specifically,

$$F(x^*) \geq F(x^{(i_m)}) - \frac{1}{2\mu_1^{(\tau+m-1)}} \|\nabla f(x^{(i_m)}) + u\|_\infty^2 \quad \forall u \in \partial g(x^{(i_m)}).$$

By the definition of $Q_i(\cdot)$ in the GS-s rule (selection rule 2.3), we further have

$$F(x^*) \geq F(x^{(i_m)}) - \frac{1}{2\mu_1^{(\tau+m-1)}} \max_{i \in [d]} Q_i(x^{(i_m)})^2. \tag{A.7}$$

Recall Lemma A.1.2, we have

$$F(x^{(i_m+1)}) \leq F(x^{(i_m)}) - \frac{1}{2L_{\max}} \max_{i \in [d]} Q_i(x^{(i_m)})^2.$$

Plug the above equation into (A.7)

$$\begin{aligned}
F(x^*) &\geq F(x^{(i_m)}) - \frac{L_{\max}}{\mu_1^{(\tau+m-1)}} (F(x^{(i_m+1)}) - F(x^{(i_m)})) \\
\Rightarrow F(x^{(i_m+1)}) - F^* &\leq \left(1 - \frac{\mu_1^{(\tau+m)}}{L_{\max}}\right) (F(x^{(i_m)}) - F^*).
\end{aligned}$$

By applying the above inequality recursively, we get

$$\begin{aligned} F(x^{(t)}) - F^* &\leq \prod_{m=1}^{k} \left(1 - \frac{\mu_1^{(\tau+m-1)}}{L_{\max}}\right)(F(0) - F^*) \\ &\leq \prod_{i=1}^{\lceil \frac{t}{2} \rceil} \left(1 - \frac{\mu_1^{(\tau+i-1)}}{L_{\max}}\right)(F(0) - F^*), \end{aligned}$$

which completes the proof.

□

Proof of Theorem 2.3.3

Proof. This proof is essentially the same as Theorem 2.3.1. The difference is that, by the definition of the Δ-GS-s rule (selection rule 2.2), the Lemma A.1.2 becomes

$$F(x^{(t+1)}) - F(x^{(t)}) \leq -\frac{\Delta}{2L_{\max}} \max_{i \in [d]} Q_i(x^{(t)})^2$$

at each good step t.

Knowing that $\text{supp}(x^{(t)}) \subset W_\Delta$, we have $|\text{supp}(x^*) \cup \text{supp}(x^{(t)})| \leq |W_\Delta| \ \forall t > 0$. Then we can incorporate the new Lemma into the analysis of Theorem 2.3.1 and get

$$\begin{aligned} F(x^{(t)}) - F^* &\leq \left(1 - \frac{\Delta \mu_1^{(|W_\Delta|)}}{L_{\max}}\right)^{\lceil \frac{t}{2} \rceil} (F(0) - F^*) \\ &\leq \left(1 - \frac{\Delta \mu_2}{|W_\Delta| L_{\max}}\right)^{\lceil \frac{t}{2} \rceil} (F(0) - F^*). \end{aligned}$$

□

Proposition A.1.1. *Let*

$$\tilde{x}^{(j)} := \underset{\text{supp}(x) \subseteq W_j}{\arg\min}\ f(x) + g(x),$$

and Q_i defined as (A.2) *(the GS-s rule). Then Q_i's are continuous at $\tilde{x}^{(j)}$ on the support W_j for all $i, j \in [d]$.*

Proof. Given $i, j \in [d]$. By the optimality condition, we know that

$$Q_i(\tilde{x}^{(j)}) = 0 \qquad \forall i \in W_j.$$

We consider 2 cases — $i \in W_j$ and $i \notin W_j$. For $i \in W_j$, following Nutini et al. (2017)'s analysis, we know that as $x \to \tilde{x}^{(j)}$, $Q_i(x) \to 0$ since $\tilde{x}^{(j)}$ is the optimal solution on the support W_j.

For $i \notin W_j$, denote $\partial g_i(0) = [l_i, u_i]$, then $Q_i(x) = \min\{|\nabla_i f(x) + \delta| \mid \delta \in [l_i, u_i]\}$ since $x_i = 0$ ($\text{supp}(x) = W_j$). We know that $\nabla_i f(x)$'s are continuous functions, therefore it is easy to conclude that $Q_i(x)$ is continuous on W_j for $i \notin W_j$.

To sum up, we finish the proof. □

Proof of Theorem 2.3.4

Proof. **Preliminaries:**

Given $\Delta > 0$, we sort the elements of $W_\Delta = \{i_1, i_2, ..., i_m\}$ by the time they first enter the working set W_Δ ,i.e., i_1 is the first coordinate being selected and i_2 is the second coordinate being included in W_Δ, etc.

We denote the t-th iterate from the Δ-GCD algorithm as $x^{(t)}$ and the t-th iterate from the totally corrective greedy algorithm (TCGA) as $\tilde{x}^{(t)}$. Let $W^\sharp = \{\tilde{i}_1, \tilde{i}_2, \ldots, \tilde{i}_k\}$, its elements are also sorted by the time when they enter the working set.

A claim: for any $j \leq k$, there exist $\varepsilon_j > 0$ such that $\forall \Delta < \varepsilon_j$, the first j elements in W_Δ is the same as the first j elements in $W^\sharp$.

Proof. We prove this claim by induction, when $j = 1$, $\forall \Delta \leq 1$, Δ-GCD and the TCGA both select the coordinate $\arg\max_{i \in [d]} Q_i(0)$ in the first iteration, thus the claim is trivially true in the base case.

Assume that the claim is true with some $j > 0$, then for $j+1$:

By Proposition A.1.1, we know that Q_i is continuous on W_j at $\tilde{x}^{(j)}$, further we assumed that $\arg\max_{i \in [d]} Q_i(\tilde{x}^{(j)})$ is singleton. Therefore there exist $\varepsilon' > 0$ such that $\forall \|x - \tilde{x}^{(j)}\| \leq \varepsilon', \arg\max_{i \in [d]} Q_i(x) = \tilde{i}_{j+1}$.

By the uniqueness (recall that F is strongly convex) of $\tilde{x}^{(j)}$:

$$\tilde{x}^{(j)} := \underset{\text{supp}(x) \subseteq W_j}{\arg\min}\ f(x) + g(x)$$

and the optimality condition, we also know that $\exists\ \delta > 0$ such that $\forall x \in \mathbb{R}^d$ satisfy $\text{supp}(x) \subseteq W_j$ and $\max_{i \in W_j} Q_i(x) \leq \delta$, we have $\|x - x^{(j)}\| \leq \varepsilon'$.

Denote $Q_i(x^{(t)})$ (recall $x^{(t)}$ is generated from Δ-GCD) is bounded by some constant $B\ \forall t > 0$.

Then, by setting $\Delta \leq (\min\{\varepsilon_j, \delta/B\})^2$, when i_{j+1} first enter W_Δ at some iteration t, we have

$$\arg\max_{i \in W_j} Q_i(x^{(t)}) \leq \sqrt{\Delta} \arg\max_{i \in [d]} Q_i(x^{(t)}) \leq \frac{\delta}{B} B = \delta,$$

also by the induction assumption, we know that $\text{supp}(x^{(t)}) \subseteq W_j$. Putting these two conditions together, we get $\|x^{(t)} - x^j\| \leq \varepsilon'$ and thus $\arg\max_{i \in [d]} Q_i(x^{(t)}) = \tilde{i}_{j+1}$, which implies that $i_{j+1} = \tilde{i}_{j+1}$. This completes the proof of this claim. □

Back to the proof:

Following the claim, we know that $\exists\ \varepsilon_k > 0$ such that for $\forall \Delta < \varepsilon_k$, the first k elements in W_Δ is just $W^\sharp$.

By the nondegeneracy assumption i.e., $\delta_i > 0\ \forall x_i^* = 0$ and continuity of $Q_i, \nabla f$, we know that $\exists\ \varepsilon'' > 0$ such that $\forall \|x - x^*\| < \varepsilon''$ (note that $\tilde{x}^{(k)} = x^*$), $|\nabla_i f(x) - \nabla_i f(x^*)| \leq \delta_i\ \forall x_i^* = 0$ and this further implies $Q_i(x) = 0\ \forall i \notin W^\sharp$ (note that $\text{supp}(x^*) \in W^\sharp$).

Again, there exist $\delta'' > 0$ such that $\forall x \in \mathbb{R}^d$ satisfy $\text{supp}_{W^\sharp}(x)$ and $\max_{i \in W^\sharp} Q_i(x) \leq \delta''$, we have $\|x - x^*\| \leq \varepsilon''$.

Thus for $\Delta \leq \min\{\varepsilon_k, \delta''\}$, the first k elements in W_Δ will be $W^\sharp$, and any coordinate $i \notin W^\sharp$ can not be included in W_Δ. Therefore $W_\Delta = W^\sharp$. □

Proof of Theorem 2.3.2

Proof. Given the number of iteration t, denote $\mathcal{Z}_t = \{i \in [d] \mid x_i^{(t')} = 0\ \forall t' < t\}$, which is the entries of $x^{(t)}$ that filled with 0's. and $\mathcal{V}_t = \{i \in [d] \mid x_i^* = 0 \quad \text{and} \quad |\nabla_i f(x^{(t')}) - \nabla_i f(x^*)| \leq \delta_i\ \forall t' \geq t\}$.

From Lemma 2.3.1 (in the main text), we know that any coordinates in $\mathcal{Z}_t \cap \mathcal{V}_t$

will always stay at 0 and thus cannot be in W, that is

$$\begin{aligned} & W \subset [d] \backslash (\mathcal{Z}_t \cap \mathcal{V}_t) \ \ \forall t > 0 \\ \Rightarrow & |W| \leq \min_{t \in [d]} \{d - |\mathcal{Z}_t \cap \mathcal{V}_t|\}. \end{aligned} \tag{A.8}$$

Recall the definition of the set of good steps until the t-th iteration $\mathcal{G}_t \subset [t]$:

$$\begin{aligned} |\mathcal{V}_t| &= \sum_{i=1}^{d} \mathbf{1}\{x_i^* = 0 \quad \text{and} \quad |\nabla_i f(x^{(t')}) - \nabla_i f(x^*)| \leq \delta_i \ \ \forall t' \geq t\} \\ &\geq \sum_{i=1}^{d} \mathbf{1}\{\|\nabla f(x^{(t')}) - \nabla f(x^*)\|_\infty \leq \delta_i \ \ \forall t' \geq t\} - \tau \\ &\overset{\text{(i)}}{\geq} \sum_{i=1}^{d} \mathbf{1}\{L_\infty \|x^{(t')} - x^*\|_1 \leq \delta_i \ \ \forall t' \geq t\} - \tau \\ &\geq \sum_{i=1}^{d} \mathbf{1}\left\{L_\infty \sup_{t' \geq t} \|x^{(t)} - x^*\|_1 \leq \delta_i\right\} - \tau, \end{aligned} \tag{A.9}$$

where (i) follows from the definition ∞-norm smoothness.

By the definition of $\mathcal{G}_t$ in section A.1, we also have $|\mathcal{Z}_t| \geq d - |\mathcal{G}_t|$, and further

$$\begin{aligned} |\mathcal{Z}_t \cap \mathcal{V}_t| &= |\mathcal{Z}_t| + |\mathcal{V}_t| - |\mathcal{Z}_t \cup \mathcal{V}_t| \\ &\geq d - |\mathcal{G}_t| + |\mathcal{V}_t| - d \\ &\geq |\mathcal{V}_t| - |\mathcal{G}_t|. \end{aligned} \tag{A.10}$$

Plug the above result in (A.8), we get

$$\begin{aligned} |W| &\leq \min_{t>0} \{d - |\mathcal{V}_t| + |\mathcal{G}_t|\} \\ &\leq \min_{t>0} \left\{ d + \tau - \sum_{i=1}^{d} \mathbf{1}\{L_\infty \sup_{t' \geq t} \|x^{(t')} - x^*\|_1 \leq \delta_i\} + |\mathcal{G}_t| \right\} \\ &\leq \min_{t \in [d]} \left\{ d + \tau - \sum_{i=1}^{d} \mathbf{1}\{L_\infty \sup_{t' \geq t} \|x^{(t')} - x^*\|_1 \leq \delta_i\} + t \right\} \\ &= \min_{t \in [d]} B_t + t, \end{aligned} \tag{A.11}$$

where B_t is defined as $B_t := d + \tau - p_\delta\left(L_\infty \sup_{i\geq t}\left\{\|x^{(i)} - x^*\|_1\right\}\right)$ in Theorem 2.3.2. □

Proof of Corollary 2.3.1

Proof. Similar to the proof of Theorem 2.3.2, denote $\mathcal{Z}_t = \{i \in [d] \mid x_i^{(t')} = 0 \; \forall t' < t\}$, which is the entries of $x^{(t)}$ that filled with 0's. and $\mathcal{V}_t = \{i \in [d] \mid |\nabla_i f(x^{(t')}) - \nabla_i f(x^*)| \leq \delta_i \; \forall t' \geq t\}$.

From Lemma 2.3.1 (in the main text), we know that any coordinates in $\mathcal{Z}_t \cap \mathcal{V}_t$ will always stay at 0 and thus cannot be in W, that is

$$\begin{aligned} & W \subset [d] \backslash (\mathcal{Z}_t \cap \mathcal{V}_t) \;\; \forall t > 0 \\ \Rightarrow & |W| \leq \min_{t \in [d]} \{d - |\mathcal{Z}_t \cap \mathcal{V}_t|\}. \end{aligned} \tag{A.12}$$

Recall the definition of the set of good steps until the t-th iteration $\mathcal{G}_t \subset [t]$.

$$\begin{aligned} |\mathcal{V}_t| &= \sum_{i=1}^{d} \mathbf{1}\{x_i^* = 0 \quad \text{and} \quad |\nabla_i f(x^{(t')}) - \nabla_i f(x^*)| \leq \delta_i \;\; \forall t' \geq t\} \\ &\geq \sum_{i=1}^{d} \mathbf{1}\{\|\nabla f(x^{(t')}) - \nabla f(x^*)\|_\infty \leq \delta_i \;\; \forall t' \geq t\} - \tau \\ &\overset{\text{(i)}}{\geq} \sum_{i=1}^{d} \mathbf{1}\{L_\infty \|x^{(t')} - x^*\|_1 \leq \delta_i \;\; \forall t' \geq t\} - \tau \\ &\overset{\text{(ii)}}{\geq} \sum_{i=1}^{d} \mathbf{1}\left\{L_\infty \sqrt{\frac{2}{\mu_1}\left(F(x^{(t)}) - F(x^*)\right)} \leq \delta_i \;\; \forall t' \geq t\right\} - \tau \\ &\overset{\text{(iii)}}{=} \sum_{i=1}^{d} \mathbf{1}\left\{L_\infty \sqrt{\frac{2}{\mu_1}\left(F(x^{(t)}) - F(x^*)\right)} \leq \delta_i\right\} - \tau \\ &\overset{\text{(iv)}}{=} p_\delta\left(L_\infty \sqrt{\frac{2}{\mu_1}\left(F(x^{(t)}) - F(x^*)\right)}\right) - \tau \\ &\overset{\text{(v)}}{\geq} p_\delta\left(L_\infty \sqrt{\frac{2}{\mu_1}\prod_{i=1}^{|\mathcal{G}_t|}\left(1 - \frac{\mu_1^{(\tau+i-1)}}{L}\right)(F(0) - F^*)}\right) - \tau, \end{aligned} \tag{A.13}$$

where (i) follows from the ∞-norm smoothness assumption, (ii) is from μ_1 strongly

convex, (iii) is true since $F(x^{(t)})$ is a decreasing sequence, (iv) is by the definition of $p_\delta(\cdot)$, (v) directly follows from Theorem 2.3.1.

By the definition of $\mathcal{G}_t$, we also have $|\mathcal{Z}_t| \geq d - |\mathcal{G}_t|$, and further

$$\begin{aligned} |\mathcal{Z}_t \cap \mathcal{V}_t| &= |\mathcal{Z}_t| + |\mathcal{V}_t| - |\mathcal{Z}_t \cup \mathcal{V}_t| \\ &\geq d - |\mathcal{G}_t| + |\mathcal{V}_t| - d \\ &\geq |\mathcal{V}_t| - |\mathcal{G}_t|. \end{aligned} \tag{A.14}$$

Plug the above result in (A.12), we get

$$\begin{aligned} |W| &\leq \min_{t>0} \{d - |\mathcal{V}_t| + |\mathcal{G}_t|\} \\ &\leq \min_{t>0} \left\{ d + \tau - \left(L_\infty \sqrt{\frac{2}{\mu_1} \prod_{i=1}^{|\mathcal{G}_t|} \left(1 - \frac{\mu_1^{(\tau+i-1)}}{L}\right) (F(0) - F^*)} \right) + |\mathcal{G}_t| \right\} \\ &\leq \min_{t \in [d]} \left\{ d + \tau - \left(L_\infty \sqrt{\frac{2}{\mu_1} \prod_{i=1}^{t} \left(1 - \frac{\mu_1^{(\tau+i-1)}}{L}\right) (F(0) - F^*)} \right) + t \right\} \\ &= \min_{t \in [d]} B_t + t, \end{aligned} \tag{A.15}$$

where B_t is defined as $B_t := d + \tau - p_\delta \left(\sqrt{\frac{2L_\infty^2}{\mu_1} \prod_{i=0}^{t-1} \left(1 - \frac{\mu_1^{(\tau+i)}}{L}\right) (F(0) - F^*)} \right)$ in Theorem 2.3.2. □

A.2 Proofs for Section 2.4

Proof of Proposition 2.4.1

Proof. From the construction of sets $\mathcal{A}^{(t)}$, it is straightforward to see that

$$\mathcal{A}^{(1)} \supseteq \mathcal{A}^{(2)} \supseteq \ldots,$$

which shows that $\{\mathcal{A}^{(t)}\}_{t=1}^{\infty}$ is a monotone sequence. By Rockafellar and Wets (2009, Exercise 4.3), the Painleveé-Kuratowski set limit

$$\mathcal{A}^{(\infty)} = \lim_{t\to\infty} \mathcal{A}^{(t)}$$

is well-defined.

First, we show that $\mathcal{F}_{\mathcal{A}}(M^*y^*) \subseteq \mathcal{A}^{(\infty)}$. By Theorem 2.4.1, we know that $\mathcal{F}_{\mathcal{A}}(M^*y^*) \subseteq \mathcal{A}^{(t)}$ for all t. Therefore, it follows that $\mathcal{F}_{\mathcal{A}}(M^*y^*) \subseteq \mathcal{A}^{(\infty)}$.

Next, we show that $\mathcal{A}^{(\infty)} \subseteq \mathcal{F}_{\mathcal{A}}(M^*y^*)$. Consider $a \in \mathcal{A}^{(\infty)}$. Since $\{\mathcal{A}^{(t)}\}_{t=1}^{\infty}$ is a monotone sequence, there exist $T > 0$ such that

$$a \in \mathcal{A}^{(t)}, \quad \forall t \geq T.$$

By the construction of $\mathcal{A}^{(t)}$, we know that $\mathcal{A}^{(t)} \subseteq \mathcal{F}_{\mathcal{A}}(M^*y^{(t)}, \varepsilon)$ for all t, and thus we can conclude that

$$\langle a, M^*y^{(t)}\rangle \geq \sigma_{\mathcal{A}}(M^*y^{(t)}) - \varepsilon, \quad \forall t \geq T.$$

Now by taking limits with respect to t to both sides of the inequality, we can conclude that

$$\langle a, M^*y^*\rangle \geq \sigma_{\mathcal{A}}(M^*y^*),$$

which implies that $a \in \mathcal{F}_{\mathcal{A}}(M^*y^*)$. □

Appendix B

Appendix for Chapter 3

B.1 Standard facts

B.1.1 Scalar inequalities

Fact B.1.1. *For any $a > 0$ and $b, x \in \mathbb{R}$, we have $-ax^2 + bx \le b^2/4a$.*

Fact B.1.2. $e^{-x} \le 1 - x + \frac{x^2}{2}$ *for* $x \ge 0$.

Fact B.1.3. $\sum_{i=1}^{t} \frac{1}{\sqrt{i}} \le 2\sqrt{t} - 1$ *for* $t \ge 1$.

Fact B.1.4. $\log(x) \le x - 1$ *for* $x \ge 0$.

The following proposition is a variant of an inequality that is frequently used in online learning; see, e.g., Auer et al. (2002b, Lemma 3.5), McMahan (2017, Lemma 4).

Proposition B.1.1. *Let $u > 0$ and $a_1, a_2, \ldots, a_T \in [0, u]$. Then*

$$\sum_{t=1}^{T} \frac{a_t}{\sqrt{u + \sum_{i<t} a_i}} \;\le\; 2\sqrt{\sum_{t=1}^{T} a_t}.$$

Although it is easy to prove this inequality by induction, the following proof may provide more intuition. The proof is based on a generic lemma on approximating sums by integrals.

Lemma B.1.1 (Sums with chain rule)**.** *Let $S \subseteq \mathbb{R}$ be an interval. Let $F : S \to \mathbb{R}$ be concave and differentiable on the interior of S. Let $u \geq 0$ and let $A : \{0, 1, \ldots, T\} \to S$ satisfy $A(i) - A(i-1) \in [0, u]$ for each $1 \leq i \leq T$. Then*

$$\sum_{i=1}^{T} F'\big(u + A(i-1)\big) \cdot (A(i) - A(i-1)) \;\leq\; F(A(T)) - F(A(0)).$$

As $u \to 0$, the left-hand side becomes comparable to $\int_0^T F'(A(x))A'(x)\,dx$, an expression that has no formal meaning since A is only defined on integers. If this expression existed, it would equal the right-hand side by the chain rule.

Proof of Lemma B.1.1. Since F is concave, $f := F'$ is non-increasing. Fix any $1 \leq i \leq T$ and observe that $f(x) \geq f(A(i)) \geq f(u + A(i-1))$ for all $x \leq A(i)$. Thus

$$f(u + A(i-1)) \cdot (A(i) - A(i-1)) \;\leq\; \int_{A(i-1)}^{A(i)} f(x)\,dx \;=\; F(A(i)) - F(A(i-1)).$$

Summing over i, the right-hand side telescopes, which yields the result. □

Proof of Proposition B.1.1. Apply Lemma B.1.1 with $S = \mathbb{R}_{\geq 0}$, $F(x) = 2\sqrt{x}$ and $A(i) = \sum_{1 \leq j \leq i} a_j$. □

Proposition B.1.2. *Let $x, y, \alpha, \beta > 0$.*

$$\begin{aligned} \textit{If} \qquad & x - y \;\leq\; \alpha\sqrt{x} + \beta \\ \textit{then} \qquad & x - y \;\leq\; \alpha\sqrt{y} + \beta + \alpha\sqrt{\beta} + \alpha^2. \end{aligned}$$

Proof. The proposition's hypothesis yields

$$y + \beta + \frac{\alpha^2}{4} \;\geq\; x - \alpha\sqrt{x} + \frac{\alpha^2}{4} \;=\; \left(\sqrt{x} - \frac{\alpha}{2}\right)^2.$$

Taking the square root and rearranging,

$$\sqrt{x} \;\leq\; \sqrt{y + \beta + \frac{\alpha^2}{4}} + \frac{\alpha}{2}.$$

Squaring both sides and rearranging,

$$\begin{aligned} x &\le y+\alpha\sqrt{y+\beta+\frac{\alpha^2}{4}}+\beta+\frac{\alpha^2}{2} \\ &\le y+\alpha\sqrt{y}+\alpha\sqrt{\beta}+\beta+\alpha^2, \end{aligned}$$

by subadditivity of the square root. □

B.1.2 Bregman divergence properties

The following lemma collects basic facts regarding the Bregman divergence induced by a mirror map of the Legendre type. See Zhang (2013).

Lemma B.1.2. *The Bregman divergence induced by Φ satisfies the following properties:*

- *$D_\Phi(x,y)$ is convex in x;*
- *$\nabla\Phi(\nabla\Phi^*(z)) = z$ and $\nabla^*\Phi(\nabla\Phi(x)) = x$ for all x and z;*
- *$D_\Phi(x,y) = D_{\Phi^*}(\nabla\Phi(y),\nabla\Phi(x))$ for all x and y.*

Proposition B.1.3. *If Φ is ρ-strongly convex with respect to $\|\cdot\|$ then $D_\Phi(x,y) \ge \frac{\rho}{2}\|x-y\|^2$.*

Differences of Bregman divergences

Recall that in (3.2) we defined the notation

$$D_\Phi\left(\begin{smallmatrix} a \\ b \end{smallmatrix}; c\right) \coloneqq D_\Phi(a,c) - D_\Phi(b,c) = \Phi(a) - \Phi(b) - \langle \nabla\Phi(c), a-b\rangle.$$

This has several useful properties, which we now discuss.

Proposition B.1.4. *$D_\Phi\left(\begin{smallmatrix} a \\ b \end{smallmatrix}; p\right)$ is linear in $\hat{p}$. In particular,*

$$D_\Phi\left(\begin{smallmatrix} a \\ b \end{smallmatrix}; \nabla\Phi^*(\hat{p}-\hat{q})\right) = D_\Phi\left(\begin{smallmatrix} a \\ b \end{smallmatrix}; p\right) + \langle \hat{q}, a-b\rangle \qquad \forall \hat{q} \in \mathbb{R}^d.$$

Proof. Immediate from the definition. □

Proposition B.1.5. *For all $a, b, c, d \in \mathcal{D}$,*

$$D_\Phi\left(\begin{smallmatrix} a \\ b \end{smallmatrix}; d\right) - D_\Phi\left(\begin{smallmatrix} a \\ b \end{smallmatrix}; c\right) \;=\; \langle \hat{c} - \hat{d}, a - b \rangle \;=\; D_\Phi\left(\begin{smallmatrix} a \\ b \end{smallmatrix}; d\right) + D_\Phi\left(\begin{smallmatrix} b \\ a \end{smallmatrix}; c\right).$$

Proof. The first equality holds from Proposition B.1.4 with $\hat{p} = \hat{c}$ and $\hat{q} = \hat{c} - \hat{d}$. The second equality holds since $D_\Phi\left(\begin{smallmatrix} b \\ a \end{smallmatrix}; c\right) = -D_\Phi\left(\begin{smallmatrix} a \\ b \end{smallmatrix}; c\right)$. □

An immediate consequence is the "generalized triangle inequality for Bregman divergence". See Bubeck (2015, eq. (4.1)), Beck and Teboulle (2003, Lemma 4.1) or Zhang (2013, eq. (3)).

Proposition B.1.6. *For all $a, b, d \in \mathcal{D}$,*

$$D_\Phi(a, d) - D_\Phi(b, d) + D_\Phi(b, a) \;=\; \langle \hat{a} - \hat{d}, a - b \rangle$$

Proof. Apply Proposition B.1.5 with $c = a$ and use $D_\Phi(a, a) = 0$. □

Proposition B.1.7. *Let $a, b, c, u, v \in \mathbb{R}^d$ satisfy $\gamma \hat{a} + (1-\gamma)\hat{b} = \hat{c}$ for some $\gamma \in \mathbb{R}$. Then*

$$\gamma D_\Phi\left(\begin{smallmatrix} u \\ v \end{smallmatrix}; a\right) + (1-\gamma) D_\Phi\left(\begin{smallmatrix} u \\ v \end{smallmatrix}; b\right) \;=\; D_\Phi\left(\begin{smallmatrix} u \\ v \end{smallmatrix}; c\right).$$

Proof. By definition of D_Φ, the claimed identity is equivalent to

$$\begin{aligned} &(1-\gamma)\big(\Phi(u) - \Phi(v) - \langle \nabla\Phi(a), u - v \rangle\big) + \gamma\big(\Phi(u) - \Phi(v) - \langle \nabla\Phi(b), u - v \rangle\big) \\ &= \big(\Phi(u) - \Phi(v) - \langle \nabla\Phi(c), u - v \rangle\big). \end{aligned}$$

This equality holds by canceling $\Phi(u) - \Phi(v)$ and by the assumption that $\nabla\Phi(c) = (1-\gamma)\nabla\Phi(a) + \gamma\nabla\Phi(b)$. □

The following proposition is the "Pythagorean theorem for Bregman divergence". Recall that $\Pi_{\mathcal{X}}^{\Phi}(y) = \operatorname*{arg\,min}_{u \in \mathcal{X}} D_\Phi(u, y)$. Proofs may be found in Bubeck (2015, Lemma 4.1) or Zhang (2013, eq. (17)).

Proposition B.1.8. *Let $\mathcal{X} \subset \mathbb{R}^d$ be a convex set. Let $p \in \mathbb{R}^d$ and $\pi = \Pi_{\mathcal{X}}^{\Phi}(p)$. Then*

$$D_\Phi\left(\begin{smallmatrix} z \\ \pi \end{smallmatrix}; p\right) \;\geq\; D_\Phi\left(\begin{smallmatrix} z \\ \pi \end{smallmatrix}; \pi\right) \;=\; D_\Phi(z, \pi) \qquad \forall z \in \mathcal{X}.$$

A generalization of the previous proposition can be obtained by using the linearity property.

Proposition B.1.9. *Let $\mathcal{X} \subset \mathbb{R}^d$ be a convex set. Let $p \in \mathbb{R}^d$ and $\pi = \Pi^{\Phi}_{\mathcal{X}}(p)$. Then*

$$D_\Phi(\begin{smallmatrix} v \\ \pi \end{smallmatrix}; \nabla\Phi^*(\hat{p} - \hat{q})) \;\geq\; D_\Phi(\begin{smallmatrix} v \\ \pi \end{smallmatrix}; \Phi^*(\hat{\pi} - \hat{q})) \qquad \forall v \in \mathcal{X}, \hat{q} \in \mathbb{R}^d.$$

Proof.

$$\begin{aligned} D_\Phi(\begin{smallmatrix} v \\ \pi \end{smallmatrix}; \nabla\Phi^*(\hat{p} - \hat{q})) &= D_\Phi(\begin{smallmatrix} v \\ \pi \end{smallmatrix}; p) + \langle \hat{q}, v - \pi \rangle && \text{(by Proposition B.1.4)} \\ &\geq D_\Phi(\begin{smallmatrix} v \\ \pi \end{smallmatrix}; \pi) + \langle \hat{q}, v - \pi \rangle && \text{(by Proposition B.1.8)} \\ &= D_\Phi(\begin{smallmatrix} v \\ \pi \end{smallmatrix}; \nabla\Phi^*(\hat{\pi} - \hat{q})) && \text{(by Proposition B.1.4).} \quad \square \end{aligned}$$

B.2 Proofs for Section 3.3.1

Proof of Proposition 3.3.1

Proof. First we apply Proposition B.1.6 with $a = x$, $b = x'$ and $d = \nabla\Phi^*(\hat{x} - \hat{q})$ to obtain

$$\begin{aligned} D_\Phi(\begin{smallmatrix} x \\ x' \end{smallmatrix}; w) &= \langle \hat{x} - \hat{d}, x - x' \rangle - D_\Phi(x', x) \\ &= \langle \hat{q}, x - x' \rangle - D_\Phi(x', x) \\ &\overset{\text{(i)}}{\leq} \|\hat{q}\|_* \|x - x'\| - \frac{\rho}{2}\|x - x'\|^2 \\ &\overset{\text{(ii)}}{\leq} \|\hat{q}\|_*^2 / 2\rho, \end{aligned} \qquad \square$$

where (i) is from the definition of dual norm and Proposition B.1.3, (ii) is by Fact B.1.1.

B.3 Proofs for Section 3.3.2

Proposition 3.3.3. *Let $a, b \in \mathcal{X}$ and $c \in \mathcal{D}$. Then $D_\Phi(\begin{smallmatrix} a \\ b \end{smallmatrix}; c) \leq \Lambda(a, c)$.*

Proof of Proposition 3.3.3

Proof. Since $a, b \in \mathcal{X}$ we have $\|a\|_1 = \|b\|_1 = 1$. Then

$$\begin{aligned} D_\Phi(\tbinom{a}{b}; c) &= D_{\mathrm{KL}}(a,c) - D_{\mathrm{KL}}(b,c) \\ &= \big(D_{\mathrm{KL}}(a,c) + 1 - \|c\|_1 + \ln\|c\|_1\big) - \big(D_{\mathrm{KL}}(b,c) + 1 - \|c\|_1 + \ln\|c\|_1\big) \\ &= \Lambda(a,c) - \Lambda(b,c) \qquad \text{(by definition of } \Lambda) \\ &\le \Lambda(a,c) \qquad \text{(by Proposition 3.3.2).} \end{aligned}$$

□

Proposition 3.3.4. *Let $a \in \mathcal{X}$, $\hat{q} \in [0,1]^d$ and $\eta > 0$. Then $\Lambda(a, \nabla\Phi^*(\hat{a} - \eta\hat{q})) \le \eta^2\langle a, \hat{q}\rangle/2$.*

Proof. Let $b = \nabla\Phi^*(\hat{a} - \eta\hat{q})$. By (3.24), $b_i = a_i \exp(-\eta\hat{q}_i)$. Then

$$\begin{aligned} \Lambda(a, \nabla\Phi^*(\hat{a} - \eta\hat{q})) &= \sum_{i=1}^d a_i \ln(a_i/b_i) + \ln\|b\|_1 \\ &= \sum_{i=1}^d \eta a_i \hat{q}_i + \ln\Big(\sum_{i=1}^d a_i \exp(-\eta\hat{q}_i)\Big) \\ &\le \sum_{i=1}^d \eta a_i \hat{q}_i + \sum_{i=1}^d a_i \exp(-\eta\hat{q}_i) - 1 \qquad \text{(by Fact B.1.4)} \\ &\le \sum_{i=1}^d \eta a_i \hat{q}_i + \sum_{i=1}^d a_i \left(1 - \eta\hat{q}_i + \frac{\eta^2\hat{q}_i^2}{2}\right) - 1 \qquad \text{(byFact B.1.2)} \\ &\le \eta^2 \sum_{i=1}^d a_i \hat{q}_i/2, \end{aligned}$$

using $\sum_{i=1}^d a_i = 1$ (since $a \in \mathcal{X}$) and $\hat{q}_i^2 \le \hat{q}_i$ (since $\hat{q} \in [0,1]^d$). □

B.4 Proofs for Section 3.4

At many points throughout this section we will need to talk about optimality condition for problems where we minimize a convex function over a convex set. Such conditions depend on the *normal cone* of the set on which the optimization is taking place.

Definition B.4.1. The normal cone to $C \subseteq \mathbb{R}^d$ at $x \in \mathbb{R}^d$ is the set $N_C(x) := \{s \in \mathbb{R}^d \mid \langle s, y - x \rangle \leq 0 \; \forall y \in C\}$.

Lemma B.4.1 (Rockafellar, 1970, Theorem 27.4)**.** *Let $h \colon \mathcal{C} \to \mathbb{R}$ be a closed convex function such that $(\operatorname{ri} \mathcal{C}) \cap (\operatorname{ri} \mathcal{X}) \neq \emptyset$. Then, $x \in \arg\min_{z \in \mathcal{X}} h(z)$ if and only if there is $g \in \partial h(x)$ such that $-g \in N_{\mathcal{X}}(x)$.*

Using the above result allows us to derive a useful characterization of points that realize the Bregman projections.

Lemma B.4.2. *Let $y \in \mathcal{D}$ and $x \in \bar{\mathcal{D}}$. Then $x = \Pi^{\Phi}_{\mathcal{X}}(y)$ if and only if $x \in \mathcal{D} \cap \mathcal{X}$ and $\nabla\Phi(y) - \nabla\Phi(x) \in N_{\mathcal{X}}(x)$.*

Proof. Suppose $x \in \mathcal{D} \cap \mathcal{X}$ and $\nabla\Phi(y) - \nabla\Phi(x) \in N_{\mathcal{X}}(x)$. Since $\nabla\Phi(y) - \nabla\Phi(x) = -\nabla(D_{\Phi}(\cdot, y))(x)$, by Lemma B.4.1 we conclude that $x \in \arg\min_{z \in \mathcal{X}} D(z, y)$. Now suppose $x = \Pi^{\Phi}_{\mathcal{X}}(y)$. By Lemma B.4.1 together with the definition of Bregman divergence, this is the case if and only if there is $-g \in \partial\Phi(x)$ such that $-(g - \nabla\Phi(y)) \in N_{\mathcal{X}}(x)$. Since Φ is of Legendre type we have $\partial\Phi(z) = \emptyset$ for any $z \notin \mathcal{D}$ (see Rockafellar, 1970, Theorem 26.1). Thus, $x \in \mathcal{D}$ and $g = \nabla\Phi(x)$ since Φ is differentiable. Finally, $x \in \mathcal{X}$ by the definition of Bregman projection. □

Before proceeding to the proof of the results from Section 3.4, we need to state on last result about the relation of subgradients and conjugate functions.

Lemma B.4.3 (Rockafellar, 1970, Theorem 23.5)**.** *Let $f \colon \mathcal{X} \to \mathbb{R}$, let $x \in \mathcal{X}$ and let $\hat{y} \in \mathbb{R}^d$. Then $\hat{y} \in \partial f(x)$ if and only if x attains $\sup_{x \in \mathbb{R}^d}(\langle \hat{y}, x \rangle - f(x)) = f^*(\hat{y})$.*

Proof of Proposition 3.4.1

Proof. Let $t \geq 1$ and let $F_t \colon \mathcal{D} \to \mathbb{R}$ be the function being minimized on the right-hand side of (3.31). By definition we have $x^{(t+1)} = \Pi^{\Phi}_{\mathcal{X}}(y^{(t+1)})$. Using the optimality conditions of the Bregman projection, we have

$$x^{(t+1)} = \Pi^{\Phi}_{\mathcal{X}}(y^{(t+1)}) \iff \hat{y}^{(t+1)} - \hat{x}^{(t+1)} \in N_{\mathcal{X}}(x^{(t+1)}), \qquad \text{(by Lemma B.4.2)}$$

By further using the definitions from Algorithm 3 we get

$$\begin{aligned}\hat{y}^{(t+1)} - \hat{x}^{(t+1)} &= \gamma_t(\hat{x}^{(t)} - \eta_t g_t) + (1-\gamma_t)\hat{x}^{(1)} - \hat{x}^{(t+1)} \\ &= \gamma_t(\hat{x}^{(t)} - \hat{x}^{(t+1)} - \eta_t g_t) + (1-\gamma_t)(\hat{x}^{(1)} - \hat{x}^{(t+1)}) \\ &= -\gamma_t\big(\nabla(D_\Phi(\cdot, x^{(t)}))(x^{(t+1)}) + \eta_t g_t\big) - (1-\gamma_t)\nabla(D_\Phi(\cdot, x^{(1)}))(x^{(t+1)}) \\ &= -\nabla F_t(x^{(t+1)})\end{aligned}$$

Thus, we have $-\nabla F_t(x^{(t+1)}) \in N_{\mathcal{X}}(x^{(t+1)})$. By the optimality conditions from Lemma B.4.1 we conclude that $x^{(t+1)} \in \arg\min_{x\in\mathcal{X}} F_t(x)$, as desired. □

Proof of Theorem 3.4.1

Theorem 3.4.1 is an easy consequence of the following proposition.

Proposition B.4.1. *Let $\{f_t\}_{t\geq 1}$ with $f_t : \mathcal{X} \to \mathbb{R}$ be a sequence of convex functions and let $\eta : \mathbb{N} \to \mathbb{R}_{>0}$ be non-increasing. Let $\{x^{(t)}\}_{t\geq 1}$ and $\{g_t\}_{t\geq 1}$ be as in Algorithm 3. Define $\gamma^{[i,t]} := \prod_{j=i}^{t} \gamma_j$ for every $i,t \in \mathbb{N}$. Then, there are $\{p_t\}_{t\geq 1}$ with $p_t \in N_X(x^{(t)})$ for each $t \geq 1$ such that $\forall t \geq 0$,*

$$\{x^{(t+1)}\} = \arg\min_{x\in\mathcal{X}} \Big(\sum_{i=1}^{t} \gamma^{[i,t]} \langle \eta_i g_i + p_i, x\rangle - \Big(\gamma^{[1,t]} + \sum_{i=1}^{t} \gamma^{[i+1,t]}(1-\gamma_i)\Big)\langle \hat{x}^{(1)}, x\rangle + \Phi(x)\Big). \tag{B.1}$$

Proof. First of all, in order to prove (B.1) we claim it suffices to prove that there are $\{p_t\}_{t\geq 1}$ with $p_t \in N_X(x^{(t)})$ for each $t \geq 1$ such that

$$\hat{y}^{(t+1)} = -\sum_{i=1}^{t} \gamma^{[i,t]}(\eta_i g_i + p_i) + \Big(\gamma^{[1,t]} + \sum_{i=1}^{t} \gamma^{[i+1,t]}(1-\gamma_i)\Big)\hat{x}^{(1)}, \qquad \forall t \geq 0. \tag{B.2}$$

To see the sufficiency of this claim, note that

$$
\begin{aligned}
& x^{(t+1)} = \Pi_{\mathcal{X}}^{\Phi}(y^{(t+1)})) \\
\iff & \hat{y}^{(t+1)} - \hat{x}^{(t+1)} \in N_{\mathcal{X}}(x^{(t+1)}) && \text{(Lemma B.4.2)} \\
\iff & \hat{y}^{(t+1)} \in \partial(\Phi + \delta(\cdot \,|\, \mathcal{X}))(x^{(t+1)}) && (\partial(\delta(\cdot \,|\, \mathcal{X}))(x) = N_{\mathcal{X}}(x)) \\
\iff & x^{(t+1)} \in \operatorname*{arg\,max}_{x \in \mathbb{R}^d} \big(\langle \hat{y}^{(t+1)}, x \rangle - \Phi(x) - \delta(x \,|\, \mathcal{X}) \big) && \text{(Lemma B.4.3)} \\
\iff & x^{(t+1)} \in \operatorname*{arg\,min}_{x \in \mathcal{X}} \big(- \langle \hat{y}^{(t+1)}, x \rangle + \Phi(x) \big).
\end{aligned}
$$

The above together with eq. (B.2) yields eq. (B.1). Let us now prove eq. (B.2) by induction on $t \geq 0$.

For $t = 0$, eq. (B.2) holds trivially. Let $t > 0$. By definition, we have $\hat{y}^{(t+1)} = (1-\gamma_t)(\hat{x}^{(t)} - \eta_t g_t) + \gamma_t \hat{x}^{(1)}$. At this point, to use the induction hypothesis, we need to write $\hat{x}^{(t)}$ in function of $\hat{y}^{(t)}$. From the definition of Algorithm 3, we have $x^{(t)} = \Pi_X^{\Phi}(y_t)$. By Lemma B.4.2, the latter holds if and only if $\hat{y}^{(t)} - \hat{x}^{(t)} \in N_{\mathcal{X}}(x^{(t)})$. That is, there is $p_t \in N_{\mathcal{X}}(x^{(t)})$ such that $\hat{x}^{(t)} = \hat{y}^{(t)} - p_t$. Plugging these facts together and using our induction hypothesis we have

$$
\begin{aligned}
& \hat{y}^{(t+1)} \\
=& \gamma_t(\hat{x}^{(t)} - \eta_t g_t) + (1-\gamma_t)\hat{x}^{(1)} = \gamma_t(\hat{y}^{(t)} - \eta_t g_t - p_t) + (1-\gamma_t)\hat{x}^{(1)} \\
\overset{\text{I.H.}}{=} & \gamma_t \Big(- \sum_{i=1}^{t-1} \gamma^{[i,t-1]}(\eta_i g_i + p_i) - \eta_t g_t - p_t + \Big(\gamma^{[1,t-1]} + \sum_{i=1}^{t-1} \gamma^{[i+1,t-1]}(1-\gamma_i) \Big) \hat{x}^{(1)} \Big) \\
& \quad + (1-\gamma_t)\hat{x}^{(1)} \\
=& - \sum_{i=1}^{t} \gamma^{[i,t]}(\eta_i g_i + p_i) + \Big(\gamma^{[1,t]} + \sum_{i=1}^{t} \gamma^{[i+1,t]}(1-\gamma_i) \Big) \hat{x}^{(1)},
\end{aligned}
$$

and this finishes the proof of eq. (B.2). □

Proof (of Theorem 3.4.1). Define $\gamma^{[i,t]}$ for every $i,t \in \mathbb{N}$ as in Proposition B.4.1. If $\gamma_t = 1$ for all $t \geq 1$, then $\gamma^{[i,t]} = 1$ for any $t,i \geq 1$. Moreover, if $\gamma_t = \frac{\eta_{t+1}}{\eta_t}$ for every $t \geq 1$, then for every $t,i \in \mathbb{N}$ with $t \geq i$ we have $\gamma^{[i,t]} = \frac{\eta_{t+1}}{\eta_i}$, which yields

$\gamma^{[i,t]}(\eta_i g_i + p_i) = \eta_t g_i + \frac{1}{\eta_i} p_i$ and

$$\begin{aligned}\gamma^{[1,t]} + \sum_{i=1}^{t} \gamma^{[i+1,t]}(1-\gamma_i) &= \frac{\eta_{t+1}}{\eta_1} + \sum_{i=1}^{t} \frac{\eta_{t+1}}{\eta_{i+1}} \Big(1 - \frac{\eta_{i+1}}{\eta_i}\Big) \\ &= \frac{\eta_{t+1}}{\eta_1} + \eta_{t+1} \sum_{i=1}^{t} \Big(\frac{1}{\eta_{i+1}} - \frac{1}{\eta_i}\Big) = 1.\end{aligned}$$

□

Appendix C

Appendix for Chapter 4

C.1 Proofs for Section 4.3

Proof of Proposition 4.3.2

Proof. Given $x_1, x_2 \in \mathbb{R}^d$, $\forall i \in [n]$,

$$
\begin{aligned}
& \|\partial f_i(x_2) - \partial f_i(x_1)\| \\
\leq\ & \|\ell'(h_i(x_2))\partial h_i(x_2) - \ell'(h_i(x_1))\partial h_i(x_1)\| \\
\leq\ & \|\ell'(h_i(x_2))\partial h_i(x_2) - \ell'(h_i(x_2))\partial h_i(x_1) + \ell'(h_i(x_2))\partial h_i(x_1) - \ell'(h_i(x_1))\partial h_i(x_1)\| \\
\leq\ & \|\ell'(h_i(x_2))\partial h_i(x_2) - \ell'(h_i(x_2))\partial h_i(x_1)\| + \|\ell'(h_i(x_2))\partial h_i(x_1) - \ell'(h_i(x_1))\partial h_i(x_1)\| \\
\leq\ & \|\ell'(h_i(x_2))(\partial h_i(x_2) - \partial h_i(x_1))\| + \|(\ell'(h_i(x_2)) - \ell'(h_i(x_1)))\partial h_i(x_1)\| \\
\leq\ & |\ell'(h_i(x_2))|\|\partial h_i(x_2) - \partial h_i(x_1))\| + |\ell'(h_i(x_2)) - \ell'(h_i(x_1))|\|\partial h_i(x_1)\| \\
\leq\ & 2L|\ell'(h_i(x_2))| + |h_i(x_2) - h_i(x_1)| \times L \\
\overset{\text{(i)}}{\leq}\ & 2L\sqrt{2f_i(x_2)} + L^2\|x_2 - x_1\|, \qquad \text{(C.1)}
\end{aligned}
$$

where (i) follows the same argument as the proof of Proposition 4.3.1:

$$|\ell'(h_i(x_2))| = |\ell'(h_i(x_2)) - \ell'(0)| \leq \sqrt{2(\ell(h_i(x_2)) - \ell(0))} = \sqrt{2f_i(x_2)}.$$

Exchange x_1 and x_2, we can also get

$$\|\partial f_i(x_2) - \partial f_i(x_1)\| \leq 2L\sqrt{2f_i(x_1)} + L^2\|x_2 - x_1\|. \tag{C.2}$$

Combining eq. (C.2) and eq. (C.1), we finished the proof for eq.(4.5).

Given $x_1, x_2 \in \mathbb{R}^d$, note that $f_i(x)$ is almost every differentiable by Rademacher's Theorem, thus

$$\begin{aligned} f_i(x_2) &= f_i(x_1) + \int_0^1 \langle \partial f_i(x_1 + \tau(x_2 - x_1)), x_2 - x_1 \rangle d\tau \\ &= f_i(x_1) + \langle \partial f_i(x_1), x_2 - x_1 \rangle + \int_0^1 \langle \partial f_i(x_1 + \tau(x_2 - x_1)) - \partial f_i(x_1), x_2 - x_1 \rangle d\tau \end{aligned} \tag{C.3}$$

Note that

$$\begin{aligned} & \int_0^1 \langle \partial f_i(x_1 + \tau(x_2 - x_1)) - \partial f_i(x_1), x_2 - x_1 \rangle d\tau \\ \leq & \int_0^1 \|\partial f_i(x_1 + \tau(x_2 - x_1)) - \partial f_i(x_1)\| \|x_2 - x_1\| d\tau \\ \leq & \int_0^1 \left(2L\sqrt{2f_i(x_1)} + L^2\|\tau(x_2 - x_1)\| \right) \|x_2 - x_1\| d\tau \\ \leq & \ 2L\|x_2 - x_1\|\sqrt{2f_i(x_1)} + \frac{L^2}{2}\|x_2 - x_1\|^2. \end{aligned} \tag{C.4}$$

Plug eq. (C.4) into eq. (C.3), we finished the proof for eq. (4.6). □