EXPLOITING SMOOTHNESS IN STATISTICAL LEARNING, SEQUENTIAL PREDICTION, AND STOCHASTIC OPTIMIZATION

By

Mehrdad Mahdavi

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science and Engineering - Doctor of Philosophy

2014

ABSTRACT

EXPLOITING SMOOTHNESS IN STATISTICAL LEARNING, SEQUENTIAL PREDICTION, AND STOCHASTIC OPTIMIZATION

By

Mehrdad Mahdavi

In the last several years, the intimate connection between convex optimization and learning problems, in both statistical and sequential frameworks, has shifted the focus of algorithmic machine learning to examine this interplay. In particular, on one hand, this intertwinement brings forward new challenges in reassessment of the performance of learning algorithms under the assumptions imposed by convexity such as Lipschitzness, strong convexity, and smoothness. On the other hand, emergence of datasets of an unprecedented size, demands the development of efficient optimization algorithms to tackle large-scale learning problems.

The overarching goal of this thesis is to reassess the smoothness of loss functions in statistical learning, sequential prediction/online learning, and stochastic optimization and explicate its consequences. In particular we examine how leveraging smoothness of loss function could be beneficial or detrimental in these settings in terms of sample complexity, statistical consistency, regret analysis, and convergence rate.

In the statistical learning framework, we investigate the sample complexity of learning problems when the loss function is smooth and strongly convex and the learner is provided with the target risk as a prior knowledge. We establish that under these assumptions, by exploiting the smoothness of loss function, we are able to improve the sample complexity of learning exponentially.

We also investigate the smoothness from the viewpoint of **statistical consistency** and show that in sharp contrast to optimization and generalization where the smoothness is favorable because of its computational and theoretical virtues, the smoothness of surrogate loss function might deteriorate the binary excess risk. Motivated by this negative result, we provide a unified analysis of three types of errors including optimization error, generalization bound, and the error in translating convex excess risk into a binary excess risk, and underline the conditions that smoothness might be preferred.

We then turn to elaborate the importance of smoothness in sequential prediction/online learning. We introduce a new measure to assess the performance of online learning algorithms which is referred to as **gradual variation**. The gradual variation is measured by the sum of the distances between every two consecutive loss functions and is more suitable for gradually evolving environments such as stock prediction. Under smoothness assumption, we devise novel algorithms for online convex optimization with regret bounded by gradual variation.

Finally, we investigate how to exploit the smoothness of loss function in convex optimization. We propose a novel optimization paradigm that is referred to as **mixed optimization** which interpolates between stochastic and full gradient methods and is able to exploit the smoothness of loss functions to obtain faster convergence rates in stochastic optimization, and condition number independent accesses of full gradients in deterministic optimization.

We also propose efficient **projection-free** optimization algorithms to tackle the computational challenge arising from the projection steps which are required at each iteration of most existing gradient based optimization methods to ensure the feasibility of intermediate solutions. In stochastic optimization setting, by introducing and leveraging smoothness, we develop novel methods which only require one projection at the final iteration. In online learning setting, we consider online convex optimization with soft constraints where the constraints are allowed to be satisfied on long term.

To my parents, Asieh and Rashid.

ACKNOWLEDGMENTS

First and foremost, I feel indebted to my advisor, Professor Rong Jin, for his guidance, encouragement, and inspiring supervision throughout the course of this research work. His patience, extensive knowledge, and creative thinking have been the source of inspiration for me. He was available for advice or academic help whenever I needed and gently guided me for deeper understanding, no matter how late or inconvenient the time is. When I was struggling to quit my Ph.D. at Sharif University to join Rong's group, I was not sure about my decision, but after four and a half years, I am happy to say that I did not make a wrong decision. It's hard to express how thankful I am for his unwavering support over the last years.

I would like to take on this opportunity to thank my thesis committee members Pan-Ning Tan, Ambuj Tewari, and Eric Torng who have accommodated my timing constraints despite their full schedules, and provided me with precious feedback for the presentation of the results, in both written and oral form.

During my Ph.D. studies, I had the pleasure of collaborating with many researchers from each and every one of which I had things to learn, and the quality of my research was considerably enhanced by these interactions. I would like to thank Tianbao Yang and Lijun Zhang for all the discussions we had and the fun moments we spent on doing research and attending conferences. The results of Chapter 5 and Chapter 8 represent part of the fruits of these collaborations. I also spent a summer as intern at Microsoft Research working with Ofer Dekel and a summer at NEC Research Labs working with Shenghuo Zhu. I learned a lot from them and would like to express my gratitude for having me as an intern. I also would like to thank Elad Hazan, Satyen Kale, Phil Long, Shai Shalev-Shwartz, and Ohad

Shamir for some helpful email correspondence.

Living in East Lansing without my good friends would not have been easy. I want to thank all my friends in the department and outside the department. I wish I could name you all.

Last but definitely not least, I want to express my deepest gratitude to my beloved parents and my dearest siblings. Their love and unwavering support have been crucial to my success, and a constant source of comfort and counsel. Special thanks to my parents for abiding by my absence in last five years.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Chapter 1

Introduction

In machine learning the goal is to learn from labeled examples in order to predict the labels of unseen examples. That is, given a training set, we aim to learn a hypothesis, or a classifier that assigns labels to samples that have never been observed by the algorithm. Efficiently finding a hypothesis based on the training set which minimizes some measure of performance is the main focus of machine learning.

In order to study the learning problem is a mathematical framework, it is necessary to define the framework in which the algorithm is to function. Basically there are two frameworks that have gained significant popularity within the last two decades: the *statistical learning* framework and the *sequential prediction* or online learning framework. In both settings *mathematicical optimization* theory plays an important role by providing a unified framework to investigate the *computational* issues of learning algorithms. Additionally, tools from convex optimization underline the analysis in algorithmic machine learning that targets most of the practical learning algorithms in both frameworks.

This chapter is devoted to an overview of these three broad topics of statistical learning, sequential prediction/online learning, and convex optimization, aiming to develop a general correspondence between the first two and convex optimization. In particular by characterizing sample complexity, statistical consistency, regret analysis, and convergence rate in terms of the properties of loss functions such as Lipschitzness, strong convexity, and smoothness, we elaborate the importance of smoothness [1] and explicate its consequences. Here we move towards the definitions in a fairly non-technical manner and the formal definitions will be given in Chapter 2.

1.1 Classical Statistical Learning

We begin by stating the basic problem of binary classification in the standard passive supervised learning setting (also called batch learning). In binary classification, the learning algorithm is given a set of labeled examples $\mathcal{S} = ((\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n))$ drawn independent and identically distributed (i.i.d.) from a fixed but unknown distribution $\mathcal{D}$ over the space $\Xi = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the instance space and $\mathcal{Y}$ is the label (target) space. The goal, with the help of provided labeled examples, is to output a hypothesis or classifier h from a predefined hypothesis class $\mathcal{H} = \{h : \mathcal{X} \mapsto \mathcal{Y}\}$ that does well on unseen examples coming from the same distribution. In other words, we would like to find a hypothesis to generalize well from the training set to the entire domain of examples.

To measure the performance of a classifier $h \in \mathcal{H}$ on unseen samples, we utilize a loss function $\ell : \mathcal{H} \times \Xi \mapsto \mathbb{R}_+$. The mostly used loss function in binary classification problem with instance space defined as $\Xi = \mathbb{R}^d \times \{-1, +1\}$ is the 0-1 loss $\ell(h, (\mathbf{x}, y)) = \mathbb{I}[h(\mathbf{x}) \neq y]$, where $\mathbb{I}[\cdot]$ is the indicator function. The risk of a particular classifier $h \in \mathcal{H}$ is the probability that the classifier does not predict the correct label on a random data point generated by the underlying distribution $\mathcal{D}$, i.e., $\mathcal{L}_{\mathcal{D}}(h) = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[h(\mathbf{x}) \neq y] = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell(h, (\mathbf{x}, y))]$. This

[1] The precise definition will be given later. We say that a continuously differentiable function $f : \mathbb{R}^d \mapsto \mathbb{R}$ is β-smooth if its gradient is Lipschitz with constant β, i.e., $|\nabla f(\mathbf{w}) - \nabla f(\mathbf{w'})| \leq \beta \|\mathbf{w} - \mathbf{w'}\|$.

performance measure is also called *generalization error* or *true risk* in statistical learning community. An equivalent way to express the generalization bound is the *sample complexity* analysis. Roughly speaking, the sample complexity of an algorithm is the number of examples which is sufficient to ensure that, with probability at least $1 - \delta$ (w.r.t. the random choice of $\mathcal{S}$), the algorithm picks a hypothesis with an error that is at most ϵ from the optimal one. The difference between the risk of a particular classifier h and of the optimal classifier $h_* = \arg\min_{h \in \mathcal{H}} \mathcal{L}_{\mathcal{D}}(h)$ is called the excess risk of h, i.e., $\mathcal{E}(h) = \mathcal{L}_{\mathcal{D}}(h) - \mathcal{L}_{\mathcal{D}}(h_*)$.

Since the underlying distribution $\mathcal{D}$ is unknown to the learner, it is impossible for learner to directly minimize the generalization error or true risk. Therefore, one has to resort to using the training data in $\mathcal{S}$ to estimate the probabilities of error for a classifiers in $\mathcal{H}$. This alternative approach is known as the Empirical Risk Minimization (ERM) method and aims to pick a hypothesis which has small error on the training set, i.e., empirical risk. The performance of the empirical risk minimization has been throughly investigated and well understood using tools from empirical process theory. It is a well stablished fact that a problem is learnable with ERM method if and if only if the empirical error for all hypothesis in $\mathcal{H}$ converges uniformly to the true risk. Furthermore, the uniform convergence holds if the complexity of hypothesis class $\mathcal{H}$ satisfies some combinatorial characteristics. This is one of the main achievements of statical learning theory to characterize, and establish necessary and sufficient conditions for the learnability of learning problems using the ERM rule.

While the ERM method is theoretically appealing, from a practical point of view one would like to consider problems that are efficiently learnable which is referred to as *computational complexity* of learning algorithm. This issue becomes more important by noting the fact that in many cases ERM approach suffers from substantial problems such as computational requirements in minimizing 0-1 loss over training set. Indeed, solving the ERM problem for 0-1 loss function is known to be an NP-hard problem. Consequently, it is natural to consider loss functions that act as surrogates for the non-convex 0-1 loss, and lead to practical algorithms. Of course, such a surrogate loss must be reasonably related to the original binary loss function since otherwise this approach fails. For classification problem, good surrogate loss functions have been recently identified, and the relationship between the excess classification risk and the excess risk of these surrogate loss functions has been exactly described.

An important family of learning problems that can be learnt efficiently are called `Convex Learning Problems`. In general, a convex learning problem is a setting in which the surrogate loss function and the hypothesis space $\mathcal{H}$ are both convex. This setting encompasses an enormous variety of well-know practical learning algorithms such as regression, support vector machines (SVMs), boosting, and logistic regression, where these algorithms differ in the type of the convex loss function being used as the surrogate of the 0-1 loss.

Interestingly, for convex learning problems the ERM rule, of minimizing the empirical convex loss over a convex domain $\mathcal{H}$, becomes a convex optimization problem, making an intimate connection between machine learning and mathematical optimization. Learnability in this setting departs from the learnability via ERM method and strongly depends on the characteristics of convex domain such as boundedness and the analytical properties (curvature) of loss function such as Lipschitzness, smoothness (i.e, differentiable with Lipschitz gradients), and strong convexity (i.e., at any point one can find a convex quadratic lower bound for the function). Beyond learnability, the sample complexity of learning algorithms can also be characterized in terms of the analytical properties of loss function. Therefore, smoothness and strong convexity of convex surrogate loss function play a crucial role in characterizing learnability and analysis of sample complexity of convex learning problems.

1.1.1 Smoothness and sample complexity

While the main focus of statistical learning theory was on understanding learnability and sample complexity by investigating the complexity of hypothesis class in terms of known combinatorial measures under uniform convergence property, recent advances in online learning and optimization theory opened a new trend in understanding the generalization ability of learning algorithms in terms of the characteristics of loss functions being used in convex learning problems. In particular, a staggering number of results have focused on strong convexity of loss function and obtained better generalization bounds which are referred to as *fast* rates. In terms of smoothness of loss function, recently it has shown that under this assumption, it is possible to obtain *optimistic* rates (in the sense that smooth losses yield better generalization bounds when the problem is easier) which are more appealing than the case the convex surrogate loss is Lipschitz continuous. This motivates us to take a step forward in this direction and investigate the smoothness of loss functions in more depth.

1.1.2 Smoothness and binary excess risk

As noted above, the convex surrogates of the 0-1 loss are highly preferred because of the computational and theoretical virtues that convexity brings in. Since the choice of convex surrogates could significantly affect the binary excess risk, the relation between risk bounds in terms of binary 0-1 loss and it is corresponding convex surrogate has been the focus of learning community over the last decade. It was delivered that the binary excessive risk can be upper bound by the convex excessive risk through a transform function that only depends on the surrogate convex loss.

Although a great deal of work has been devoted to understanding the relation between binary excess risk and convex excess risk, there remain a variety of open problems. In particular, this transformation is well understood under mild conditions such as convexity and it is unclear how other properties of convex surrogates such as smoothness may affect this relation. This becomes more critical if we consider smooth surrogates as witnessed by the fact that the smoothness is further beneficial both computationally- by attaining an optimal convergence rate for optimization error, and in a statistical sense- by providing an improved optimistic rate for generalization bound. Given these positive news of using smooth convex surrogates, an open research question is how the smoothness of a convex surrogate will affect the binary excess risk. So we are thrived to investigate the impact of the smoothness of a convex loss function on transforming the excess risk in terms of the convex surrogate loss into the binary excess risk.

1.2 Sequential Prediction/Game Theoretic Learning

An alternative paradigm to analyze the learning problems is the sequential or online learning that can be phrased as a repeated two-player game between the learner and the adversary, making an intimate connection between learning and adversarial game theory.

In sequential prediction/online learning framework, the learner is faced with a sequence of samples appearing at discrete time intervals and is required to make predictions sequentially. In contrast to statistical setting, in which, the data source is typically assumed to be i.i.d. with an unknown distribution, in the online framework we relax or eliminate any stochastic assumptions imposed on the samples and they might be chosen adversarially. As a result, online learning framework is better suited for adversarial and interactive learning tasks such as spam email detection and stock market prediction where decisions of the learner could negatively affect future instances the learner receives.

By dropping the statistical assumptions on the observed sequence, it is not immediately clear how the prediction problem can be made meaningful and which goals are reasonable. One popular possibility is to measure the performance of the learner by the loss he/she has accumulated during the learning process and compare it to the loss of best fixed solution. The cumulative loss suffered on a sequence of rounds is the sum of instantaneous losses suffered on each one of the rounds in the sequence. In particular, the goal becomes to minimize the gap between the cumulative loss of the online learner and the loss of a strategy that selects the best action fixed in hindsight. This performance gap is called the *regret*. The analysis of regret mainly focuses on investigating how the regret depends on the length of the time horizon the game proceeds. We note that the best fixed action is chosen form a comparator class of predictors against which the learner will be compared and can only be computed in full knowledge of the sequence of loss functions.

Regret analysis stands in stark contrast to the statistical framework in which the learner is evaluated based on his/her accuracy after seeing all training examples, making the online learning setting inherently harder. The theoretical utility of online learning has long been appreciated. More recently, it has become the mainstay of optimization, where it serves as computational platform from which a variety of large-scale learning problems can be solved. The analogous of statistical learnability in online setting is referred to as Hannan consistency. A hypothesis class is learnable in online setting, i.e., Hannan consistent, if for any sequence of samples, there exists an algorithm which attains sub-linear regret in terms of number of rounds the interaction proceeds. Interestingly, unlike statistical learning theory, the analysis of online learning is mostly algorithmic where efficient algorithms are proposed to solve the learning problem and its performance is analyzed to guarantee the Hannan consistency.

Recently tools from convex optimization made it possible to capture many online learning problems under a generic problem template, and in many circumstances obtain improved regret bounds. This unified framework, which is referred to as `Online Convex Optimization`, assumes that the learner is forced to make decisions from a convex set and the adversary is supposed to play convex functions. Additionally, it has been demonstrated that the curvature of convex loss functions played by the adversary such as strong convexity gives a great advantage to the player to attain better regret bounds. Surprisingly, tools from online optimization also provided insights to get better convergence rates or more efficient algorithms for some stochastic and deterministic convex optimization problems.

1.2.1 Smoothness and regret bounds

Unlike strong convexity, the smoothness of loss functions is not a desirable property in online setting as it yields the same regret bounds as the loss functions being Lipschitz continuous. However, there are scenarios that the smoothness of the sequence of loss functions played by the adversary becomes important. One such scenario is the online learning from loss functions that might have some patterns and not being fully adversarial. For example, the weather condition or the stock price at one moment may have some correlation with the next and their difference is usually small, while abrupt changes only occur sporadically. Therefore devising online convex optimization algorithms which can take into account the gradual behavior of the environment and at the same time protect against the worst case sequences would be more desireable. In terms of regret analysis, this translates to having algorithms with regret bounded in terms of *variation* of loss functions instead of time horizon that is main measure in the standard setting of sequential prediction. In these evolving settings, the smoothness of loss function becomes critical. More importantly, no gradual variation bound is achievable if the loss functions are no longer smooth. This necessitates the need to develop online methods that exploit the smoothness assumption in the learning process or in the analysis to obtain improved regret bounds in terms of variation of the sequence of loss functions and underlines our motivation in this thesis.

1.3 Convex Optimization and Learning

In the problem of convex optimization, we are interested in minimizing a given convex function $f : \mathbb{R}^d \mapsto \mathbb{R}$ form a predefined family of convex functions $\mathcal{F}$ over a convex set $\mathcal{W} \subseteq \mathbb{R}^d$. The goal is to find an *approximate* solution with an accuracy ϵ, i.e., finding a $\widehat{\mathbf{w}} \in \mathcal{W}$ where $f(\widehat{\mathbf{w}}) - \min_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}) \leq \epsilon$. A typical optimization algorithm initially chooses a point from the feasible convex set $\mathbf{w}_0 \in \mathcal{W}$ and iteratively updates these points based on some information about the function at hand until it achieves the desired accuracy.

To capture the efficiency of an optimization procedure, we follow the black-box model of optimization [2]. In this model we assume that there exists an oracle which provides information about the query points such as function value, gradient, and second gradient (i.e, Hessian). The number of queries issued to an oracle to find a solution with a predefined level of accuracy is called *oracle complexity* when it is stated in terms of desired accuracy ϵ or equivalently *convergence rate* when it is stated in terms of the number of queries.

As already mentioned, learning problems under both statistical and online learning frameworks can be directly formulated as optimization problems. In statistical setting and espe-

[2] As indicated by Yurii Nesterov in his seminal book [121], in general, optimization problems are unsolvable and we need to relax the goal to make it reachable.

cially convex learning problems, the learning algorithm corresponds to the optimization algorithm that solves the minimization problem of picking a hypothesis from the set of hypotheses that minimizes empirical loss over training sample. Similarly, in the online convex optimization, the online learner iteratively chooses decisions from a closed, bounded and non-empty convex set and encounters convex cost functions.

Formulating and investigating both statistical and online learning problems in the context of convex optimization makes an intimate connection between learning and mathematical optimization. Therefore, the study of fast iterative methods for approximately solving convex programming problems is a central focus of research in convex optimization, with important applications in machine learning, and many other areas of computer science. The usefulness of convex optimization in the development of various learning algorithms is well established in the past several years. Additionally, challenges exist in machine learning applications demand the development of new optimization algorithms.

In optimization for supervised machine learning and in particular the empirical risk minimization paradigm with convex surrogates and gradient information, there exist two regimes in which popular algorithms tend to operate: *deterministic* (also known as batch optimization or full gradient method) regime in which whole training data are used to compute the gradient at each iteration and *stochastic* regime which samples a fixed number of training samples per iteration, typically a single training sample, to compute the gradient at each iteration. Although stochastic optimization methods suffer from the low convergence rate in comparison to batch methods, the lightweight computation per iteration makes them attractive for many large-scale learning problems. Hence, with the increasing amount of data that is available for training, stochastic convex optimization has emerged as the most scalable approach for large-scale machine learning which is known to yield moderately accurate solutions in a relatively short time.

We emphasize that the role of convex optimization goes beyond computational issues and it also provides tools to characterize the learnability in convex learning problems via efficient stochastic optimization algorithms for learning these problems.

Analogous to both statistical and online learning frameworks, the curvature of function to be optimized, significantly affects the convergence rate of optimization methods. Perhaps the most extensively studied are strong convexity and smoothness of function.

1.3.1 Smoothness and convergence rate

Exploiting smoothness of loss function, in particular in stochastic optimization, to obtain better convergence rate has been one the main research challenges in recent years. Despite enormous advances in exploiting smoothness in deterministic optimization, it has not been utilized in stochastic optimization. In particular, stochastic optimization of smooth loss functions exhibits the same convergence rate as stochastic optimization under Lipschitzness assumption of function. Therefore, this thesis is motivated by the need of developing stochastic optimization algorithms with better convergence rates under smoothness assumption. The key question is whether or not smoothness property of loss functions could be leveraged to develop much faster stochastic optimization methods.

1.3.2 Smoothness and projection-free optimization

At the core of many iterative constrained optimization algorithms in both online and stochastic convex optimization is a projection step to ensure the feasibility of solutions for intermediate iterations. This is a serious deficiency, since in many applications the projection onto the constrained domain might be computationally expensive and sometimes as hard as solving the original optimization problem. It is therefore of considerable interest to devise optimization methods which do not require projection steps or need a bounded number of projection operations. At it will became clear later in this thesis, by smoothing a strongly convex objective function, we are able to reduce the number of projections into a single projection at the end of the optimization process. In contrast to the other parts of the thesis where we assume and exploit the smoothness, this is the only result that injects and leverages the smoothness to gain from the merits of smoothness to be able to devise more efficient algorithms.

1.4 Main Contributions

In this section we shall elaborate on the main problems considered in this thesis and our key contributions to address these problems. A common theme in all of the algorithms is that they exploit the smoothness of loss function for more efficient methods.

1.4.1 Statistical Learning

- **Logarithmic sample complexity for learning from smooth and strongly convex losses with target risk.** The first problem we consider in this thesis has a statistical nature. In particular, we consider learning in passive setting but with a slight modification. We assume that the target expected loss, also referred to as *target risk*, is provided in advance for learner as prior knowledge. Unlike most studies in the learning theory that only incorporate the prior knowledge into the generalization bounds, we are able to explicitly utilize the target risk in the learning process.

Our analysis reveals a surprising result on the sample complexity of learning: by exploiting the target risk in the learning algorithm, we show that when the loss function is both smooth and strongly convex, the sample complexity reduces to $O\left(\log\left(\frac{1}{\epsilon}\right)\right)$, an exponential improvement compared to the sample complexity $O(\frac{1}{\epsilon})$ for learning with strongly convex loss functions.

Unlike the previous works on sample complexity, the proof of our result is constructive and is based on a computationally efficient stochastic optimization algorithm which makes it practically interesting. The proposed ClippedSGD algorithm uses knowledge of the target risk to appropriately clip gradients obtained from a stochastic oracle. The clipping is beneficial because it reduces the variance in stochastic gradients and makes it possible to reduce the sample complexity. This happens under the assumption that the loss function is smooth and strongly convex.

- **Statistical consistency of smoothed hinge loss.** The second problem we address in statistical learning setting is to investigate the relation between the excess risk that can be achieved by minimizing the empirical binary risk and the excess risk of smooth convex surrogates.

  As mentioned earlier, convex surrogates of the 0-1 loss are highly preferred because of the computational and theoretical virtues that convexity brings in. This is of more importance if we consider smooth surrogates as witnessed by the fact that the smoothness is further beneficial both computationally- by attaining an *optimal* convergence rate for optimization, and in a statistical sense- by providing an improved *optimistic* rate for generalization bound. However, we investigate the smoothness property from the viewpoint of statistical consistency and show how it affects the binary excess risk for smoothed hinge loss. In particular, we intend to answer the following fundamental questions:

”*How does the smoothness of surrogate convex loss affect the binary excess risk? Considering the advantages of smooth losses in terms of optimization and generalization, is it beneficial or detrimental in terms of statistical consistency? Under what conditions on these three types of errors it is better to use smooth losses?*”

We show that in contrast to optimization and generalization errors that favor the choice of smooth surrogate loss, the smoothness of loss function may deteriorate the binary excess risk. Motivated by this negative result, we provide a unified analysis that integrates optimization error, generalization bound, and the error in translating convex excess risk into a binary excess risk when examining the impact of smoothness on the binary excess risk. We show that under favorable conditions appropriate choice of smooth convex surrogate loss will result in a binary excess risk that is better than $O(1/\sqrt{n})$ which is unimprovable for general non-smooth Lipschitz losses.

1.4.2 Sequential Prediction/Online Learning

- **Regret bounded by gradual variation for smooth online convex optimization.** As our third problem, we study the online convex optimization problem under the assumption that even the loss functions are arbitrary, but there is a hidden pattern that can be exploited in learning process. Therefore, an interesting question that inspires our work in the analysis of online learning algorithms is the following:

  ”*Can we have online algorithms that can take advantage of benign sequences and at the same time protect against the adversarial sequences?*”

To answer this question, we introduce the *gradual variation*, measured by the sum of the distances between every two consecutive loss functions, to asses the performance of online learning algorithms in gradually evolving environments such as stock prediction. We propose two novel algorithms, an Improved Follow the Regularized Leader (IFTRL) algorithm and an Online Mirror Prox (OMP) method, that achieve a regret bound which only scales as the square root of the gradual variation for the linear and general smooth convex loss functions.

To establish the main results, we discuss a lower bound for online gradient descent, and a necessary condition on the smoothness of the cost functions for obtaining a gradual variation bound. For the closely related problem of prediction with expert advice, we show that an online algorithm modified from the multiplicative update algorithm can also achieve a similar regret bound for a different measure of deviation. Finally, for loss functions which are strongly convex in applications such as portfolio management problem, we show a regret which is only logarithmic in terms of the gradual variation.

The gradual variation- in addition to its intrinsic interest as an extension of regret analysis- has several specific consequences. First, since gradual variation lower bounds the regret bound, devising algorithm with small gradual variation is also guarantees to achieve small regret. Second, algorithms with small gradual variation are specifically designed to attain small variation, therefore they can capture the correlation between loss functions if it exist and boost the performance.

- **Gradual variation for composite online convex optimization.** As an impossibility result for obtaining gradual variation for convex losses, we show that for non-smooth functions when the only information presented to the learner is the first order information about the cost functions, it is impossible to obtain a regret bounded by gradual variation. However, we show that a gradual variation bound is achievable for a special class of non-smooth functions that is composed of a smooth component and a non-smooth component.

We consider two categories for the non-smooth component. In the first category, we assume that the non-smooth component is a fixed function and is relatively easy such that the composite gradient mapping can be solved without too much computational overhead compared to gradient mapping. In the second category, we assume that the non-smooth component can be written as an explicit maximization structure. In general, we consider a time-varying non-smooth component, present a primal-dual prox method, and prove a min-max regret bound by gradual variation. When the non-smooth components are equal across all trials, the usual regret is bounded by the min-max bound plus a variation in the non-smooth component.

1.4.3 Stochastic Optimization

- **Improved convergence rate for stochastic optimization of smooth losses.** We then turn to exploiting smoothness in stochastic optimization. Recently stochastic optimization methods have experienced a renaissance in the design of fast algorithms for large-scale learning problems. Unlike the optimization methods based on full gradients, the smoothness assumption was not exploited by most of stochastic optimization methods. More importantly, for general Lipschitz continuous convex functions, simple stochastic optimization methods such as stochastic gradient descent exhibit the same convergence rate as that for the smooth functions, implying that smoothness of the loss function is essentially not very useful and can not be exploited in stochastic optimization. Therefore, by noting this significant gap between the convergence rate of optimization of smooth functions in stochastic and deterministic optimization, the natural question arises is:

"*Can smoothness property of function be exploited to speed up the convergence rate of stochastic optimization of smooth functions?*".

We will provide an affirmative answer to this question. In particular, we propose a novel optimization paradigm which interpolates between stochastic and full gradient methods and is able to exploit the smoothness of loss functions in optimization process to obtain faster rates. The results show an intricate interplay between stochastic and deterministic convex optimization. The MixedGrad algorithm we propose fits in the mixed optimization paradigm and is an alternation of deterministic and stochastic gradient steps, with different of frequencies for each type of steps. We show that it attains an $O(1/T)$ convergence rate for smooth losses.

- **Condition number independent accesses of full gradient oracle for smooth and strongly convex optimization.** The optimal iteration complexity of the gradient based algorithm for smooth and strongly convex objectives is $O(\sqrt{\kappa}\log 1/\epsilon)$, where κ is the conditional number (the ratio of strong convexity to smoothness parameters). Despite its linear convergence rate in terms of target accuracy ϵ, in the case that the optimization problem is ill-conditioned, we need to evaluate a larger number of full gradients, which could be computationally expensive. Therefore, a natural question is: "*Can we manage the dependency on the condition number and devise optimization methods independent of condition number in accessing the full gradient oracle?*"

We show that in the mixed optimization regime introduced in this thesis, we may also leverage the smoothness assumption of loss functions to devise algorithms with iteration complexities that are independent of condition number in accessing the full gradient oracle. We utilize the idea of mixed optimization in progressively reducing the variance of stochastic gradients to optimize smooth and strongly convex functions, and propose the Epoch Mixed Gradient Descent (EMGD) algorithm that is independent of condition number in accessing the full gradients. Similar to the MixedGrad algorithm, a distinctive step in EMGD is the mixed gradient descent, where we use a combination of the full gradient and the stochastic gradient to update the intermediate solutions. By performing a fixed number of mixed gradient descents, we are able to improve the sub-optimality of the solution by a constant factor, and thus achieve a linear convergence rate. Theoretical analysis shows that EMGD is able to find an ϵ-optimal solution by computing $O(\log 1/\epsilon)$ full gradients and $O(\kappa^2 \log 1/\epsilon)$ stochastic gradients.

We also provide experimental evidence complementing our theoretical results for classification problem on few medium-sized data sets.

- **Efficient projection-free online and stochastic convex optimization.** Another problem we address in this thesis is efficient projection-free optimization methods for stochastic and online convex optimization. Our motivation stems from the observation that most of the gradient-based optimization algorithms require a projection onto the convex set $\mathcal{W}$ from which the decisions are made. While the projection is straightforward for simple shapes (e.g., Euclidean ball), for arbitrary complex sets this is the main computational bottleneck and may be inefficient in practice. For instance, for many applications in machine learning such as metric learning, the convex domain is the positive semidefinite cone for which is the projection step requires a full eigendecomposition. For many other problems, the projection step is itself an offline optimization problem and might be as hard as solving the original optimization problem. This observation immediately leads to the following question that inspires our work:

"*To what extent it is possible to reduce the number of expensive projection steps in online and stochastic optimization? Can we trade expensive projection steps off for other types of light computational operations?.*"

We consider this problem in two settings: stochastic optimization and online convex optimization. In stochastic setting, we develop novel stochastic optimization algorithms that do not need intermediate projections. Instead, only one projection at the last iteration is needed to obtain a feasible solution in the given domain. Our theoretical analysis shows that with a high probability, the proposed algorithms achieve an $O(1/\sqrt{T})$ convergence rate for general convex optimization, and an $O(\ln T/T)$ rate for strongly convex optimization under mild conditions about the domain and the objective function. The key insight which underlines the proposed projection-free algorithm for strongly convex functions is smoothing the objective function. This is in contrast to other problems in this thesis where we try to leverage the smoothness of objective, while here we introduce smoothness to gain from its computational virtues in alleviating the projection steps.

In online setting, we consider an alternative online convex optimization problem. Instead of requiring that decisions belong to a constrained convex domain for all rounds, we only require that the constraints, which define the convex set, be satisfied in the long run. By turning the problem into an online convex-concave optimization problem, we propose an efficient algorithm which achieves an $O(\sqrt{T})$ regret bound and an $O(T^{3/4})$ bound for the violation of constraints. Then we modify the algorithm in order to guarantee that the constraints are satisfied in the long run. This gain is achieved at the price of getting $O(T^{3/4})$ regret bound. We also prove an impossibility result which shows that simple ideas such as augmenting the objective function with penalized constraints fail to solve the problem and results in a linear bound $O(T)$ for either the regret or the violation of the constraints.

1.5 Thesis Overview

The remainder of this thesis is organized as follows. Chapter 2 lays out the foundation for the rest of the thesis. In particular, we provide a survey of some of the background material from statistical learning, sequential prediction/online learning theory, and as well as convex optimization. It will become clear in this chapter that there exist deep connections between these three areas.

The first part of the thesis focuses on the statistical learning, investigating the sample complexity of learning when the target risk is known to the learner and the consistency of smoothed hinge loss. In Chapter 3 we focus on statistical learning with target risk under the assumption that the loss function is smooth and strongly convex. Chapter 4 investigates the consistency of smoothed hinge loss and provides negative and positive results on transforming its excess risk to a binary excess risk.

The second part of the thesis is on sequential prediction/online learning and introduces the gradual variation measure to asses the performance of online convex optimization algorithms in gradually evolving environments. Chapter 5 discusses the necessity of smoothness to obtain regret bounds in terms of gradual variation, followed by two efficient algorithms to obtain gradual variation bounds for smooth online convex optimization problems. The adaption to other settings such as expert advice problem and strongly convex loss functions are also discussed. The extension of results to special composite loss functions with a smooth component is discussed in Chapter 6.

The third part of the thesis is devoted to devising efficient stochastic optimization algorithms by leveraging the smoothness of loss functions. We propose the mixed optimization paradigm for stochastic optimization in Chapter 7 and extend to smooth and strongly convex losses in Chapter 8. The stochastic optimization methods with bounded projections and online optimization with soft constraints are elaborated in Chapter 9.

Finally, the appendix summarizes rather standard things on convex analysis and concentration inequalities that are used in the proof of results in the thesis and is mainly for reference. In order to facilitate independent reading of various chapters, some of the definitions from convex analysis are even repeated several times.

1.6 Bibliographic Notes

Some of the results in this dissertation have appeared in prior publications. The material in Chapter 3 is based on a work published in Conference on Learning Theory (COLT)[102] and the content of Chapter 4 is new [108]. The material in Chapter 5 and Chapter 6 come from [41] which is published at COLT and its extended version has been recently published in Machine Learning journal [150]. The results in Chapter 7 and Chapter 8 follow [107] and [152], respectively, which are appeared in Advances in Neural Information Processing Systems (NIPS). The content of Chapter 9 is mostly compiled from [106], [104], and [105] which are published at NIPS and Journal of Machine Learning Research (JMLR).

Chapter 2

Preliminaries

The goal of this chapter is to give a gentle and formal overview of the material related to the work has been done in this thesis. In particular, we will discuss key concepts and questions in statistical learning, online learning, and convex optimization and will highlight the role of analytical properties of loss functions such as Lipschitzness, strong convexity, and smoothness in all these settings. The exposition given here is necessarily very brief and the detailed discussion will be provided in the relevant chapters.

2.1 Statistical Learning

2.1.1 Statistical Learning Model

In a typical supervised statistical learning problem (also known as passive learning or batch learning), we are given an instance space $\mathcal{X}$, and a space $\mathcal{Y}$ of labels or target values. Each element in the data domain $\mathcal{X}$ represents an object to be classified, e.g., the content of an email in spam detection application or the features of an image in vision applications. The target space $\mathcal{Y}$ can be either discrete $\mathcal{Y} = \{-1, +1\}$, as in the case of classification, or continuous $\mathcal{Y} = \mathbb{R}$, as in the case of regression. To model learning problems in statistical or probabilistic setting, we assume that the product space $\Xi \equiv \mathcal{X} \times \mathcal{Y}$ is endowed with a probability measure $\mathcal{D}$ which is unknown to the learner. However, it is possible to sample an arbitrary number of pairs $\mathcal{S} = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)) \in \Xi^n$ according to the underlying distribution $\mathcal{D}$. We term this set of examples the training set, or the training sample. The existence of distribution $\mathcal{D}$ is necessary to ensure that the already collected samples $\mathcal{S}$ have something in common with the new and unseen data.

A hypothesis or classifier $h : \mathcal{X} \mapsto \mathcal{Y}$ is a function that assigns labels $h(\mathbf{x}) \in \mathcal{Y}$ to any instance $\mathbf{x} \in \mathcal{X}$ such that the assigned label is a good approximation of the possible response y to an arbitrary instance $\mathbf{x}$ generated according to the distribution $\mathcal{D}$. In other words, the hypothesis h captures the functional relationship between the input instances and the output, which in turn makes it possible to predict the output value for future input instances.

In light of no free lunch theorem [148], learning is impossible unless we make assumptions regarding the nature of the problem at hand. Therefore, when approaching a particular learning problem, it is desirable to take into account some prior knowledge we might have about our problem. In this regard, we assume that the learning algorithm is confined to a predetermined set of candidate hypotheses $\mathcal{H} = \{\mathcal{X} \mapsto \mathcal{Y}\}$ to which we wish to compare the result of our learning algorithm. In a specific learning context, the hypothesis class can represent our beliefs on the true nature of the classification rule for the problem. For example the hypothesis class for binary classification might be a subset of a vector space with bounded norm which represents the linear classifiers, i.e., $\mathcal{H} = \{\mathbf{x} \mapsto \text{sign}(\langle \mathbf{w}, \mathbf{x} \rangle + b),\ \mathbf{w} \in \mathbb{R}^d,\ b \in \mathbb{R},\ \|\mathbf{w}\| \leq R\}$.

In order to measure the performance of a learning algorithm, we usually use a loss function $\ell : \mathcal{H} \times \Xi \mapsto \mathbb{R}_+$. The instantaneous loss incurred by a learning algorithm on instance $\mathbf{z} = (\mathbf{x}, y) \in \Xi$ for picking hypothesis $h \in \mathcal{H}$ is given by $\ell(h, \mathbf{z})$. For example, in *binary classification* problems, $\Xi = \mathcal{X} \times \{-1, 1\}$, $\mathcal{H}$ is a set of functions $h : \mathcal{X} \mapsto \{-1, 1\}$, and the loss function is the binary or 0-1 loss defined as $\ell_{0-1}(h(\mathbf{x}), y) = \mathbb{I}[h(\mathbf{x}) \neq y]$, where $\mathbb{I}[\cdot]$ is the indicator function that takes value 1 if its argument is true and 0 otherwise. In *regression* or real classification problems, where the goal is to predict real valued labels $\mathcal{Y} = \mathbb{R}$, the common loss function used to evaluate the performance of a regressor h on sample $\mathbf{z} = (\mathbf{x}, y)$ is the squared loss $\ell(h, \mathbf{z}) = (h(\mathbf{x}) - y)^2$.

A classifier is constructed on the basis of the n independent and identically distributed (i.i.d) samples $\mathcal{S} = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n))$ from Ξ. The ultimate goal of a typical learning algorithm is to pick a classifier $h \in \mathcal{H}$ that is competitive with respect to the best hypothesis from $\mathcal{H}$ with respect to the *expected risk* or *generalization error* defined as:

$$\mathcal{L}_{\mathcal{D}}(h) = \mathbb{E}_{\mathbf{z} \sim \mathcal{D}}[\ell(h, \mathbf{z})],$$

where $\mathbb{E}[\cdot]$ denotes the expectation with respect to the (unknown) probability distribution $\mathcal{D}$ underlying our samples and $\ell(h, (\mathbf{x}, y))$ is the binary loss $\mathbb{I}[h(\mathbf{x}) \neq y]$ for classification and squared loss $(h(\mathbf{x}) - y)^2$ for regression problem. For binary classification the generalization error of a hypotheses $h : \mathcal{X} \mapsto \{-1, +1\}$ is simply the probability that it predicts the wrong label on a randomly drawn instance from Ξ, i.e., $\mathcal{L}_{\mathcal{D}}(h) = \mathbb{P}_{(\mathbf{x}, y) \sim \mathcal{D}}[h(\mathbf{x}) \neq y]$.

The difference between the risk of a particular classifier h and of the optimal classifier $h_* = \arg\min_{h \in \mathcal{H}} \mathcal{L}_{\mathcal{D}}(h)$ is called the *excess risk* of h, i.e.,

$$\mathcal{E}(h) = \mathcal{L}_{\mathcal{D}}(h) - \mathcal{L}_{\mathcal{D}}(h_*).$$

In designing any typical solution to a supervised machine learning problem, there are few key questions that must be considered. The first of these concerns approximation that characterizes how rich the solution space $\mathcal{H}$ is to approximate the true underlying model. The second fundamental issue concerns estimation that characterizes how well the obtained solution performs in making future predictions on unseen data and how much training samples suffices to find the solution. The third key question concerns the computational efficiency that characterizes how efficiently can we make use of the training data to choose an accurate hypothesis.

The basic model to analyze learning algorithms in computational learning theory is the Probably Approximately Correct (PAC) model proposed by the pioneering work of Valiant [144]. It applies to learning binary valued functions and uses the 0-1 loss under realizability assumption, i.e., the algorithm gets samples that are consistent with a hypothesis in a fixed class $\mathcal{H}$, $\exists\, h_* \in \mathcal{H}; \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[h_*(\mathbf{x}) = y] = 1$. In PAC model we bound the loss of the algorithm with a high probability over the random draw of samples. A decision-theoretic extension of the PAC framework which is known as *agnostic learning* is introduced by [87] that generalizes the PAC model to general loss functions and without assuming realizability assumption as defined below:

Definition 2.1 (Agnostic PAC Learnability). *A hypothesis class $\mathcal{H}$ is agnostic PAC learnable with respect to Ξ and a loss function $\ell : \mathcal{H}\times\Xi \rightarrow \mathbb{R}_+$, if there exists a function $m_{\mathcal{H}} : (0,1)^2 \rightarrow \mathbb{N}$ and a learning algorithm $\mathcal{A}$ with the following property: for every $\epsilon, \delta \in (0,1)$ and for any distribution $\mathcal{D}$ over the domain Ξ, when running the algorithm $\mathcal{A}$ on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$, the algorithm returns $h \in \mathcal{H}$ such that, with probability of at least $1 - \delta$,*

$$\mathcal{L}_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} \mathcal{L}_{\mathcal{D}}(h') + \epsilon.$$

If further $\mathcal{A}$ runs in poly$(1/\epsilon, 1/\delta, n)$, then $\mathcal{H}$ is said to be efficiently agnostic PAC-learnable.

The goal of the PAC framework is to understand how large a data set needs to be in order to give good generalization. It also provides bounds for the computational cost of learning. In agnostic PAC learnability there are two fundamental questions that need to be addressed carefully: these are *computational efficiency* and *sample complexity*.

The computational aspect of learning measures the amount of computation required to implement a learning algorithm. The sample complexity of an algorithm is the number of examples which is sufficient to ensure that, with probability at least $1-\delta$ (w.r.t. the random choice of $\mathcal{S}$), the algorithm picks a hypothesis with an error that is at most ϵ from the optimal one. We note that while computational complexity concerns the efficiency of learning, the sample complexity is a statistical measure and concerns the difficulty of learning from the hypothesis $\mathcal{H}$ with respect to the underlying distribution $\mathcal{D}$. An equivalent way to present the sample complexity is to give a generalization bound. It states that with probability at least $1-\delta$, to attain a risk $\mathcal{L}_{\mathcal{D}}(h)$ which departs from the optimal risk $\min_{h' \in \mathcal{H}} \mathcal{L}_{\mathcal{D}}(h')$ by at most ϵ, is upper bounded by some quantity that depends on the sample size n and δ.

Sample complexity of passive learning is well established and goes back to early works in the learning theory where the lower bounds $\Omega\left(\frac{1}{\epsilon}(\log\frac{1}{\epsilon} + \log\frac{1}{\delta})\right)$ and $\Omega\left(\frac{1}{\epsilon^2}(\log\frac{1}{\epsilon} + \log\frac{1}{\delta})\right)$ were obtained in classic PAC and general agnostic PAC settings, respectively [52, 26, 7]. It worth emphasizing that the PAC framework is a distribution-free model and we are interested in sample-complexity guarantees that hold regardless of the distribution $\mathcal{D}$ from which examples are drawn.

2.1.2 Empirical Risk Minimization

Since in the probabilistic setting, we assume that there is an underlying probability distribution $\mathcal{D}$ over the sample space $\mathcal{X} \times \mathcal{Y}$ which captures the relationship between the samples given to the algorithm during training and the new instances it will receive in the future;

the training examples must be 'representative' in some way of the examples to be seen in the future. Clearly, learning is hopeless if there is no correlation between past and present rounds.

Note however that the distribution $\mathcal{D}$ is not known to the learner; the learner sees the distribution only through the training examples $\mathcal{S} = (\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_n) \in \Xi^n$, and based on these examples, must learn to predict well on new instances from the same distribution. Therefore, we cannot compute the generalization error directly and machine learning aims to find estimators based on the observed data samples $\mathcal{S}$.

A simple and well-known learning approach is the empirical risk minimization (ERM) method. Basically, the idea of ERM is to replace the unknown true risk

$$\mathcal{L}_{\mathcal{D}} = \mathbb{P}_{(\mathbf{x},y)\sim\mathcal{D}}[h(\mathbf{x}) \neq y]$$

by its empirical counterpart rooted in the training set $\mathcal{S}$ and minimize this empirical risk as defined below:

$$\mathcal{L}_{\mathcal{S}}(h) = \frac{1}{n}\Big|\{i : i \in [n] \;\text{ and }\; h(\mathbf{x}_i) \neq y_i\}\Big|.$$

The empirical error $\mathcal{L}_{\mathcal{S}}(h)$ of any hypothesis $h \in \mathcal{H}$ is its average error over the training samples in $\mathcal{S}$, while the generalization error $\mathcal{L}_{\mathcal{D}}(h)$ is its expected error based on a random sample realized by the distribution $\mathcal{D}$. We note that the empirical error is a useful quantity, since it can easily be determined from the training data and it provides a simple estimate of the true error. The empirical loss over the training data provides an estimate whose loss is close to the optimal loss if the class $\mathcal{H}$ is sufficiently large so that the loss of the best function in $\mathcal{H}$ is close to the optimal loss and is small enough so that finding the best candidate in $\mathcal{H}$ based on the data is computationally feasible. In this regard, generalization error bounds give an upper bound on the difference between the true and empirical error of functions in a given class, which holds with high probability with respect to the sampling of the training set.

Then, our task becomes to evaluate the expected risk relying on the empirical error. Having this quantity bounded, a learning algorithm may choose the hypothesis that is the most accurate on the sample, and is guaranteed that its loss on the distribution will also be low. The statistical learning theory is concerned with characterizing learnability and providing bounds on the deviations of this estimate from the expected error. One of its main achievements is a complete characterization of the necessary and sufficient conditions for generalization of ERM, and for its consistency.

A fundamental answer, formally proven for supervised classification and regression, is that learnability is equivalent to uniform convergence, and that if a problem is learnable, it is learnable via empirical risk minimization.

Definition 2.2 (Uniform Convergence)**.** *A hypothesis class $\mathcal{H}$ has the uniform convergence property with respect to Ξ and the loss function $\ell : \mathcal{H} \times \Xi \mapsto \mathbb{R}_+$, if for any probability distribution $\mathcal{D}$ over Ξ, there exists a function $m_{\mathcal{H}} : (0,1)^2 \mapsto \mathbb{N}$ such that for any sample $\mathcal{S}$ of size $m_{\mathcal{H}}(\epsilon, \delta)$ drawn i.i.d based on $\mathcal{D}$, with probability at least $1-\delta$, $\forall h \in \mathcal{H}$ it holds:*

$$\left|\mathcal{L}_{\mathcal{S}}(h) - \mathcal{L}_{\mathcal{D}}(h)\right| \leq \epsilon.$$

Hence, the crucial step towards proving learnability is to obtain a result on the uniform convergence of sample errors to true errors. Uniform convergence of empirical quantities to their mean provides ways to bound the gap between the expected risk and the empirical risk by the complexity of the hypothesis set. Hence, the complexity of the hypothesis class $\mathcal{H}$ is the critical factor in determining the distribution-free sample-complexity of a supervised learning problem. Several complexity measures for hypothesis classes have been proposed, each providing a different type of guarantee including the Vapnik-Chervonenkis (VC) dimension [146] and the Rademacher complexity [15, 91]. The main virtue of the Vapnik-Chervonenkis theorem and Rademacher complexity is that they convert the problem of uniform deviations of empirical averages into a combinatorial and data dependent problems, respectively.

We note that uniform convergence arguments is not the only possible way to characterize learnability. Since the first results of Vapnik and Chervonenkis on uniform laws of large numbers for classes of binary valued functions, there has been a considerable amount of work aiming at obtaining generalizations and refinements of these bounds. These techniques include sample compression [55], algorithmic stability [32], and PAC-Bayesian analysis [112] which also have been shown for characterizing learnability and proving generalization bounds. We will also discuss the stochastic optimization machinery [133] to characterize learnability in general settings later in this chapter.

2.1.3 Surrogate Loss Functions and Statistical Consistency

Although ERM approach has a lot of theoretical merits, since we should seek to minimize the training error based on 0-1 loss, it typically is a combinatorial problem, leading to NP-hard optimization problem which is not computationally realizable.

A common practice to circumvent this difficulty is to revert to minimize a surrogate loss function, i.e., to replace the indicator function by a surrogate function and find the minimizer with respect to this surrogate function. Obviously, the surrogate loss needs to be computationally easy to optimize, while close in some sense to the 0-1 loss. In partic-

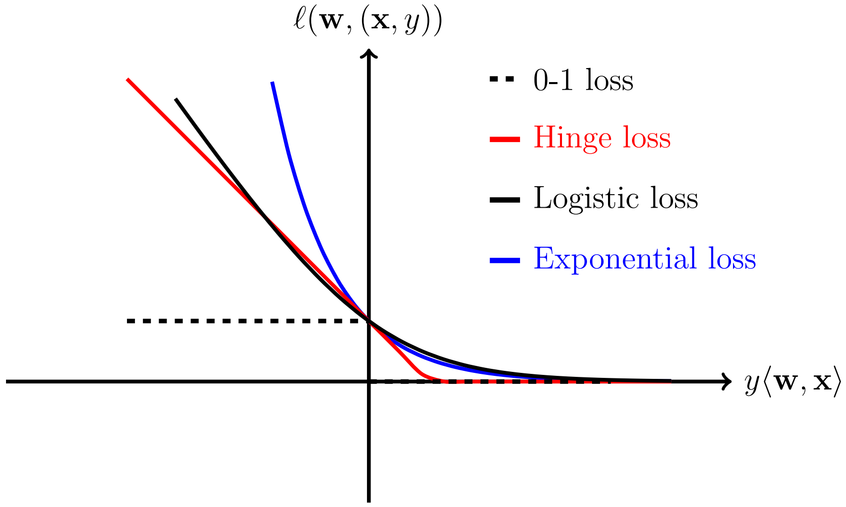


Figure 2.1: Illustrations of the 0-1 loss function, and three surrogate convex loss functions: hinge loss, logistic loss, and exponential loss as scalar functions of $y\langle \mathbf{w}, \mathbf{x} \rangle$.

ular, if the surrogate function is assumed to be convex, it allows the optimization to be performed efficiently with only modest computational resources. Examples of such surrogate loss functions for 0-1 loss include logistic loss $\ell_{\text{log}}(h, (\mathbf{x}, y)) = \log(1 + \exp(-yh(\mathbf{x})))$ in logistic regression [61], hinge loss $\ell_{\text{hinge}}(h, (\mathbf{x}, y)) = \max(0, 1 - yh(\mathbf{x}))$ in support vector machines (SVMs) [43] and exponential loss $\ell_{\text{exp}}(h, (\mathbf{x}, y)) = \exp(-yh(\mathbf{x}))$ in boosting (e.g., AdaBoost [58]). When the hypothesis class $\mathcal{H}$ consists of functions that are linear in a parameter vector $\mathbf{w}$, i.e., linear classifiers, these loss functions are depicted in Figure 2.1.

Having defined the surrogate loss functions, then the task is to minimize the relaxed empirical loss in terms of the surrogate losses. However, in practice, the ubiquitous approach to find the solution is the regularized empirical risk minimization which adds a regularization function $\mathcal{R}(h)$ to the objective and solves

$$h_{\mathcal{S}} \in \arg\min_{h \in \mathcal{H}} \left\{ \mathcal{L}_{\mathcal{S}}(h) + \mathcal{R}(h) \equiv \frac{1}{n} \sum_{i=1}^{n} \ell(h, \mathbf{z}_i) + \mathcal{R}(h) \right\}. \tag{2.1}$$

The goal of introducing regularizer is to prevent over-fitting. Of course, given some training data, it is always possible to build a function that fits exactly the data. But, in the presence of noise, this may lead to a poor performance on unseen instances. An immediate consequence of adding the regularization term is to favor simpler classifiers to increase its generalization capability. Some of the commonly used regularizers in the literature are $\mathcal{R}(h) = \|h\|_2^2$, and $\mathcal{R}(h) = \|h\|_1$. We note that solving the optimization problem in (2.1) is a convex optimization problem for which efficient algorithms exist to find a near optimal solution in a reasonable amount of time.

Although the idea of replacing the non-convex 0-1 loss function with convex surrogate loss functions seems appealing and resolves the efficiency issue of the ERM method, but it has statistical consequences that must be balanced against the computational virtues of convexity. The question then is how well does minimizing such a convex surrogate perform relative to minimizing the actual classification error. Statistical consistency concerns this issue. Consistency requires convergence of the empirical risk to the expected risk for the minimizer of the empirical risk together with convergence of the expected risk to the minimum risk achievable by functions in $\mathcal{H}$ [14]. An important line of research in statistical learning theory focused on relating the convex excess risk to the binary excess risk. It is known that under mild conditions, the classifier learned by minimizing the empirical loss of convex surrogate is consistent to the Bayes classifier [156, 101, 80, 97, 141, 14]. For instance, it was shown in [14] that the necessary and sufficient condition for a convex loss $\ell(\cdot)$ to be consistent with the binary loss is that $\ell(\cdot)$ is differentiable at origin and $\ell'(0) < 0$. It was further established in the same work that the binary excessive risk can be upper bound by the convex excess risk through a ψ-transform that depends on the surrogate convex loss $\ell(\cdot)$. A detailed elaboration of this issue will be given in Chapter 4 where we examine the statistical consistentency of smooth convex surrogates.

2.1.4 Convex Learning Problems

We now turn our attention to *convex learning* problems where Ξ be an arbitrary measurable set, $\mathcal{H}$ be a closed, convex subset of a vector space, and the loss function $\ell(h, \mathbf{z})$ be convex w.r.t. its first argument. This family of learning encompasses a rich body of existing learning methods for which efficient algorithms exists such as support vector machines, boosting and logistic regression. Convex learning problems makes an important family of learning problems, mainly because most of what we can learn efficiently falls into this family.

Before diving into formal definition, we need to familiarize ourselves with the following definitions about convex analysis [28, 121] which will come in handy throughout this dissertation (for standard definitions about convex analysis see Appendix **??**).

Definition 2.3 (Convexity). *A set $\mathcal{W}$ in a vector space is convex if for any two vectors $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$, the line segment connecting two points is contained in $\mathcal{W}$ as well. In other fords, for any $\lambda \in [0,1]$, we have that $\lambda\mathbf{w} + (1-\lambda)\mathbf{w}' \in \mathcal{W}$. A function $f : \mathcal{W} \mapsto \mathbb{R}$ is said to be convex if $\mathcal{W}$ is convex and for every $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$ and $\alpha \in [0,1]$,*

$$f(\lambda\mathbf{w} + (1-\lambda)\mathbf{w}') \leq \lambda f(\mathbf{w}) + (1-\lambda) f(\mathbf{w}').$$

A continuously differentiable function is convex if $f(\mathbf{w}) \geq f(\mathbf{w}') + \langle \nabla f(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle$ for all $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$. If f is non-smooth then this inequality holds for any sub-gradient $\mathbf{g} \in \partial f(\mathbf{w}')$.

The formal definition of convex learning problems is given below.

Definition 2.4 (Convex Learning Problem). *A learning problem with hypothesis space $\mathcal{H}$,* *instance space $\Xi = \mathcal{X} \times \mathcal{Y}$, and the loss function $\ell : \Xi \times \mathcal{H} \mapsto \mathbb{R}_+$ is said to be convex if the hypothesis class $\mathcal{H}$ is a parametrized convex set $\mathcal{H} = \{h_{\mathbf{w}} : \mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle : \mathbf{w} \in \mathbb{R}^d, \|\mathbf{w}\| \leq R\}$ and for all $\mathbf{z} = (\mathbf{x}, y) \in \Xi$, the loss function $\ell(\cdot, \mathbf{z})$ is a non-negative convex function.*

In the remainder of thesis, when it is clear from the context, we will represent the hypothesis class with $\mathcal{W}$ and simply use vector $\mathbf{w}$ to represent $h_{\mathbf{w}}$, rather than working with hypothesis $h_{\mathbf{w}}$.

We note that for convex learning problems, the ERM rule becomes a convex optimization problem which can be efficiently solved. This stands in sharp contrast to non-convex loss functions such as 0-1 loss for which solving the ERM rule is computationally cumbersome and known to be NP-hard. Obviously, this efficiency comes at a price and not every convex learning problem is guaranteed to be learnable and convexity by itself is not sufficient for learnability. This requires to impose more assumptions on the setting to ensure the learnability of the problem. In particular, it can been shown that if the hypothesis space $\mathcal{W}$ is bounded and the loss function is Lipschtiz or smooth as formally defined below, then the convex learning problem is learnable [133, 130].

Definition 2.5 (Lipschitzness)**.** *A function $f : \mathcal{W} \mapsto \mathbb{R}$ is ρ-Lipschtiz over the set $\mathcal{W}$ if for every $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$ we have that $|f(\mathbf{w}) - f(\mathbf{w}')| \leq \rho||\mathbf{w} - \mathbf{w}'||$.*

Definition 2.6 (Smoothness)**.** *A differentiable loss function $f : \mathcal{W} \mapsto \mathbb{R}$ is said to be β-smooth with respect to a norm $\| \cdot \|$, if it holds that*

$$f(\mathbf{w}) \leq f(\mathbf{w}') + \langle \nabla f(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle + \frac{\beta}{2}\|\mathbf{w} - \mathbf{w}'\|^2, \ \forall \ \mathbf{w}, \mathbf{w}' \in \mathcal{W}. \tag{2.2}$$

We note that smoothness also follows if the gradient of the loss function is β-Lipschtiz, i.e., $\|\nabla f(\mathbf{w}) - \nabla f(\mathbf{w'})\| \leq \beta \|\mathbf{w} - \mathbf{w'}\|$. Smooth functions arise, for instance, in logistic and least-squares regression, and in general for learning linear predictors where the loss function has a Lipschitz continuous gradient.

There has been an upsurge of interest over the last decade in finding tight upper bounds on the sample complexity of convex learning problems by utilizing prior knowledge on the curvature of the loss function, that led to stronger generalization bounds in agnostic PAC setting. In [95] *fast* rates obtained for squared loss, exploiting the strong convexity of this loss function, which only holds under pseudo-dimensionality assumption. With the recent development in online strongly convex optimization [68], fast rates approaching $O(\frac{1}{\epsilon} \log \frac{1}{\delta})$ for convex Lipschitz strongly convex loss functions has been obtained in [83, 140, 82] and for exponentially concave loss functions in [103]. For smooth non-negative loss functions, [138] improved the sample complexity to *optimistic* rates for non-parametric learning using the notion of local Rademacher complexity [13].

2.2 Sequential Prediction/Online Learning

The statistical model discussed above, first assumes the existence of a stochastic model for generating instances according to the underlying distribution $\mathcal{D}$, and then samples a training set $\mathcal{S} = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n))$ and investigates the ERM strategy to find a hypothesis $h \in \mathcal{H}$ which generalizes well on unseen instances. Although, this model is valid for cases for which a tractable statistical model reasonably describes the underlying process, but it may be unrealistic in practical problems where the process is hard to model from a statistical viewpoint and may even react to the learners's decisions, e.g., applications such as portfolio management, computational finance, and whether prediction.

Sequential prediction/online learning (also known as universal prediction of individual sequences) is a strand of learning theory avoiding making any stochastic assumptions about the way the observations are generated and the goal is to develop prediction methods that are robust in the sense that they work well even in the worst case. For instance, in the problem of online portfolio management [18], an online investor wants to distribute her wealth on a set of available financial instruments without any assumption on the market outcome in advance. Obviously in applications of this kind, the main challenge is that the learner can not make any statistical assumption about the process generating the instances and the data are continuously evolving or adversarially changing. Online learning or sequential prediction is an elegant paradigm to capture these problems that alleviates the statistical assumption usually made in statistical setting (see e.g., [38] and [129] for through discussion).

2.2.1 Mistake Bound Model and Regret Analysis

The problem of sequential prediction may be cast as a repeated game between a decision maker- also called the forecaster- and an environment- also called adversary. In this model, learning proceeds in T consecutive rounds, as we see examples one by one. At the beginning of round t, the learning algorithm $\mathcal{A}$ has the hypothesis $h_t \in \mathcal{H}$ and the adversary picks an instance $\mathbf{z}_t = (\mathbf{x}_t, y_t)$. The adversary at round t can select the instance $\mathbf{z}_t \in \Xi$ in an adversarial worst case fashion based on previous instances $\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}$ and based on previous hypotheses $h_1, \ldots, h_{t-1}$ selected by the learner. Then, the learner receives the instance $\mathbf{x}_t$ and predicts $h_t(\mathbf{x}_t)$. At the end of the round, the true label y_t is revealed to the learner and $\mathcal{A}$ makes a mistake if $h_t(\mathbf{x}_t) \neq y_t$. Unlike statistical setting, here the prediction task is sequential: the outcomes are only revealed one after another; at time t, the learner guesses the next outcome y_t before it is revealed. The algorithm then updates its hypothesis, if necessary, to h_{t+1} and this continues till time T.

The sequential prediction model discussed above, which is reminiscent of the framework of competitive analysis [27], is known as *mistake bound model* and was introduced to learning community in [98]. In this model the goal of the learner is to sequentially deduce information from previous rounds so as to improve its predictions on future rounds. When the algorithm is conservative (or lazy), meaning that the algorithm only changes its hypothesis when it makes a mistake is called *mistake driven*. We note that many seemingly unrelated problems fit in the framework of the abstract sequential decision problem including online prediction problems in the experts model [37], Perceptron like classification algorithms [25], Winnow algorithm [98], and learning in repeated game playing [59]- to name a few. Here we list few sequential prediction problems to better illustrate the setting.

Example 1: Online classification and regression. As an illustrative example let us consider the *online binary classification* problem where at each round t the learner receives an instance $\mathbf{x}_t \in \mathcal{X}$ as input and is required to predict a label $\hat{y}_t \in \mathcal{Y} = \{-1, +1\}$ for the input instance. Then, the learner receives the correct label $y_t \in \mathcal{Y}$ and suffers the 0-1 loss: $\mathbb{I}[y_t \neq \hat{y}_t]$. As another example, let us consider *online regression* problems where at each round t a feature vector $\mathbf{x}_t \in \mathbb{R}^d$ is given to the online learner, and a value $y_t \in \mathbb{R}$ has to be estimated using linear predictors with bounded norm, i.e, $\mathcal{H} = \{\mathbf{x} \mapsto \mathbf{w}, \mathbf{x} : \mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\| \leq R\}$, that the learner predicts $\langle \mathbf{w}, \mathbf{x}_t \rangle$. The loss function at round t for a predictor $\mathbf{w}$ is $\ell_t(\mathbf{w}) = (y_t - \langle \mathbf{w}_t, \mathbf{x}_t \rangle)^2$.

Example 2: Prediction with expert advice. Every day the manager of a company should decide to produce one of the K different products without knowing the market demand in advance. At the end of the day, he will be informed the gain achieved by selling the product but nothing about the potential income from other products. The goal of the manager is to maximize the income of company over a sequence of many periods. This is a problem of repeated decision-making which is called learning from expert advice where the objective functions to be optimized are unknown and revealed (perhaps only partially) in an online manner. In the general prediction with expert advice game a learner competes against $K \in \mathbb{N}$ experts in a game consisting of T rounds. Each round t, each expert reveals a prediction from $\mathcal{Y} = \{0, 1\}$. The learner form its own prediction by sampling an expert from $h_t = \mathbf{w}_t \in \mathcal{H} \equiv \Delta_K$, where Δ_K is the set of probabilities over K experts (i.e., simplex). The true outcome y_t is then revealed and the learner and all of the experts receive a penalty depending on how well their prediction fits with the revealed outcome. The aim of the learner in this game is to incur a cumulative loss over all rounds that is not much worse than the best expert.

One natural measure of the quality of learning in mistake bound model of sequential setting is the number of worst case mistakes the learner makes. In particular, under the realizability assumption (i.e. where there exists a hypothesis in $\mathcal{H}$ which performs perfectly on the sequence), the learner's goal becomes to have a bounded number of mistakes which is known as *mistake bound.* The optimal mistake bound for a hypothesis class $\mathcal{H}$ is the minimum mistake bound over all learning algorithms $\mathcal{A}$ on the worst case sequence of examples:

$$\text{Mistake}(\mathcal{A}, \mathcal{H}, \Xi, T) = \min_{h_1, h_2, \cdots, h_T \in \mathcal{H}} \max_{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_T \in \mathcal{X}} \sum_{t=1}^{T} \mathbb{I}[h_t(\mathbf{x}_t) \neq h(\mathbf{x}_t)], \tag{2.3}$$

where $h_1, h_2, \cdots, h_T \in \mathcal{H}$ is the sequence of hypothesis generated by $\mathcal{A}$.

We say that a hypothesis class $\mathcal{H}$ is learnable in the online learning model if there exists an online learning algorithm $\mathcal{A}$ with a finite worst case mistake bound, no matter how long the sequence of examples T is. We note that the mistake bound model, in comparison to PAC model, is strong in the sense that it does not depend on any assumption about the instances. It is also remarkable that despite the inherent differences between PAC and mistake bound frameworks, mistake bounds have corresponding risk bounds that are not worse, and sometimes better, than those obtainable with a direct statistical approach. In particular, by a simple reduction, it is straightforward to show that if an algorithm $\mathcal{A}$ learns a hypothesis class $\mathcal{H}$ in the mistake bound model, then $\mathcal{A}$ also learns $\mathcal{H}$ in the probably approximately correct model.

We note that due to impossibility theorem by Cover [45], any online predictor that makes deterministic predictions is doomed to achieve a sub-linear regret universally for all sequences. To circumvent this obstacle, two typical solutions have been examined which are randomization and convexification. In the former we allow the learner to make randomized predictions, making the algorithm unpredictable against the adversary, and in the latter we replace the non-convex 0-1 loss with a cover surrogate loss function (see e.g., [24] and [129]).

Similar to statistical learning, we can also generalize online setting to the agnostic (a.k.a. non-realizable) setting where there is no classifier in $\mathcal{H}$ which performs perfectly on the sequence. In this case an adversary can make the cumulative loss of our online learning algorithm arbitrarily large. To overcome this deficiency, the performance of the forecaster is compared to some notion of "how well it could have performed". In particular, the performance of the online learner is compared to that of the best single decision for the sequence, in hindsight, chosen from the hypothesis in $\mathcal{H}$. This brings us to the objective which is commonly known as *regret* which is formally defined by:

$$\mathrm{Regret}(\mathcal{A}, \mathcal{H}, \Xi, T) = \sum_{t=1}^{T} \ell(h_t, (\mathbf{x}_t, y_t)) - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell(h, (\mathbf{x}_t, y_t)) \tag{2.4}$$

where $\ell(h, (\mathbf{x}, y))$ is the loss function to measure the discrepancy between the prediction $h(\mathbf{x})$ and the corresponding observed element, e.g., 0-1 loss function $\mathbb{I}[h(\mathbf{x}) \neq y]$ for binary classification.

Regret measures the difference between the cumulative loss of the learner's strategy and the minimum possible loss had the sequence of loss functions been known in advance and the learner could choose the best fixed action in hindsight. In particular, we are interested in rates of increase of $\mathrm{Regret}(\mathcal{A}, \mathcal{H}, \Xi, T)$ in terms of T. When this is sub-linear in the number of rounds, that is, $o(T)$, we call the solution Hannan consistent [38], implying that the learner's average per-round loss approaches the average per-round loss of the best fixed action in hindsight. It is noticeable that the performance bound must hold for any sequence of loss functions, and in particular if the sequence is chosen adversarially.

Online learnability. In the online setting, the analogous of PAC learnability was addressed by Littlestone [98] who described a combinatorial characterization of hypothesis classes that are learnable in mistake bound model under realizability assumption. The extension of these results to agnostic online setting was addressed in [20]. Recall that, in the PAC model, VC-dimension of hypothesis class $\mathcal{H}$ characterizes learnability of $\mathcal{H}$ if we ignore computational considerations. Moroever, VC-dimension characterizes learnability in the agnostic PAC model as well. In online setting what is known as Littlestone's dimension plays the same rule. Recently, notion of Sequential Rademacher complexity has been introduced to characterizing online learnability which plays the similar role as the Rademacher complexity in statistical learning theory [126].

2.2.2 Online Convex Optimization and Regret Bounds

The online convex optimization (OCO) framework generalizes many known online learning problems in the realm of sequential prediction and repeated game playing. Among these are online classification and sequential portfolio optimization. The unified setting of OCO was introduced in [63] and the exact term was used before in [158]. Since the introduction of OCO, there have been a dizzying number of extensions and variants that is the focus of this section.

Assume we are given a fixed convex set $\mathcal{W}$ and some set of convex functions $\mathcal{F}$ on $\mathcal{W}$. In OCO, a decision maker is iteratively required to choose a decision $\mathbf{w}_t \in \mathcal{W}$. After making the decision $\mathbf{w}_t$ at round t, a convex loss function $f_t \in \mathcal{F}$ is chosen by adversary and the decision maker incurs loss $f_t(\mathbf{w}_t)$. The loss function is chosen completely arbitrarily and even in an adversarial manner given the current and past decisions of the decision maker. Online linear optimization is a special case of OCO in which the set $\mathcal{F}$ is the set of linear functions, i.e., $\mathcal{F} = \{\mathbf{w} \mapsto \langle \mathbf{f}, \mathbf{w} \rangle : \mathbf{f} \in \mathbb{R}^d\}$. The goal of online convex optimization is to come up with a sequence of solutions $\mathbf{w}_1, \ldots, \mathbf{w}_T$ that minimizes the *regret*, which is defined as the difference in the cost of the sequence of decisions accumulated up to the trial T made by the learner and the cost of the best fixed decision in hindsight, i.e.

$$\text{Regret}(\mathcal{A}, \mathcal{W}, \mathcal{F}, T) = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}).$$

Based on the type of feedback revealed to learner by adversary at the end of each iteration, we distinguish two types of OCO problem. In the *full information* OCO, after suffering the loss, the decision maker gets full knowledge of the function $f_t(\cdot)$. In the *partial information* setting (also bandit OCO), the decision maker only learns the value $f_t(\mathbf{w}_t)$ and does not gain any other knowledge about $f_t(\cdot)$.

We also distinguish between *oblivious* and *adapive* adversaries. In the oblivious or non-adaptive model, adversary is assumed to know our algorithm, and can pick the worst possible sequence of cost functions for it. However this sequence must be fixed in advance before game starts; during the game, adversary receives no feedback about our chosen decisions. In the more powerful adaptive model, the adversary is assumed to know not only our algorithm, but also the history of the game up to the current round. In other words, at the end of each round t, our decision $\mathbf{w}_t$ is revealed to adversary, and the next cost function $f_t(\cdot)$ may depend arbitrarily on $\mathbf{w}_1, \cdots, \mathbf{w}_t$.

The design of algorithms for regret minimization in OCO setting recently has been influenced by tools from the convex optimization. It has long been known that special kinds of loss functions permit tighter regret bounds than other loss functions. Two most important family of loss functions that has been considered are convex Lipschtiz and strongly convex functions.

Before presenting the known results on regret bounds for different families of loss functions, we will first need the definition of strong convexity as:

Definition 2.7 (Strong Convexity). *A loss function $f : \mathcal{W} \mapsto \mathbb{R}$ is said to be α-strongly convex w.r.t a norm $\|\cdot\|$, if there exists a constant $\alpha > 0$ (often called the modulus of strong convexity) such that, for any $\lambda \in [0, 1]$ and for all $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$, it holds that*

$$f(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') \leq \alpha f(\mathbf{w}) + (1-\lambda) f(\mathbf{w}') - \frac{1}{2}\lambda(1-\lambda)\alpha\|\mathbf{w} - \mathbf{w}'\|^2.$$

If $f(\mathbf{w})$ is twice differentiable, then an equivalent definition of strong convexity is $\nabla^2 f(\mathbf{w}) \succeq \alpha\mathbf{I}$ which indicates that the smallest eigenvalue of the Hessian of $f(\mathbf{w})$ is uniformly lower bounded by α everywhere. When $f(\mathbf{w})$ is differentiable, the strong convexity is equivalent to

$$f(\mathbf{w}) \geq f(\mathbf{w}') + \langle \nabla f(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle + \frac{\alpha}{2}\|\mathbf{w} - \mathbf{w}'\|^2, \; \forall\, \mathbf{w}, \mathbf{w}' \in \mathcal{W}.$$

Henceforth, we shall review few algorithms for OCO and state their regret bounds under different assumptions on the curvature of the sequence of the adversarial loss functions.

2.2.2.1 Online Gradient Descent

We start with the first, perhaps the simplest online convex optimization algorithm, which nevertheless captures the spirit of the idea behind the most of existing methods. This algorithm applies to the most general setting of online convex optimization and is referred to as Online Gradient Descent (OGD). The OGD method is rooted in the standard gradient descent algorithm and was introduced to the online setting by Zinkevich [158].

The OGD method starts with an arbitrary decision $\mathbf{w}_1 \in \mathcal{W}$, and iteratively modifies it according to the cost functions that are encountered so far as follows:

```
Online Gradient Descent (OGD)

Input: convex set $\mathcal{W}$, step size $\eta > 0$
Initialize: $\mathbf{w}_1 \in \mathcal{W}$
for $t = 1, 2, \ldots, T$
    Play $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\big(\mathbf{w}_t - \eta \nabla f_t(\mathbf{w}_t)\big)$
    Receive loss function $f_{t+1}(\cdot)$ and incur $f_{t+1}(\mathbf{w}_{t+1})$
end for
```

Here $\Pi_{\mathcal{W}}(\cdot)$ denotes the orthogonal projection onto the convex set $\mathcal{W}$.

The OGD algorithm is straightforward to implement, and updates take time $O(d)$ given the gradient. However, the projection step might be computationally cumbersome for complex domains $\mathcal{W}$. The following theorem states that the regret bound for OGD method when applied to convex Lipschitz functions is on the order of $O(\sqrt{T})$ which has been proven to be tight up to constant factors [1].

Theorem 2.8. *Let $f_1, f_2, \ldots, f_T$ be an arbitrary sequence of convex, differentiable functions defined over the convex set $\mathcal{W} \subseteq \mathbb{R}^d$. Let $G = \max_{t\in[T]} ||\nabla f_t(\mathbf{w})||$ and $R = \max_{\mathbf{w},\mathbf{w}'\in\mathcal{W}} ||\mathbf{w}-\mathbf{w}'||$. Then OGD with step size $\eta_t = \frac{R}{G\sqrt{t}}$ achieves the following guarantee for any $\mathbf{w} \in \mathcal{W}$, for all $T \geq 1$.*

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}) \leq \frac{\|\mathbf{w} - \mathbf{w}_1\|^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T} \|\nabla f_t(\mathbf{w}_t)\|^2 = O(RG\sqrt{T}) \tag{2.5}$$

In fact it is possible to show that OGD can attain a logarithmic regret $O(\log T)$ for strongly convex functions by appropriately tuning the step sizes as stated below.

Theorem 2.9. *Let $f_1, f_2, \ldots, f_T : \mathcal{W} \to \mathbb{R}$ be an arbitrary sequence of α-strongly convex functions. Under the same conditions as Theorem 2.8 with step size $\eta_t = \frac{1}{\alpha t}$, the OGD algorithm achieves the following regret:*

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}) \leq G^2 \sum_{t=1}^{T} \frac{1}{\alpha t} \leq \frac{G^2}{2\alpha}(1 + \log T) = O(\log T) \tag{2.6}$$

Remark 2.10. *In the above algorithm, the updating rule for OGD uses the gradients of the loss functions $\nabla f_t(\mathbf{w})$ at each iteration to update the solution. In fact it is not required to assume that the loss functions are differentiable and it suffices to assume that the loss functions only have a sub-gradient everywhere in the domain $\mathcal{W}$, making the algorithm suitable for non-smooth settings. In particular for any $\mathbf{g}_t \in \partial f_t(\mathbf{w}_t)$ where $\partial f_t(\mathbf{w}_t)$ is the set of sub-gradient at point $\mathbf{w}_t$, the algorithm is able to achieve the same regret bounds in both cases.*

2.2.2.2 Follow The Perturbed Leader

The first efficient algorithm for the general online *linear* optimization problems is due to Hannan [65], and was subsequently rediscovered and clarified in [86]. The Follow The Perturbed Leader (FTPL) algorithm assumes that there is an oracle that can efficiently solve the offline optimization problem. Having access to such an oracle, the FTPL selects the decision that appears to be the best so far, but for a version of actual cost vectors which have been perturbed by the addition of some noise. The addition of random noise to the observed cost functions has the effect of slowing down the algorithm so that instead of tracking small fluctuations in cost functions, it has the tendency to stick to the same decision unless there is a compelling reason to switch to another decision.

Although the FTPL algorithm works in a similar setting as OGD, but there is a crucial difference between them which makes FTPL more suitable for online combinatorial learning. In particular, the decision set $\mathcal{W}$ does not need to be convex as long as the offline optimization problem can be solved efficiently. This is a significant advantage of the FTPL approach, which can be utilized to tackle more general problems with discrete decision spaces. The FTPL method for online decision making relies on a linear optimization procedure $\mathcal{M}$ over the set $\mathcal{W}$ that computes $\mathcal{M}(\mathbf{f}) = \arg\min_{\mathbf{w}\in\mathcal{W}}\langle \mathbf{f}, \mathbf{w}\rangle$ for all $\mathbf{f} \in \mathbb{R}^d$. Then, FTPL chooses $\mathbf{w}_t$ by first drawing a perturbation $\boldsymbol{\mu}_t \in [0, \frac{1}{\eta}]^d$ uniformly at random, and computing $\mathbf{w}_{t+1} = \mathcal{M}\left(\sum_{\tau=1}^{t} \mathbf{f}_\tau + \boldsymbol{\mu}_t\right)$.

```
Follow The Perturbed Leader (FTPL)

Input: a general domain $\mathcal{W}$, step size $\eta > 0$, offline linear oracle $\mathcal{M}$ over $\mathcal{W}$
Initialize: $\mathbf{w}_1 \in \arg\min_{\mathbf{w}\in\mathcal{W}} \mathcal{M}(\mathbf{0})$
for $t = 1, 2, \ldots, T$
    Draw $\boldsymbol{\mu}_t \in [0, \frac{1}{\eta}]^d$ uniformly at random
    Play $\mathbf{w}_{t+1} = \mathcal{M}\left(\sum_{\tau=1}^{t} \mathbf{f}_\tau + \boldsymbol{\mu}_t\right)$
    Receive loss vector $\mathbf{f}_{t+1} \in \mathbb{R}^d$ and incur $\langle \mathbf{f}_{t+1}, \mathbf{w}_{t+1}\rangle$
end for
```

The regret of FTPL algorithm is stated in the following theorem.

Theorem 2.11 (Regret Bound for FTPL). *Let $\mathbf{f}_1, \ldots, \mathbf{f}_T$ be an arbitrary sequence of linear loss functions from unit ball and let $\mathbf{w}_1, \ldots, \mathbf{w}_T$ be the sequence of decisions generated by the FTPL constrained to a general set $\mathcal{W}$. Then for any $\mathbf{w} \in \mathcal{W}$, the FTPL algorithm with*

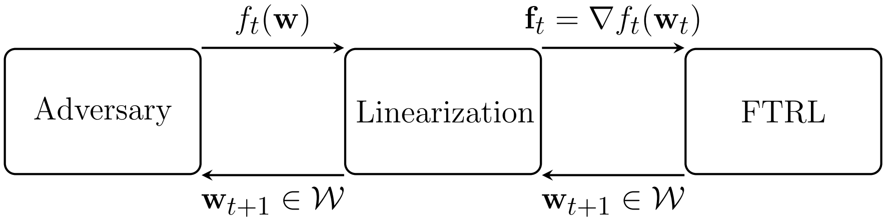

$f_t(\mathbf{w})$
$\mathbf{f}_t = \nabla f_t(\mathbf{w}_t)$
Adversary
Linearization
FTRL
$\mathbf{w}_{t+1} \in \mathcal{W}$
$\mathbf{w}_{t+1} \in \mathcal{W}$

Figure 2.2: Reduction of general online convex optimization problem to online optimization with linear functions.

parameter $\eta = \sqrt{R/GT}$ *satisfies:*

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \mathbf{f}_t, \mathbf{w}_t \rangle\right] - \sum_{t=1}^{T} \langle \mathbf{f}_t, \mathbf{w} \rangle \;\leq\; 2\sqrt{RGT}\ ,$$

where $\max_{\mathbf{w}} |\langle \mathbf{f}_t, \mathbf{w} \rangle| \leq G, t \in [T]$ *is an upper bound on the magnitude of the rewards, and* R *is an upper bound on the* ℓ_1 *diameter of* $\mathcal{W}$*, i.e.,* $R \geq \max_{\mathbf{w},\mathbf{w}' \in \mathcal{W}} \|\mathbf{w} - \mathbf{w}'\|_1$.

2.2.2.3 Follow The Regularized Leader

A natural modification of the basic FTPL algorithm or fictitious play in game theory is the Follow The Regularized Leader (FTRL) algorithm, in which we minimize the loss on all past rounds plus a regularization term. Regularization is an alternative to perturbation to stabilize decisions during the prediction. Naturally, different regularization functions will yield different algorithms for different applications.

In FTRL we assume that the loss functions are linear $f_t(\mathbf{w}) = \langle \mathbf{f}_t, \mathbf{w} \rangle, \mathbf{f}_t \in \mathbb{R}^d$ and the generalization to convex loss functions can be accomplished by *linearization* strategy. This reductions is shown in Figure 2.2. The main idea behind linearization trick is that if an algorithm $\mathcal{A}$ is able to achieve good regret against linear loss functions, then $\mathcal{A}$ can be used to achieve good regret against sequences of convex loss functions as well. To see this, we note that from the definition of convexity, i.e, $f_t(\mathbf{w}) \geq f_t(\mathbf{w}_t) + \langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle$, we can feed the learner $\mathcal{A}$ with linear loss functions $f'_t(\mathbf{w}) = \langle \nabla f_t(\mathbf{w}_t), \mathbf{w} \rangle$ and then guarantee that $f_t(\mathbf{w}_t) - f_t(\mathbf{w}) \leq \langle \mathbf{f}_t, \mathbf{w}_t - \mathbf{w} \rangle$. Therefore, any bound on the regret of linear loss functions directly translate to a regret bound for convex losses.

The FTRL algorithm at each round t solves an offline optimization problem based based on the sum of loss functions up to time $t-1$ and a regularization function. Let $\mathcal{R}(\mathbf{w})$ be a strongly convex differentiable function. The detailed steps of FTRL are as follows [3]:

```
Follow the Regularized Leader (FTRL)

Input: convex set $\mathcal{W}$, step size $\eta > 0$, regularization $\mathcal{R}(\mathbf{w})$
Initialize: $\mathbf{w}_1 \in \arg\min_{\mathbf{w} \in \mathcal{W}} \mathcal{R}(\mathbf{w})$
for $t = 1, 2, \ldots, T$
  Play $\mathbf{w}_{t+1} = \arg\min_{\mathbf{w} \in \mathcal{W}} \left[ \sum_{s=1}^{t} \langle \mathbf{f}_s, \mathbf{w} \rangle + \mathcal{R}(\mathbf{w}) \right]$
  Receive loss function $f_{t+1}(\cdot)$ and incur $f_{t+1}(\mathbf{w}_{t+1})$
  Set $\mathbf{f}_{t+1} = \nabla f_{t+1}(\mathbf{w}_{t+1})$
end for
```

It is noticeable that what FTRL implements at each iteration is the regularized empirical risk minimization over the previous trials as we saw in the statistical learning. In online setting, the regularizer has the role of forcing the consecutive solutions is to stay closer to each other, similar role role the perturbation plays in FTPL algorithm. Furthermore, different choices of the regularizer lead to different algorithms. For instance, in the simplest case if we let $\mathcal{R}(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2$, then the FTRL behaves same as OGD algorithm.

It is not hard to prove a simple bound on the regret of the FTRL algorithm for a given strongly convex regularization function $\mathcal{R}(\mathbf{w})$ and the learning rate η.

Theorem 2.12 (Regret Bound for FTRL)**.** *Let* $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_T \in \mathbb{R}^d$ *be an arbitrary sequence of linear loss functions over convex set* $\mathcal{W} \subseteq \mathbb{R}^d$*. Let* $\mathcal{R} : \mathcal{K} \mapsto \mathbb{R}$ *be a 1-strongly convex function with respect to a norm* $\|\cdot\|$*. Let* $R = \max_{\mathbf{w} \in \mathcal{W}} \mathcal{R}(\mathbf{w}) - \mathcal{R}(\mathbf{w}_1)$ *and* $\max_{t \in [T]} \|\mathbf{f}_t\|_* \leq G$*. Then for any* $\mathbf{w} \in \mathcal{W}$ *by setting* $\eta = \sqrt{\frac{R}{G^2 T}}$ *the regret of FTRL is bounded by:*

$$\sum_{t=1}^{T} \langle \mathbf{f}_t, \mathbf{w}_t \rangle - \sum_{t=1}^{T} \langle \mathbf{f}_t, \mathbf{w} \rangle \leq \frac{\mathcal{R}(\mathbf{w}) - \mathcal{R}(\mathbf{w}_1)}{\eta} + \eta \sum_{t=1}^{T} \|\mathbf{f}_t\|_*^2 = O(G\sqrt{RT}) \tag{2.7}$$

2.2.2.4 Online Mirror Descent

Another algorithm for OCO problem is the online version of celebrated proximal point algorithm in offline convex optimization. As mentioned before, the implicit goal of regularization used in the FTRL algorithm is to control by how much the consecutive solutions differ from each other. The proximal point algorithm is designed with the explicit goal of keeping $\mathbf{w}_{t+1}$ as close as possible to $\mathbf{w}_t$. The closeness of two solutions are measured by the Bregman divergence induced by a strongly convex Legendre function $\Phi(\cdot)$ defined over convex domain $\mathcal{K}$. A Legendre function is a strictly convex functions with continuous partial derivatives and gradient blowing up at the boundary of its domain (see Appendix **??** for detailed discussion). We assume that $\mathcal{W} \cap \mathcal{K} \neq \emptyset$. Online proximal point method solves an optimization problem which expresses the tradeoff between the distance from the old solution and the loss suffered by by the current convex function as:

$$\mathbf{w}_{t+1} = \arg \min_{\mathbf{w} \in \mathcal{W} \cap \mathcal{K}} \left[\eta_t f_t(\mathbf{w}) + \mathsf{B}_\Phi(\mathbf{w}, \mathbf{w}_t) \right], \tag{2.8}$$

where $\mathsf{B}_\Phi(\mathbf{w}, \mathbf{w}_t)$ is the Bregman divergence induced by $\Phi(\cdot)$ (Definition A.17). When the loss functions are linear, i.e., $f_t(\mathbf{w}) = \langle \mathbf{f}_t, \mathbf{w} \rangle$ for some $\mathbf{f}_t \in \mathbb{R}^d$, or one replaces the objective $f_t(\mathbf{w})$ in (2.8) with its linearized term, i.e., $f_t(\mathbf{w}) \approx f_t(\mathbf{w}_t) + \langle \mathbf{w} - \mathbf{w}_t, \nabla f_t(\mathbf{w}_t) \rangle$, the proximal point method becomes the Online Mirror Descent (OMD) algorithm as detailed below:

```
Online Mirror Descent (OMD)

Input: convex sets $\mathcal{W}$, step size $\eta > 0$, Legendre function $\Phi : \mathcal{K} \mapsto \mathbb{R}$
Initialize: $\mathbf{w}_1 \in \arg\min_{\mathbf{w} \in \mathcal{K}} \Phi(\mathbf{w})$
for $t = 1, 2, \ldots, T$
    Play $\mathbf{w}_{t+1} = \arg\min_{\mathbf{w} \in \mathcal{W} \cap \mathcal{K}} \left[ \eta_t \langle \mathbf{w} - \mathbf{w}_t, \nabla f_t(\mathbf{w}_t) \rangle + \mathsf{B}_\Phi(\mathbf{w}, \mathbf{w}_t) \right]$
    Receive loss function $f_{t+1}(\cdot)$ and incur $f_{t+1}(\mathbf{w}_{t+1})$
end for
```

Theorem 2.13 (Regret Bound for OMD)**.** *Let $f_1, f_2, \ldots, f_T$ be an arbitrary sequence of convex, differentiable functions defined over the convex set $\mathcal{W} \subseteq \mathbb{R}^d$. Let $G = \max_t ||\nabla f_t(\mathbf{w}_t)||_*$ and $R = \max_{\mathbf{w}, \mathbf{w}' \in \mathcal{W}} ||\mathbf{w} - \mathbf{w}'||$. Let $\Phi : \mathcal{K} \mapsto \mathbb{R}$ be a Legendre function which is 1-strongly convex w.r.t the norm $\| \cdot \|$ and $\mathcal{W} \cap \mathcal{K} \neq \emptyset$. Then the regret of the OMD can be bounded by*

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}) \leq \frac{\mathsf{B}_\Phi(\mathbf{w}, \mathbf{w}_1)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla f_t(\mathbf{w}_t)\|_*^2 = O(RG\sqrt{T}), \tag{2.9}$$

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

We note that many classical online learning algorithms can be viewed as variants of OMD, generally either with the Euclidean geometry such as Perceptron algorithm and OGD, or in the simplex geometry, using an entropic distance generating function such as Winnow [98] and Online Exponentiated Gradient algorithm [90].

2.2.2.5 Online Newton Step

As mentioned in the analysis of OGD algorithm, it attains a logarithmic regret $O(\log T)$ if the sequence of loss functions which have bounded gradient and are strongly convex. Another case in which we can obtain logarithmic regret is the case of exp-concave loss functions (i.e., the function $\exp(-\alpha f(\mathbf{w}))$ is concave for some α). Exp-concavity is weaker condition than the bounded gradient and strong convexity. Online Newton Step (ONS) [68] is the adaption of Newton method for convex optimization to online setting and is able to achieve logarithmic regret when learned on exp-concave functions which makes is more general than OGD.

```
Online Newton Step (ONS)

Input: convex sets $\mathcal{W}$, step size $\eta > 0$
Initialize: $\mathbf{w}_1 \in \arg\min_{\mathbf{w}\in\mathcal{K}} \Phi(\mathbf{w})$
for $t = 1, 2, \ldots, T$
    Play $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}^{\mathbf{A}_t}\left(\mathbf{w}_t - \eta_t \mathbf{A}_t^{-1}[\nabla f_t(\mathbf{w}_t)]\right)$ where $\mathbf{A}_{t-1} = \sum_{s=1}^{t-1} \nabla f_s(\mathbf{w}_s)\nabla f_s(\mathbf{w}_s)^\top$
    Receive loss function $f_{t+1}(\cdot)$ and incur $f_{t+1}(\mathbf{w}_{t+1})$
end for
```

Here $\Pi_{\mathcal{W}}^{\mathbf{A}}(\cdot)$ is the projection induced by the a matrix $\mathbf{A}$, i.e., $\Pi_{\mathcal{W}}^{\mathbf{A}}(\mathbf{w}) = \min_{\mathbf{z}\in\mathcal{W}}(\mathbf{z} - \mathbf{w})^\top\mathbf{A}(\mathbf{z} - \mathbf{w})$. We note that compared to the provisos algorithms which only exploit first-order information about the loss functions, the *analysis* of ONS method is based on second order information, i.e. the second derivatives of the loss functions, whereas the implementation of ONS relies only on first-order information.

The following result shows that the ONS method achieves a logarithmic regret for exp-concave functions which is stronger result than the performance of OGD for strongly convex losses.

Theorem 2.14 (Regret Bound for ONS)**.** *Let* $f_1, f_2, \ldots, f_T$ *be an arbitrary sequence of* α*-exp-concave functions defined over the convex set* $\mathcal{W} \subseteq \mathbb{R}^d$ *with* $G = \max_t ||\nabla f_t(\mathbf{w}_t)||$*. Let* $R = \max_{\mathbf{w},\mathbf{w}' \in \mathcal{W}} ||\mathbf{w} - \mathbf{w}'||$*. Then the ONS algorithm achieves the following regret bound for any* $\mathbf{w} \in \mathcal{W}$*:*

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}) \leq \Big(\frac{1}{\alpha} + GR\Big) d \log T = O(d \log T) \tag{2.10}$$

We note that also this result seems interesting, but for some of the functions of interest such as logistic loss the exp-cancavity parameter α could be exponentially large in d [113]. The exponential dependence on the diameter of the feasible set can make this bound worse than the $O(\sqrt{T})$ bound obtained by OGD.

2.2.3 Variational Regret Bounds

Most previous works, including those discussed above, considered the most general setting in which the loss functions could be arbitrary and possibly chosen in an adversarial way. However, the environments around us may not always be adversarial, and the loss functions may have some patterns which can be considered to achieve a smaller regret. Consequently, it is objected that requiring an algorithm to have a small regret for all sequences leads to results that are too loose to be practically interesting. As a result, the bounds obtained for worst case scenarios become pessimistic for these regular sequences. Therefore, it would be desirable to develop algorithms that yield tighter bounds for more regular sequences, while still providing protection against worst case sequences. To this end, we need to replace the number of rounds appeared in the regret bound with some other notion of performance. In particular, this new measure should depend on variation in the sequence of costs functions emitted to the learner. Having such an algorithm guarantees that if the cost sequence has low variation, the algorithm would be able to perform better.

One work along this direction is that of [69]. For the online linear optimization problem, in which each loss function is linear $f_t(\mathbf{w}) = \langle \mathbf{f}_t, \mathbf{w} \rangle$ and can be seen as a vector, they considered the case in which the loss functions have a small variation, defined as

$$\text{Variation}(\mathcal{A}, \mathcal{W}, \mathcal{F}, T) = \sum_{t=1}^{T} \|\mathbf{f}_t - \boldsymbol{\mu}\|_2^2,$$

where $\boldsymbol{\mu} = \sum_{t=1}^{T} \mathbf{f}_t / T$ is the average of the loss functions. For this, they showed that a regret of $O(\sqrt{\text{Variation}})$ can be achieved, and they also have an analogous result for the prediction with expert advice problem. According to this definition, a small Variation$(\mathcal{A}, \mathcal{W}, \mathcal{F}, T)$ means that most of the loss functions center around some fixed loss function $\boldsymbol{\mu}$. This seems to model a stationary environment, in which all of the loss functions are produced according to some fixed distribution.

The variation bound is defined in terms of total difference between individual linear cost vectors to their mean. In Chapter 5 of this thesis, we introduce another measure which is called *gradual variation*. Gradual variation is more general and applies to environments which may be evolving but is a somewhat gradual way. For example, the weather condition or the stock price at one moment may have some correlation with the next and their difference is usually small, while abrupt changes only occur sporadically. Formally, the gradual variation of a sequence of loss functions is defined as:

$$\text{GradualVarition}(\mathcal{A}, \mathcal{W}, \mathcal{F}, T) = \sum_{t=1}^{T-1} \max_{\mathbf{w} \in \mathcal{W}} \|\nabla f_{t+1}(\mathbf{w}) - \nabla f_t(\mathbf{w})\|_2^2.$$

It is easy to verify that the gradual variation lower bounds the variation bound and hence algorithms with regret bounded by gradual variation are more adaptive to regular patterns than algorithm with bounded variational regret bounds.

2.2.4 Bandit Online Convex Optimization

In bandit OCO, once the online learner commits to the decision $\mathbf{w}_t$ at round t, he does not have access to the function $f_t(\cdot)$ chosen by adversary and instead receives the scalar loss $f_t(\mathbf{w}_t)$ he suffers at point $\mathbf{w}_t$. In the optimization community, this problem usually known as zeroth-order or derivative-free convex optimization as we only have access to function values to solve the optimization problem [79, 136]. A simple approach for bandit OCO which was the main dilemma in most existing works is to utilize a reduction to the full information OCO setting. To do so, one needs to approximate the gradient of the loss functions at each iteration based on the observed scalar loss and feed it to the full information algorithm. This reduction has been illustrated in Figure 2.3. A simple idea to estimate the gradients has been utilized in [54] and a modified gradient descent approach for bandit OCO has been presented that attains $O(T^{3/4})$ regret bound. The key idea of their algorithm is to compute the stochastic approximation of the gradient of cost functions by single point evaluation of the cost functions. The main observation was that one can estimate the gradient of a function $f(\mathbf{w})$ by taking a random vector $\mathbf{u}$ from unit sphere $\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^d; \|\mathbf{x}\| = 1\}$ and

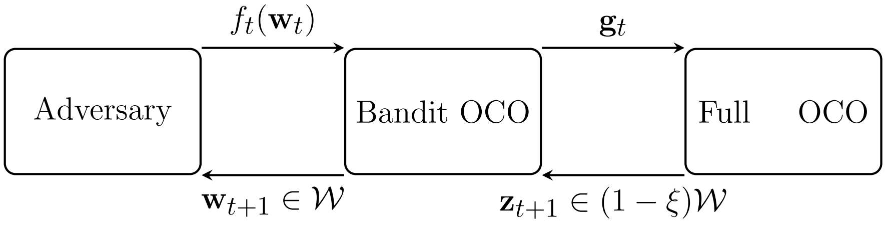

$f_t(\mathbf{w}_t)$
$\mathbf{g}_t$
Adversary
Bandit OCO
Full OCO
$\mathbf{w}_{t+1} \in \mathcal{W}$
$\mathbf{z}_{t+1} \in (1-\xi)\mathcal{W}$

Figure 2.3: Reduction of bandit online convex optimization to online convex optimization with full information. The full OCO needs to play from a shrinked domain $(1-\xi)\mathcal{W}$ to ensure that the sampled points belong to the domain.

scaling it by $f(\mathbf{w}+\delta\mathbf{u})$, i.e. $\hat{\mathbf{g}} = f(\mathbf{w}+\delta\mathbf{u})\mathbf{u}$. Then $\mathbb{E}[\hat{\mathbf{g}}]$ is proportional to the gradient of a *smoothed* version of $f(\cdot)$ defined as $\hat{f}(\mathbf{w}) = \mathbb{E}_{\mathbf{v}\in\mathbb{B}}[f(\mathbf{w}+\delta\mathbf{v})]$ for $\mathbf{v}$ random from the unit ball $\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^d; \|\mathbf{x}\| \leq 1\}$. To ensure that the sampled query points belong the domain $\mathcal{W}$, the OGD is run over a shrinked domain $(1-\xi)\mathcal{W}$ where we further assume that $r\mathbb{B} \subseteq \mathcal{W} \subseteq R\mathbb{B}$.

Expected Online Gradient Descent (EOGD)

```
Input: convex set $\mathcal{W}$, step size $\eta > 0$, $\delta$, $r$, and $\xi$
Initialize: $\mathbf{z}_0 = \mathbf{0}$
for $t = 1, 2, \ldots, T$
  Pick a random unit vector $\mathbf{u}_t$ uniformly at random
  Play $\mathbf{w}_t = \mathbf{z}_t + \delta\mathbf{u}_t$ and observe $f_t(\mathbf{w}_t)$
  Update $\mathbf{z}_{t+1} = \Pi_{(1-\xi)\mathcal{W}}\Big(\mathbf{z}_t - \eta f_t(\mathbf{w}_t)\mathbf{u}_t\Big)$
end for
```

Theorem 2.15 (Regret Bound for EOGD)**.** *Let $f_1, f_2, \ldots, f_T$ be a sequence of convex, differentiable functions defined over the convex domain $\mathcal{W} \subseteq \mathbb{R}^d$ where $r\mathbb{B} \subseteq \mathcal{W} \subseteq R\mathbb{B}$. Let $\mathbf{g}_1, \ldots, \mathbf{g}_n$ are vector-valued random variables with $\mathbb{E}[\mathbf{g}_t|\mathbf{w}_t] = \mathbb{E}[\nabla f_t(\mathbf{w}_t)]$ and $\|\mathbf{g}_t\| \leq G$,* *for some $G > 0$. Then, for $\eta = \frac{R}{G\sqrt{n}}$, $\xi = \delta/r$, and $\delta = T^{-1/4}\sqrt{\frac{RdGr}{3(rG+C)}}$*

$$\mathbf{E}\left[\sum_{t=1}^{T} f_t(\mathbf{w}_t)\right] - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq O(T^{3/4})$$

This regret bound is later improved to $O(T^{2/3})$ [9] for online bandit linear optimization. More recently, [46] proposed an inefficient algorithm for online bandit linear optimization with the optimal regret bound $O(poly(d)\sqrt{T})$ based on multi-armed bandit algorithm. The key disadvantage of [46] is that it is not computationally efficient. Abernethy, Hazan, and Rakhlin [2] presented an efficient randomized algorithm with an optimal regret bound $O(poly(d)\sqrt{T})$ that exploits the properties of self-concordant barrier regularization.

For general online bandit convex optimization, [5] proposed optimal algorithms in a multi-point bandit setting, in which multiple points can be queried for the cost values. With multiple queries, they showed that the EOGD algorithm can give an $O(\sqrt{T})$ expected regret bound. The key idea of multiple point bandit online convex optimization, proposed in [5], is to approximate the gradient using two point function evaluations. More specifically, at each iteration t we randomly choose a unit direction $\mathbf{u}_t$ and measure the function values at points $\mathbf{w}_t + \delta\mathbf{u}_t$ and $\mathbf{w} - \delta\mathbf{u}_t$, i.e. $\ell_t(\mathbf{w}_t + \delta\mathbf{u}_t)$ and $\ell_t(\mathbf{w}_t - \delta\mathbf{u}_t)$, where $\delta > 0$ is a small perturbation that is $O(1/T)$. Given two point function evaluation, we approximate the gradient $\nabla f_t(\mathbf{w}_t)$ by $\mathbf{g}_t = \frac{d}{2\delta}\left(f_t(\mathbf{w}_t + \delta\mathbf{u}_t) - f_t(\mathbf{w}_t - \delta\mathbf{u}_t)\right)\mathbf{u}_t$. The nice property of this sampling strategy is that the norm of sampled gradient is no longer dependent on δ, i.e., $\|\mathbf{g}_t\| \leq dG$, and yield the same regret bound as OGD.

2.2.5 From Regret to Risk Bounds

So far we have dealt with two different models for learning: statistical and sequential or online. The online setting is in stark contrast to the statistical setting in few aspects. First, in statistical learning or the batch model there is a strict division between the training phase and the testing phase. In contrast, in the online model, training and testing occur together all at the same time since every example acts both as a test of what was learned in the past and as a training example for improving our predictions in the future. This requires that the online learner to be adaptive to the environment.

A second key difference relates to the generation of examples. Statistical learning scenario follows the key assumption that the distribution over data points is fixed over time, both for training and test points, and samples are assumed to be drawn i.i.d. from an underlying distribution $\mathcal{D}$. Furthermore, the goal is to learn a hypothesis with a small expected loss or generalization error. In contrast, in online setting there is no notion of generalization and algorithms are measured using a mistake bound model and the notion of regret, which are based on worst-case or adversarial assumption where the adversary deliberately trying to ruin the learners performance.

Finally, we distinguish between the processing model of statistical and online learning settings. Online algorithms process one sample at a time and can thus be significantly more efficient both in time and space and more practical than batch algorithms, when processing modern data sets of several million or billion points. hence, these algorithms are more suitable for large scale learning. This stands in contrast to statistical learning algorithms such as ERM and it would be tempting to switch to online learning algorithm.

Given the close relationship between these two settings and clear advantage of online learning from computational viewpoint, a paramount question is "*whether or not algorithms developed in the sequential setting can be used for statistical learning with guaranteed generalization bound?*". More precisely, can we devise algorithms that exhibits the desirable characteristics of online learning but also has good generalization properties. Since a regret bound holds for all sequences of training samples, it also holds for an i.i.d. sequence. What remains to be done is to extract a single hypothesis out of the sequence produced by the sequential method, and to convert the regret guarantee into a guarantee about generalization. Such a process has been dubbed an `Online-to-Batch Conversion` and foreshadows a key achievement, which is any online learning algorithm with sub linear regret can be converted into a batch algorithm. Here we introduce two methods to convert an online algorithm that attains low regret into a batch learning algorithm that attains low risk. Such online-to-batch conversions are interesting both from the practical and the theoretical perspectives [36, 48, 83].

Formally, let $f_1, f_2, \cdots, f_T$ be an i.i.d sequence of loss functions, $f_t : \mathcal{W} \mapsto \mathbb{R}$. In statistical setting one can think of each loss function $f_t(\mathbf{w})$ as $f_t(\mathbf{w}) = \ell(\mathbf{w}, (\mathbf{x}_t, y_t))$ for a fixed loss function $\ell : \mathcal{W} \times \Xi \mapsto \mathbb{R}_+$ and a random instance $(\mathbf{x}_t, y_t) \in \Xi$ sampled following the underlying distribution $\mathcal{D}$. We feed these loss functions to an online learning algorithm $\mathcal{A}$ and assume that the online learner produces a sequence $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_T \in \mathcal{W}$ of hypothesis. The goal is to construct a single hypothesis $\hat{\mathbf{w}} \in \mathcal{W}$ with small generalization error. Here we consider two solutions for this problem: *randomized conversion* and *averaging*.

The simplest conversion scheme is to simply choose a random hypothesis uniformly at random from the sequence of hypothesis $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_T$. At the first glance this idea seems naive, but it has few desirable properties. First, the average loss of the algorithm is an unbiased estimate of the expected risk of $\hat{\mathbf{w}}$, i.e., $\mathbf{E}[\ell(\hat{\mathbf{w}})] = (1/T) \sum_{t=1}^{T} f_t(\mathbf{w}_t)$. Second, the conversion is applicable regardless of any convexity assumption. Finally, in expectation, the excess loss of $\hat{\mathbf{w}}$ is upper bounded by the average per-round regret of online learner, i.e.,

$$\mathcal{L}_{\mathcal{D}}(\hat{\mathbf{w}}) - \min_{\mathbf{w} \in \mathcal{W}} \mathcal{L}_{\mathcal{D}}(\mathbf{w}) \leq \frac{\mathbf{E}\big[\text{Regret}(\mathcal{A}, \mathcal{W}, \mathcal{F}, T)\big]}{T}.$$

An alternative solution which is only applicable to learning from convex loss functions over convex hypothesis spaces, is to output the average solution $\hat{\mathbf{w}} = (1/T) \sum_{t=1}^{T} \mathbf{w}_t$. This conversion is also enjoys the same properties as the randomized conversion with an additional important feature. That is, we can able to show high-probability bounds on the excess risk provided that loss functions are bounded.

2.3 Convex Optimization

A generic convex optimization problem may be written as

$$\min f(\mathbf{w}) \quad \text{subject to} \quad \mathbf{w} \in \mathcal{W},$$

where $f : \mathbb{R}^d \mapsto \mathbb{R}$ chosen from a specific family of functions $\mathcal{F}$ is a proper convex function, and $\mathcal{W} \subseteq \mathbb{R}^d$ is nonempty, compact, and convex set which is also called the constraint or feasible set. We denote by $\mathbf{w}_*$ the optimal solution to above problem and assume that it exists, i.e., $\mathbf{w}_* = \arg\min_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w})$. Ideally, the goal of an optimization algorithm is to compute the optimal solution, but almost always it is impossible to compute an exact $\mathbf{w}_*$ in finite time. hence, we turn to find an ϵ-approximate solution. A solution $\mathbf{w} \in \mathcal{W}$ is an ϵ sun-optimal if $f(\mathbf{w}) - \min_{\mathbf{w}' \in \mathcal{W}} f(\mathbf{w}') \leq \epsilon$.

For a given family $\mathcal{F}$ of convex functions over the feasible set $\mathcal{W}$, our primary focus is to determine the efficiency of an optimization procedure to produce sub-optimal solutions. To analyze the efficiency of convex optimization algorithm one typically follows the oracle model of optimization which lies in the heart of the complexity theory of convex optimization [119, 121].

2.3.1 Oracle Complexity of Optimization

A typical convex optimization procedure initially picks some point in the feasible convex set $\mathcal{W}$ and iteratively updates these points based on some local information about the function it calculates around these successive points. The method can decide which points to query at based on the results of earlier queries, and tries to use as few queries as possible to achieve its task. The crucial question we are interested to answer about a specific optimization problem is the number of queries the algorithm makes to find an ϵ-accurate solution. The oracle complexity is a general model to analyze the computational complexity of optimization algorithms.

In the oracle model, there is an oracle $\mathcal{O}$ and an information set $\mathcal{I}$. The oracle $\mathcal{O}$ is simply a function $\psi : \mathcal{W} \mapsto \mathcal{I}$ that for any query point $\mathbf{w} \in \mathcal{W}$ returns an output from $\mathcal{I}$. The information set provided to the algorithms varies depending on the type of the oracle. In particular, a zero-order oracle returns $f(\mathbf{w})$ for a given query $\mathbf{w} \in \mathcal{W}$, first-order oracle returns gradient $\mathcal{I} = \{\nabla f(\mathbf{w})\}$ (respectively a sub-gradient $\mathcal{I} = \{\mathbf{g} \in \partial f(\mathbf{w})\}$ if the function is not differentiable), and a second-order oracle return the Hessian at the queried point. We also distinguish between noisy (or stochastic) and exact (or deterministic) oracle models. In the noisy oracle model, the information returned by the oracle are corrupted with zero-mean noise with bounded variance.

The algorithm iteratively updates the solution based on the information accumulated in

Oracle	Lipschitz	Lipschitz & Strongly	Smooth	Smooth & Strongly
Deterministic	$\frac{\rho}{\epsilon^2}$	$\frac{\rho^2}{\alpha\epsilon}$	$\frac{\beta}{\sqrt{\epsilon}}$	$\sqrt{\kappa}\log\frac{\alpha}{\epsilon}$
Stochastic	$\frac{\rho}{\epsilon^2}$	$\frac{\rho^2}{\alpha^2\epsilon}$	$\frac{\beta}{\epsilon}+\frac{\rho}{\epsilon^2}$	$\sqrt{\kappa}\log\left(\frac{\beta}{\epsilon}\right)+\frac{1}{\alpha\epsilon}$

Table 2.1: Lower bound on the oracle complexity for stochastic/deterministic first-order optimization methods. Here ρ, α, and β are the Lipschitzness, strong convexity, and smoothness parameters, respectively. The parameter κ is the condition number of function and is defined as $\kappa = \beta/\alpha$.

previous iterations. In particular, in optimization with zero and first-order exact oracle model which is the main focus of large scale optimization methods, an optimization method updates the solution using $\mathbf{w}_t = \phi_t(\mathbf{w}_0, \ldots, \mathbf{w}_{t-1}, \nabla f(\mathbf{w}_0), \ldots, \nabla f(\mathbf{w}_{t-1}), f(\mathbf{w}_0), \ldots, f(\mathbf{w}_{t-1}))$ where $\phi_t : \times \cup_{s=1}^{t} \mathcal{I}_s \mapsto \mathcal{W}$ is updating mechanism utilized by the optimization algorithm at iteration t to determine the next query point $\mathbf{w}_t$. Roughly speaking, we measure complexity of an algorithm by the number of queries that it makes to a prescribed oracle for computing the final solution.

Given a positive integer T corresponding to the number of iterations, the minimax oracle optimization error after T steps, over a set of functions $\mathcal{F}$, is defined as follows:

$$\mathrm{OracleComplexity}(\mathcal{F}, \mathcal{W}, \mathcal{O}, T) = \inf_{\psi} \sup_{f \in \mathcal{F}} \left(f(\mathbf{w}_T) - \inf_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}) \right).$$

In other words, the minimax oracle complexity is the best possible rate of convergence (as a function of the number of queries) for the optimization error when one restricts to black-box procedures in order to guarantee delivering an ϵ-accurate solution to any function $f \in \mathcal{F}$.

A large body of literature is devoted to obtaining rates of convergence of specific procedures for various set of convex functions $\mathcal{F}$ of interest (essentially smooth/non-smooth, and strongly convex/non-strongly convex) and different types of oracles (essentially noisy or stochastic/deterministic or exact, zero order or derivative free, first order, and second order). The oracle complexity of first-order deterministic and stochastic oracle models are summarized in Table 2.1 for different family of loss functions elicited from [121] for deterministic and from [4, 119] for stochastic optimization. The algorithms which attain these lower bounds will be discussed later.

2.3.2 Deterministic Convex Optimization

Here we briefly review the optimization algorithms in the *first-order* oracle model which are called gradient based methods for simplicity. More precisely, we assume that the only information the optimization methods can learn about the particular problem instance is the values and derivatives of these components $(f(\mathbf{w}), \nabla f(\mathbf{w}))$ at query points $\mathbf{w} \in \mathcal{W}$. Recently, first-order methods have experienced a renaissance in the design of fast algorithms for large-scale optimization problems. This is due the fact that although higher order methods such as interior point methods [118] have linear convergence rate, but this fast rate comes at the cost of more expensive iterations, typically requiring the solution of a system of linear equations in the input variables. Consequently, the cost of each iteration typically grows at least quadratically with the problem dimension, making interior point methods impractical for very-large-scale convex programs.

The convergence rate of gradient based methods usually depends on the properties of the objective function to be optimized. When the objective function is strongly convex and smooth, it is well known that gradient descent methods can achieve a geometric convergence rate [33]. When the objective function is smooth but not strongly convex, the optimal convergence rate of a gradient descent method is $O(1/T^2)$, and is achieved by the Nesterov's methods [93]. For the objective function which is strongly convex but not smooth, the convergence rate becomes $O(1/T)$ [134]. For general non-smooth objective functions, the optimal rate of any first order method is $O(1/\sqrt{T})$. Although it is not improvable in general, recent studies are able to improve this rate to $O(1/T)$ by exploring the special structure of the objective function [123, 122]. In addition, several methods are developed for composite optimization, where the objective function is written as a sum of a smooth and a non-smooth function [94, 93, 96]. The proof of coming results can be found in [121] and in the reference papers.

2.3.2.1 Gradient Descent Method

Perhaps the simplest and most intuitive algorithm for deterministic optimization is gradient decent (GD) method which which was proposed by Cauchy in 1846 [35] [1]. To find a solution within the domain $\mathcal{W}$ that optimizes the given objective function $f(\mathbf{w})$, GD computes the gradient of $f(\mathbf{w})$ by querying a first-order deterministic oracle, and updates the solution by moving it in the opposite direction of the gradient. To ensure that the solution stays within the domain $\mathcal{W}$, GD has to project the updated solution back into the $\mathcal{W}$ at every iteration.

[1] The original Cauchy's algorithm uses the direction that descends most and the best step-size which convergences slowly. Afterwards, a lot of researches have been done on how to choose the step-size for more efficient algorithms [64, 16]

```
Projected Gradient Descent (GD)

Input: convex set $\mathcal{W}$, $\eta > 0$, function $f \in \mathcal{F}$, first-order oracle $\mathcal{O}$
Initialize: $\mathbf{w}_1 \in \mathcal{W}$
for $t = 1, 2, \dots, T$
  Query the oracle $\mathcal{O}$ at point $\mathbf{w}_t$ to get $\nabla f(\mathbf{w}_t)$
  Update $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}\left(\mathbf{w}_t - \eta \nabla f(\mathbf{w}_t)\right)$
end for
```

Theorem 2.16 (Convergence Rate of GD)**.** *Assume that $f \in \mathcal{F}$ be a convex function defined over the convex domain $\mathcal{W} \subseteq \mathbb{R}^d$. Let $\mathbf{w}_* = \arg\min_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w})$ be the optimal solution. Then, for the convergence rate of GD algorithm*

- *if f be ρ-Lipschitz, by setting $\eta = \frac{R}{\rho\sqrt{t}}$ we have*

$$f\left(\frac{1}{T}\sum_{t=1}^{T} \mathbf{w}_t\right) - f(\mathbf{w}) \leq \frac{\rho\|\mathbf{w}_* - \mathbf{w}_1\|}{\sqrt{T}}.$$

- *if f be β-smooth by setting $\eta = \frac{1}{\beta}$ we have*

$$f(\mathbf{w}_T) - f(\mathbf{w}_*) \leq \frac{2\beta\|\mathbf{w}_* - \mathbf{w}_1\|^2}{T}.$$

- *if f be β-smooth and α-strongly convex, and $\kappa = \frac{\beta}{\alpha}$ be the condition number of f, by* *setting* $\eta = \frac{2}{\alpha+\beta}$ *we have*

$$f(\mathbf{w}_T) - f(\mathbf{w}_*) \leq \frac{\beta}{2}\|\mathbf{w}_1 - \mathbf{w}_*\|^2 \left(\frac{\kappa - 1}{\kappa + 1}\right)^T.$$

By comparing the rates obtained in Theorem 2.16 to the lower bounds in Table 2.1, one can realize that the GD obtains the optimal bound only for Lipschitz functions. We also note that by examining the bounds it turn out that the GD method is independent of the dimension of the convex domain $\mathcal{W}$ as long as the Euclidean norm of solutions and gradients are independent of the ambient dimension of convex domain $\mathcal{W}$ which makes it attractive for optimization in high dimension. The computational bottleneck of the projected GD is often the projection step which is a convex optimization problem by itself and might be expensive for many domains. In Chapter 9 we propose efficient optimization methods which do not require intermediate projection steps.

2.3.2.2 Accelerated Gradient Descent Method

The convergence rate of GD method for optimization smooth loss functions is $O(1/T)$ which is far away from the lower bound $O(1/T^2)$ discussed before. Nesterov showed in 1983 that we can improve the convergence rate of GD without using anything more than gradient information at various points of the domain. Accelerated GD [120, 121, 123] bridges the gap between the lower bound for smooth optimization and lower bound provided by oracle complexity with a simple twist of GD method and is able to obtain the optimal $O(1/T^2)$ convergence rate for minimizing smooth functions.

```
Accelerated Gradient Descent (AGD)

Input: $\eta > 0$, function $f \in \mathcal{F}$, first-order oracle $\mathcal{O}$
Initialize: $\mathbf{w}_0 = \mathbf{z}_0 = \mathbf{0}$, $\lambda_0 = 0$
for $t = 1, 2, \ldots, T$
    Query the oracle $\mathcal{O}$ at point $\mathbf{w}_t$ to get $\nabla f(\mathbf{w}_t)$
    Set $\eta_s = \frac{1}{2}\left(1 + \sqrt{1 + 4\eta_{t-1}^2}\right)$, and $\gamma_t = \frac{1-\eta_t}{\eta_{t+1}}$.
    Update $\mathbf{z}_{t+1} = \mathbf{z}_t - \frac{1}{\beta}\nabla f(\mathbf{w}_t)$
    Update $\mathbf{w}_{t+1} = (1 - \gamma_t)\mathbf{z}_{t+1} + \gamma_t \mathbf{z}_t$
end for
```

The following theorem shows that AGD achieves an $O(1/T^2)$ convergence rate which is tight.

Theorem 2.17 (Convergence Rate of AGD)**.** *Let $f \in \mathcal{F}$ be a convex and β-smooth function and $\mathbf{w}_*$ be the optimal solution. Then the accelerated gradient descent outputs a solution which satisfies:*

$$f(\mathbf{z}_T) - f(\mathbf{w}_*) \leq \frac{2\beta\|\mathbf{w}_1 - \mathbf{w}_*\|^2}{T^2}.$$

2.3.2.3 Mirror Descent Method

Mirror Descent (MD) is a first-order optimization procedure which generalizes the classic GD method to non-Euclidean geometries by relying on a distance generating function specific to the geometry. The original MD algorithm was developed to perform the gradient descent in spaces where the gradient only makes sense in the dual space. In this cases, the MD first maps the point $\mathbf{w}_t$ into a dual space by mapping Φ, then performs the gradient update in the dual space, and finally maps the resulting point back to the primal space. When the mapping $\Phi(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2$ then the primal and dual spaces are same and the MD performs a simple gradient descent.

```
Mirror Descent (MD)

Input: $\eta > 0$, function $f \in \mathcal{F}$, first-order oracle $\mathcal{O}$
Initialize: $\mathbf{w}_0 = \mathbf{0}$
for $t = 1, 2, \ldots, T$
  Query the oracle $\mathcal{O}$ at point $\mathbf{w}_t$ to get $\nabla f(\mathbf{w}_t)$
  Update $\nabla\Phi(\mathbf{z}_{t+1}) = \nabla\Phi(\mathbf{w}_t) - \eta\nabla f(\mathbf{w}_t)$
  Update $\mathbf{w}_{t+1} = \text{argmin}_{\mathbf{w}\in\mathcal{W}\cap\mathcal{K}}\mathsf{B}_\Phi(\mathbf{w}, \mathbf{z}_{t+1})$
end for
```

Theorem 2.18 (Convergence Rate of MD)**.** *Let Φ be a mirror map. Assume also that Φ is α-strongly convex on $\mathcal{W} \cap \mathcal{K}$ with respect to $\|\cdot\|$. Let $R = \sup_{\mathbf{w}\in\mathcal{K}\cap\mathcal{W}} \Phi(\mathbf{w}) - \Phi(\mathbf{w}_1)$ and f be convex and ρ-Lipschitz w.r.t. $\|\cdot\|$, then MD algorithm with $\eta = \frac{\rho}{R}\sqrt{\frac{2\alpha}{T}}$ satisfies*

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{w}_t\right) - \min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w}) \leq \rho R\sqrt{\frac{2}{\alpha T}}.$$

The MD algorithm can alternatively be expressed as nonlinear projected sub-gradient type method, derived from a general distance generating function (Bregmen divergence in Definition **??**) instead of the usual Euclidean squared distance as [17]:

$$\mathbf{w}_{t+1} = \min_{\mathbf{w}\in\mathcal{W}}\left\{\langle\mathbf{w}, \nabla f(\mathbf{w}_t)\rangle + \frac{1}{\eta}\mathsf{B}_\Phi(\mathbf{w}, \mathbf{w}_t)\right\}. \tag{2.11}$$

Remark 2.19. *In terms of convergence rate, the MD obtains the same rate as GD method but MD has advantage by exploiting the geometry of convex domain. More specifically, since MD method adapts to the structure of domain $\mathcal{W}$ via mapping Φ, it has less dependency on the dimensionality of the domain which could be appealing for large scale optimization problems. As an example, it is easy to verify that for optimization over simplex, i.e.,* $\Delta = \{\mathbf{w} \in \mathbb{R}^d_{++} : \sum_i w_i = 1\}$ *by using the negative entropy* $\Phi(\mathbf{w}) = \sum_{i=1}^d \log w_i$ *as the mapping function, the dependency of MD to* d *is in order of* $\log d$*, while regular GD algorithm has a linear* $O(d)$ *dependency.*

2.3.2.4 Mirror Prox Method

In the black-box oracle model the algorithm has access to the values and gradients of function, without knowing the structure of the objective function. But in many circumstances we never meet a pure black box model and have some information about the structure of the underlying function. Intestinally, the proper use of the structure of the problem can help to obtain better convergence rate for specific family of loss functions [122, 123, 117]. In particular, in [117] it been shown that for non-smooth Lipschitz continuous functions which admit a smooth saddle-point representation one can obtain a rate of convergence of order $O(1/T)$ with a properly designed gradient descent method, despite the fact that the original function is non-smooth and can not be optimized with a convergence rate better then $O(1/\sqrt{T})$ in black-box model. As an example consider the function f to be optimized is of the form $f(\mathbf{w}) = \max_{1 \leq i \leq n} f_i(\mathbf{w})$ where each individual functions $f_i(\mathbf{w}), i \in [n]$ is convex, β-smooth and ρ-Lipschitz in some norm $\|\cdot\|$. In this case the function $f(\mathbf{w})$ is not smooth and the best convergence rate one can hope in the black-box model is $O(1/\sqrt{T})$.

Let $\Phi : \mathcal{K} \to \mathbb{R}$ be a mirror map on $\mathcal{W}$ and let $\mathbf{w}_1 \in \text{argmin}_{\mathbf{w} \in \mathcal{W} \cap \mathcal{K}} \Phi(\mathbf{w})$. The mirror prox (extragradient in a specialized case) method is detailed below.

```
Extra Gradient Descent Method (EGD)

Input: $\eta > 0$, function $f \in \mathcal{F}$, first-order oracle $\mathcal{O}$
Initialize: $\mathbf{w}_1 = \mathbf{z}_1 = \mathbf{0}$
for $t = 1, 2, \ldots, T$
  Query the oracle $\mathcal{O}$ at point $\mathbf{w}_t$ to get $\nabla f(\mathbf{w}_t)$
  Update $\nabla\Phi(\mathbf{z}'_{t+1}) = \nabla\Phi(\mathbf{w}_t) - \eta\nabla f(\mathbf{w}_t)$
  Update $\mathbf{z}_{t+1} \in \text{argmin}_{\mathbf{z}\in\mathcal{W}\cap\mathcal{K}}\mathsf{B}_\Phi(\mathbf{z}, \mathbf{z}'_{t+1})$ and query the oracle to get $\nabla f(\mathbf{z}_{t+1})$
  Update $\nabla\Phi(\mathbf{w}'_{t+1}) = \nabla\Phi(\mathbf{w}_t) - \eta\nabla f(\mathbf{z}_{t+1})$
  Update $\mathbf{w}_{t+1} \in \text{argmin}_{\mathbf{w}\in\mathcal{W}\cap\mathcal{K}}\mathsf{B}_\Phi(\mathbf{w}, \mathbf{w}'_{t+1})$
end for
```

The EGD method first makes a step of MD to go from $\mathbf{w}_t$ to $\mathbf{z}_{t+1}$, and then it makes a similar step to obtain $\mathbf{w}_{t+1}$, starting again from $\mathbf{w}_t$ but this time using the gradient of f evaluated at $\mathbf{z}_{t+1}$. The following theorem exhibits the rate of convergence for EGD algorithm.

Theorem 2.20 (Convergence Rate of EGD)**.** *Let Φ be a α-strongly convex on $\mathcal{K}\cap\mathcal{W}$ with respect to $\|\cdot\|$. Let $R = \sup_{\mathbf{w}\in\mathcal{K}\cap\mathcal{W}}\Phi(\mathbf{w}) - \Phi(\mathbf{w}_1)$ and f be convex and β-smooth w.r.t. $\|\cdot\|$. Then EGD with $\eta = \frac{\alpha}{\beta}$ has a convergence rate as:*

$$f\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{z}_t\right) - \min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w}) \leq \frac{\beta R^2}{\alpha T}.$$

2.3.2.5 Conditional Gradient Descent Method

The main computation bottleneck of gradient descent methods in solving constrained optimization problems is the projection step which might be as hard as solving the original optimization problem. Surprisingly the projection step can be avoided by replacing the expensive projection operation with other kinds of light computational operations. One such an example is the Conditional Gradient Descent (CGD) method which is also known as Frank-Wolf algorithm. The Frank-Wolfe method, that was originally introduced in a paper by Frank and Wolfe from the 1950 [56], where they aimed to present an algorithm for minimizing a quadratic function over a polytope using only linear optimization steps over the feasible set.

The CGD algorithm proceeds by iteratively solving a linear optimization problem to find a direction $\mathbf{p}_t$ inside the domain $\mathcal{W}$ and updating the solution as a linear combination of the obtained direction and previous solution. This procedure guarantees that the updated solutions remain inside the feasible domain $\mathcal{W}$. This method replaces the projection step with a linear optimization problem over the constrained domain which is more efficient as long as the linear problem is easy to be solved.

Conditional Gradient Descent (CGD)

```
Input: convex set $\mathcal{W}$, $\eta > 0$, a smooth convex function $f \in \mathcal{F}$
Initialize: $\mathbf{w}_1 \in \mathcal{W}$
for $t = 1, 2, \ldots, T$
  Find $\mathbf{p}_t = \arg\min_{\mathbf{p} \in \mathcal{W}} \langle \nabla f(\mathbf{w}_t), \mathbf{p} \rangle$
  Update $\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t \mathbf{p}_t$
end for
```

The following result shows the convergence rate of CGD for smooth functions.

Theorem 2.21 (Convergence Rate of CGD)**.** *Assume that $f \in \mathcal{F}$ be a β-smooth convex function with respect to some norm $\|\cdot\|$ defined over the convex domain $\mathcal{W}$. Let $R = \sup_{\mathbf{w},\mathbf{w}'} \|\mathbf{w} - \mathbf{w}'\|$. Then by setting $\eta_t = \frac{2}{t+1}$ in CGD method, we have:*

$$f(\mathbf{w}_T) - f(\mathbf{w}_*) \leq \frac{2\beta R^2}{t+1}$$

In Chapter 9 we will show that by replacing the projection step with gradient computation of constrain function, it is possible to devise efficient stochastic optimization methods which only require a single projection at the final iteration.

2.3.3 Stochastic Convex Optimization

So far we assumed the the optimization algorithm has access to a noiseless oracle. It is more realistic to consider noisy oracles, where one does not have access to exact objective function or gradient values, but rather to their noisy estimates (usually zero mean and bounded variance). In particular, for a fixed closed convex subset $\mathcal{W} \subset \mathbb{R}^d$ of $\mathbb{R}^d$ we consider the following optimization problem:

$$\min_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}) \quad \text{for} \quad f(\mathbf{w}) = \mathbb{E}[F(\mathbf{w}, \xi)] = \int_{\Xi} F(\mathbf{w}, \xi) dP(\xi), \tag{2.12}$$

where we assume that the expected value function $f(\mathbf{w})$ is continuous and convex on $\mathcal{W}$. We note if the function $F(\mathbf{w}, \xi)$ be convex on $\mathcal{W}$, then it follows that $f(\mathbf{w})$ is also convex and the problem becomes a convex programming problem. The main difficulty in solving the stochastic optimization problem in (2.12) is that the multidimensional integral (expectation) cannot be computed with a high accuracy [115], and in statistical learning problems we usually do not know what the distribution P is. Therefore, there are two solutions to address this issue: these are stochastic approximation (SA) and the sample average approximation (SAA) methods. The main idea of SAA approach to solving stochastic programs is as follow. A sample $\xi_1, \xi_2, \cdots, \xi_n$ of n realizations of the random vector in objective is generated and the stochastic objective is approximated by the sample average function. Then, a deterministic optimization algorithm is applied to solve the approximate function. We note that we can not perform a gradient descent on $f(\mathbf{w})$ as we would need to know the underlying distribution to compute a gradient of $f(\mathbf{w})$

In SA we assume that there is an stochastic oracle $\mathcal{O}$, which, for a given point $(\mathbf{w}, \xi) \in \mathcal{W} \times \Xi$ returns an unbiased estimates of subgradient of $f(\mathbf{w})$. In other words, it returns $\mathbf{g}$ such that $\mathbb{E}[\mathbf{g}] \in \partial f(\mathbf{w})$. Stochastic optimization methods allow the optimization method to take a step which is only in expectation along the negative of the gradient. Based on this oracle, a simple algorithm to optimize the objective is Stochastic Gradient Descent (SGD). SGD is in the same spirit of GD but it replaces the true gradients with stochastic gradients in updating the solutions:

```
Stochastic Gradient Descent (SGD)

Input: convex set $\mathcal{W}$, $\eta > 0$, function $f \in \mathcal{F}$, stochastic first-order oracle $\mathcal{O}$
Initialize: $\mathbf{w}_0 = \mathbf{0}$
for $t = 1, 2, \ldots, T$
  Query the stochastic oracle $\mathcal{O}$ at point $\mathbf{w}_t$ to get $\mathbf{g}_t$ where $\mathbb{E}[\mathbf{g}_t] \in \partial f(\mathbf{w}_t)$
  Update $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}_t - \eta \mathbf{g}_t)$
end for
Return: $\hat{\mathbf{w}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{w}_t$
```

Under mild conditions as outlined below, one con show the SGD algorithm convergence to the optimal solution with convergence rate $O(1/\sqrt{T})$ with a high probability:

$$\mathbb{E}_{\xi_t}[\mathbf{g}_t] = \nabla f(\mathbf{w})$$

$$\mathbb{E}_{\xi_t}[\exp(\|\mathbf{g}_t - \nabla f(\mathbf{w})\|_*^2/\sigma^2)] \leq \exp(1).$$

It is also straightforward to generalize the the mirror descent method to stochastic setting by replacing the Euclidean distance in the update of SGD with another Bregman divergence adopted to a specific domain.

By comparing SGD method for stochastic optimization and OGD method for regret minimization, we note that both methods are closely related. Although both methods looks similar algorithmically, but there are main conceptual differences between SGD and OGD. We note that in stochastic optimization the goal is to generate a sequence of solutions which quickly convergences to the minimum of a function defined as $f(\mathbf{w}) = \mathbb{E}[F(\mathbf{w}, \xi)]$, while is online learning the goal is to generate a sequence of solutions that accumulates a small loss during the learning measured in terms of regret. In other words SGD provides an incremental solution to a stochastic optimization problem and OGD provides a solution to adopt a sequence of adversarially generated loss functions. We note that regret minimization algorithms equipped with online to batch conversion schemas discussed before settle an efficient paradigm to solve general optimization problems, but sometimes it seems essential to go beyond this barrier to obtain optimal convergence rates in stochastic setting [72, 125].

Remark 2.22. *It is remarkable that in stark contrast to deterministic optimization where the smoothness of objective function makes a significant improvement in terms of convergence rate (i.e., Theorem 2.16), in stochastic optimization the smoothness is not a desirable property as it yields the same convergence rate as the Lipschitz functions. In particular, as it has been shown in Appendix A.3, a tight analysis of stochastic mirror descent algorithm has an* $O(\frac{\beta R^2}{T} + \frac{\sigma R}{\sqrt{T}})$ *convergence rate for smooth objective functions, which is dominated by the slow* $O(1/\sqrt{T})$ *rate unless the variance of stochastic gradients becomes zero* $\sigma = 0$. *As it will be discussed in Chapter 7, the mixed optimization paradigm we introduce in thesis is able to leverage the smoothness of objection function to attain an* $O(1/T)$ *rate by accessing the full gradient oracle* $\log T$ *times on top of the* $O(T)$ *accesses of the stochastic gradient oracle.*

2.3.4 Convex Optimization for Learning Problems

Formulating statistical learning tasks and in particular convex learning problems as a convex optimization problem makes an intimate connection between learning and mathematical optimization. Therefore, optimization methods play a central role in solving machine learning problems and challenges exist in machine learning applications demand the development of new optimization algorithms.

To see this, consider the typical problem of the supervised learning consisting of a input space $\Xi = \mathcal{X} \times \mathcal{Y}$ and a suitable set of hypotheses $\mathcal{W}$ for prediction such as the set of linear predictors, i.e., $\mathcal{W} = \{\mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle : \mathbf{w} \in \mathbb{R}^d\}$. Then, the learner is provided with a training sample $\mathcal{S} = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n)) \in \Xi^n$ and is supposed to pick a hypothesis $\mathbf{w} \in \mathcal{W}$ which minimizes appropriate empirical cost over the training sample based on a predefined surrogate loss function $\ell : \mathcal{W} \times \Xi \mapsto \mathbb{R}_+$. The last step of this learning process corresponds to an optimization algorithm that solves the minimization problem of picking that hypothesis from the set of hypotheses. As a result, convex optimization forms the backbone of many algorithms for statistical learning. This formulation includes support vector machine (SVM), support vector regression (SVR), Lasso, logistic regression, and ridge regression among many others as detailed below:

- Hinge loss (Support Vector Machine)): $\sum_{i=1}^{n} \max(0, 1 - y_i \langle \mathbf{w}, \mathbf{x}_i \rangle)$.
- Logistic loss (Logistic Regression): $\min_{\mathbf{w} \in \mathcal{W}} \sum_{i=1}^{n} \log(1 + \exp(-y_i \langle \mathbf{w}, \mathbf{x}_i \rangle))$.
- Least-squares loss (Regression): $\min_{\mathbf{w} \in \mathcal{W}} \sum_{i=1}^{n} (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle)^2$.
- Exponential loss (Boosting): $\min_{\mathbf{w} \in \mathcal{W}} \sum_{i=1}^{n} \exp(-y_i \langle \mathbf{w}, \mathbf{x}_i \rangle)$.

The domain $\mathcal{W}$ in above formulations, captures the constrains on the classifier $\mathbf{w}$. Commonly considered examples are the bounded Euclidean ball $\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\|_2 \leq R\}$, bounded ℓ_1 ball $\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\|_1 \leq B\}$ or the box $\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\|_\infty \leq B\}$. We note that instead of moving the constraint into the $\mathcal{W}$, by leveraging on the theory of

Lagrangian method in constrained optimization, one can simply move the constraint into the objective and solve the unconstrained optimization problem, i.e., $\mathcal{W} = \mathbb{R}^d$.

To fully understand the application of convex optimization methods to solving machine learning problems, let us consider the following optimization problem:

$$\min_{\mathbf{w} \in \mathcal{W}} \mathcal{L}_{\mathcal{S}}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, (\mathbf{x}_i, y_i)) \tag{2.13}$$

A preliminary approach for solving the optimization problem in (2.13) is the batch gradient descent (GD) algorithm. It starts with some initial point, and iteratively updates the solution using the equation $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}_t - \eta \nabla \mathcal{L}_{\mathcal{S}}(\mathbf{w}_t))$ where

$$\nabla \mathcal{L}_{\mathcal{S}}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} \partial \ell(\mathbf{w}, (\mathbf{x}_i, y_i)) \mathbf{x}_i.$$

The main shortcoming of GD method is its high cost in computing the full gradient $\nabla \mathcal{L}_{\mathcal{S}}(\mathbf{w}_t)$, i.e., O(n) gradient computations, when the number of training examples is large. Stochastic gradient descent (SGD) alleviates this limitation of GD by sampling one (or a small set of) examples and computing a stochastic (sub)gradient at each iteration based on the sampled examples. Since the computational cost of SGD per iteration is independent of the size of the data (i.e., n), it is usually appealing for large-scale learning and optimization [29, 115, 134]. Despite of their slow rate of convergence compared with the batch methods, stochastic optimization methods have shown to be very effective for large scale and online learning problems, both theoretically [115, 94] and empirically [134]. We note although a large number of iterations is usually needed to obtain a solution of desirable accuracy, the lightweight computation per iteration makes SGD attractive for many large-scale learning problems.

2.3.5 From Stochastic Optimization to Convex Learning Theory

As mentioned earlier, most of existing learning algorithms follow the framework of empirical risk minimizer or regularized ERM, which was developed to great extent by Vapnik and Chervonenkis [146]. Essentially, ERM methods use the empirical loss over $\mathcal{S}$, i.e.,

$$\mathcal{L}_{\mathcal{S}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}, (\mathbf{x}_i, y_i)),$$

as a criterion to pick a hypothesis. From optimization viewpoint, the ERM methods resembles the widely used Sample Average Approximation (SAA) method in the optimization community when the hypothesis space and the loss function are convex. If uniform convergence holds, then the empirical risk minimizer is consistent, i.e., the population risk of the ERM converges to the optimal population risk, and the problem is learnable using ERM.

A rather different paradigm for risk minimization is stochastic optimization. Recall that the goal of learning is to approximately minimize the risk

$$\mathcal{L}_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\ell(\mathbf{w}, (\mathbf{x}, y))]. \tag{2.14}$$

However, since the distribution $\mathcal{D}$ is unknown to the learner, we can not utilize standard gradient methods to directly minimize the expected loss in (2.14). This is because we are not able to compute the gradient $\nabla\mathcal{L}_{\mathcal{D}}(\mathbf{w})$ at a particular query point $\mathbf{w}$. We note that this is different from the application of SGD for solving the optimization problem in (2.13) because in (2.13) the randomness is over the uniform sampling from the objective function which is known (essentially we have a randomized optimization method), while in (2.14) the randomness is imposed on the instance space Ξ through a distribution $\mathcal{D}$ which is unknown to the learner in advance.

In stochastic optimization all we need is not the exact gradient of objective function, but an unbiased estimate of the true gradient $\nabla \mathcal{L}_{\mathcal{D}}(\mathbf{w})$. Surprisingly, it turns out that the construction of this unbiased estimate is extremely simple for risk minimization as follows. First, we sample an instance $\mathbf{z} = (\mathbf{x}_i, y_i) \in \Xi$ according to $\mathcal{D}$ and set the stochastic gradient to be

$$\mathbf{g} = \partial \ell(\mathbf{w}, (\mathbf{x}_i, y_i))\mathbf{x}_i,$$

which will be an unbiased estimate of true gradient, i.e., $\mathbb{E}[\mathbf{g}] = \nabla \mathcal{L}_{\mathcal{D}}(\mathbf{w})$.

The beauty of SGD for direct risk minimization is that it is efficient and it delivers the same sample complexity as the ERM method. To motivate stochastic optimization as an alternative to the ERM method, [132, 131] challenged the ERM method and showed that there is a real gap between learnability and uniform convergence by investigating non-trivial problems where no uniform convergence holds, but they are still learnable using SGD algorithm [115]. These results uncovered an important relationship between learnability and stability, and showed that stability together with approximate empirical risk minimization, assures learnability [133]. Unlike ERM method in which the learnability is characterized by attendant complexity of hypothesis space, in SGD based learning, stability is a general notion to characterize learnability. In particular, in learning setting under i.i.d. samples where uniform convergence is not necessary for learnability, but where stability is both sufficient and necessary for learnability.

Chapter 3

Passive Learning with Target Risk

The setup of this chapter will be in the classical statistical learning setting discussed in Chapter 2, but with a slight modification. In particular, we assume that the target expected loss, also referred to as target risk, is provided in advance for learner as prior knowledge. Unlike most studies in the learning theory that only incorporate the prior knowledge into the generalization bounds, we are able to explicitly utilize the target risk in the learning process. By leveraging on the smoothness of loss function, our analysis reveals a surprising result on the sample complexity of learning: by exploiting the target risk in the learning algorithm, we show that when the loss function is both smooth and strongly convex, the sample complexity reduces to $O(\log\left(\frac{1}{\epsilon}\right))$, an exponential improvement compared to the sample complexity $O(\frac{1}{\epsilon})$ for learning with strongly convex loss functions. Furthermore, our proof is constructive and is based on a computationally efficient stochastic optimization algorithm, dubbed ClippedSGD, for such settings which demonstrate that the proposed algorithm is practically useful.

The remainder of the chapter is organized as follows: Section 3.1 motivates the problem and setups the notation. Section 3.2 motivates the main intuition behind the proposed algorithm. The proposed ClippedSGD algorithm and main result on its sample complexity are discussed in Section 3.3. The proof of logarithmic sample complexity is given in Section 3.4 and the omitted proofs are deferred to Section 3.5. Section 3.6 summarizes the chapter and Section 3.7 surveys the related works.

3.1 Setup and Motivation

Recall that in the standard statistical or passive supervised learning setting, we consider an input space $\Xi \equiv \mathcal{X} \times \mathcal{Y}$ where $\mathcal{X} \subseteq \mathbb{R}^d$ is the space for instances and $\mathcal{Y}$ is the set of labels, and a hypothesis class $\mathcal{H}$ from which we choose a classifier. We assume that the domain space Ξ is endowed with an unknown probability measure $\mathcal{D}$ and measure the performance of a specific hypothesis h by defining a nonnegative loss function $\ell : \mathcal{H} \times \Xi \to \mathbb{R}_+$. The risk of a hypothesis h with respect to the underlying distribution $\mathcal{D}$ is defined as:

$$\mathcal{L}_{\mathcal{D}}(h) = \mathbb{E}_{\mathbf{z} \sim \mathcal{D}}[\ell(h, \mathbf{z})].$$

Given a sample $\mathcal{S} = (\mathbf{z}_1, \cdots, \mathbf{z}_n) = ((\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)) \sim \Xi^n$, the goal of a learning algorithm is to pick a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ from $\mathcal{H}$ in such a way that its risk $\mathcal{L}_{\mathcal{D}}(h)$ is close to the minimum possible risk of a hypothesis in $\mathcal{H}$.

In the new setting we consider for learning here, we assume that before the start of the learning process, the learner has in mind a *target expected loss*, also referred to as *target risk*, denoted by ϵ_{prior} [1], and tries to learn a classifier with the expected risk of $O(\epsilon_{\text{prior}})$ by labeling a small number of training examples. We further assume the target risk ϵ_{prior} is feasible, i.e., $\epsilon_{\text{prior}} \geq \epsilon_{\text{opt}}$ where $\epsilon_{\text{opt}} = \min_{h \in \mathcal{H}} \mathcal{L}_{\mathcal{D}}(h)$. To address this problem, we develop an efficient algorithm, based on stochastic optimization, for passive learning with target risk. The most surprising property of the proposed algorithm is that when the loss function is both smooth and strongly convex, it only needs $O(d \log(1/\epsilon_{\text{prior}}))$ labeled examples to find a classifier with the expected risk of $O(\epsilon_{\text{prior}})$, where d is the dimension of data. This is a

[1] We use ϵ_{prior} instead of ϵ to emphasize the fact that this parameter is known to the learner in advance.

significant improvement compared to the sample complexity for empirical risk minimization. We note that the target risk assumption is fully exploited by the learning algorithm and stands in contrast to all those assumptions such as the nature of unknown distribution $\mathcal{D}$, sparsity, and margin that usually enter into the generalization bounds and are often perceived as a rather crude way to incorporate such assumptions.

The key intuition behind the ClippedSGD algorithm is that by knowing target risk as prior knowledge, the learner has better control over the variance in stochastic gradients, which contributes mostly to the slow convergence in stochastic optimization and consequentially large sample complexity in passive learning. The trick is to run the stochastic optimization in multiple stages with a *fixed* size and decrease the variance of stochastically perturbed gradients at each iteration by a properly designed mechanism. Another crucial feature of the proposed algorithm is to utilize the target risk ϵ_{prior} to gradually refine the hypothesis space as the algorithm proceeds. Our algorithm differs significantly from standard stochastic optimization algorithms and is able to achieve a geometric convergence rate with the knowledge of target risk ϵ_{prior}.

To analyze the sample complexity of ClippedSGD algorithm, we pursue the stochastic optimization viewpoint for risk minimization detailed in Chapter 2. Precisely, we focus on the convex learning problems for which we assume that the hypothesis class $\mathcal{H}$ is a parametrized convex set $\mathcal{H} = \{h_{\mathbf{w}} : \mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle : \mathbf{w} \in \mathbb{R}^d, \|\mathbf{w}\| \leq R\}$ and for all $\mathbf{z} = (\mathbf{x}, y) \in \Xi$, the loss function $\ell(\cdot, \mathbf{z})$ is a non-negative convex function. Thus, in the remainder we simply use vector $\mathbf{w}$ to represent $h_{\mathbf{w}}$, rather than working with hypothesis $h_{\mathbf{w}}$. We will assume throughout that $\mathcal{X} \subseteq \mathbb{R}^d$ is the unit ball so that $\|\mathbf{x}\| \leq 1$. Finally, the conditions under which we can get the desired result on sample complexity depend on analytic properties of the loss function. In particular, we assume that the loss function is strongly convex and smooth as defined in

Chapter 2 and can be found in Appendix **??**. We would like to emphasize that in our setting, we only need that the expected loss function $\mathcal{L}_{\mathcal{D}}(\mathbf{w})$ be strongly convex, without having to assume strong convexity for individual loss functions.

3.2 The Curse of Stochastic Oracle

We begin by discussing stochastic optimization for risk minimization, convex learnability, and then the main intuition that motivates the proposed algorithm.

As mentioned earlier in Chapter 2, most existing learning algorithms follow the framework of empirical risk minimizer (ERM) or regularized ERM methods that use the empirical loss over $\mathcal{S}$, i.e., $\mathcal{L}_{\mathcal{S}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{z}_i)$, as a criterion to pick a hypothesis. In regularized ERM methods, the learner picks a hypothesis that jointly minimizes $\mathcal{L}_{\mathcal{S}}(\mathbf{w})$ and a regularization function over $\mathbf{w}$.

A rather different paradigm for risk minimization is stochastic optimization. Recall that the goal of learning is to approximately minimize the risk $\mathcal{L}_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}[\ell(\mathbf{w}, \mathbf{z})]$. However, since the distribution $\mathcal{D}$ is unknown to the learner, we can not utilize standard gradient methods to minimize the expected loss. Stochastic optimization methods circumvent this problem by allowing the optimization method to take a step which is only in expectation along the negative of the gradient. To directly solve $\min_{\mathbf{w}\in\mathcal{H}} \left[\mathcal{L}_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{\mathbf{z}\sim\mathcal{D}}[\ell(\mathbf{w}, \mathbf{z})]\right]$, a typical stochastic optimization algorithm initially picks some point in the feasible set $\mathcal{H}$ and iteratively updates these points based on first order perturbed gradient information about the function at those points. For instance, the widely used SGD algorithm starts with $\mathbf{w}_0 = \mathbf{0}$; at each iteration t, it queries the stochastic oracle $\mathcal{O}_s$ at $\mathbf{w}_t$ to obtain a perturbed but unbiased gradient $\mathbf{g}_t$ and updates the current solution by

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{H}} \left(\mathbf{w}_t - \eta_t \mathbf{g}_t \right),$$

where $\Pi_{\mathcal{H}}(\cdot)$ projects the solution $\mathbf{w}$ into the domain $\mathcal{H}$.

To capture the efficiency of optimization procedures in a general sense, one can use oracle complexity of the algorithm which, roughly speaking, is the minimum number of calls to any oracle needed by any method to achieve desired accuracy [121]. We note that the oracle complexity corresponds to the sample complexity of learning from the stochastic optimization viewpoint previously discussed. This viewpoint for learning theory has been taken by few very recent works [132, 131] where the ERM method has been challenged and it has been shown that there is a real gap between learnability and uniform convergence. This has been done by investigating non-trivial problems where no uniform convergence holds, but they are still learnable using SGD algorithm. These results uncovered an important relationship between learnability and stability, and showed that stability together with approximate empirical risk minimization, assures learnability [133]. Unlike ERM method in which the learnability is characterized by attendant complexity of hypothesis space, in SGD based learning, stability is a general notion to characterize learnability. In particular, in learning setting under i.i.d. samples where uniform convergence is not necessary for learnability, but where stability is both sufficient and necessary for learnability.

To motivate the main intuition behind the proposed method, we begin by stating the following theorem which provides a lower bound on the sample complexity of stochastic optimization algorithms that is taken from [119].

Theorem 3.1 (Lower Bound on Oracle Complexity). *Suppose* $\mathcal{L}_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{\mathbf{z} \sim \mathcal{D}}[\ell(\mathbf{w}, \mathbf{z})]$

is α-strongly and β-smooth convex function defined over convex domain $\mathcal{H}$. Let $\mathcal{O}_s$ be a stochastic oracle that for any point $\mathbf{w} \in \mathcal{H}$ returns an unbiased estimate $\mathbf{g}$, i.e., $\mathbb{E}[\mathbf{g}] = \nabla \mathcal{L}_{\mathcal{D}}(\mathbf{w})$, such that $\mathbb{E}\left[\|\mathbf{g} - \nabla \mathcal{L}_{\mathcal{D}}(\mathbf{w})\|^2\right] \leq \sigma^2$ holds. Then for any stochastic optimization algorithm $\mathcal{A}$ to find a solution $\widehat{\mathbf{w}}$ with ϵ accuracy respect to the optimal solution $\mathbf{w}_*$, i.e., $\mathbb{E}\left[\mathcal{L}_{\mathcal{D}}(\widehat{\mathbf{w}}) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}_*)\right] \leq \epsilon$, the number of calls to $\mathcal{O}_s$ is lower bounded by

$$\mathcal{O}(1) \left(\sqrt{\frac{\beta}{\alpha}} \log \left(\frac{\beta \|\mathbf{w}_0 - \mathbf{w}_*\|^2}{\epsilon} \right) + \frac{\sigma^2}{\alpha \epsilon} \right). \tag{3.1}$$

The first term in (3.1) comes from deterministic oracle complexity and the second term is due to noisy gradient information provided by stochastic oracle $\mathcal{O}_s$. As indicated in (3.1), the slow convergence rate for stochastic optimization is due to the variance in stochastic gradients, leading to at least $O\left(\sigma^2/\epsilon\right)$ queries to be issued. We note that the idea of mini-batch [44, 50], although it reduces the variance in stochastic gradients, does not reduce the oracle complexity.

We close this section by informally presenting why logarithmic sample complexity is, in principle, possible, under the assumption that target risk is known to the learner $\mathcal{A}$. To this end, consider the setting of Theorem 3.1 and assume that the learner $\mathcal{A}$ is given the prior accuracy ϵ_{prior} and is asked to find an ϵ_{prior}-accurate solution. If it happens that the variance of stochastic oracle $\mathcal{O}_s$ has the same magnitude as ϵ_{prior}, i.e., $\mathbb{E}\left[\|\mathbf{g} - \nabla \mathcal{L}_{\mathcal{D}}(\mathbf{w})\|^2\right] \leq \epsilon_{\text{prior}}$, then from (3.1) it follows that the second term vanishes and the learner $\mathcal{A}$ needs to issue only $O\left(\log 1/\epsilon_{\text{prior}}\right)$ queries to find the solution. But, since there is no control on the stochastic oracle $\mathcal{O}_s$, except that the variance of stochastic gradients are bounded, $\mathcal{A}$ needs a mechanism to manage the variance of perturbed gradients at each iteration in order to alleviate the influence of noisy gradients. One strategy is to replace the unbiased estimate of gradient with a biased one, which unfortunately may yield loose bounds. To overcome this problem, we introduce a strategy that shrinks the solution space with respect to the target risk ϵ_{prior} to control the damage caused by biased estimates.

As an illustrative example to see how the knowledge of target risk is helpful, we consider a simple one dimensional regression problem with loss function $\ell(w, x) = (wx - b)^2$ where b is a random variable that can either be δ or 1 with $\Pr[b = \delta] = 1 - \delta^2$. Here we choose δ to be a very small value $\delta \ll 1$. The loss function is non-negative, smooth, and strongly convex and is appropriate for our setting. For this setting we have, $\epsilon_{\text{opt}} \leq \mathbb{E}_b[\ell(0)] = \delta^2 \times 1 + (1 - \delta^2) \times \delta^2 \leq 2\delta^2$ which can be arbitrarily small. For this example, the solution obtained by ERM with a small number of training examples will be on order of δ and therefore its expected risk will be on the order of δ^2. However, from the viewpoint of the learner, this expected risk is *unknown* unless the learner could figure out $\Pr(b = 1) = \delta^2$, which unfortunately requires an order of $1/\delta^2$ samples. On the other hand, by having the target feasible risk as prior knowledge the learner is able to find out $\Pr(b = 1)$ with a small number of samples.

3.3 The ClippedSGD Algorithm

In this section we proceed to describe the proposed algorithm and state the main result on its sample complexity.

3.3.1 The Algorithm Description

We now turn to describing our algorithm. Interestingly, our algorithm is quite dissimilar to the classic stochastic optimization methods. It proceeds by running the algorithm online on fixed chunks of examples, and using the intermediate hypotheses and target risk ϵ_{prior} to gradually refine the hypothesis space. As mentioned above, we assume in our setting that the target expected risk ϵ_{prior} is provided to the learner a priori. We further assume the target risk ϵ_{prior} is feasible for the solution within the domain $\mathcal{H}$, i.e., $\epsilon_{\text{prior}} \geq \epsilon_{\text{opt}}$. The proposed algorithm explicitly takes advantage of the knowledge of expected risk ϵ_{prior} to attain an $O\left(\log(1/\epsilon_{\text{prior}})\right)$ sample complexity.

Throughout we shall consider linear predictors of form $\langle \mathbf{w}, \mathbf{x} \rangle$ and assume that the loss function of interest $\ell(\langle \mathbf{w}, \mathbf{x} \rangle, y)$ is β-smooth. It is straightforward to see that $\mathcal{L}_{\mathcal{D}}(\mathbf{w}) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\ell(\langle \mathbf{w}, \mathbf{x} \rangle, y)\right]$ is also β-smooth. In addition to the smoothness of the loss function, we also assume that $\mathcal{L}_{\mathcal{D}}(\mathbf{w})$ to be α-strongly convex. We denote by $\mathbf{w}_*$ the optimal solution that minimizes $\mathcal{L}_{\mathcal{D}}(\mathbf{w})$, i.e., $\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathcal{H}} \mathcal{L}_{\mathcal{D}}(\mathbf{w})$, and denote its optimal value by ϵ_{opt}.

Let $(\mathbf{x}_t, y_t), t = 1, \ldots, T$ be a sequence of i.i.d. training examples. The proposed algorithm divides the T iterations into the m stages, where each stage consists of T_1 training examples, i.e., $T = mT_1$. Let $(\mathbf{x}_k^t, y_k^t)$ be the tth training example received at stage k, and let η be the step size used by all the stages. At the beginning of each stage k, we initialize the solution $\mathbf{w}$ by the average solution $\widehat{\mathbf{w}}_k$ obtained from the last stage, i.e.,

$$\widehat{\mathbf{w}}_k = \frac{1}{T_1} \sum_{t=1}^{T_1} \mathbf{w}_k^t, \tag{3.2}$$

where $\mathbf{w}_k^t$ denotes the tth solution at stage k. Another feature of the proposed algorithm is a domain shrinking strategy that adjusts the domain as the algorithm proceeds using intermediate hypotheses and target risk. We define the domain $\mathcal{H}_k$ used at stage k as

$$\mathcal{H}_k = \left\{\mathbf{w} \in \mathcal{H} : \|\mathbf{w} - \widehat{\mathbf{w}}_k\| \leq \Delta_k\right\}, \tag{3.3}$$

where Δ_k is the domain size, whose value will be discussed later. Similar to the SGD method, at each iteration of stage k, we receive a training example $(\mathbf{x}_k^t, y_k^t)$, and compute the gradient $\mathbf{g}_k^t = \ell'\left(\langle \mathbf{w}_k^t, \mathbf{x}_k^t \rangle, y_t\right)\mathbf{x}_k^t$. Instead of using the gradient directly, a clipped version of the gradient, denoted by $\mathbf{v}_k^t = \text{clip}\left(\gamma_k, \mathbf{g}_k^t\right)$, will be used for updating the solution. More specifically, the clipped vector $\mathbf{v}_k^t \in \mathbb{R}^d$ is defined as

$$[\mathbf{v}_k^t]_i = \text{clip}\left(\gamma_k, \left[\mathbf{g}_k^t\right]_i\right) = \text{sign}\left(\left[\mathbf{g}_k^t\right]_i\right) \min\left(\gamma_k, \left|\left[\mathbf{g}_k^t\right]_i\right|\right), i = 1, \ldots, d \tag{3.4}$$

where $\gamma_k = 2\xi\beta\Delta_k$ with $\xi \geq 1$. Given the clipped gradient $\mathbf{v}_k^t$, we follow the standard framework of stochastic gradient descent, and update the solution by

$$\mathbf{w}_k^{t+1} = \Pi_{\mathcal{H}_k}\left(\mathbf{w}_k^t - \eta \mathbf{v}_k^t\right). \tag{3.5}$$

The purpose of introducing the clipped version of the gradient is to effectively control the variance in stochastic gradients, an important step toward achieving the geometric convergence rate. At the end of each stage, we will update the domain size by explicitly exploiting the target expected risk ϵ_{prior} as

$$\Delta_{k+1} = \sqrt{\varepsilon \Delta_k^2 + \tau \epsilon_{\text{prior}}}\ , \tag{3.6}$$

where $\varepsilon \in (0, 1)$ and $\tau \in (0, 1)$ are two parameters, both of which will be discussed later.

Algorithm 1 gives the detailed steps for the proposed method. The three important aspects of Algorithm 1, all crucial to achieve a geometric convergence rate, are highlighted as follows:

- Each stage of the proposed algorithm is comprised of the same number of training

Algorithm 1 ClippedSGD Algorithm

```
Input:
  • step size $\eta$
  • stage size $T_1$
  • number of stages $m$
  • target expected risk $\epsilon_{\text{prior}}$
  • parameters $\varepsilon \in (0,1)$ and $\tau \in (0,1)$ used for updating domain size $\Delta_k$
  • parameter $\xi \geq 1$ used to clip the gradients
Initialize: $\widehat{\mathbf{w}}_1 = 0$, $\Delta_1 = R$, and $\mathcal{H}_1 = \mathcal{H}$
for $k = 1, \ldots, m$ do
    Set $\mathbf{w}_k^t = \widehat{\mathbf{w}}_k$ and $\gamma_k = 2\xi\beta\Delta_k$
    for $t = 1, \ldots, T_1$ do
        Receive training example $(\mathbf{x}_t, y_t)$
        Compute the gradient $\mathbf{g}_k^t$ and
        Clip the gradient $\mathbf{g}_k^t$ to $\mathbf{v}_k^t$ using
            $[\mathbf{v}_k^t]_i = \text{sign}\left(\left[\mathbf{g}_k^t\right]_i\right) \min\left(\gamma_k, \left|\left[\mathbf{g}_k^t\right]_i\right|\right), i = 1, \ldots, d$
        Update the solution by $\mathbf{w}_k^{t+1} = \Pi_{\mathcal{H}_k}\left(\mathbf{w}_k^t - \eta\mathbf{v}_k^t\right)$
    end for
    Update $\Delta_k$ using (3.6).
    Compute the average solution $\widehat{\mathbf{w}}_{k+1}$ according to (3.2)
    Shrink the domain $\mathcal{H}_{k+1}$ using the expression in (3.3).
end for
```

examples. This is in contrast to the epoch gradient algorithm [72] which divides m iterations into exponentially increasing epochs, and runs SGD with averaging on each epoch. Also, in our case the learning rate is fixed for all iterations.

- The proposed algorithm uses a clipped gradient for updating the solution in order to better control the variance in stochastic gradients; this stands in contrast to the SGD method, which uses original gradients to update the solution.

- The proposed algorithm takes into account the targeted expected risk and intermediate hypotheses when updating the domain size at each stage. The purpose of domain shrinking is to reduce the damage caused by biased gradients that resulted from clipping operation.

3.3.2 Main Result on Sample Complexity

The main theoretical result on the performance of the ClippedSGD algorithm is given in the following theorem.

Theorem 3.2 (Convergence Rate). *Assume that the hypothesis space $\mathcal{H}$ is compact and the loss function ℓ is α-strongly convex and β-smooth. Let $T = mT_1$ be the size of the sample and ϵ_{prior} be the target expected loss given to the learner in advance such that $\epsilon_{\text{opt}} \leq \epsilon_{\text{prior}}$ holds. Given $\varepsilon \in (0,1)$ and $\tau \in (0,1)$, set ξ, η, and T_1 as*

$$\xi = \frac{4\beta}{\alpha\tau},\ T_1 = 4\max\left\{\frac{\xi^3\beta d + 2\xi\beta\sqrt{d}}{\varepsilon\alpha}\ln\frac{ms}{\delta}, \frac{16\xi^2\beta^2}{\alpha^2\varepsilon^2}\right\},\ \eta = \frac{1}{2\xi\beta\sqrt{T_1}},$$

where

$$s = \left\lceil \log_2 \frac{\xi\beta R^2}{\epsilon_{\text{prior}}} \right\rceil. \tag{3.7}$$

After running Algorithm 1 over m stages, we have, with a probability $1-\delta$,

$$\mathcal{L}_{\mathcal{D}}(\widehat{\mathbf{w}}_{m+1}) \leq \frac{\beta R^2}{2}\varepsilon^m + \left(1 + \frac{\tau}{1-\varepsilon}\right)\epsilon_{\text{prior}},$$

implying that only $O(d\log[1/\epsilon_{\text{prior}}])$ training examples are needed in order to achieve a risk of $O(\epsilon_{\text{prior}})$.

We note that comparing to the bound in Theorem 3.1, for Algorithm 1 the level of error to which the linear convergence holds is not determined by the noise level in stochastic gradients, but by the target risk. In other words, the algorithm is able to tolerate the noise by knowing the target risk as prior knowledge and achieves a linear convergence to the level of the target risk even when the variance of stochastic gradients is much larger than the target risk. In addition, although the result given in Theorem 3.2 assumes a bounded domain with $\|\mathbf{w}\| \leq R$, however, this assumption can be lifted by effectively exploring the strong convexity of the loss function and further assuming that the loss function is Lipschitz continuous with constant G, i.e., $|\mathcal{L}_{\mathcal{D}}(\mathbf{w}_1) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}_2)| \leq G\|\mathbf{w}_1 - \mathbf{w}_2\|, \ \forall \ \mathbf{w}_1, \mathbf{w}_2 \in \mathcal{H}$. More specifically, the fact that the $\mathcal{L}_{\mathcal{D}}(\mathbf{w})$ is α-strongly convex with first order optimality condition, from Lemma A.14 for the optimal solution $\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathcal{H}} \mathcal{L}_{\mathcal{D}}(\mathbf{w})$, we have

$$\mathcal{L}_{\mathcal{D}}(\mathbf{w}) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}_*) \geq \frac{\alpha}{2}\|\mathbf{w} - \mathbf{w}_*\|^2, \ \ \forall \mathbf{w} \in \mathcal{H}.$$

This inequality combined with Lipschitz continuous assumption implies that for any $\mathbf{w} \in \mathcal{H}$ the inequality $\|\mathbf{w} - \mathbf{w}_*\| \leq R_* := 2G/\alpha$ holds, and therefore we can simply set $R = R_*$. We also note that this dependency can be resolved with a weaker assumption than Lipschitz continuity, which only depends on the gradient of loss function at origin. To this end, we define $|\ell'(0, y)| = G$. Using the fact that $\mathcal{L}_{\mathcal{D}}(\mathbf{w})$ is α-strongly, it is easy to verify that $\frac{\alpha}{2}\|\mathbf{w}_*\|^2 - G\|\mathbf{w}_*\| \leq 0$, leading to $\|\mathbf{w}_*\| \leq R_* := \frac{2}{\alpha}G$ and, therefore, we can simply set $R = R_*$.

We now use our analysis of Algorithm 1 to obtain a sample complexity analysis for learning smooth strongly convex problems with a bounded hypothesis class. To make it easier to parse, we only keep the dependency on the main parameters d, α, β, T, and ϵ_{prior} and hide the dependency on other constants in $\mathcal{O}(\cdot)$ notation. Let $\widehat{\mathbf{w}}$ denote the output of Algorithm 1. By setting $\varepsilon = 0.5$ and letting $c = O(\tau)$ to be an arbitrary small number, Theorem 3.2 yields the following:

Corollary 3.3 (Sample Complexity). *Under the same conditions as Theorem 3.2, by running Algorithm 1 for minimizing $\mathcal{L}_{\mathcal{D}}(\mathbf{w})$ with a number of iterations (i.e., number of training examples) T, if it holds that,*

$$T \geq \mathcal{O}\left(d\kappa^4\left(\log\frac{1}{\epsilon_{\text{prior}}}\log\log\frac{1}{\epsilon_{\text{prior}}} + \log\frac{1}{\delta}\right)\right)$$

where $\kappa = \beta/\alpha$ denotes the condition number of the loss function and d is the dimension of data, then with a probability $1-\delta$, $\widehat{\mathbf{w}}$ attains a risk of $O(\epsilon_{\text{prior}})$, i.e., $\mathcal{L}_{\mathcal{D}}(\widehat{\mathbf{w}}) \leq (1+c)\epsilon_{\text{prior}}$.

As an example of a concrete problem that may be put into the setting of the present work is the regression problem with squared loss. It is easy to show that average square loss function is Lipschitz continuous with a Lipschitz constant $\beta = \lambda_{\max}(X^\top X)$ which denotes the largest eigenvalue of matrix $X^\top X$ where X is the data matrix. The strong convexity is guaranteed as long as the population data covariance matrix is not rank-deficient and its minimum eigenvalue is lower bounded by a constant $\alpha > 0$. For this problem, the optimal minimax sample complexity is known to be $O(\frac{1}{\epsilon})$, but as it implies from Corollary 3.3, by the knowledge of target risk ϵ_{prior}, it is possible to reduce the sample complexity to $O(\log(1/\epsilon_{\text{prior}}))$.

Remark 3.4. *It is indeed remarkable that the sample complexity of Theorem 3.2 has $\kappa^4 = (\beta/\alpha)^4$ dependency on the condition number of the loss function, which is worse than the $\sqrt{\beta/\alpha}$ dependency in the lower bound in (3.1). Also, the explicit dependency of sample complexity on dimension d makes the proposed algorithm inappropriate for non-parametric settings.*

3.4 Analysis of Sample Complexity

Now we turn to proving the main theorem. The proof will be given in a series of lemmas and theorems where the proof of few are given in the Section 3.5. The proof makes use of the Bernstein inequality for martingales, idea of peeling process, self-bounding property of smooth loss functions, standard analysis of stochastic optimization, and novel ideas to derive the claimed sample complexity for the proposed algorithm.

The proof of Theorem 3.2 is by induction and we start with the key step given in the following theorem.

Theorem 3.5. *Assume $\epsilon_{\text{prior}} \geq \epsilon_{\text{opt}}$. For a fixed stage k, if $\|\widehat{\mathbf{w}}_k - \mathbf{w}_*\| \leq \Delta_k$, then, with a probability $1-\delta$, we have*

$$\|\widehat{\mathbf{w}}_{k+1} - \mathbf{w}_*\|^2 \leq a\Delta_k^2 + b\epsilon_{\text{prior}}$$

where

$$a = \frac{2}{\alpha T_1}\left(2\xi\beta\sqrt{T_1} + \left[\xi^3\beta d + 2\xi\beta\sqrt{d}\right]\ln\frac{s}{\delta}\right), \quad b = \frac{8}{\alpha\xi} \tag{3.8}$$

and s is given in (3.7), provided that $\xi \geq 16\beta/\alpha$ and $\eta = 1/(2\xi\beta\sqrt{T_1})$ hold.

Taking this statement as given for the moment, we proceed with the proof of Theorem 3.2, returning later to establish the claim stated in Theorem 3.5.

Proof of Theorem 3.2. By setting a and b in (3.8) in Theorem 3.5 as $a \leq \varepsilon$ and $b \leq 2\tau/\beta$, we have $\xi \geq 4\beta/(\alpha\tau)$ and

$$T_1 \leq \frac{2}{\alpha\varepsilon}\left(2\xi\beta\sqrt{T_1} + \left[\xi^3\beta d + 2\xi\beta\sqrt{d}\right]\ln\frac{s}{\delta}\right)$$

implying that

$$T_1 \geq 4 \max \left\{ \frac{\xi^3 \beta d + 2\xi\beta\sqrt{d}}{\varepsilon\alpha} \ln \frac{s}{\delta}, \frac{16\xi^2\beta^2}{\alpha^2\varepsilon^2} \right\}.$$

Thus, using Theorem 3.5 and the definition of ξ and T_1, we have, with a probability $1-\delta$,

$$\Delta_{k+1}^2 \leq \varepsilon\Delta_k^2 + \frac{2\tau}{\beta}\epsilon_{\text{prior}}.$$

After m stages, with a probability $1-m\delta$, we have

$$\Delta_{m+1}^2 \leq \varepsilon^m \Delta_1^2 + \frac{2\tau}{\beta}\epsilon_{\text{prior}} \sum_{i=0}^{m-1} \varepsilon^i \leq \varepsilon^m \Delta_1^2 + \frac{2\tau}{\beta(1-\varepsilon)}\epsilon_{\text{prior}}.$$

By the β-smoothness of $\mathcal{L}_{\mathcal{D}}(\mathbf{w})$, it implies that

$$\begin{aligned} \mathcal{L}_{\mathcal{D}}(\widehat{\mathbf{w}}_{m+1}) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}_*) \leq \frac{\beta}{2}\|\widehat{\mathbf{w}}_{m+1} - \mathbf{w}_*\|^2 \quad &\leq \quad \frac{\beta}{2}\varepsilon^m\Delta_1^2 + \frac{\tau}{1-\varepsilon}\epsilon_{\text{prior}}, \\ &\leq \quad \frac{\beta R^2}{2}\varepsilon^m + \frac{\tau}{1-\varepsilon}\epsilon_{\text{prior}}, \end{aligned}$$

where the last inequality follows from $\Delta_1 \leq R$. The bound stated in the theorem follows the assumption that $\mathcal{L}_{\mathcal{D}}(\mathbf{w}_*) = \epsilon_{\text{opt}} \leq \epsilon_{\text{prior}}$. □

We now turn to proving Theorem 3.5. To bound $\|\widehat{\mathbf{w}}_{k+1} - \mathbf{w}_*\|$ in terms of Δ_k, we start with the standard analysis of online learning. In particular, from the strong convexity assumption of $\mathcal{L}_{\mathcal{D}}(\mathbf{w})$ and updating rule (3.5) we have,

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}_*) &\leq \langle \nabla \mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t), \mathbf{w}_k^t - \mathbf{w}_* \rangle - \frac{\alpha}{2}\|\mathbf{w}_k^t - \mathbf{w}_*\|^2 \\
&= \langle \mathbf{v}_k^t, \mathbf{w}_k^t - \mathbf{w}_* \rangle + \langle \nabla \mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t) - \mathbf{v}_k^t, \mathbf{w}_k^t - \mathbf{w}_* \rangle - \frac{\alpha}{2}\|\mathbf{w}_t - \mathbf{w}_*\|^2 \\
&\leq \frac{\|\mathbf{w}_k^t - \mathbf{w}_*\|^2 - \|\mathbf{w}_k^{t+1} - \mathbf{w}_*\|^2}{2\eta} + \frac{\eta d}{2}\gamma_k^2 \\
&\quad + \underbrace{\langle \nabla \mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t) - \mathbf{v}_k^t, \mathbf{w}_k^t - \mathbf{w}_* \rangle}_{\triangleq v_k^t} - \frac{\alpha}{2}\|\mathbf{w}_t - \mathbf{w}_*\|^2, \qquad (3.9)
\end{aligned}$$

where the last step follows from $\|\mathbf{v}_k^t\| \leq \gamma_k\sqrt{d}$. By adding all the inequalities of (7.1) at stage k, we have

$$\begin{aligned}
\sum_{t=1}^{T_1} \mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}_*) &\leq \frac{\|\widehat{\mathbf{w}}_k - \mathbf{w}_*\|^2}{2\eta} + \frac{d\eta}{2}\gamma_k^2 T_1 + \sum_{t=1}^{T_1} v_k^t - \frac{\alpha}{2}\sum_{t=1}^{T_1}\|\mathbf{w}_t - \mathbf{w}_*\|^2 \\
&\leq \frac{\Delta_k^2}{2\eta} + \frac{d\eta}{2}\gamma_k^2 T_1 + V_k - \frac{\alpha}{2} W_k, \qquad (3.10)
\end{aligned}$$

where V_k and W_k are defined as $V_k = \sum_{t=1}^{T_1} v_k^t$ and $W_k = \sum_{t=1}^{T_1} \|\mathbf{w}_k^t - \mathbf{w}_*\|^2$, respectively. In order to bound V_k, using the fact that $\nabla \mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t) = \mathbb{E}_t[\mathbf{g}_k^t]$, we rewrite V_k as

$$\begin{aligned}
V_k &= \sum_{t=1}^{T_1} \underbrace{\langle -\mathbf{v}_k^t + \mathbb{E}_t[\mathbf{v}_k^t], \mathbf{w}_k^t - \mathbf{w}_* \rangle}_{\triangleq d_k^t} + \sum_{t=1}^{T_1} \underbrace{\langle \mathbb{E}_t\left[\mathbf{g}_k^t\right] - \mathbb{E}_t[\mathbf{v}_k^t], \mathbf{w}_k^t - \mathbf{w}_* \rangle}_{\triangleq e_k^t} \\
&= D_k + E_k,
\end{aligned}$$

where $D_k = \sum_{t=1}^{T_1} d_k^t$ and $E_k = \sum_{t=1}^{T_1} e_k^t$ which represent the variance and bias of the clipped gradient $\mathbf{v}_k^t$, respectively. We now turn to separately upper bound each term.

The following lemma bounds the variance term D_k using the Bernstein inequality for martingale. Its proof can be found in Section 3.5.

Lemma 3.6. *For any $L > 0$ and $\mu > 0$, we have*

$$\Pr\left(W_k \leq \frac{\epsilon_{\text{prior}} T_1}{2\mu\beta}\right) + \Pr\left(D_k \leq \frac{1}{L} W_k + \left(L\gamma_k^2 d + \gamma_k \Delta_k \sqrt{d}\right) \ln \frac{s}{\delta}\right) \geq 1 - \delta$$

where s is given by

$$s = \left\lceil \log_2 \frac{8\beta\mu R^2}{\epsilon_{\text{prior}}} \right\rceil.$$

The following lemma bounds E_k using the self-bounding property of smooth functions and the proof is deferred to Section 3.5.2.

Lemma 3.7.

$$E_k \leq \frac{4T_1}{\xi} \epsilon_{\text{opt}} + \frac{4\beta}{\xi} W_k \leq \frac{4T_1}{\xi} \epsilon_{\text{prior}} + \frac{4\beta}{\xi} W_k.$$

Note that without the knowledge of ϵ_{prior}, we have to bound ϵ_{opt} by $\Omega(1)$, resulting in a very loose bound for the bias term E_k. It is knowledge of the target expected risk ϵ_{prior} that allows us to come up with a significantly more accurate bound for the bias term E_k, which consequentially leads to a geometric convergence rate.

We now proceed to bound $\sum_{t=1}^{T_1} \mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}_*)$ using the two bounds in Lemma 3.6 and 3.7. To this end, based on the result obtained in Lemma 3.6, we consider two scenarios. In the first scenario, we assume

$$W_k \leq \frac{\epsilon_{\text{prior}} T_1}{2\mu\beta} \tag{3.11}$$

In this case, we have

$$\sum_{t=1}^{T_1} \mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}_*) \leq \frac{\beta}{2} W_k \leq \frac{\epsilon_{\text{prior}}}{2\mu} T_1. \tag{3.12}$$

In the second scenario, we assume

$$D_k \leq \frac{1}{L} W_T + \left(L\gamma_k^2 d + \gamma_k \Delta_k \sqrt{d} \right) \ln \frac{s}{\delta}. \tag{3.13}$$

In this case, by combining the bounds for D_k and E_k and setting $L = \frac{\xi}{4\beta}$, we have

$$\begin{aligned} V_k &\leq \frac{8\beta}{\xi} W_k + \left(\frac{\xi d}{4\beta} \gamma_k^2 + \gamma_k \Delta_k \sqrt{d} \right) \ln \frac{s}{\delta} + \frac{4T_1}{\xi} \epsilon_{\text{prior}} \\ &= \frac{8\beta}{\xi} W_k + \left(\xi^3 \beta d + 2\xi\beta\sqrt{d} \right) \Delta_k^2 \ln \frac{s}{\delta} + \frac{4T_1}{\xi} \epsilon_{\text{prior}}, \end{aligned}$$

where the last equality follows from the fact $\gamma_k = 2\xi\beta\Delta_k$. If we choose ξ such that $\frac{8\beta}{\xi} \leq \frac{\alpha}{2}$ or $\xi \geq \frac{16\beta}{\alpha} > 1$ holds, we get

$$V_k \leq \frac{\alpha}{2} W_k + \left(\xi^3 \beta d + 2\xi\beta\sqrt{d} \right) \Delta_k^2 \ln \frac{s}{\delta} + \frac{4T_1}{\xi} \epsilon_{\text{prior}}$$

Substituting the above bound for V_k into the inequality of (3.10), we have

$$\sum_{t=1}^{T_1} \mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}_*) \leq \frac{\Delta_k^2}{2\eta} + \frac{\eta}{2} \gamma_k^2 T_1 + \left(\xi^3 \beta d + 2\xi\beta\sqrt{d} \right) \Delta_k^2 \ln \frac{s}{\delta} + \frac{4T_1}{\xi} \epsilon_{\text{prior}}$$

By choosing η as $\eta = \frac{\Delta_k}{\gamma_k \sqrt{T_1}} = \frac{1}{2\xi\beta\sqrt{T_1}}$, we have

$$\mathcal{L}_{\mathcal{D}}(\widehat{\mathbf{w}}_{k+1}) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}_*) \leq \frac{1}{T_1} \left(2\xi\beta\sqrt{T_1} + \left[\xi^3 \beta d + 2\xi\beta\sqrt{d} \right] \ln \frac{s}{\delta} \right) \Delta_k^2 + \frac{4}{\xi} \epsilon_{\text{prior}}. \tag{3.14}$$

By combining the bounds in (3.12) and (3.14), under the assumption that at least one of the two conditions in (3.11) and (3.13) is true, by setting $\mu = B/8$, we have

$$\mathcal{L}_{\mathcal{D}}(\widehat{\mathbf{w}}_{k+1}) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}_*) \leq \frac{1}{T_1}\left(2\xi\beta\sqrt{T_1} + \left[\xi^3\beta d + 2\xi\beta\sqrt{d}\right]\ln\frac{s}{\delta}\right)\Delta_k^2 + \frac{4}{\xi}\epsilon_{\text{prior}},$$

implying

$$\|\widehat{\mathbf{w}}_{k+1} - \mathbf{w}_*\| \leq \frac{2}{\alpha T_1}\left(2\xi\beta\sqrt{T_1} + \left[\xi^3\beta d + 2\xi\beta\sqrt{d}\right]\ln\frac{s}{\delta}\right)\Delta_k^2 + \frac{8}{\alpha\xi}\epsilon_{\text{prior}}.$$

We complete the proof by using Lemma 3.6, which states that the probability for either of the two conditions hold is no less than $1-\delta$.

3.5 Proofs of Sample Complexity

3.5.1 Proof of Lemma 3.6

The proof is based on the Bernstein's inequality for martingales which can be found in Lemma A.25. Define martingale difference $d_k^t = \left\langle \mathbf{w}_k^t - \mathbf{w}_*, \mathbb{E}_t[\mathbf{v}_k^t] - \mathbf{v}_k^t \right\rangle$ and martingale $D_k = \sum_{t=1}^{T_1} d_k^t$. Let Σ_T^2 denote the conditional variance as

$$\begin{aligned}\Sigma_T^2 = \sum_{t=1}^{T_1}\mathbb{E}_t\left[(d_k^t)^2\right] &\leq \sum_{t=1}^{T_1}\mathbb{E}_t\left[\left\|\mathbb{E}_t[\mathbf{v}_k^t] - \mathbf{v}_k^t\right\|^2\right]\|\mathbf{w}_k^t - \mathbf{w}_*\|^2 \\ &\leq \sum_{t=1}^{T} d\gamma_k^2\|\mathbf{w}_k^t - \mathbf{w}\|^2 = d\gamma_k^2 W_k,\end{aligned}$$

which follows from the Cauchy's Inequality and the definition of clipping.

Define $M = \max_t |d_k^t| \leq 2\sqrt{d}\gamma_k\Delta_k$. To prove the inequality in Lemma 3.6, we follow the idea of peeling process [92]. Since $W_k \leq 4R^2T_1$, we have

$$
\begin{aligned}
&\Pr\left(D_k \geq 2\gamma_k\sqrt{W_k d\rho} + \sqrt{2}M\rho/3\right) \\
&= \Pr\left(D_k \geq 2\gamma_k\sqrt{W_k d\rho} + \sqrt{2}M\rho/3, W_k \leq 4R^2T_1\right) \\
&= \Pr\left(D_k \geq 2\gamma_k\sqrt{W_k d\rho} + \sqrt{2}M\rho/3, \Sigma_T^2 \leq \gamma_k^2 dW_k, W_k \leq 4R^2T_1\right) \\
&\leq \Pr\left(D_k \geq 2\gamma_k\sqrt{W_k d\rho} + \sqrt{2}M\rho/3, \Sigma_T^2 \leq \gamma_k^2 dW_k, W_k \leq \epsilon_{\text{prior}}T_1/(2\beta\mu)\right) \\
&+ \sum_{i=1}^{s}\Pr\left(D_k \geq 2\gamma_k\sqrt{W_k d\rho} + \sqrt{2}M\rho/3, \Sigma_T^2 \leq \gamma_k^2 dW_k, \frac{\epsilon_{\text{prior}}2^{i-1}T_1}{2\beta\mu} < W_k \leq \frac{\epsilon_{\text{prior}}2^{i}T_1}{2\beta\mu}\right) \\
&\leq \Pr\left(W_k \leq \frac{\epsilon_{\text{prior}}T_1}{2\beta\mu}\right) \\
&+ \sum_{i=1}^{s}\Pr\left(D_k \geq \sqrt{\frac{\epsilon_{\text{prior}}2^{i+1}T_1\gamma_k^2 d}{2\beta\mu}\rho} + \frac{\sqrt{2}}{3}M\rho, \Sigma_T^2 \leq \frac{\epsilon_{\text{prior}}2^{i}T_1\gamma_k^2 d}{2\beta\mu}\right) \\
&\leq \Pr\left(W_k \leq \frac{\epsilon_{\text{prior}}T_1}{2\beta\mu}\right) + se^{-\rho},
\end{aligned}
$$

where s is given by

$$s = \left\lceil \log_2 \frac{8\beta\mu R^2}{\epsilon_{\text{prior}}} \right\rceil .$$

The last step follows the Bernstein inequality for martingales. We complete the proof by setting $\rho = \ln(s/\delta)$ and using the fact that

$$2\gamma_k\sqrt{W_k\rho d} \leq \frac{1}{L}W_k + \gamma_k^2\rho dL.$$

3.5.2 Proof of Lemma 3.7

To bound E_k, we need the following two lemmas. The first lemma bounds the deviation of the expected value of a clipped random variable from the original variable, in terms of its variance (Lemma A.2 from [74]).

Lemma 3.8. *Let X be a random variable, let $\widetilde{X} = \mathrm{clip(X, C)}$ and assume that $|\mathbb{E}[X]| \leq C/2$ for some $C > 0$. Then*

$$|\mathbb{E}[\widetilde{X}] - \mathbb{E}[X]| \leq \frac{2}{C}\,|\mathrm{Var[X]}|$$

Another key observation used for bounding E_k is the fact that for any non-negative β-smooth convex function, we have the following self-bounding property. We note that this self-bounding property has been used in [138] to get better (optimistic) rates of convergence for non-negative smooth losses.

Lemma 3.9. *For any β-smooth non-negative function $f : \mathbb{R} \to \mathbb{R}$, we have $|f'(w)| \leq \sqrt{4\beta f(w)}$*

Proof. See Appendix **??** □

Proof of Lemma 3.7. To apply the above lemmas, we write e_k^t as

$$e_k^t \;=\; \sum_{i=1}^{d} \mathbb{E}_t\left[\ell'(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t)[\mathbf{x}_k^t]_i - \mathrm{clip}\left(\gamma_k, \ell'(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t)[\mathbf{x}_k^t]_i\right)\right][\mathbf{w}_k^t - \mathbf{w}_*]_i$$

In order to apply Lemma 3.8, we check if the following condition holds

$$\gamma_k \geq 2\left|\mathbb{E}_t\left[\ell'\left(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t\right)[\mathbf{x}_k^t]_i\right]\right| \tag{3.15}$$

Since

$$\begin{aligned}
&\left|\mathbb{E}_t\left[\ell'\left(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t\right)[\mathbf{x}_k^t]_i\right]\right| \\
\leq\;& \left|\mathbb{E}_t\left[\left\{\ell'\left(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t\right) - \ell'\left(\langle \mathbf{w}_*, \mathbf{x}_k^t\rangle, y_t\right)\right\}[\mathbf{x}_k^t]_i\right]\right| + \left|\mathbb{E}_t\left[\ell'\left(\langle \mathbf{w}_*, \mathbf{x}_k^t\rangle, y_t\right)[\mathbf{x}_k^t]_i\right]\right| \\
\leq\;& \beta\|\mathbf{w}_k^t - \mathbf{w}_*\| \leq \beta\Delta_k
\end{aligned}$$

where the last inequality follows from $\mathbb{E}_t\left[\ell'\left(\langle \mathbf{w}_*, \mathbf{x}_k^t\rangle, y_t\right)[\mathbf{x}_k^t]_i\right] = 0$ since $\mathbf{w}_*$ is the minimizer of $\mathcal{L}_{\mathcal{D}}(\mathbf{w})$, we thus have

$$\gamma_k = 2\xi\beta\Delta_k \geq 2\beta\Delta_k \geq 2\left|\mathbb{E}_t\left[\ell'\left(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t\right)[\mathbf{x}_k^t]_i\right]\right|$$

where $\xi \geq 1$, implying that the condition in (3.15) holds. Thus, using Lemma 3.8, we have

$$\begin{aligned} e_k^t &\leq \sum_{i=1}^{d} \left|[\mathbf{w}_k^t - \mathbf{w}_*]_i\right| \frac{1}{\gamma_k}\mathbb{E}_t\left[\left(\ell'(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t)[\mathbf{x}_k^t]_i\right)^2\right] \\ &\leq \frac{2\|\mathbf{w}_k^t - \mathbf{w}_*\|_\infty}{\gamma_k}\mathbb{E}_t\left[\left(\ell'(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t)\right)^2\right] \end{aligned}$$

Using Lemma 3.9 to upper bound the right hand side, we further simplify the above bound for e_k^t as

$$\begin{aligned} e_k^t &\leq \frac{8\beta\|\mathbf{w}_k^t - \mathbf{w}_*\|_\infty}{\gamma_k}\mathbb{E}_t\left[\ell\left(\langle \mathbf{w}_k^t, \mathbf{x}_k^t\rangle, y_t\right)\right] \\ &= \frac{8\beta\|\mathbf{w}_k^t - \mathbf{w}_*\|_\infty}{\gamma_k}\mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t) \\ &\leq \frac{8\beta\Delta_k}{\gamma_k}\mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t) \\ &= \frac{4}{\xi}\mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t) \end{aligned}$$

where the second inequality follows from $\|\mathbf{w}_k^t - \mathbf{w}_*\|_\infty \leq \|\mathbf{w}_k^t - \mathbf{w}_*\| \leq \Delta_k$. Therefore we obtain

$$
\begin{aligned}
E_k = \sum_{t=1}^{T_1} e_k^t \leq \frac{4}{\xi}\sum_{t=1}^{T_1}\mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t) &= \frac{4}{\xi}\sum_{t=1}^{T_1}\mathcal{L}_{\mathcal{D}}(\mathbf{w}_*) + \frac{4}{\xi}\sum_{t=1}^{T_1}\mathcal{L}_{\mathcal{D}}(\mathbf{w}_k^t) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}_*) \\
&\leq \frac{4T_1}{\xi}\mathcal{L}_{\mathcal{D}}(\mathbf{w}_*) + \frac{4\beta}{\xi}\sum_{t=1}^{T_1}\|\mathbf{w}_k^t - \mathbf{w}_*\|^2 \\
&= \frac{4T_1}{\xi}\mathcal{L}_{\mathcal{D}}(\mathbf{w}_*) + \frac{4\beta}{\xi}W_k,
\end{aligned}
$$

where the second inequality follows from the smoothness assumption of $\mathcal{L}_{\mathcal{D}}(\mathbf{w})$. □

3.6 Summary

In this chapter, we have studied the sample complexity of passive learning when the target expected risk is given to the learner as prior knowledge. The crucial fact about target risk assumption is that, it can be fully exploited by the learning algorithm and stands in contrast to most common types of prior knowledges that usually enter into the generalization bounds and are often perceived as a rather crude way to incorporate such assumptions. We showed that by explicitly employing the target risk ϵ_{prior} in a properly designed stochastic optimization algorithm, it is possible to attain the given target risk ϵ_{prior} with a logarithmic sample complexity $\log\left(\frac{1}{\epsilon_{\text{prior}}}\right)$, under the assumption that the loss function is both strongly convex and smooth.

There are various directions for future research. The current study is restricted to the parametric setting where the hypothesis space is of finite dimension. It would be interesting to see how to achieve a logarithmic sample complexity in a non-parametric setting where hypotheses lie in a functional space of infinite dimension. Evidently, it is impossible to extend the current algorithm for the non-parametric setting; therefore additional analysis tools are needed to address the challenge of infinite dimension arising from the non-parametric setting. It is also an interesting problem to relate target risk assumption we made here to the low noise margin condition which is often made in active learning for binary classification since both settings appear to share the same sample complexity. However it is currently unclear how to derive a connection between these two settings. We believe this issue is worthy of further exploration and leave it as an open problem.

3.7 Bibliographic Notes

Sample complexity of passive learning is well established and goes back to early works in the learning theory where the lower bounds $\Omega\left(\frac{1}{\epsilon}(\log\frac{1}{\epsilon}+\log\frac{1}{\delta})\right)$ and $\Omega\left(\frac{1}{\epsilon^2}(\log\frac{1}{\epsilon}+\log\frac{1}{\delta})\right)$ were obtained in classic PAC and general agnostic PAC settings, respectively [52, 26, 7]. There has been an upsurge of interest over the last decade in finding tight upper bounds on the sample complexity by utilizing prior knowledge on the analytical properties of the loss function, that led to stronger generalization bounds in agnostic PAC setting. In [95] *fast* rates obtained for squared loss, exploiting the strong convexity of this loss function, which only holds under pseudo-dimensionality assumption. With the recent development in online strongly convex optimization [68], fast rates approaching $\mathcal{O}(\frac{1}{\epsilon}\log\frac{1}{\delta})$ for convex Lipschitz strongly convex loss functions has been obtained in [140, 82]. For smooth non-negative loss functions, [138] improved the sample complexity to *optimistic* rates

$$\mathcal{O}\left(\frac{1}{\epsilon}\left(\frac{\epsilon_{\text{opt}}+\epsilon}{\epsilon}\right)\left(\log^3\frac{1}{\epsilon}+\log\frac{1}{\delta}\right)\right)$$

for non-parametric learning using the notion of local Rademacher complexity [13], where ϵ_{opt} is the optimal risk.

The proposed ClippedSGD algorithm is related to the recent studies that examined the learnability from the viewpoint of stochastic convex optimization. In [139, 133], the authors presented learning problems that are learnable by stochastic convex optimization but not by empirical risk minimization (ERM). Our work follows this line of research. The proposed algorithm achieves the sample complexity of $O(d \log(1/\epsilon_{\text{prior}}))$ by explicitly incorporating the target expected risk ϵ_{prior} into the stochastic convex optimization algorithm. It is however difficult to incorporate such knowledge into the framework of ERM. Furthermore, it is worth noting that in [127, 139, 126, 20], the authors explored the connection between online optimization and statistical learning in the opposite direction. This was done by exploring the complexity measures developed in statistical learning for the learnability of online learning. We note that our work does not contradict the lower bound in [138] because a *feasible* target risk ϵ_{prior} is given in our learning setup and is fully exploited by the proposed algorithm. Knowing that the target risk ϵ_{prior} is feasible makes it possible to improve the sample complexity from $\mathcal{O}(1/\epsilon_{\text{prior}})$ to $\mathcal{O}(\log(1/\epsilon_{\text{prior}}))$. We also note that although the logarithmic sample complexity is known for active learning [66, 12], we are unaware of any existing passive learning algorithm that is able to achieve a logarithmic sample complexity by incorporating any kind of prior knowledge.

The proposed algorithm is also closely related to the recent works that stated $O(1/n)$ is the optimal convergence rate for stochastic optimization when the objective function is strongly convex [76, 72, 125]. In contrast, the proposed algorithm is able to achieve a geometric convergence rate for a target optimization error. Similar to the previous argument, our result does not contradict the lower bound given in [72] because of the knowledge of a feasible optimization error. Moreover, in contrast to the multistage algorithm in [72] where the size of stages increases exponentially, in our algorithm, the size of each stage is fixed to be a constant.

Chapter 4

Statistical Consistency of Smoothed Hinge Loss

In Chapter 2 we discussed that convex surrogates of the 0-1 loss are highly preferred because of the computational and theoretical virtues that convexity brings in and most prominent practical methods studied in machine learning make significant use of convexity. This is of more importance if we consider smooth surrogates as witnessed by the fact that the smoothness is further beneficial both computationally- by attaining an *optimal* convergence rate for optimization, and in a statistical sense- by providing an improved *optimistic* rate for generalization bound.

This chapter concerns itself with the statistical consistency of smooth convex surrogates. The statistical consistency finds general quantitative relationships between the excess risk errors associated with convex and those associated with 0-1 loss. Consistency results provide reassurance that optimizing a surrogate does not ultimately hinder the search for a function that achieves the binary excess risk, and thus allow such a search to proceed within the scope of computationally efficient algorithms. Statistical consistency of surrogates under conditions such as convexity is a well studied problem in learning community and quantitative relationships between binary risk and convex excess risk has been established. In this chapter we investigate the smoothness property from the viewpoint of statistical consistency and show how it affects the binary excess risk. We show that in contrast to optimization and generalization errors that favor the choice of smooth surrogate loss, the smoothness of loss function may degrade the binary excess risk. Motivated by this negative result, we provide a unified analysis that integrates optimization error, generalization bound, and the error in translating convex excess risk into a binary excess risk when examining the impact of smoothness on the binary excess risk. We show that under favorable conditions appropriate choice of smooth convex loss will result in a binary excess risk that is better than $O(1/\sqrt{n})$.

The reminder of this paper is organized as follows. In Section 4.1 we set up notation and describe the setting. Section 4.2 briefly discusses the classification-calibrated convex surrogate losses on which our analysis relies. We derive the ψ-transform for smoothed hinge loss and elaborate its binary excess risk in Section 4.3. Section 4.4 provides a unified analysis of three types of errors and derives conditions in terms of smoothness to obtain better rates for the binary excess risk. The omitted proofs are included in Section 4.5. Section 4.6 concludes the paper.

4.1 Motivation

Let $\mathcal{S} = ((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_n, y_n))$ be a set of i.i.d. samples drawn from an unknown distribution $\mathcal{D}$ over $\Xi = \mathcal{X} \times \{-1, +1\}$, where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ is an instance and $y_i \in \{-1, +1\}$ is the binary class assignment for $\mathbf{x}_i$. Let $\kappa(\cdot, \cdot)$ be an universal kernel and let $\mathcal{H}_\kappa$ be the Reproducing Kernel Hilbert Space (RKHS) endowed with kernel $\kappa(\cdot, \cdot)$. According to [157], $\mathcal{H}_\kappa$ is a rich function space whose closure includes all the smooth functions. We consider predictors from $\mathcal{H}_\kappa$ with bounded norm to form the measurable function class $\mathcal{H} = \{h \in \mathcal{H}_\kappa : \|h\|_{\mathcal{H}_\kappa} \leq B\}$.

For a function $h : \mathcal{X} \mapsto \mathbb{R}$, the risk of h is defined as:

$$\mathcal{L}_{\mathcal{D}}(h) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\mathbb{I}[yh(\mathbf{x}) \leq 0]\right] = \mathbb{P}\left[yh(\mathbf{x}) \leq 0\right].$$

Let h_* be the optimal classifier that attains the minimum risk, i.e.

$$h_* = \arg\min_{h} \mathbb{P}\left[yh(\mathbf{x}) \leq 0\right].$$

We assume $h_* \in \mathcal{H}_\kappa$ with $\|h_*\|_{\mathcal{H}_\kappa} \leq B$. This boundedness condition is satisfied for any RKHS with a bounded kernel (i.e. $\sup_{\mathbf{x}\in\mathcal{X}} \kappa(\mathbf{x},\mathbf{x}) \leq B$). Henceforth, let $\mathcal{L}_{\mathcal{D}}^*$ stand for the minimum achievable risk by the optimal classifier h_*, i.e., $\mathcal{L}_{\mathcal{D}}^* = \mathcal{L}_{\mathcal{D}}(h_*)$. Define the *binary excess risk* for a prediction function $h \in \mathcal{H}$ as

$$\mathcal{E}(h) = \mathcal{L}_{\mathcal{D}}(h) - \mathcal{L}_{\mathcal{D}}^*.$$

Our goal is to efficiently learn a prediction function $h \in \mathcal{H}$ from the training examples in $\mathcal{S}$ that minimizes the binary excess risk $\mathcal{E}(h)$. Many studies of binary excess risk assume that the optimal classifier $h \in \mathcal{H}$ is learned by minimizing the empirical binary risk, $\min_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[y_i h(\mathbf{x}_i) \leq 0]$, an approach that is usually referred to as Empirical Risk Minimization (ERM) [145]. To understand the generalization performance of the classifier learned by ERM, it is important to have upper bounds on the excess risk of the empirical minimizer that hold with a high probability and that take into account complexity measures of classification functions. It is well known that, under certain conditions, direct empirical classification error minimization is consistent [145] and achieves a fast convergence rate under low noise situations [109].

One shortcoming of the ERM based approaches is that they need to minimize 0-1 loss, leading to non-convex optimization problems that are potentially NP-hard [1] [8, 75]. A common practice to circumvent this difficulty is to replace the indicator function $\mathbb{I}[\cdot \leq 0]$ with some convex loss $\phi(\cdot)$ and find the optimal solution by minimizing the convex surrogate loss. Examples of such surrogate loss functions for 0-1 loss include logit loss $\phi_{\log}(h;(\mathbf{x},y)) = \log(1+\exp(-yh(\mathbf{x})))$ in logistic regression [61], hinge loss $\phi_{\text{Hinge}}(h;(\mathbf{x},y)) = \max(0, 1-yh(\mathbf{x}))$ in support vector machine (SVM) [43] and exponential loss $\phi_{\exp}(h;(\mathbf{x},y)) = \exp(-yh(\mathbf{x}))$ in AdaBoost [58]. Given a convex surrogate loss function $\phi : \mathbb{R} \mapsto \mathbb{R}_+$ (e.g., hinge loss, exponential loss, or logistic loss) we define the risk with respect to the convex loss ϕ (convex risk or ϕ-risk) as

$$\mathcal{L}_{\mathcal{D}}^{\phi}(h) = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\phi(yh(\mathbf{x}))].$$

Similarly we define the *optimal* ϕ-risk as $\mathcal{L}_{\mathcal{D}}^{\phi,*} = \inf_{h\in\mathcal{H}} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}[\phi(yh(\mathbf{x}))]$. The *excess ϕ-risk* or *convex excess risk* of a classifier $h \in \mathcal{H}$ with respect to the convex surrogate loss $\phi(\cdot)$ is defined as

$$\mathcal{E}_{\phi}(h) = \mathcal{L}_{\mathcal{D}}^{\phi}(h) - \mathcal{L}_{\mathcal{D}}^{\phi,*}.$$

An important line of research in statistical learning theory focused on relating the convex excess risk $\mathcal{E}_{\phi}(h)$ to the binary excess risk $\mathcal{E}(h)$ that will be elaborated in next section.

It is known that under mild conditions, the classifier learned by minimizing the empirical loss of convex surrogate is consistent to the Bayes classifier [156, 101, 80, 97, 141, 14]. For

[1] We note that several works [84, 85] provide efficient algorithms for direct 0-1 empirical error minimization but under strong (unrealistic) assumptions on data distribution or label generation.

instance, it was shown in [14] that the necessary and sufficient condition for a convex loss $\phi(\cdot)$ to be consistent with the binary loss is that $\phi(\cdot)$ is differentiable at origin and $\phi'(0) < 0$. It was further established in the same work that the binary excessive risk can be upper bound by the convex excess risk through a ψ-transform that depends on the surrogate convex loss $\phi(\cdot)$.

Since the choice of convex surrogates could significantly affect the binary excess risk, in this chapter, we will investigate the impact of the smoothness of a convex loss function on the binary excess risk. This is motivated by the recent results that show the advantages of using smooth convex surrogates in reducing the optimization complexity and the generalization error bound. More specifically, [121, 143] show that a faster convergence rate (i.e., $O(1/T^2)$) can be achieved by first order methods when the objective function to be optimized is convex and smooth such as accelerated gradient descent method introduced in Chapter 2; in [138], the authors show that a smooth convex loss will lead to a better optimistic generalization error bound rooted in the self-bounding property of smooth losses (Lemma A.12). Given the positive news of using smooth convex surrogates, an open research question is how the smoothness of a convex surrogate will affect the binary excess risk. The answer to this question, as will be revealed later, is negative: the smoother the convex loss, the poorer approximation will be for the binary excess risk. Thus, the second contribution of this work is to integrate these results for smooth convex losses, and examine the overall effect of replacing 0-1 loss with a smooth convex loss when taking into account three sources of errors, i.e. the optimization error, the generalization error, and the error in translating the convex excess risk into the binary risk. As we will show, under favorable conditions, appropriate choice of smooth convex loss will result a binary excess risk better than $O(1/\sqrt{n})$.

4.2 Classification Calibration and Surrogate Risk Bounds

Although it is computationally convenient to minimize the empirical risk based on a convex surrogate, the ultimate goal of any classification method is to find a function $h \in \mathcal{H}_\kappa$ that minimizes the binary loss. Therefore, it is crucial to investigate the conditions which guarantee that if the ϕ-risk of h gets close to the optimal $\mathcal{L}_{\mathcal{D}}^{\phi,*}$, the binary risk of h will also approach the optimal binary risk $\mathcal{L}_{\mathcal{D}}^{*}$. This question has been an active trend in statistical learning theory over the last decade where the necessary and sufficient conditions have been established for relating the binary excess risk to a convex excess risk [156, 101, 80, 97, 141, 14].

In this chapter we follow the strategy introduced in [14] in order to relate the binary excess risk to the excess ϕ-risk. Their methodology, through the notion of *classification calibration*, allows us to find quantitative relationship between the excess risk associated with ϕ and the excess risk associated with 0-1 loss. It is established in [14] that the binary excessive risk can be bounded by the convex excess risk, based on the convex loss function ϕ, through a ψ-transform.

Definition 4.1. *Given a loss function $\phi : \mathbb{R} \mapsto [0, \infty)$, define the function $\psi : [0, 1] \mapsto [0, \infty)$ by*

$$\tilde{\psi}(z) = H^{-}\left(\frac{1+z}{2}\right) - H\left(\frac{1+z}{2}\right)$$

where

$$H^{-}(\eta) = \inf_{\alpha:\alpha(2\eta-1)\leq 0} \left(\eta\phi(\alpha) + (1-\eta)\phi(-\alpha)\right) \text{ and } H(\eta) = \inf_{\alpha\in\mathbb{R}} \left(\eta\phi(\alpha) + (1-\eta)\phi(-\alpha)\right).$$

The transform function $\psi : [0, 1] \mapsto [0, \infty)$ is defined to be the convex closure of $\tilde{\psi}$.

The following theorem from [14, Theorem 1] shows that the binary excess risk can be bounded by the convex excess risk using transform function $\psi : [0,1] \mapsto [0,\infty)$ that depends on the surrogate convex loss function.

Theorem 4.2. *For any non-negative loss function $\phi(\cdot)$, any measurable function $h \in \mathcal{H}$, and any probability distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$, there is a nondecreasing function $\psi : [0,1] \mapsto [0,\infty)$ that*

$$\psi(\mathcal{L}_{\mathcal{D}}(h) - \mathcal{L}_{\mathcal{D}}^{*}) \leq \mathcal{L}_{\mathcal{D}}^{\phi}(h) - \mathcal{L}_{\mathcal{D}}^{\phi,*} \tag{4.1}$$

holds. Here the minimization is taken over all measurable functions.

Definition 4.3. *A convex loss ϕ is classification-calibrated if, for any $\eta \neq 1/2$,*

$$H^{-}(\eta) > H(\eta).$$

This condition is essentially an extension of [156, Theorem 2.1] and can be viewed as a form of Fisher consistency that is appropriate for classification.

It has been shown in [14] that the necessary and sufficient condition for a convex loss $\phi(z)$ to be classification-calibrated is if it is differentiable at the origin and $\phi'(0) < 0$. In particular, for a certain convex function $\phi(\cdot)$, the ψ-transform can be computed by

$$\psi(z) = \inf_{\alpha z \leq 0} \left(\frac{1+z}{2}\phi(\alpha) + \frac{1-z}{2}\phi(-\alpha) \right) - \inf_{\alpha \in \mathbb{R}} \left(\frac{1+z}{2}\phi(\alpha) + \frac{1-z}{2}\phi(-\alpha) \right),$$

that can be further simplified as $\psi(z) = \phi(0) - H\left(\frac{1+z}{2}\right)$ when ϕ is classification-calibrated. Examples of ψ-transform for the convex surrogate functions of known practical algorithms mentioned before are as follows: (i) for hinge loss $\phi(\alpha) = \max(0, 1-\alpha)$, $\psi(z) = |z|$, (ii) for exponential loss $\phi(\alpha) = e^{-\alpha}$, $\psi(z) = 1 - \sqrt{1-z^2} \geq z^2/2$, and (iii) for truncated quadratic loss $\phi(\alpha) = [\max(0, 1-\alpha)]^2$, $\phi(z) = z^2$.

Remark 4.4. *We note that the inequality in (4.1) provides insufficient guidance on choosing appropriate loss function. A few brief comments are appropriate. First, it does not measure explicitly how the choice of the convex surrogate $\phi(\cdot)$ affects the excess risk $\mathcal{L}^{\phi}_{\mathcal{D}}(h) - \mathcal{L}^{\phi,*}_{\mathcal{D}}$. Second, it does not take into account the impact of loss function on optimization efficiency, an important issue for practitioners when dealing with big data. It is thus unclear, from Theorem 4.2, how to choose an appropriate loss function that could result in a small generalization error for the binary loss when the computational time is limited. In this chapter, we address these limitations by examining a family of convex losses that are constructed by smoothing the hinge loss function using different smoothing parameters. We study the binary excessive risk of the learned classification function by taking into account errors in optimization, generalization, and translation of convex excess risk into binary excess risk.*

4.3 Binary Excess Risk for Smoothed Hinge Loss

As stated before, to efficiently learn a prediction function $h \in \mathcal{H}$, we will replace the binary loss with a smooth convex loss. Since hinge loss is one of the most popular loss functions used in machine learning and is the loss of choice for classification problems in terms of the margin error [19], in this work, we will focus on the smoothed version of the hinge loss. Another advantage of using the hinge loss is that its ψ-transform is a linear function. Compared with the ψ-transforms of other popular convex loss functions (e.g. exponential loss and truncated square loss) that are mostly quadratic, using the hinge loss as convex surrogate will lead to a tighter bound for the binary excess risk.

The smoothed hinge loss considered in this chapter is defined as

$$\phi(z;\gamma) = \max_{\alpha \in [0,1]} \alpha(1-z) + \frac{1}{\gamma}\mathcal{R}(\alpha), \tag{4.2}$$

where $\mathcal{R}(\alpha) = -\alpha \log \alpha - (1-\alpha)\log(1-\alpha)$ and $\gamma > 0$ is the smoothing parameter. It is straightforward to verify that the loss function in (4.2) can be simplified as

$$\phi(z;\gamma) = \frac{1}{\gamma}\log(1 + \exp(\gamma(1-z))).$$

It is not immediately clear from Theorem 4.2 how the relationship between smooth convex excess risk $\mathcal{E}_{\phi}(\cdot)$ and binary excess risk is affected by the smoothness parameter γ. In addition, as discussed in [14], whereas conditions such as convexity and smoothness have natural relationship to optimization and generalization, it is not immediately obvious how properties such as convexity and smoothness of convex surrogate relates to statistical consequences. In what follows, we show that, indeed smoothness of loss function has a *negative* statistical consequence and can degrade the binary excess risk.

4.3.1 ψ-Transform for Smoothed Hinge Loss

The first step in our analysis is to derive the ψ-transform for the loss function defined in (4.2) as stated in the following theorem.

Theorem 4.5. *The ψ-transform of smoothed hinge loss with smoothing parameter γ is given* *by*

$$\psi(\eta;\gamma) = -\frac{1+\eta}{2\gamma}\log\left(\frac{1}{1+e^{\gamma}}\left[1+e^{\gamma}\frac{C_1}{1+\eta}\right]\right) - \frac{1-\eta}{2\gamma}\log\left(\frac{1}{1+e^{\gamma}}\left[1+e^{\gamma}\frac{C_2}{1-\eta}\right]\right)$$

where C_1 and C_2 are defined as $C_1 = -\eta e^{\gamma}+\sqrt{\eta^2 e^{2\gamma}+1-\eta^2}$ and $C_2 = \eta e^{\gamma}+\sqrt{\eta^2 e^{2\gamma}+1-\eta^2}$.

The ψ-transform given in Theorem 4.5 is too complicated to be useful. The theorem below provides a simpler bound for the ψ-transform in terms of the smoothness parameter γ.

Theorem 4.6. *For $\eta \in (-1,1)$, we have*

$$\psi(\eta;\gamma) \geq |\eta| - \frac{1}{\gamma}\log\frac{1}{|\eta|}.$$

Remark 4.7. *The bound obtained in Theorem 4.6 demonstrates that when γ approaches to infinity, the ψ-transform for smoothed hinge loss $\phi(\eta;\gamma)$ becomes $|\eta|$. According to [14], the ψ-transform for the hinge loss is $\psi(\eta) = |\eta|$. Therefore, this result is consistent with the ψ-transform for smoothed hinge loss, which is the limit of $\phi(z;\gamma)$ as γ approaches infinity.*

4.3.2 Bounding $\mathcal{E}(h)$ based on $\mathcal{E}_\phi(h)$

Based on the transform function $\psi(\cdot;\gamma)$ that is computed for smoothed hinge loss with smoothing parameter γ, we are now in the position to bound its corresponding binary excess risk $\mathcal{E}(h)$. Our main result in this section is the following theorem that shows how binary excess risk can be bounded by the excess ϕ-risk for smoothed hinge loss.

Theorem 4.8. *Consider any measurable function $h \in \mathcal{H}$ and the smoothed hinge loss $\phi(\cdot)$ with parameter γ defined in (4.2). Then, binary excess risk $\mathcal{E}(h)$ can be bounded by the* *smooth convex excess risk* $\mathcal{E}_\phi(h)$ *as*

$$\mathcal{E}(h) \leq \mathcal{E}_\phi(h) + \frac{\mathcal{E}_\phi(h)}{1+\gamma\mathcal{E}_\phi(h)} \log \frac{1}{\mathcal{E}_\phi(h)}.$$

Proof. Using the result from Theorem 4.2, we have $\mathcal{E}_\phi(h) \geq \psi(\mathcal{E}(h);\gamma)$ and therefore an immediate result from the ψ-transform for smoothed hinge loss that is obtained in Theorem 4.6 indicates

$$\mathcal{E}(h) + \frac{1}{\gamma} \log \mathcal{E}(h) \leq \mathcal{E}_\phi(h).$$

Define $\Delta = \mathcal{E}(h) - \mathcal{E}_\phi(h)$. We have

$$\Delta + \frac{1}{\gamma}\log(\Delta + \mathcal{E}_\phi(h)) = \Delta + \frac{1}{\gamma}\log \mathcal{E}_\phi(h) + \frac{1}{\gamma}\log\left(1 + \frac{\Delta}{\mathcal{E}_\phi(h)}\right) \leq 0.$$

Based on the $\log(1+x) \leq x$ inequality, the sufficient condition for the above inequality to hold is to have

$$\Delta + \frac{\Delta}{\gamma\mathcal{E}_\phi(h)} \leq \frac{1}{\gamma}\log\frac{1}{\mathcal{E}_\phi(h)}$$

and therefore

$$\Delta \leq \frac{\gamma^{-1}}{1+(\gamma\mathcal{E}_\phi(h))^{-1}} \log\frac{1}{\mathcal{E}_\phi(h)} = \frac{\mathcal{E}_\phi(h)}{1+\gamma\mathcal{E}_\phi(h)}\log\frac{1}{\mathcal{E}_\phi(h)}.$$

The final bound is obtained by substituting $\mathcal{E}(h) - \mathcal{E}_\phi(h)$ for Δ in the left hand side of above inequality. □

As indicated by Theorem 4.8, the smaller the smoothing parameter γ, the poorer the approximation is in bounding the binary excess $\mathcal{E}(h)$ with smooth convex excess risk $\mathcal{E}_\phi(h)$. On the other hand, the smoothness of loss function has been proven to be beneficial in terms of optimization error and generalization bound. The mixture of negative and positive results for using smooth convex surrogates motivates us to develop an integrated bound for binary excess risk that takes into account all types of errors. One of the main contributions of this work is to show that under favorable conditions, with appropriate choice of smoothing parameter, the smoothed hinge loss will result in a bound for the binary excess risk better than $O(1/\sqrt{n})$.

4.4 A Unified Analysis

Using the smoothed hinge loss, we define the convex loss for a prediction function $h \in \mathcal{H}$ as $\mathcal{L}_{\mathcal{D}}^{\phi}(h) = \mathbb{E}[\phi(yh(\mathbf{x}); \gamma)]$. Let h_{γ}^{*} be the optimal classifier that minimizes $\mathcal{L}_{\mathcal{D}}^{\phi}(h)$. Similar to the case of binary loss, we assume $h_{\gamma}^{*} \in \mathcal{H}_{\kappa}$ with $\|h_{\gamma}^{*}\| \leq B$. The *smooth convex excess risk* for a given prediction function $h \in \mathcal{H}$ is then given by $\mathcal{E}_{\phi}(h) = \mathcal{L}_{\mathcal{D}}^{\phi}(h) - \mathcal{L}_{\mathcal{D}}^{\phi}(h_{\gamma}^{*})$. Given the smooth convex loss $\phi(z; \gamma)$ in (4.2), we find the optimal classifier by minimizing the empirical convex loss, i.e. $\min_{h \in \mathcal{H}_{\kappa}, \|h\|_{\mathcal{H}_{\kappa}} \leq B} \mathcal{L}_{\mathcal{S}}^{\phi}(h)$, where the empirical convex loss $\mathcal{L}_{\mathcal{S}}^{\phi}(h)$ is given by

$$\mathcal{L}_{\mathcal{S}}^{\phi}(h) = \frac{1}{n} \sum_{i=1}^{n} \phi(y_i h(\mathbf{x}_i); \gamma). \tag{4.3}$$

Let $\widehat{h}$ be the solution learned from solving the empirical convex loss over training examples. There are three sources of errors that affect bounding the binary excess risk $\mathcal{E}(\widehat{h})$. First, since $\widehat{h}$ is obtained by numerically solving an optimization problem, the error in estimating the optimal solution, which we refer to as optimization error [2], will affect $\mathcal{E}(\widehat{h})$. Additionally,

[2] We note that in literature the error in estimating the optimal solution for empirical minimization is usually referred to as *e*stimation error. We emphasize it as optimization error because different convex surrogates could lead to very different iteration complexities and consequentially different optimization efficiency.

since the binary excess risk can be bounded by a nonlinear transform of the convex excess risk, both the bound for $\mathcal{E}_\phi(\widehat{h})$ and the error in approximating $\mathcal{E}(\widehat{h})$ with $\mathcal{E}_\phi(\widehat{h})$ will affect the final estimation of $\mathcal{E}(\widehat{h})$. We aim at investigating how the smoothing parameter γ affect all these three types of errors. As it is investigated in Theorem 4.8, a smaller smoothing parameter γ will result in a poorer approximation of $\mathcal{E}(\widehat{h})$. On the other hand, a smaller smoothing parameter γ will result in a smaller estimation error and a smaller bound for $\mathcal{E}_\phi(\widehat{h})$. Based on the understanding of how smoothing parameter γ affects the three errors, we identify the choice of γ that results in the best tradeoff between all three error and consequentially a binary excess risk $\mathcal{E}(\widehat{h})$ better than $O(1/\sqrt{n})$.

To investigate how the smoothing parameter γ affects the binary excess risk $\mathcal{E}(\widehat{h})$, we intend to unify three types of errors. The analysis is comprised of two components, i.e. bounding the binary excess risk $\mathcal{E}(h)$ by a smooth convex excess risk $\mathcal{E}_\phi(h)$ that has been established in Theorem 4.8 and bounding $\mathcal{E}_\phi(h)$ for a solution h that is suboptimal in minimizing the empirical convex loss $\mathcal{L}^\phi_\mathcal{S}(h)$ that is the focus of this section.

4.4.1 Bounding Smooth Excess Convex Risk $\mathcal{E}_\phi(h)$

We now turn to bounding the excess ϕ-risk $\mathcal{E}_\phi(h)$ for the smoothed hinge loss. To bound $\mathcal{E}_\phi(h)$ we need to consider two types of errors: optimization error due to the approximate optimization of the empirical ϕ-risk, and the generalization error bound for the empirical risk minimizer. After obtaining these two errors for smooth convex surrogates, we provide a unified bound on the excess ϕ-risk $\mathcal{E}_\phi(h)$ of empirical convex risk minimizer in terms of n.

We begin by bounding the error arising from solving the optimization problem numerically. One nice property of smoothed hinge loss function is that both its first order and second order derivatives are bounded, i.e.

$$|\phi'(z;\gamma)| = \left|\frac{\exp(\gamma(1-z)}{1+\exp(\gamma(1-z))}\right| \leq 1, \quad \phi''(z;\gamma) = \gamma\frac{\exp(\gamma(1-z))}{(1+\exp(\gamma(1-z)))^2} \leq \frac{\gamma}{4}.$$

Due to the smoothness of $\phi(z;\gamma)$, we can apply the accelerated optimization algorithm [121, 143] to achieve an $O(1/k^2)$ convergence rate for the optimization, where k is the number of iterations the optimization algorithm proceeds (see e.g., accelerated gradient descent algorithm in Subsection 2.3.2.2). More specifically, we will apply Algorithm 1 from [143] to solve the numerical optimization problem in (4.3) over the convex domain $\mathcal{H} = \{h \in \mathcal{H}_\kappa : \|h\|_{\mathcal{H}_\kappa} \leq B\}$ which results in the following updating rules at sth iteration:

$$\begin{aligned} g_s &= (1-\theta_s)h_s + \theta_s f_s \\ f_{s+1} &= \arg\min_{f\in\mathcal{H}} \left(\langle \nabla \mathcal{L}^{\phi}_{\mathcal{S}}(g_s), f - g_s\rangle + \frac{\theta_s}{2}\|f - f_s\|_{\mathcal{H}_\kappa} \right) \\ h_{s+1} &= (1-\theta_s)h_s + \theta_s f_{s+1}. \end{aligned} \tag{4.4}$$

The following theorem that follows immediately from [143, Corollary 1] and the fact $\phi''(z;\gamma) \leq \gamma/4$, bounds the optimization error for the optimization problem after k iterations.

Lemma 4.9. *Let $\widehat{h} = h_{k+1}$ be the solution obtained by running accelerated gradient descent method (i.e., updating rules in (4.4)) to solve the optimization problem in (4.3) after k iterations with $\theta_0 = 1$ and $\theta_k = 2/(k+2)$ for $k \geq 1$. We have*

$$\mathcal{L}^{\phi}_{\mathcal{S}}(\widehat{h}) \leq \min_{\|h\|_{\mathcal{H}_\kappa}\leq B} \mathcal{L}^{\phi}_{\mathcal{S}}(h) + \frac{\gamma B^2}{(k+2)^2}.$$

We now turn to understanding the generalization error for the smooth convex loss. There are many theoretical results giving upper bounds of the generalization error. However, a recent result [138] has showed that it is possible to obtain optimistic rates for generalization bound of smooth convex loss (in the sense that smooth losses yield better generalization bounds when the problem is easier), which are more appealing than the generalization of simple Lipschitz continuous losses. The following theorem from [138, Theorem 1] bounds the generalization error for any solution $h \in \mathcal{H}$ when the learning has been performed by a smooth convex surrogate $\phi(\cdot)$.

Lemma 4.10. *With a probability $1-\delta$, for any $\|h\|_{\mathcal{H}_\kappa} \leq B$, we have*

$$\begin{aligned} \mathcal{L}_{\mathcal{D}}^{\phi}(h) - \mathcal{L}_{\mathcal{S}}^{\phi}(h) &\leq K_1 \left(\frac{(B+\gamma B^2)t}{n} + \sqrt{\mathcal{L}_{\mathcal{S}}^{\phi}(h) \frac{(B+\gamma B^2)t}{n}} \right) \\ \mathcal{L}_{\mathcal{D}}^{\phi}(h) - \mathcal{L}_{\mathcal{S}}^{\phi}(h) &\leq K_2 \left(\frac{(B+\gamma B^2)t}{n} + \sqrt{\mathcal{L}_{\mathcal{D}}^{\phi}(h) \frac{(B+\gamma B^2)t}{n}} \right). \end{aligned}$$

where $t = \log(1/\delta) + \log^3 n$ and K_1 and K_2 are universal constants.

The bound stated in this lemma is *optimistic* in the sense that it reduces to $\tilde{O}(1/\sqrt{n})$ when the problem is difficult and be better when the problem is easier, approaching $\tilde{O}(1/n)$ for linearly separable data, i.e., $\mathcal{L}_{\mathcal{D}}^{\phi,*} = 0$ in the second inequality. These two lemmas essentially enable us to transform a bound on the optimization error and generalization bound into a bound on the convex excess risk. In particular, by combining Lemma 4.9 with Lemma 4.10, we have the following theorem that bounds the smooth convex excess risk $\mathcal{E}_\phi(\widehat{h}) = \mathcal{L}_{\mathcal{D}}^{\phi}(\widehat{h}) - \mathcal{L}_{\mathcal{D}}^{\phi}(h_\lambda^*)$ for the empirical convex risk minimizer.

Theorem 4.11. *Let $\widehat{h}$ be the solution output from updating rules in (4.4) after k iterations. Then, with a probability at least $1-\delta$, we have*

$$\mathcal{E}_\phi(\widehat{h}) \leq \frac{\gamma B^2}{(k+2)^2} + K \left(\frac{(B+\gamma B^2)t}{n} + \sqrt{\mathcal{L}_{\mathcal{D}}^{\phi,*} \frac{(B+\gamma B^2)t}{n}} + \sqrt{\frac{\gamma B^2 (B+\gamma B^2)t}{(k+2)^2 n}} \right)$$

where K is a universal constant, $t = \log(1/\delta) + \log^3 n$, and $\mathcal{L}_{\mathcal{D}}^{\phi,*} = \min_{\|h\|_{\mathcal{H}_\kappa} \leq B} \mathcal{L}_{\mathcal{D}}^{\phi}(h)$.

Since our overall interest is to understand how the smoothing parameter γ affects the convergence rate of excess risk in terms of n, the number of training examples, it is better to parametrize both the number of iterations k and smoothing parameter γ in n, and bound the $\mathcal{E}_\phi(\widehat{h})$ only in terms of n. This is given in the following corollary.

Corollary 4.12. *Assume $\gamma \geq 1$ and $B \geq 1$. Paramertize k and γ in terms of n as $k+2 = n^{\alpha/2}$ and $\gamma = n^\beta$. Then, with a probability at least $1-\delta$,*

$$\mathcal{E}_\phi(\widehat{h}) \leq C(B,t)\left(n^{\beta-\alpha} + n^{\beta-1} + n^{\beta-(\alpha+1)/2} + [\mathcal{L}_{\mathcal{D}}^{\phi,*}]^{1/2} n^{(\beta-1)/2}\right) \tag{4.5}$$

where $C(B,t)$ is a constant depending on both B and t with $t = \log(1/\delta) + \log^3 n$.

The bound given in (4.5) depends on $\mathcal{L}_{\mathcal{D}}^{\phi,*}$. We would like to further characterize $\mathcal{L}_{\mathcal{D}}^{\phi,*}$ in terms of γ. First, we have

$$\begin{aligned} \phi(z;\gamma) &= \max_{\alpha\in[0,1]} \max(0, 1-z) + \frac{1}{\gamma}\mathcal{R}(\alpha) \\ &\leq \max_{\alpha\in[0,1]} \max(0,1-z) + \frac{1}{\gamma}\log 2 = \phi_{\text{Hinge}}(z) + \frac{\log 2}{\gamma}, \end{aligned}$$

where $\phi_{\text{Hinge}}(z) = \max(0, 1-z)$ is the hinge loss. As a result, we have

$$\mathcal{L}_{\mathcal{D}}^{\phi,*} \leq \mathcal{L}_{\mathcal{D}}^{\text{Hinge},*} + \frac{\log 2}{\gamma}$$

where $\mathcal{L}_{\mathcal{D}}^{\text{Hinge},*} = \min\limits_{\|h\|_{\mathcal{H}_\kappa} \leq B} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[\phi_{\text{Hinge}}(yh(\mathbf{x}))\right]$ is the optimal risk with respect to the hinge loss. In general, we will assume

$$\mathcal{L}_{\mathcal{D}}^{\phi,*} \leq \mathcal{L}_{\mathcal{D}}^{\text{Hinge},*} + \frac{a}{\gamma^{1+\xi}} \tag{4.6}$$

where $a > 0$ is a constant and $\xi \geq 0$ characterizes how fast $\mathcal{L}_{\mathcal{D}}^{\phi,*}$ will converge to $\mathcal{L}_{\mathcal{D}}^{\text{Hinge},*}$ with increasing γ. To see why the assumption in (4.6) is sensible, consider the case when the optimal classifier $h_{\text{Hinge}}^* = \arg\min_{\|h\|_{\mathcal{H}_\kappa} \leq B} \mathcal{L}_{\mathcal{D}}^{\text{Hinge}}(h)$ can perfectly classify all the data points with margin ϵ, in which we have

$$\mathcal{L}_{\mathcal{D}}^{\phi,*} \leq \mathcal{L}_{\mathcal{D}}^{\text{Hinge},*} + O\left(\frac{e^{-\epsilon\gamma}}{\gamma}\right)$$

which satisfy the condition in (4.6) with arbitrarily large ξ. It is easy to verify that the condition (4.6) holds with $\xi > 0$ if h_{Hinge}^* can perfectly classify $O(1 - \gamma^{-1-\xi})$ percentage of data with margin ϵ.

Using the assumption in (4.6), we have the following result that characterizes the smooth convex excess risk bound $\mathcal{E}_\phi(\widehat{h})$ stated in terms of the parameters α, δ and $\mathcal{L}_{\mathcal{D}}^{\text{Hinge},*}$.

Theorem 4.13. *Assume $\alpha \geq 1/2$. Set β as*

$$\beta = \frac{\min(1/2, \alpha - 1/2)}{1 + \xi}.$$

With a probability $1 - \delta$, we have

$$\mathcal{E}_\phi(\widehat{h}) \leq O(n^{-\tau_1} + [\mathcal{L}_{\mathcal{D}}^{\text{Hinge},*}]^{1/2} n^{-\tau_2})$$

where

$$\tau_1 = \frac{1 + 2\xi \min(1, \alpha)}{2(1 + \xi)}, \quad \tau_2 = \frac{1/2 + \xi}{2(1 + \xi)}$$

Proof. Replacing $\mathcal{L}_{\mathcal{D}}^{\phi,*}$ in Corollary 4.12 with the expression in (4.6), we have, with a probability $1 - \delta$,

$$\mathcal{E}_\phi(\widehat{h}) \leq C(R, t, a) \left(n^{\beta - \alpha} + n^{\beta - 1} + n^{\beta - (\alpha+1)/2} + [\mathcal{L}_{\mathcal{D}}^{\text{Hinge},*}]^{1/2} n^{(\beta-1)/2} + n^{-1/2 - \xi\beta} \right)$$

We first consider the case when $\alpha > 1$. In this case, we have

$$\mathcal{E}_\phi(\widehat{h}) \leq O\left(n^{\beta-1} + n^{-1/2-\xi\beta} + [\mathcal{L}_{\mathcal{D}}^{\text{Hinge},*}]^{1/2} n^{(\beta-1)/2} \right)$$

By choosing $\beta - 1 = -1/2 - \xi\beta$, we have $\beta = \frac{1/2}{1+\xi}$ and

$$\mathcal{E}_\phi(\widehat{h}) \leq O(n^{-(1/2+\xi)/(1+\xi)} + [\mathcal{L}_{\mathcal{D}}^{\text{Hinge},*}]^{1/2} n^{-(1/2+\xi)/[2(1+\xi)]}$$

In the second case, we have $\alpha \in [1/2, 1]$. Hence we have

$$\mathcal{E}_\phi(\widehat{h}) \leq O\left(n^{\beta-\alpha} + [\mathcal{L}_{\mathcal{D}}^{\text{Hinge},*}]^{1/2} n^{(\beta-1)/2} + n^{-1/2-\xi\beta} \right)$$

By setting $\beta - \alpha = -1/2 - \xi\beta$, we have $\beta = \frac{\alpha - 1/2}{1+\xi}$ and

$$\mathcal{E}_\phi(\widehat{h}) \leq O\left(n^{-\frac{\xi\alpha + 1/2}{1+\xi}} + [\mathcal{L}_{\mathcal{D}}^{\text{Hinge},*}]^{1/2} n^{-(1/2+\xi)/[2(1+\xi)]} \right).$$

We complete the proof by combining the results for the two cases. □

4.4.2 Bounding Binary Excess Risk $\mathcal{E}(h)$

We now combine the results from Theorem 4.8 and Corollary 4.12 to bound $\mathcal{E}(h)$.

Theorem 4.14. *Assume $\alpha \geq 1/2$. For a failure probability $\delta \in (0, 1)$, define n_0 as*

$$n_0 \leq K_3(B, \delta) \left(\frac{1}{\mathcal{L}_{\mathcal{D}}^{\text{Hinge},*}} \right)^{1/(2\tau_1 - 2\tau_2)}$$

where $K_3(B, \delta)$ is a constant depending on B and δ, and τ_1 and τ_2 are defined in Theorem 4.13. Set β as that in Theorem 4.13 if $n \leq n_0$ and 0, otherwise. Then, with a probability $1 - \delta$, we have

$$\mathcal{E}_\phi \leq \begin{cases} K_4(B, \delta) n^{-\tau_1} \log n & n \leq n_0 \\ K_5(B, \delta) n^{-1/2} \log n & n > n_0 \end{cases}$$

where $K_4(B, \delta)$ and $K_5(B, \delta)$ are constants depending on B and δ.

Theorem 4.14 follows from Theorem 4.8 and similar analysis for Theorem 4.13, from which we have

$$\mathcal{E}(\widehat{h}) = \mathcal{L}_{\mathcal{D}}(\widehat{h}) - \mathcal{L}_{\mathcal{D}}^{*} \leq O\left(\min\left(\gamma^{-1}, \mathcal{L}_{\mathcal{D}}^{\phi}(\widehat{h}) - \mathcal{L}_{\mathcal{D}}^{\phi,*} \right) \log n \right)$$

Remark 4.15. *According to Theorem 4.14, when the number of training examples n is not too large, for the binary excess risk of empirical minimizer we have, with a high probability,*

$$\mathcal{E}(\widehat{h}) \leq O(n^{-\tau_1} \log n).$$

In the case when $\xi > 0$ and $\alpha > 1/2$ (i.e. when the number of optimization iterations is *larger than $\sqrt{n}$ and $\mathcal{L}_{\mathcal{D}}^{\phi,*}$ converges to $\mathcal{L}_{\mathcal{D}}^{\text{Hinge},*}$ faster than $1/\gamma$), we have $\tau_1 > 1/2$, implying that using a smooth convex loss will lead to a generalization error bound better than $O(n^{-1/2})$ when the number of training examples is limited. This implies that for smooth loss function to achieve a binary excess error to the extent which is achievable by corresponding non-smooth loss we can run the first order optimization method for a less number of iterations. This is because our result examines the binary excess risk by taking into account the optimization complexity.*

We also note $1/(2\tau_1 - 2\tau_2)$ is given by

$$\frac{1}{2\tau_1 - 2\tau_2} = \frac{1+\xi}{1/2 + \xi\min(1, 2\alpha - 1)}$$

When $\alpha \leq 3/4$, we have $n_0 \geq K_3(B,\delta)[\mathcal{L}_{\mathcal{D}}^{\text{Hinge},*}]^{-2}$, which could be a large number when $\mathcal{L}_{\mathcal{D}}^{\text{Hinge},*}$ is very small.

4.5 Proofs of Statistical Consistency

4.5.1 Proof of Theorem 4.5

We first compute

$$z = \arg\min_{z'} \frac{1+\eta}{2}\phi(z';\gamma) + \frac{1-\eta}{2}\phi(-z';\gamma)$$

By setting the derivative to be zero, we have

$$\frac{1+\eta}{1+\exp(-\gamma(1-z))} = \frac{1-\eta}{1+\exp(-\gamma(1+z))}$$

and therefore

$$(1+\eta)\exp(-\gamma z) - (1-\eta)\exp(\gamma z) + 2\eta\exp(\gamma) = 0.$$

Solving the equation, we obtain

$$\exp(-\gamma z) = \frac{-\eta\exp(\gamma) + \sqrt{\eta^2\exp(2\gamma) + (1-\eta^2)}}{1+\eta}$$

and

$$\exp(\gamma z) = \frac{\eta\exp(\gamma) + \sqrt{\eta^2\exp(2\gamma) + (1-\eta^2)}}{1-\eta}.$$

It is easy to verify that $\operatorname{sgn}(z) = \operatorname{sgn}(\eta)$. This is because if $\eta > 0$, we have

$$\exp(-\gamma z) \leq \frac{\sqrt{1-\eta^2}}{1+\eta} = \sqrt{\frac{1-\eta}{1+\eta}} < 1$$

and therefore $z > 0$. On the other hand, when $\eta < 0$, we have

$$\exp(\gamma z) = \frac{1+\eta}{-\eta\exp(\gamma) + \sqrt{\eta^2\exp(2\gamma) + (1-\eta^2)}} \leq \sqrt{\frac{1+\eta}{1-\eta}} < 1,$$

and therefore $z < 0$. Using the solution for z, we compute $\phi(\eta)$ as

$$\begin{aligned}
\psi(\eta;\gamma) &= \frac{1+\eta}{2}\phi(z;\gamma) + \frac{1-\eta}{2}\phi(z;\gamma) - \min_z \frac{1+\eta}{2}\phi(z;\gamma) + \frac{1-\eta}{2}\phi(z;\gamma) \\
&= -\frac{1+\eta}{2\gamma}\log\frac{1+\exp(\gamma(1-z))}{1+\exp(\gamma)} - \frac{1-\eta}{2\gamma}\log\frac{1+\exp(\gamma(1+z))}{1+\exp(\gamma)}.
\end{aligned}$$

By defining constants $C_1 = -\eta e^{\gamma} + \sqrt{\eta^2 e^{2\gamma} + 1 - \eta^2}$ and $C_2 = \eta e^{\gamma} + \sqrt{\eta^2 e^{2\gamma} + 1 - \eta^2}$, we can rewrite the transform function $\psi(\eta;\gamma)$ as

$$\psi(\eta;\gamma) = -\frac{1+\eta}{2\gamma}\log\left(\frac{1}{1+e^{\gamma}}\left[1 + e^{\gamma}\frac{C_1}{1+\eta}\right]\right) - \frac{1-\eta}{2\gamma}\log\left(\frac{1}{1+e^{\gamma}}\left[1 + e^{\gamma}\frac{C_2}{1-\eta}\right]\right).$$

4.5.2 Proof of Theorem 4.6

Since the expression for $\psi(\eta;\gamma)$ is symmetric in terms η, we will only consider the case when $\eta > 0$. First, we have

$$\frac{C_1 e^{\gamma}}{1+\eta} = \frac{1-\eta}{\eta + \sqrt{\eta^2 + (1-\eta^2)e^{-2\gamma}}} \le \frac{1-\eta}{2\eta}.$$

Similarly, we have

$$\frac{C_2 e^{\gamma}}{1-\eta} = \frac{e^{\gamma}}{1-\eta}\left(\eta e^{\gamma} + \sqrt{\eta^2 e^{2\gamma} + 1 - \eta^2}\right) \le \frac{1+\eta}{1-\eta}e^{2\gamma}$$

Thus, we have

$$\begin{aligned}
\psi(\eta;\gamma) &\ge \frac{1+\eta}{2\gamma}\log(1+e^{\gamma}) - \frac{1+\eta}{2\gamma}\log\left(\frac{1-\eta}{2\eta}\right) - \frac{1-\eta}{2\gamma}\log\left(\frac{1+\eta}{1-\eta}e^{\gamma}\right) \\
&\ge \eta - \frac{1+\eta}{2\gamma}\log\left(\frac{1-\eta}{2\eta}\right) - \frac{1-\eta}{2\gamma}\log\left(\frac{1+\eta}{1-\eta}\right) \\
&\ge \eta - \frac{1}{\gamma}\log\left(\frac{1-\eta^2}{4\eta} + \frac{1+\eta}{2}\right) = \eta - \frac{1}{\gamma}\log\left(\frac{1}{4\eta} + \frac{\eta}{4} + \frac{1}{2}\right)
\end{aligned}$$

where the last inequality follows from the concaveness of $\log(\cdot)$ function. As a result when $\eta \in (-1, 1)$ we have

$$\frac{1}{4\eta} + \frac{\eta}{4} + \frac{1}{2} \le \frac{1}{\eta},$$

which completes the proof.

4.5.3 Proof of Theorem 4.11

Applying Lemmas 4.9 and 4.10 to the solution to the empirical convex risk minimizer $\widehat{h}$, we have

$$
\begin{aligned}
\mathcal{L}_{\mathcal{D}}^{\phi}(\widehat{h}) &\leq \mathcal{L}_{\mathcal{S}}^{\phi}(\widehat{h}) + K_1 \left(\frac{(B+\gamma B^2)t}{n} + \sqrt{\mathcal{L}_{\mathcal{S}}^{\phi}(\widehat{h}) \frac{(B+\gamma B^2)t}{n}} \right) \qquad (4.7) \\
&\leq \; \mathcal{L}_{\mathcal{S}}^{\phi}(h_{\gamma}^{*}) + \frac{\gamma B^2}{(k+2)^2} + K_1 \left(\frac{(B+\gamma B^2)t}{n} + \sqrt{\mathcal{L}_{\mathcal{S}}^{\phi}(h_{\gamma}^{*}) \frac{(B+\gamma B^2)t}{n}} + \sqrt{\frac{\gamma B^2 (B+\gamma B^2)t}{(k+2)^2 n}} \right)
\end{aligned}
$$

On the other hand, by the application of the Bernstein's inequality [30], with probability at least $1-\delta$ we have

$$
\begin{aligned}
\mathcal{L}_{\mathcal{S}}^{\phi}(h_{\gamma}^{*}) - \mathcal{L}_{\mathcal{D}}^{\phi}(h_{\gamma}^{*}) &\leq \frac{4B \log \frac{1}{\delta}}{n} + \sqrt{\frac{4\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left[\left(\phi(y h_{\gamma}^{*}(\mathbf{x});\gamma) - \mathcal{L}_{\mathcal{D}}^{\phi}(h_{\gamma}^{*}) \right)^2 \right] \log \frac{1}{\delta}}{n}} \qquad (4.8) \\
&\leq \frac{4B \log \frac{1}{\delta}}{n} + \sqrt{\frac{8B \mathcal{L}_{\mathcal{D}}^{\phi}(h_{\gamma}^{*}) \log \frac{1}{\delta}}{n}}.
\end{aligned}
$$

We conclude the proof by plugging in (7.7) with (7.6), replacing the constants with a new universal constant K, and noting that $t = \log \frac{1}{\delta} + \log^3 n$.

4.6 Summary

In this chapter we have investigated how the smoothness of loss function being used as the surrogate of 0-1 loss function in empirical risk minimization affects the excess binary risk. While the relation between convex excess risk and binary excess risk being provably established previously under weakest possible condition such as differentiability, it was not immediately obvious how smoothness of convex surrogate relates to statistical consequences. This chapter made first step towards understanding this affect. In particular, in contrast to optimization and generalization analysis that favor smooth surrogate losses, our results revealed that smoothness degrades the binary excess risk. To investigate guarantees on which the smoothness would be a desirable property, we proposed a unified analysis that integrates errors in optimization, generalization, and translating convex excess risk into binary excess risk. Our result shows that under favorable conditions and with appropriate choice of smoothness parameter, a smoothed hinge loss can achieve a binary excess risk that is better than $O(1/\sqrt{n})$.

Chapter 5

Regret Bounded by Gradual Variation

The focus so far in this thesis has been on statistical learning where we assumed that the learner is provided with a pool of i.i.d training examples according to a fixed and unknown distribution $\mathcal{D}$ from the instance space $\Xi = \mathcal{X} \times \mathcal{Y}$ and is asked to output a hypothesis $h \in \mathcal{H}$ which achieves a good generalization performance. This statistical assumption permits the estimation of the generalization error and the uniform convergence theory provides basic guarantees on the correctness of future predictions.

We turn now to the sequential prediction setting in which no statistical assumption is made about the sequence of observations. In particular, we consider the online convex optimization problem introduced in Chapter 2 where the ultimate goal is to devise efficient algorithms in adversarial environments with sub-linear regret bounds in terms of the number of rounds the game proceeds. We have seen a wide variety of algorithms such as Follow The Perturbed Leader (FTPL) for linear and combinatorial online learning problems, and a simple Online Gradient Descent (OGD), Follow The Regularized Leader (FTRL), and Online Mirror Descent (OMD) algorithms for general convex functions which attain an $O(\sqrt{T})$ and $O(\log T)$ regret bounds for Lipschitz continuous and strongly convex functions, respectively.

Most previous works, including those discussed above, considered the most general setting in which the loss functions could be arbitrary and possibly chosen in an adversarial way. However, the environments around us may not always be fully adversarial, and the loss functions may have some patterns which can be exploited for achieving a smaller regret. For example, the weather condition or the stock price at one moment may have some correlation with the next and their difference is usually small, while abrupt changes only occur sporadically. Consequently, it is objected that requiring an algorithm to have a small regret for all sequences leads to results that are too loose to be practically interesting, and the bounds obtained for worst case scenarios become pessimistic for these regular sequences. Recently, it has been shown that the regret of the FTRL algorithm for online *linear* optimization can be bounded by the total variation of the cost vectors rather than the number of rounds. This result is appealing for the scenarios where the sequence of loss functions have a pattern and are not fully adversarial.

In this chapter we extend this result to general online convex optimization and introduce a new measure referred to as *gradual variation* to capture the variation of consecutive convex functions. We show that the total variation bound is not necessarily small when the cost functions change slowly, and the gradual variation lower bounds the total variation. To establish the main results, we discuss a lower bound on the performance of the FTRL that maintains only one sequence of solutions, and a necessary condition on smoothness of the cost functions for obtaining a gradual variation bound. We then present two novel algorithms, improved FTRL and Online Mirror Prox (OMP), that bound the regret by the gradual variation of cost functions. Unlike previous approaches that maintain a single sequence of solutions, the proposed algorithms maintain two sequences of solutions that makes it possible to achieve a gradual variation-based regret bound for online convex optimization. We also extend the main results two-fold: (i) we present a general method to obtain a gradual variation bound measured by general norms rather than the ℓ_2 norm and specialize it to three online learning settings, namely online linear optimization, prediction with expert advice, and online strictly convex optimization; (ii) we develop a deterministic algorithm for online bandit optimization in multipoint bandit setting based on the proposed OMP algorithm.

5.1 Variational Regret Bounds

Recall that in online convex optimization problem, at each trial t, the learner is asked to predict the decision vector $\mathbf{w}_t$ that belongs to a bounded closed convex set $\mathcal{W} \subseteq \mathbb{R}^d$; it then receives a cost function $f_t : \mathcal{W} \to \mathbb{R}_+$ from a family of convex functions $\mathcal{F}$ and incurs a cost of $f_t(\mathbf{w}_t)$ for the submitted solution. The goal of online convex optimization is to come up with a sequence of solutions $\mathbf{w}_1, \ldots, \mathbf{w}_T \in \mathcal{W}$ that minimizes the regret, which is defined as the difference in the cost of the sequence of decisions accumulated up to the trial T made by the learner and the cost of the best fixed decision in hindsight, i.e.

$$\text{Regret}_T = \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}). \tag{5.1}$$

The goal of online convex optimization is to design algorithms that predict, with a small regret, the solution $\mathbf{w}_t$ at the tth trial given the (partial) knowledge about the past cost functions $f_1, f_2, \cdots, f_{t-1} \in \mathcal{F}$.

As already mentioned, generally most previous studies of online convex optimization bound the regret in terms of the number of trials T. In particular for general convex Lipschitz continuous and strongly convex functions regret bounds of $O(\sqrt{T})$ and $O(\log T)$ have been established, respectively, which are known to minimax optimal. However, it is expected that the regret should be low in an unchanging environment or when the cost functions are somehow correlated. Ideally, the tightest rate for the regret should depend on the variance of the sequence of cost functions rather than the number of rounds T. Consequently, the bounds obtained for worst case scenarios in terms of number of iterations become pessimistic for these regular sequences and too loose to be practically interesting. Therefore, it is of great interest to derive a variation-based regret bound for online convex optimization in an adversarial setting.

Recently [69] made a substantial progress in this route and proved a variation-based regret bound for online *linear* optimization by tight analysis of FTRL algorithm with an appropriately chosen step size. A similar regret bound is shown in the same paper for prediction from expert advice by slightly modifying the multiplicative weights algorithm. In this chapter, we take one step further and contribute to this research direction by developing algorithms for general framework of online *convex* optimization with variation-based regret bounds.

When all the cost functions are linear, i.e., $f_t(\mathbf{w}) = \langle \mathbf{f}_t, \mathbf{w} \rangle$, where $\mathbf{f}_t \in \mathbb{R}^d$ is the cost vector in trial t, online convex optimization becomes online linear optimization. Many decision problems can be cast into online linear optimization problem, such as prediction from expert advice [37], online shortest path problem [142]. The first variation-based regret bound for online linear optimization problems in an adversarial setting has been shown in [69]. Hazan and Kale's algorithm for online *linear* optimization is based on the framework of FTRL. At each trial, the decision vector $\mathbf{w}_t$ is given by solving the following optimization problem:

$$\mathbf{w}_t = \arg\min_{\mathbf{w} \in \mathcal{W}} \sum_{\tau=1}^{t-1} \langle \mathbf{f}_\tau, \mathbf{w} \rangle + \frac{1}{2\eta} \|\mathbf{w}\|_2^2, \tag{5.2}$$

where $\mathbf{f}_t$ is the cost vector received at trial t after predicting the decision $\mathbf{w}_t$, and η is a step size. They bound the regret by the variation of cost vectors defined as

$$\text{Variation}_T = \sum_{t=1}^{T} \|\mathbf{f}_t - \boldsymbol{\mu}\|_2^2, \tag{5.3}$$

where $\boldsymbol{\mu} = 1/T \sum_{t=1}^{T} \mathbf{f}_t$. By assuming $\|\mathbf{f}_t\|_2 \leq 1, \forall t$ and setting the value of η to $\eta = \min(2/\sqrt{\text{Variation}_T}, 1/6)$, they showed that the regret of FTRL can be bounded by

$$\sum_{t=1}^{T} \langle \mathbf{f}_t, \mathbf{w}_t \rangle - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} \langle \mathbf{f}_t, \mathbf{w} \rangle \leq \begin{cases} 15\sqrt{\text{Variation}_T} & \text{if } \sqrt{\text{Variation}_T} \geq 12 \\ 150 & \text{if } \sqrt{\text{Variation}_T} \leq 12 \end{cases}. \tag{5.4}$$

From (5.4), we can see that when the variation of the cost vectors is small (less than 12), the regret is a constant, otherwise it is bounded by the variation $O(\sqrt{\text{Variation}_T})$. This result indicates that online linear optimization in the adversarial setting is as efficient as in the stationary stochastic setting.

5.2 Gradual Variation and Necessity of Smoothness

Here we introduce a new measure to characterize the efficiency of online learning algorithms in evolving environments which is termed as gradual variation. The motivation of defining gradual variation stems from two observations: one is practical and the other one is technical raised by the limitation of extending the results in [69] to general convex functions. From practical point of view, we are interested in a more general scenario, in which the environment may be evolving but in a somewhat gradual way. For example, the weather condition or the stock price at one moment may have some correlation with the next and their difference is usually small, while abrupt changes only occur sporadically.

Algorithm 2 Linearalized Follow The Regularized Leader for OCO

```
Input: $\eta > 0$
Initialize: $\mathbf{w}_1 = 0$
for $t = 1, \ldots, T$ do
    Predict $\mathbf{w}_t$ by
        $\mathbf{w}_t = \arg\min_{\mathbf{w} \in \mathcal{W}} \sum_{\tau=1}^{t-1} \langle \mathbf{f}_\tau, \mathbf{w} \rangle + \frac{1}{2\eta} \|\mathbf{w}\|_2^2$
    Receive cost function $f_t(\cdot)$ and incur loss $f_t(\mathbf{w}_t)$
    Compute $\mathbf{f}_t = \nabla f_t(\mathbf{w}_t)$
end for
```

In order to understand the limitation of extending the results in [69], let us apply the results to general convex loss functions. This is an important problem in its own as online convex optimization generalizes online linear optimization by replacing linear cost functions with non-linear convex cost functions and covers many other sequential decision making problems. For instance, it has found applications in portfolio management [6] and online classification [88]. In online portfolio management problem, an investor wants to distribute his wealth over a set of stocks without knowing the market output in advance. If we let $\mathbf{w}_t$ denote the distribution on the stocks and $\mathbf{r}_t$ denote the price relative vector, i.e., $r_t[i]$ denote the the ratio of the closing price of stock i on day t to the closing price on day $t-1$, then an interesting function is the logarithmic growth ratio, i.e. $\sum_{t=1}^{T} \log(\langle \mathbf{w}_t, \mathbf{r}_t \rangle)$, which is a concave function to be maximized.

Since the results in [69] were developed for linear loss functions, a straightforward approach is to use the first order approximation for convex loss functions, i.e., $f_t(\mathbf{w}) \simeq f_t(\mathbf{w}_t) + \langle \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t \rangle$, and replace the linear loss vector with the gradient of the loss function $f_t(\mathbf{w})$ at $\mathbf{w}_t$. The resulting algorithm is shown in Algorithm 2. Using the convexity of loss function $f_t(\mathbf{w})$, we have

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq \sum_{t=1}^{T} \langle \mathbf{f}_t, \mathbf{w}_t \rangle - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} \langle \mathbf{f}_t, \mathbf{w} \rangle. \tag{5.5}$$

If we assume $\|\nabla f_t(\mathbf{w})\|_2 \leq 1, \forall t \in [T], \forall \mathbf{w} \in \mathcal{W}$, we can apply Hazan and Kale's variation-based bound in (5.4) to bound the regret in (5.5) by the variation of the cost functions as:

$$\text{Variation}_T = \sum_{t=1}^{T} \|\mathbf{f}_t - \boldsymbol{\mu}\|_2^2 = \sum_{t=1}^{T} \left\| \nabla f_t(\mathbf{w}_t) - \frac{1}{T} \sum_{\tau=1}^{T} \nabla f_\tau(\mathbf{w}_\tau) \right\|_2^2. \tag{5.6}$$

To better understand Variation_T in (5.6), we rewrite it as

$$\begin{aligned}
\text{Variation}_T &= \sum_{t=1}^{T} \left\| \nabla f_t(\mathbf{w}_t) - \frac{1}{T} \sum_{\tau=1}^{T} \nabla f_\tau(\mathbf{w}_\tau) \right\|_2^2 \\
&= \frac{1}{2T} \sum_{t,\tau=1}^{T} \|\nabla f_t(\mathbf{w}_t) - \nabla f_\tau(\mathbf{w}_\tau)\|^2 \\
&\leq \frac{1}{T} \sum_{t=1}^{T} \sum_{\tau=1}^{T} \|\nabla f_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_\tau)\|_2^2 + \frac{1}{T} \sum_{t=1}^{T} \sum_{\tau=1}^{T} \|\nabla f_t(\mathbf{w}_\tau) - \nabla f_\tau(\mathbf{w}_\tau)\|_2^2 \\
&= \text{Variation}_T^1 + \text{Variation}_T^2.
\end{aligned}$$

We see that the variation Variation_T is bounded by two parts: Variation_T^1 essentially measures the smoothness of individual cost functions, while Variation_T^2 measures the variation in the gradients of cost functions. Let us consider an easy setting when all cost functions are identical. In this case, Variation^2_T vanishes, and Variation_T is equal to $\text{Variation}^1_T/2$, i.e.,

$$\begin{aligned}
\text{Variation}_T &= \frac{1}{2T}\sum_{t,\tau=1}^{T} \|\nabla f_t(\mathbf{w}_t) - \nabla f_\tau(\mathbf{w}_\tau)\|^2 \\
&= \frac{1}{2T}\sum_{t,\tau=1}^{T} \|\nabla f_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_\tau)\|^2 \\
&= \frac{\text{Variation}^1_T}{2}.
\end{aligned}$$

As a result, the regret of the FTRL algorithm for online convex optimization may still be bounded by $O(\sqrt{T})$ regardless of the smoothness of the cost functions.

To address this challenge, we develop two novel algorithms for online convex optimization that bound the regret by the variation of cost functions. In particular, we would like to bound the regret of online convex optimization by the variation of cost functions defined as follows:

$$\text{GradualVariation}_T = \sum_{t=1}^{T-1} \max_{\mathbf{w}\in\mathcal{W}} \|\nabla f_{t+1}(\mathbf{w}) - \nabla f_t(\mathbf{w})\|_2^2. \tag{5.7}$$

Note that the variation in (5.7) is defined in terms of gradual difference between individual cost function to its previous one, while the variation in (5.3) is defined in terms of total difference between individual cost vectors to their mean. Therefore we refer to the variation defined in (5.7) as *gradual variation*, and to the variation defined in (5.3) as *total variation*. It is straightforward to show that when $f_t(\mathbf{w}) = \langle \mathbf{f}_t, \mathbf{w}\rangle$, the gradual variation $\text{GradualVariation}_T$ is upper bounded by the total variation Variation_T defined with a constant factor:

$$\sum_{t=1}^{T-1} \|\mathbf{f}_{t+1} - \mathbf{f}_t\|_2^2 \leq \sum_{t=1}^{T-1} 2\|\mathbf{f}_{t+1} - \boldsymbol{\mu}\|_2^2 + 2\|\mathbf{f}_t - \boldsymbol{\mu}\|_2^2 \leq 4\sum_{t=1}^{T} \|\mathbf{f}_t - \boldsymbol{\mu}\|_2^2.$$

On the other hand, we can not bound the total variation by the gradual variation up to a constant. This is verified by the following example. Let us assume that the adversary plays a fixed function $\mathbf{f}$ for the first half of the iterations and another different function $\mathbf{g}$ for the second half of the iterations, i.e., $\mathbf{f}_1 = \cdots = \mathbf{f}_{T/2} = \mathbf{f}$ and $\mathbf{f}_{T/2+1} = \cdots = \mathbf{f}_T = \mathbf{g} \neq \mathbf{f}$. Then, in this simple scenario the total variation of the sequence of cost functions in (5.3) is given by

$$\text{Variation}_T = \sum_{t=1}^{T} \|\mathbf{f}_t - \boldsymbol{\mu}\|_2^2 = \frac{T}{2}\left\|\mathbf{f} - \frac{\mathbf{f}+\mathbf{g}}{2}\right\|_2^2 + \frac{T}{2}\left\|\mathbf{g} - \frac{\mathbf{f}+\mathbf{g}}{2}\right\|_2^2 = \Omega(T),$$

while the gradual variation defined in (5.7) is a constant given by

$$\text{GradualVariation}_T = \sum_{t=1}^{T-1} \|\mathbf{f}_{t+1} - \mathbf{f}_t\|_2^2 = \|\mathbf{f} - \mathbf{g}\|_2^2 = O(1).$$

Based on the above analysis, we claim that the regret bound by gradual variation is usually tighter than total variation. In particular, the following theorem shows a lower bound on the performance of the FTRL in terms of gradual variation. Unlike the standard setting of online learning where the FTLR achieves the optimal regret bound for Lipschitz continuous and strongly convex losses, it is not capable of achieving regret bounded by gradual variation. The result of this theorem motivates us to develop new algorithms for online convex optimization to achieve a gradual variation bound of $O(\sqrt{\text{GradualVariation}_T})$. For the ease of exposition we use GV_T to denote the gradual variation after T iterations.

Theorem 5.1. *The regret of FTRL is at least* $\Omega(\min(GV_T, \sqrt{T}))$.

Proof. Let $\mathbf{f}$ be any unit vector passing through $\mathbf{w}_1$. Let $s = \lfloor 1/\eta \rfloor$, so that if we use $\mathbf{f}_t = \mathbf{f}$ for every $t \leq s$, each such $\mathbf{z}_{t+1} = \mathbf{w}_1 - t\eta\mathbf{f}$ still remains in $\mathcal{W}$ and thus $\mathbf{w}_{t+1} = \mathbf{z}_{t+1}$. Next, we analyze the regret by considering the following three cases depending on the range of s.

Case I: $s \geq \sqrt{T}$.

First, when $s \geq \sqrt{T}$, we choose $\mathbf{f}_t = \mathbf{f}$ for t from 1 to $\lfloor s/2 \rfloor$ and $\mathbf{f}_t = 0$ for the remaining t. Clearly, the best strategy of the offline algorithm is to play $\mathbf{w} = -\mathbf{f}$. On the other hand, since the learning rate η is too small, the strategy $\mathbf{w}_t$ played by GD, for $t \leq \lfloor s/2 \rfloor$, is far away from $\mathbf{w}$, so that $\langle \mathbf{f}_t, \mathbf{w}_t - \mathbf{w} \rangle \geq 1 - t\eta \geq 1/2$. Therefore, the regret is at least $\lfloor s/2 \rfloor (1/2) = \Omega(\sqrt{T})$.

Case II: $0 < s < \sqrt{T}$.

Second, when $0 < s < \sqrt{T}$, the learning rate is high enough so that FTRL may overreact to each loss vector, and we make it pay by flipping the direction of loss vectors frequently. More precisely, we use the vector $\mathbf{f}$ for the first s rounds so that $\mathbf{w}_{t+1} = \mathbf{w}_1 - t\eta\mathbf{f}$ for any $t \leq s$, but just as $\mathbf{w}_{s+1}$ moves far enough in the direction of $-\mathbf{f}$, we make it pay by switching the loss vector to $-\mathbf{f}$, which we continue to use for s rounds. Note that $\mathbf{w}_{s+1+r} = \mathbf{w}_{s+1-r}$ but $\mathbf{f}_{s+1+r} = -\mathbf{f}_{s+1-r}$ for any $r \leq s$, so $\sum_{t=1}^{2s} \langle \mathbf{f}_t, \mathbf{w}_t - \mathbf{w}_1 \rangle = \langle \mathbf{f}_{s+1}, \mathbf{w}_{s+1} - \mathbf{w}_1 \rangle \geq \Omega(1)$. As $\mathbf{w}_{2s+1}$ returns back to $\mathbf{w}_1$, we can see the first $2s$ rounds as a period, which only contributes $\|2f\|_2^2 = 4$ to the deviation. Then we repeat the period for τ times, where $\tau = \lfloor \mathrm{GV}_T/4 \rfloor$ if there are enough rounds, with $\lfloor T/(2s) \rfloor \geq \lfloor \mathrm{GV}_T/4 \rfloor$, to use up the gradual variation GV_T, and $\tau = \lfloor T/(2s) \rfloor$ otherwise. For any remaining round t, we simply choose $\mathbf{f}_t = 0$. As a result, the total regret is at least $\Omega(1) \cdot \tau = \Omega(\min\{\mathrm{GV}_T/4, T/(2s)\}) = \Omega(\min\{\mathrm{GV}_T, \sqrt{T}\})$.

Case III: $s = 0$.

Finally, when $s = 0$, the learning rate is so high that we can easily make GD pay by flipping the direction of the loss vector in each round. More precisely, by starting with $\mathbf{f}_1 = -\mathbf{f}$, we can have $\mathbf{w}_2$ on the boundary of $\mathcal{W}$, which means that if we then alternate between $\mathbf{f}$ and $-\mathbf{f}$, the strategies FTRL plays will alternate between $\mathbf{w}_3$ and $\mathbf{w}_2$ which have a constant distance from each other. Then following the analysis in the second case, one can show that the total regret is at least $\Omega(\min\{\mathrm{GV}_T, T\})$. □

Assumption 5.2. *In this study, we assume smooth cost functions with Lipschitz continuous gradients, i.e., there exists a constant $L > 0$ such that*

$$\|\nabla f_t(\mathbf{w}) - \nabla f_t(\mathbf{w}')\|_2 \leq L\|\mathbf{w} - \mathbf{w}'\|_2, \forall \mathbf{w}, \mathbf{w}' \in \mathcal{W}, \forall t. \tag{5.8}$$

We would like to emphasize that our assumption about the smoothness of cost functions is necessary to achieve the variation-based bound stated in this chapter. To see this, consider the special case of $f_1(\mathbf{w}) = \cdots = f_T(\mathbf{w}) = f(\mathbf{w})$. If we are able to achieve a regret bound which scales as the square roof of the gradual variation, for any sequence of convex functions, then for the special case where all the cost functions are identical, we have

$$\sum_{t=1}^{T} f(\mathbf{w}_t) \leq \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} f(\mathbf{w}) + O(1),$$

implying that $\widehat{\mathbf{w}}_T = \sum_{t=1}^{T} \mathbf{w}_t / T$ approaches the optimal solution at the rate of $O(1/T)$. This contradicts the lower complexity bound (i.e., $O(1/\sqrt{T})$) for any optimization method which only uses first order information about the cost functions [119, Theorem 3.2.1] (see also Table 2.1). This analysis indicates that the smoothness assumption is necessary to attain variation based regret bound for general online convex optimization problem. We would like to emphasize the fact that this contradiction holds when only the gradient information about the cost functions is provided to the learner and the learner *may* be able to achieve a variation-based bound using second order information about the cost functions, which is not the focus of this chapter.

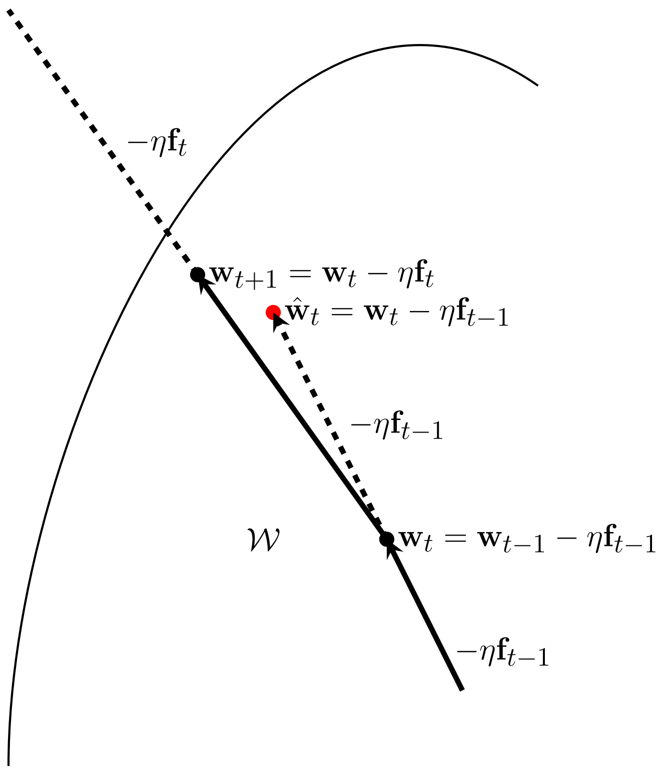

$-\eta\mathbf{f}_t$
$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\mathbf{f}_t$
$\hat{\mathbf{w}}_t = \mathbf{w}_t - \eta\mathbf{f}_{t-1}$
$-\eta\mathbf{f}_{t-1}$
$\mathcal{W}$
$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta\mathbf{f}_{t-1}$
$-\eta\mathbf{f}_{t-1}$

Figure 5.1: Illustration of the main idea behind the proposed improved FTRL and online mirror prox methods to attain regret bounds in terms of gradual variation for linear loss functions. The learner plays the decision $\hat{\mathbf{w}}_t$ instead of $\mathbf{w}_t$ to suffer less regret when the consecutive loss functions are gradually evolving.

5.3 The Improved FTRL Algorithm

As mentioned earlier, the ultimate goal of this chapter is to have algorithms that can take advantage of benign sequences in gradually evolving environments and at the same time protect against the adversarial sequences. However, the impossibility result we showed in the previous section and in particular Theorem 5.1, demonstrated that the existing algorithms such as OGD and in general the family of follow the regularized leader algorithms fail to attain a regret bounded by the gradual variation of the loss functions. Motivated by this negative result, we now turn to proposing two algorithms for online convex optimization that are able to attain regret bounds in terms of gradual variation. The first algorithm is an improved FTRL and the second one is based on the mirror prox method introduced in Chapter 2. One common feature shared by the two algorithms is that both of them maintain two sequences of solutions: decision vectors $\mathbf{w}_{1:T} = (\mathbf{w}_1, \cdots, \mathbf{w}_T)$ and searching vectors $\mathbf{z}_{1:T} = (\mathbf{z}_1, \cdots, \mathbf{z}_T)$ that facilitate the updates of decision vectors. Both algorithms share almost the same regret bound except for a constant factor.

All of our algorithms are based on the following idea, which we illustrate using the online linear optimization problem as an example which is graphically depicted in Figure 5.1. For general linear functions, the online gradient descent algorithm is known to achieve an optimal regret, which plays in round t the point $\mathbf{w}_t = \Pi_{\mathcal{W}}(\mathbf{w}_{t-1} - \eta \mathbf{f}_{t-1})$. Now, if the loss functions have a small deviation, $\mathbf{f}_{t-1}$ may be close to $\mathbf{f}_t$, so in round t, it may be a good idea to play a point which moves further in the direction of $-\mathbf{f}_{t-1}$ as it may make its inner product with $\mathbf{f}_t$ (which is its loss with respect to $\mathbf{f}_t$) smaller. In fact, it can be shown that if one could play the point $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}_t - \eta \mathbf{f}_t)$ in round t, a very small regret could be achieved, i.e., $\sum_{t=1}^{T} \langle \mathbf{w}_{t+1}, \mathbf{f}_t \rangle - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} \langle \mathbf{w}, \mathbf{f}_t \rangle \leq O(1)$, but in reality one does not have $\mathbf{f}_t$ available before round t to compute $\mathbf{w}_{t+1}$. On the other hand, if $\mathbf{f}_{t-1}$ is a good estimate of $\mathbf{f}_t$, the point $\hat{\mathbf{w}}_t = \Pi_{\mathcal{W}}(\mathbf{w}_t - \eta \mathbf{f}_{t-1})$ should be a good estimate of $\mathbf{w}_{t+1}$ too. The point $\hat{\mathbf{w}}_t$ can actually be computed before round t since $\mathbf{f}_{t-1}$ is available, so our algorithm plays $\hat{\mathbf{w}}_t$ in round t. As it will be clear later in this chapter, our algorithms for the prediction with expert advice problem and the online convex optimization problem use the same idea to be able to achieve regret bounds stated in terms of gradual variation of the sequence of losses.

To facilitate the discussion, besides the variation of cost functions defined in (5.7), we

Algorithm 3 Improved FTRL (IFTRL) Algorithm

```
**Input**: $\eta \in (0, 1]$
**Initialize:**: $\mathbf{z}_0 = \mathbf{0}$ and $f_0(\mathbf{w}) = 0$
**for** $t = 1, \ldots, T$ **do**
    Predict $\mathbf{w}_t$ by
        $\mathbf{w}_t = \arg\min_{\mathbf{w} \in \mathcal{W}} \left\{ \langle \mathbf{w}, \nabla f_{t-1}(\mathbf{z}_{t-1}) \rangle + \frac{L}{2\eta} \|\mathbf{w} - \mathbf{z}_{t-1}\|_2^2 \right\}$
    Receive cost function $f_t(\cdot)$ and incur loss $f_t(\mathbf{w}_t)$
    Update $\mathbf{z}_t$ by
        $\mathbf{z}_t = \arg\min_{\mathbf{w} \in \mathcal{W}} \left\{ \langle \mathbf{w}, \sum_{\tau=1}^{t} \nabla f_\tau(\mathbf{z}_{\tau-1}) \rangle + \frac{L}{2\eta} \|\mathbf{w}\|_2^2 \right\}$
**end for**
```

define another variation, named *extended gradual variation*, as follows

$$\text{EGV}_{T,2}(\mathbf{y}_{1:T}) = \sum_{t=0}^{T-1} \|\nabla f_{t+1}(\mathbf{y}_t) - \nabla f_t(\mathbf{y}_t)\|_2^2 \leq \|\nabla f_1(\mathbf{y}_0)\|_2^2 + \text{GV}_T, \tag{5.9}$$

where $f_0(\mathbf{w}) = 0$, the sequence $(\mathbf{y}_0, \ldots, \mathbf{y}_T)$ is either $(\mathbf{z}_0, \ldots, \mathbf{z}_T)$ (as in the improved FTRL) or $(\mathbf{w}_0, \ldots, \mathbf{w}_T)$ (as in the online mirror prox method) and the subscript 2 means the variation is defined with respect to ℓ_2 norm. When all cost functions are identical, GV_T becomes zero and the extended variation $\text{EGV}_{T,2}(\mathbf{y}_{1:T})$ is reduced to $\|\nabla f_1(\mathbf{y}_0)\|_2^2$, a constant independent from the number of trials. In the sequel, we use the notation $\text{EGV}_{T,2}$ for simplicity.

Our results show that for online convex optimization with L-smooth cost functions, the regrets of the proposed algorithms can be bounded as follows

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq O\left(\sqrt{\text{EGV}_{T,2}}\right) + \text{constant}. \tag{5.10}$$

We now turn to presenting our first algorithm, dubbed IFTRL, which is a simple modification of the FTRL algorithm and show that its regret bounded by the gradual variation.

The improved FTRL algorithm for online convex optimization is presented in Algorithm 3. Without loss of generality, we assume that the decision set $\mathcal{W}$ is contained in a unit ball $\mathbb{B} = \{\mathbf{x} \in \mathbb{R}^d : \|\mathbf{x}\| \leq 1\}$, i.e., $\mathcal{W} \subseteq \mathbb{B}$, and $0 \in \mathcal{W}$. Note that in step 6, the searching vectors $\mathbf{z}_t$ are updated according to the FTRL algorithm after receiving the cost function $f_t(\mathbf{w})$. To understand the updating procedure for the decision vector $\mathbf{w}_t$ specified in step 4, we rewrite it as

$$\mathbf{w}_t = \arg\min_{\mathbf{w} \in \mathcal{W}} \left\{ f_{t-1}(\mathbf{z}_{t-1}) + \langle \mathbf{w} - \mathbf{z}_{t-1}, \nabla f_{t-1}(\mathbf{z}_{t-1}) \rangle + \frac{L}{2\eta} \|\mathbf{w} - \mathbf{z}_{t-1}\|_2^2 \right\}. \tag{5.11}$$

Notice that

$$\begin{aligned} f_t(\mathbf{w}) &\leq f_t(\mathbf{z}_{t-1}) + \langle \mathbf{w} - \mathbf{z}_{t-1}, \nabla f_t(\mathbf{z}_{t-1}) \rangle + \frac{L}{2} \|\mathbf{w} - \mathbf{z}_{t-1}\|_2^2 \\ &\leq f_t(\mathbf{z}_{t-1}) + \langle \mathbf{w} - \mathbf{z}_{t-1}, \nabla f_t(\mathbf{z}_{t-1}) \rangle + \frac{L}{2\eta} \|\mathbf{w} - \mathbf{z}_{t-1}\|_2^2, \end{aligned}$$

where the first inequality follows the smoothness condition in (5.8) and the second inequality follows from the fact $\eta \leq 1$. The inequality (5.12) provides an upper bound for $f_t(\mathbf{w})$ and therefore can be used as an approximation of $f_t(\mathbf{w})$ for predicting $\mathbf{w}_t$. However, since $\nabla f_t(\mathbf{z}_{t-1})$ is unknown before the prediction, we use $\nabla f_{t-1}(\mathbf{z}_{t-1})$ as a surrogate for $\nabla f_t(\mathbf{z}_{t-1})$, leading to the updating rule in (5.11). It is this approximation that leads to the variation bound. The following theorem states the regret bound of Algorithm 3.

Theorem 5.3. *Let $f_1, f_2, \ldots, f_T$ be a sequence of convex functions with L-Lipschitz continuous gradients. By setting $\eta = \min\left\{1, L/\sqrt{\mathrm{EGV}_{T,2}}\right\}$, we have the following regret bound* *for the IFTRL in Algorithm 3:*

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq \max\left(L, \sqrt{\mathrm{EGV}_{T,2}}\right).$$

Remark 5.4. *Comparing with the variation bound in (5.7) for the FTRL algorithm, the smoothness parameter L plays the same role as Variation^1_T that accounts for the smoothness of cost functions, and term $\mathrm{EGV}_{T,2}$ plays the same role as Variation^2_T that accounts for the variation in the cost functions. Compared to the FTRL algorithm, the key advantage of the improved FTRL algorithm is that the regret bound is reduced to a constant when the cost functions change only by a constant number of times along the horizon. Of course, the extended variation $\mathrm{EGV}_{T,2}$ may not be known apriori for setting the optimal η, we can apply the standard doubling trick [38] to obtain a bound that holds uniformly over time and is a factor at most 8 from the bound obtained with the optimal choice of η. The details are provided later in this chapter.*

5.4 The Online Mirror Prox Algorithm

The second algorithm we present to attain regret bounds in terms of gradual variation is based on the prox method we introduced in Chapter 2 for non-smooth convex optimization. We generalize the prox method for online convex optimization that shares the same order of regret bound as the improved FTRL algorithm. The detailed steps of the Online Mirror Prox (OMP) method are shown in Algorithm 4, where we use an equivalent form of updates for $\mathbf{w}_t$ and $\mathbf{z}_t$ in order to compare to Algorithm 3 . The OMP method is closely related to the prox method in [117] by maintaining two sets of vectors $\mathbf{w}_{1:T}$ and $\mathbf{z}_{1:T}$, where $\mathbf{w}_t$ and

Algorithm 4 Online Mirror Prox (OMP) Algorithm

```
**Input**: $\eta > 0$
**Initialize:**: $\mathbf{z}_0 = \mathbf{w}_0 = \mathbf{0}$ and $f_0(\mathbf{w}) = 0$
**for** $t = 1, \ldots, T$ **do**
    Predict $\mathbf{w}_t$ by
        $\mathbf{w}_t = \arg\min_{\mathbf{w} \in \mathcal{W}} \left\{ \langle \mathbf{w}, \nabla f_{t-1}(\mathbf{w}_{t-1}) \rangle + \frac{L}{2\eta} \|\mathbf{w} - \mathbf{z}_{t-1}\|_2^2 \right\}$
    Receive cost function $f_t(\cdot)$ and incur loss $f_t(\mathbf{w}_t)$
    Update $\mathbf{z}_t$ by
        $\mathbf{z}_t = \arg\min_{\mathbf{w} \in \mathcal{W}} \left\{ \langle \mathbf{w}, \nabla f_t(\mathbf{w}_t) \rangle + \frac{L}{2\eta} \|\mathbf{w} - \mathbf{z}_{t-1}\|_2^2 \right\}$
**end for**
```

$\mathbf{z}_t$ are computed by *gradient mappings* using $\nabla f_{t-1}(\mathbf{w}_{t-1})$, and $\nabla f_t(\mathbf{w}_t)$, respectively, as

$$\mathbf{w}_t = \arg \min_{\mathbf{w} \in \mathcal{W}} \frac{1}{2} \left\| \mathbf{w} - \left(\mathbf{z}_{t-1} - \frac{\eta}{L} \nabla f_{t-1}(\mathbf{w}_{t-1}) \right) \right\|_2^2$$

$$\mathbf{z}_t = \arg \min_{\mathbf{w} \in \mathcal{W}} \frac{1}{2} \left\| \mathbf{w} - \left(\mathbf{z}_{t-1} - \frac{\eta}{L} \nabla f_t(\mathbf{w}_t) \right) \right\|_2^2$$

The OMP differs from the IFTRL algorithm: (i) in updating the searching points $\mathbf{z}_t$, Algorithm 3 updates $\mathbf{z}_t$ by the FTRL scheme using *all* the gradients of the cost functions at $\{\mathbf{z}_\tau\}_{\tau=1}^{t-1}$, while OMP updates $\mathbf{z}_t$ by a prox method using a *single* gradient $\nabla f_t(\mathbf{w}_t)$, and (ii) in updating the decision vector $\mathbf{w}_t$, OMP uses the gradient $\nabla f_{t-1}(\mathbf{w}_{t-1})$ instead of $\nabla f_{t-1}(\mathbf{z}_{t-1})$. The advantage of OMP algorithm compared to the IFTRL algorithm is that it only requires to compute one gradient $\nabla f_t(\mathbf{w}_t)$ for each loss function; in contrast, the improved FTRL algorithm in Algorithm 3 needs to compute the gradients of $f_t(\mathbf{w})$ at two searching points $\mathbf{z}_t$ and $\mathbf{z}_{t-1}$. It is these differences that make it easier to extend the OMP to a bandit setting, which will be discussed in Section 5.7.

The following theorem states the regret bound of the online mirror prox method for online convex optimization.

Algorithm 5 Online Mirror Prox Method for General Norms

```
Input: $\eta > 0, \Phi(\mathbf{z})$
Initialize:: $\mathbf{z}_0 = \mathbf{w}_0 = \min_{\mathbf{z}\in\mathcal{W}} \Phi(\mathbf{z})$ and $f_0(\mathbf{w}) = 0$
for $t = 1, \ldots, T$ do
    Predict $\mathbf{w}_t$ by
        $$\mathbf{w}_t = \arg\min_{\mathbf{w}\in\mathcal{W}} \left\{ \langle \mathbf{w}, \nabla f_{t-1}(\mathbf{w}_{t-1})\rangle + \frac{L}{\eta}\mathsf{B}(\mathbf{w}, \mathbf{z}_{t-1}) \right\}$$
    Receive cost function $f_t(\cdot)$ and incur loss $f_t(\mathbf{w}_t)$
    Update $\mathbf{z}_t$ by
        $$\mathbf{z}_t = \arg\min_{\mathbf{w}\in\mathcal{W}} \left\{ \langle \mathbf{w}, \nabla f_t(\mathbf{w}_t)\rangle + \frac{L}{\eta}\mathsf{B}(\mathbf{w}, \mathbf{z}_{t-1}) \right\}$$
end for
```

Theorem 5.5. *Let $f_1, f_2, \ldots, f_T$ be a sequence of convex functions with L-Lipschitz continuous gradients. By setting $\eta = (1/2)\min\left\{1/\sqrt{2}, L/\sqrt{\mathrm{EGV}_{T,2}}\right\}$, we have the following regret bound for OMP in Algorithm 4*

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq 2\max\left(\sqrt{2}L, \sqrt{\mathrm{EGV}_{T,2}}\right).$$

We note that compared to Theorem 5.3, the regret bound in Theorem 5.5 is slightly worse by a factor of 2.

5.4.1 Online Mirror Prox Method with General Norms

In this subsection, we first present a general OMP method to obtain a variation bound defined in a general norm. Then we discuss three special cases: online linear optimization, prediction with expert advice, and online strictly convex optimization. The omitted proofs in this subsection can be easily duplicated by mimicking the proof of Theorem 5.5, if necessary with the help of previous analysis as mentioned in the appropriate text.

To adapt OMP to general norms other than the Euclidean norm, let $\|\cdot\|$ denote a general norm, $\|\cdot\|_*$ denote its dual norm, $\Phi(\mathbf{z})$ be a α-strongly convex function with respect to the norm $\|\cdot\|$, and $\mathsf{B}(\mathbf{w},\mathbf{z}) = \Phi(\mathbf{w}) - \big(\Phi(\mathbf{z}) + \langle \mathbf{w}-\mathbf{z}, \Phi'(\mathbf{z})\rangle\big)$ be the Bregman distance induced by function $\Phi(\mathbf{w})$. Let $f_1, f_2, \cdots, f_T$ be a sequence of smooth functions with Lipschitz continuous gradients bounded by L with respect to norm $\|\cdot\|$, i.e.,

$$\|\nabla f_t(\mathbf{w}) - \nabla f_t(\mathbf{w}')\|_* \leq L\|\mathbf{w}-\mathbf{w}'\|. \tag{5.12}$$

Correspondingly, we define the extended gradual variation based on the general norm as follows:

$$\mathrm{EGV}_T = \sum_{t=0}^{T-1} \|\nabla f_{t+1}(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t)\|_*^2. \tag{5.13}$$

Algorithm 5 gives the detailed steps for the general framework. We note that the key differences from Algorithm 4 are: $\mathbf{z}_0$ is set to $\min_{\mathbf{z}\in\mathcal{W}} \Phi(\mathbf{z})$, and the Euclidean distances in steps 4 and 6 are replaced by Bregman distances, i.e.,

$$\begin{aligned} \mathbf{w}_t &= \arg\min_{\mathbf{w}\in\mathcal{W}} \left\{ \langle \mathbf{w}, \nabla f_{t-1}(\mathbf{w}_{t-1})\rangle + \frac{L}{\eta}\mathsf{B}(\mathbf{w},\mathbf{z}_{t-1}) \right\}, \\ \mathbf{z}_t &= \arg\min_{\mathbf{w}\in\mathcal{W}} \left\{ \langle \mathbf{w}, \nabla f_t(\mathbf{w}_t)\rangle + \frac{L}{\eta}\mathsf{B}(\mathbf{w},\mathbf{z}_{t-1}) \right\}. \end{aligned} \tag{5.14}$$

The following theorem states the variation-based regret bound for the general norm framework, where R measure the size of $\mathcal{W}$ defined as

$$R = \sqrt{2(\max_{\mathbf{w}\in\mathcal{W}} \Phi(\mathbf{w}) - \min_{\mathbf{w}\in\mathcal{W}} \Phi(\mathbf{w}))}.$$

Theorem 5.6. *Let $f_1, f_2, \cdots, f_T$ be a sequence of convex functions whose gradients are L-smooth continuous, $\Phi(\mathbf{z})$ be a α-strongly convex function, both with respect to norm $\|\cdot\|$, and EGV_T be defined in (5.13). By setting $\eta = (1/2)\min\left\{\sqrt{\alpha}/\sqrt{2}, LR/\sqrt{\mathrm{EGV}_T}\right\}$, we have the* *following regret bound*

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq 2R \max\left(\sqrt{2}LR/\sqrt{\alpha}, \sqrt{\mathrm{EGV}_T}\right).$$

In the following subsections we specialize the proposed general method to few specific online learning settings.

5.4.2 Online Linear Optimization

Here we consider online linear optimization and present the algorithm and the gradual variation bound for this setting as a special case of proposed algorithm. In particular, we are interested in bounding the regret by the gradual variation

$$\mathrm{EGV}^{f}_{T,2} = \sum_{t=0}^{T-1} \|\mathbf{f}_{t+1} - \mathbf{f}_t\|_2^2,$$

where $\mathbf{f}_t, t = 1, \ldots, T$ are the linear cost vectors and $\mathbf{f}_0 = 0$. Since linear functions are smooth functions that satisfy the inequality in (5.8) for any positive $L > 0$, therefore we can apply Algorithm 4 to online linear optimization with any positive value for L [1]. The regret bound of Algorithm 4 for online linear optimization is presented in the following corollary.

Corollary 5.7. *Let $f_t(\mathbf{w}) = \langle \mathbf{f}_t, \mathbf{w}\rangle, t = 1, \ldots, T$ be a sequence of linear functions. By setting $\eta = \sqrt{1/\left(2\mathrm{EGV}^{f}_{T,2}\right)}$ and $L = 1$ in Algorithm 4, then we have*

$$\sum_{t=1}^{T} \mathbf{f}_t^\top \mathbf{w}_t - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} \langle \mathbf{f}_t, \mathbf{w}\rangle \leq \sqrt{2\mathrm{EGV}^{f}_{T,2}}.$$

[1] We simply set $L = 1$ for online linear optimization and prediction with expert advice.

Remark 5.8. *Note that the regret bound in Corollary 5.7 is stronger than the regret bound obtained in [71] for online linear optimization due to the fact that the gradual variation is smaller than the total variation.*

5.4.3 Prediction with Expert Advice

In the problem of prediction with expert advice, the decision vector $\mathbf{w}$ is a distribution over m experts, i.e., $\mathbf{w} \in \mathcal{W} = \{\mathbf{w} \in \mathbb{R}^m_+ : \sum_{i=1}^m w_i = 1\}$. Let $\mathbf{f}_t \in \mathbb{R}^m$ denote the costs for m experts in trial t. Similar to [69], we would like to bound the regret of prediction from expert advice by the gradual variation defined in infinite norm, i.e.,

$$\mathrm{EGV}^f_{T,\infty} = \sum_{t=0}^{T-1} \|\mathbf{f}_{t+1} - \mathbf{f}_t\|^2_\infty.$$

Since it is a special online linear optimization problem, we can apply Algorithm 4 to obtain a regret bound as in Corollary 5.7, i.e.,

$$\sum_{t=1}^T \mathbf{f}_t^\top \mathbf{w}_t - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^T \langle \mathbf{f}_t, \mathbf{w}\rangle \leq \sqrt{2\mathrm{EGV}^f_{T,2}} \leq \sqrt{2m\mathrm{EGV}^f_{T,\infty}}.$$

However, the above regret bound scales badly with the number of experts. We can obtain a better regret bound in $O(\sqrt{\mathrm{EGV}^f_{T,\infty}} \ln m)$ by applying the general prox method in Algorithm 5 with $\Phi(\mathbf{w}) = \sum_{i=1}^m w_i \ln w_i$ and $\mathsf{B}(\mathbf{w}, \mathbf{z}) = \sum_{i=1}^m w_i \ln(z_i/w_i)$. The two updates in Algorithm 5 become

$$\begin{aligned} w_t^i &= \frac{z_{t-1}^i \exp([\eta/L] f_{t-1}^i)}{\sum_{j=1}^m z_{t-1}^j \exp([\eta/L] f_{t-1}^j)}, i = 1, \dots, m \\ z_t^i &= \frac{z_{t-1}^i \exp([\eta/L] f_t^i)}{\sum_{j=1}^m z_{t-1}^j \exp([\eta/L] f_t^j)}, i = 1, \dots, m. \end{aligned} \tag{5.15}$$

The resulting regret bound is formally stated in the following Corollary.

Corollary 5.9. *Let $f_t(\mathbf{w}) = \langle \mathbf{f}_t, \mathbf{w}\rangle, t = 1, \ldots, T$ be a sequence of linear functions in prediction with expert advice. By setting $\eta = \sqrt{(\ln m)/\mathrm{EGV}^f_{T,\infty}}$, $L = 1$, $\Phi(\mathbf{w}) = \sum_{i=1}^m w_i \ln w_i$ and $\mathsf{B}(\mathbf{w}, \mathbf{z}) = \sum_{i=1}^m w_i \ln(w_i/z_i)$ in Algorithm 5, we have*

$$\sum_{t=1}^T \langle \mathbf{f}_t, \mathbf{w}_t\rangle - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^T \langle \mathbf{f}_t, \mathbf{w}\rangle \leq \sqrt{2\mathrm{EGV}^f_{T,\infty} \ln m}.$$

By noting the definition of $\mathrm{EGV}^{\mathrm{f}}_{\mathrm{T},\infty}$, the regret bound in Corollary 5.9 is

$$O\left(\sqrt{\sum_{t=0}^{T-1} \max_i |f^i_{t+1} - f^i_t| \ln m}\right),$$

which is similar to the regret bound obtained in [69] for prediction with expert advice. However, the definitions of the variation are not exactly the same. In [69], the authors bound the regret of prediction with expert advice by $O\left(\sqrt{\ln m \max_i \sum_{t=1}^T |f^i_t - \mu^i_t|^2} + \ln m\right)$, where the variation is the maximum total variation over all experts. To compare the two regret bounds, we first consider two extreme cases. When the costs of all experts are the same, then the variation in Corollary 5.9 is a standard gradual variation, while the variation in [69] is a standard total variation. According to the previous analysis, a gradual variation is smaller than a total variation, therefore the regret bound in Corollary 5.9 is better than that in [69]. In another extreme case when the costs at all iterations of each expert are the same, both regret bounds are constants. More generally, if we assume the maximum total variation is small (say a constant), then $\sum_{t=0}^{T-1} |f^i_{t+1} - f^i_t|$ is also a constant for any $i \in [m]$. By a trivial analysis $\sum_{t=0}^{T-1} \max_i |f^i_{t+1} - f^i_t| \leq m \max_i \sum_{t=0}^{T-1} |f^i_{t+1} - f^i_t|$, the regret bound in Corollary 5.9 might be worse up to a factor $\sqrt{m}$ than that in [69].

Remark 5.10. *It was shown in [41], both the regret bounds in Corollary 5.7 and Corollary 5.9 are optimal because they match the lower bounds for a special sequence of loss functions. In particular, for online linear optimization if all loss functions but the first* $T_k = \sqrt{\mathrm{EGV}_{T,2}^f}$ *are all-0 functions, then the known lower bound* $\Omega(\sqrt{T_k})$ *matches the upper bound in Corollary 5.7. Similarly, for prediction from expert advice if all loss functions but the first* $T'_k = \sqrt{\mathrm{EGV}_{T,\infty}^f}$ *are all-0 functions, then the known lower bound* $\Omega(\sqrt{T'_k \ln m})$ *[38] matches the upper bound in Corollary 5.9.*

5.4.4 Online Strictly Convex Optimization

In this subsection, we present an algorithm to achieve a logarithmic variation bound for online strictly convex optimization. In particular, we assume the cost functions $f_t(\mathbf{w})$ are not only smooth but also strictly convex defined formally in the following.

Definition 5.11. *For* $\beta > 0$*, a function* $f(\mathbf{w}) : \mathcal{W} \to \mathbb{R}$ *is* β*-strictly convex if for any* $\mathbf{w}, \mathbf{z} \in \mathcal{W}$

$$f(\mathbf{w}) \geq f(\mathbf{z}) + \nabla\langle f(\mathbf{z}), \mathbf{w} - \mathbf{z}\rangle + \beta(\mathbf{w} - \mathbf{z})^\top \nabla f(\mathbf{z}) \nabla f(\mathbf{z})^\top (\mathbf{w} - \mathbf{z}) \tag{5.16}$$

It is known that such a defined strictly convex function include strongly convex function and exponential concave function as special cases as long as the gradient of the function is bounded. To see this, if $f(\mathbf{w})$ is a β'-strongly convex function with a bounded gradient $\|\nabla f(\mathbf{w})\|_2 \leq G$, then

$$\begin{aligned} f(\mathbf{w}) &\geq f(\mathbf{z}) + \langle \nabla f(\mathbf{z}), \mathbf{w} - \mathbf{z}\rangle + \beta' \langle \mathbf{w} - \mathbf{z}, \mathbf{w} - \mathbf{z}\rangle \\ &\geq f(\mathbf{z}) + \langle \nabla f(\mathbf{z}), \mathbf{w} - \mathbf{z}\rangle + \frac{\beta'}{G^2}(\mathbf{w} - \mathbf{z})^\top \nabla f(\mathbf{z}) \nabla f(\mathbf{z})^\top (\mathbf{w} - \mathbf{z}), \end{aligned} \tag{5.17}$$

thus $f(\mathbf{w})$ is a (β'/G^2) strictly convex. Similarly if $f(\mathbf{w})$ is exp-concave, i.e., there exists $\alpha > 0$ such that $h(\mathbf{w}) = \exp(-\alpha f(\mathbf{w}))$ is concave, then $f(\mathbf{w})$ is a $\beta = 1/2 \min(1/(4GD), \alpha)$ strictly convex (c.f. Lemma 2 in [68]), where D is defined as the diameter of the domain. Therefore, in addition to smoothness and strict convexity we also assume all the cost functions have bounded gradients, i.e., $\|\nabla f_t(\mathbf{w})\|_2 \leq G$.

We now turn to deriving a logarithmic gradual variation bound for online strictly convex optimization. To this end, we need to change the Euclidean distance function in Algorithm 4 to a generalized Euclidean distance function. Specifically, at trial t, we let $\mathbf{H}_t = \mathbf{I} + \beta G^2 \mathbf{I} + \beta \sum_{\tau=0}^{t-1} \nabla f_\tau(\mathbf{w}_\tau) \nabla f_\tau(\mathbf{w}_\tau)^\top$ and use the generalized Euclidean distance $\mathsf{B}_t(\mathbf{w}, \mathbf{z}) = \frac{1}{2}\|\mathbf{w} - \mathbf{z}\|_{\mathbf{H}_t}^2 = \frac{1}{2}(\mathbf{w} - \mathbf{z})^\top \mathbf{H}_t (\mathbf{w} - \mathbf{z})$ in updating $\mathbf{w}_t$ and $\mathbf{z}_t$, i.e.,

$$\begin{aligned} \mathbf{w}_t &= \arg\min_{\mathbf{w} \in \mathcal{W}} \left\{ \langle \mathbf{w}, \nabla f_{t-1}(\mathbf{w}_{t-1}) \rangle + \frac{1}{2}\|\mathbf{w} - \mathbf{z}_{t-1}\|_{\mathbf{H}_t}^2 \right\} \\ \mathbf{z}_t &= \arg\min_{\mathbf{w} \in \mathcal{W}} \left\{ \langle \mathbf{w}, \nabla f_t(\mathbf{w}_t) \rangle + \frac{1}{2}\|\mathbf{w} - \mathbf{z}_{t-1}\|_{\mathbf{H}_t}^2 \right\}, \end{aligned} \tag{5.18}$$

To prove the regret bound, we can prove a similar inequality as in Lemma 5.16 by applying $\Phi(\mathbf{w}) = 1/2\|\mathbf{w}\|_{\mathbf{H}_t}^2$, which is stated as follows

$$\begin{aligned} \nabla f_t(\mathbf{w}_t)^\top (\mathbf{w}_t - \mathbf{z}) &\leq \mathsf{B}_t(\mathbf{z}, \mathbf{z}_{t-1}) - \mathsf{B}_t(\mathbf{z}, \mathbf{z}_t) \\ &\quad + \|\nabla f_t(\mathbf{w}_t) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|_{\mathbf{H}_t^{-1}}^2 - \frac{1}{2}\left[\|\mathbf{w}_t - \mathbf{z}_{t-1}\|_{\mathbf{H}_t}^2 + \|\mathbf{w}_t - \mathbf{z}_t\|_{\mathbf{H}_t}^2\right]. \end{aligned}$$

Then by applying inequality in (5.16) for strictly convex functions, we obtain the following

$$\begin{aligned} f_t(\mathbf{w}_t) - f_t(\mathbf{z}) \leq \mathsf{B}_t(\mathbf{w}, \mathbf{z}_{t-1}) - \mathsf{B}_t(\mathbf{w}, \mathbf{z}_t) - \beta\|\mathbf{w}_t - \mathbf{z}\|^2_{\mathbf{M}_t} \\ + \|\nabla f_t(\mathbf{w}_t) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|^2_{\mathbf{H}_t^{-1}} - \frac{1}{2}\left[\|\mathbf{w}_t - \mathbf{z}_{t-1}\|^2_{\mathbf{H}_t} + \|\mathbf{w}_t - \mathbf{z}_t\|^2_{\mathbf{H}_t}\right], \end{aligned} \tag{5.19}$$

where $\mathbf{M}_t = \nabla f_t(\mathbf{w}_t)\nabla f_t(\mathbf{w}_t)^\top$ and $\mathbf{H}_t = \mathbf{I} + \beta G^2 \mathbf{I} + \beta \sum_{\tau=0}^{t-1} \mathbf{M}_\tau$ as defined above. The following corollary shows that general OMP method attains a logarithmic gradual variation bound and its proof is deferred to later.

Corollary 5.12. *Let $f_1, f_2, \ldots, f_T$ be a sequence of β-strictly convex and L-smooth functions with gradients bounded by G. We assume $8dL^2 \geq 1$, otherwise we can set $L = \sqrt{1/(8d)}$. An algorithm that adopts the updates in (5.18) has a regret bounded by*

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq \frac{1+\beta G^2}{2} + \frac{8d}{\beta}\ln\max(16dL^2, \beta \mathrm{EGV}_{T,2}),$$

where $\mathrm{EGV}_{T,2} = \sum_{t=0}^{T-1} \|\nabla f_{t+1}(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t)\|_2^2$ *and d is the dimension of* $\mathbf{w} \in \mathcal{W}$.

5.4.5 Gradual Variation Bounds which Hold Uniformly over Time

As mentioned in Remark 5.4, the algorithms presented in this chapter rely on the previous knowledge of the gradual variation $\mathrm{EGV}_{T,2}$ to tune the learning rate η to obtain the optimal bound. Here, we show that the Algorithm 3 can be used as a black-box to achieve the same regret bound but without any prior knowledge of the $\mathrm{EGV}_{T,2}$. We note that the analysis here is not specific to Algorithm 3 and it is general enough to be adapted to other algorithms in the chapter too.

The main idea is to run the algorithm in epochs with a fixed learning rate $\eta_k = \eta_0/2^k$ for

kth epoch where η_0 is a fixed constant and will be decided by analysis. We denote the number of epochs by K and let b_k denote the start of kth epoch. We note that $b_{K+1} = T+1$. Within kth epoch, the algorithm ensures that the inequality $\eta_k \sum_{t=b_k}^{b_{k+1}-1} \|\nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\|_2^2 \leq L^2\eta_k^{-1}$ holds.

To this end, the algorithm computes and maintains the quantity

$$\sum_{s=b_k}^{t} \|\nabla f_{s+1}(\mathbf{z}_s) - \nabla f_s(\mathbf{z}_s)\|_2^2$$

and sets the beginning of new epoch to be

$$b_{k+1} = \min_t \sum_{s=b_k}^{t} \|\nabla f_{s+1}(\mathbf{z}_s) - \nabla f_s(\mathbf{z}_s)\|_2^2 > L^2\eta_k^{-1},$$

i.e., the first iteration for which the invariant is violated. We note that this decision can only be made after seeing the tth cost function. Therefor, we burn the first iteration of each epoch which causes an extra regret of KL^2 in the total regret. From the analysis we have:

$$\begin{aligned}\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) &\leq \sum_{k=1}^{K} \left[\sum_{t=b_k}^{b_{k+1}-1} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=b_k}^{b_{k+1}-1} f_t(\mathbf{w})\right] \\ &\leq \sum_{k=1}^{K} \frac{L}{2\eta_k} + \frac{\eta_k}{2L}\mathrm{EGV}_{b_k:b_{k+1}-1} + KL^2 \\ &\leq \sum_{k=1}^{K} \frac{L}{2\eta_k} + \frac{L}{2\eta_k} + K = 2L\sum_{k=1}^{K} \eta_k^{-1} + KL^2\end{aligned}$$

where the first inequality follows the analysis of algorithm for each epoch, the last inequality follows the invariant maintained within each phase and the constant KL^2 is due to burning the first iteration of each epoch. We now try to upper bound the last term. We first note that $\sum_{k=1}^{K} \eta_k^{-1} = \sum_{k=1}^{K-1} \eta_0^{-1}2^k + \eta_0^{-1}2^K = \eta_0^{-1}(2^K - 1) + \eta_0^{-1}2^K \leq$

$\eta_0^{-1}2^{K+1}$. Furthermore, from b_K, we know that $\eta_{K-1}\sum_{t=b_{K-1}}^{b_K}\|\nabla f_{t+1}(\mathbf{z}_t)-\nabla f_t(\mathbf{z}_t)\|_2^2 \geq L^2\eta_{K-1}^{-1}$ since the b_K is the first iteration within epoch $K-1$ which violates the invariant. Also, from the monotonicity of gradual variation one can obtain that $\eta_{K-1}\mathrm{EGV}_{T,2} \geq \eta_{K-1}\sum_{t=b_{K-1}}^{b_K}\|\nabla f_{t+1}(\mathbf{z}_t)-\nabla f_t(\mathbf{z}_t)\|_2^2 \geq L^2\eta_{K-1}^{-1}$ which indicates $\eta_{K-1}^{-1} \leq \sqrt{\mathrm{EGV_{T,2}}}/L$. Putting these together, from (5.20) we obtain:

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}}\sum_{t=1}^{T} f_t(\mathbf{w}) \leq 2L\frac{2^{K+1}}{\eta_0} + KL^2 \leq 8\sqrt{\mathrm{EGV}_{T,2}} + KL^2. \tag{5.20}$$

It remains to bound the number of epochs in terms of $\mathrm{EGV}_{T,2}$. A simple idea would be to set K to be $\lfloor \log \mathrm{EGV}_{T,2} \rfloor + 1$, since it is the maximum number of epochs that could exists. Alternatively, we can also bound K in terms of $\sqrt{\mathrm{EGV}_{T,2}}$ which worsen the constant factor in the bound but results in a bound similar to one obtained by setting optimal η.

5.5 Bandit Online Mirror Prox with Gradual Variation Bounds

Online convex optimization becomes more challenging when the learner only receives partial feedback about the cost functions. One common scenario of partial feedback is that the learner only receives the cost $f_t(\mathbf{w}_t)$ at the predicted point $\mathbf{w}_t$ but without observing the entire cost function $f_t(\cdot)$. This setup is usually referred to as bandit setting, and the related online learning problem is called online bandit convex optimization.

Recently Hazan et al [70] extended the FTRL algorithm to online bandit *linear* optimization and obtained a variation-based regret bound in the form of $O(poly(d)\sqrt{\mathrm{Variation}_T \log(T)}+$

$poly(d\log(T)))$, where Variation$_T$ is the total variation of the cost vectors. We continue this line of work by proposing algorithms for general online bandit convex optimization with a variation-based regret bound. We present a deterministic algorithm for online bandit convex optimization by extending the OPM algorithm to a multi-point bandit setting, and prove the variation-based regret bound, which is optimal when the variation is independent of the number of trials. In our bandit setting , we assume we are allowed to query $d+1$ points around the decision point $\mathbf{w}_t$.

To develop a variation bound for online bandit convex optimization, we follow [5] by considering the multi-point bandit setting, where at each trial the player is allowed to query the cost functions at multiple points. We propose a deterministic algorithm to compete against the *completely adaptive* adversary that can choose the cost function $f_t(\mathbf{w})$ with the knowledge of $\mathbf{w}_1, \cdots, \mathbf{w}_t$. To approximate the gradient $\nabla f_t(\mathbf{w}_t)$, we query the cost function to obtain the cost values at $f_t(\mathbf{w}_t)$, and $f_t(\mathbf{w}_t+\delta\mathbf{e}_i), i = 1, \cdots, d$, where $\mathbf{e}_i$ is the ith standard base in $\mathbb{R}^d$. Then we compute the estimate of the gradient $\nabla f_t(\mathbf{w}_t)$ by

$$\mathbf{g}_t = \frac{1}{\delta}\sum_{i=1}^{d}\left(f_t(\mathbf{w}_t+\delta\mathbf{e}_i) - f_t(\mathbf{w}_t)\right)\mathbf{e}_i. \tag{5.21}$$

It can be shown that [5], under the smoothness assumption in (5.8),

$$\|\mathbf{g}_t - \nabla f_t(\mathbf{w}_t)\|_2 \leq \frac{\sqrt{d}L\delta}{2}. \tag{5.22}$$

To prove the regret bound, besides the smoothness assumption of the cost functions, and the boundness assumption about the domain $\mathcal{W} \subseteq \mathcal{B}$, we further assume that (i) there exists $r \leq 1$ such that $r\mathcal{B} \subseteq \mathcal{W} \subseteq \mathcal{B}$, and (ii) the cost function themselves are Lipschitz continuous, i.e., there exists a constant G such that

Algorithm 6 Deterministic Online Bandit Convex Optimization

```
Input: $\eta$, $\alpha$, $\delta > 0$
Initialize:: $\mathbf{z}_0 = \mathbf{0}$ and $f_0(\mathbf{w}) = 0$
for $t = 1, \ldots, T$ do
    Compute $\mathbf{w}_t$ by
        $\mathbf{w}_t = \arg\min_{\mathbf{w} \in (1-\alpha)\mathcal{W}} \left\{ \langle \mathbf{w}, \mathbf{g}_{t-1} \rangle + \frac{G}{2\eta} \|\mathbf{w} - \mathbf{z}_{t-1}\|_2^2 \right\}$
    Observe $f_t(\mathbf{w}_t), f_t(\mathbf{w}_t + \delta \mathbf{e}_i), i = 1, \cdots, d$
    Update $\mathbf{z}_t$ by
        $\mathbf{z}_t = \arg\min_{\mathbf{w} \in (1-\alpha)\mathcal{W}} \left\{ \langle \mathbf{w}, \mathbf{g}_t \rangle + \frac{G}{2\eta} \|\mathbf{w} - \mathbf{z}_{t-1}\|_2^2 \right\}$
end for
```

$$|f_t(\mathbf{w}) - f_t(\mathbf{w}')| \leq G\|\mathbf{w} - \mathbf{w}'\|_2, \forall \mathbf{w}, \mathbf{w}' \in \mathcal{W}, \forall t. \tag{5.23}$$

For our purpose, we define another gradual variation of cost functions by

$$\mathrm{EGV}_T^c = \sum_{t=0}^{T-1} \max_{\mathbf{w} \in \mathcal{W}} |f_{t+1}(\mathbf{w}) - f_t(\mathbf{w})|. \tag{5.24}$$

Unlike the gradual variation defined in (5.9) that uses the gradient of the cost functions, the gradual variation in (5.24) is defined according to the values of cost functions. The reason why we bound the regret by the gradual variation defined in (5.24) by the values of the cost functions rather than the one defined in (5.9) by the gradient of the cost functions is that in the bandit setting, we only have point evaluations of the cost functions. The following theorem states the regret bound for Algorithm 6.

Theorem 5.13. *Let $f_t(\cdot), t = 1, \ldots, T$ be a sequence of G-Lipschitz continuous convex functions, and their gradients are L-Lipschitz continuous. By setting $\delta = \sqrt{\frac{4d \max(\sqrt{2}G, \sqrt{\mathrm{EGV}_T^c})}{(\sqrt{d}L + G(1 + 1/r))T}}$, $\eta = \frac{\delta}{4d} \min\left\{\frac{1}{\sqrt{2}}, \frac{G}{\sqrt{\mathrm{EGV}_T^c}}\right\}$, and $\alpha = \delta/r$, we have the following regret bound for Algo-*

rithm 6

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq 4\sqrt{\max\left(\sqrt{2}G, \sqrt{\mathrm{EGV}_T^c}\right) d\,(dL + G/r)\,T}. \tag{5.25}$$

Remark 5.14. *Similar to the regret bound in [5](Theorem 9), Algorithm 6 also gives the optimal regret bound $O(\sqrt{T})$ when the variation is independent of the number of trials. Our regret bound has a better dependence on d (i.e., d) compared with the regret bound in [5] (i.e., d^2).*

5.6 Proofs of Gradual Variation

5.6.1 Proof of Theorem 5.3

To prove Theorem 5.3, we first present the following lemma.

Lemma 5.15. *Let $f_1, f_2, \ldots, f_T$ be a sequence of convex functions with L-Lipschitz continuous gradients. By running Algorithm 3 over T trials, we have*

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) \leq \min_{\mathbf{w}\in\mathcal{W}} \left[\frac{L}{2\eta}\|\mathbf{w}\|_2^2 + \sum_{t=1}^{T} f_t(\mathbf{z}_{t-1}) + \langle \mathbf{w} - \mathbf{z}_{t-1}, \nabla f_t(\mathbf{z}_{t-1})\rangle\right]$$
$$+ \frac{\eta}{2L}\sum_{t=0}^{T-1} \|\nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\|_2^2.$$

With this lemma, we can easily prove Theorem 5.3 by exploring the convexity of $f_t(\mathbf{w})$.

Proof of Theorem 5.3. By using $\|\mathbf{w}\|_2 \leq 1, \forall \mathbf{w} \in \mathcal{W} \subseteq \mathcal{B}$, and the convexity of $f_t(\mathbf{w})$, we have

$$\min_{\mathbf{w}\in\mathcal{W}} \left\{ \frac{L}{2\eta}\|\mathbf{w}\|_2^2 + \sum_{t=1}^{T} f_t(\mathbf{z}_{t-1}) + \langle \mathbf{w} - \mathbf{z}_{t-1}, \nabla f_t(\mathbf{z}_{t-1})\rangle \right\} \leq \frac{L}{2\eta} + \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}).$$

Combining the above result with Lemma 5.15, we have

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq \frac{L}{2\eta} + \frac{\eta}{2L}\mathrm{EGV}_{T,2}.$$

By choosing $\eta = \min(1, L/\sqrt{\mathrm{EGV}_{T,2}})$, we have the regret bound claimed in Theorem 5.3.

□

The Lemma 5.15 is proved by induction. The key to the proof is that $\mathbf{z}_t$ is the optimal solution to the strongly convex minimization problem in Lemma 5.15, i.e.,

$$\mathbf{z}_t = \arg\min_{\mathbf{w}\in\mathcal{W}} \left[\frac{L}{2\eta}\|\mathbf{w}\|_2^2 + \sum_{\tau=1}^{t} f_\tau(\mathbf{z}_{\tau-1}) + \langle \mathbf{w} - \mathbf{z}_{\tau-1}, \nabla f_\tau(\mathbf{z}_{\tau-1})\rangle \right]$$

Proof of Lemma 5.15. We prove the inequality by induction. When $T = 1$, we have $\mathbf{w}_1 = \mathbf{z}_0 = 0$ and

$$\begin{aligned}
&\min_{\mathbf{w}\in\mathcal{W}} \left[\frac{L}{2\eta}\|\mathbf{w}\|_2^2 + f_1(\mathbf{z}_0) + \langle \mathbf{w} - \mathbf{z}_0, \nabla f_1(\mathbf{z}_0)\rangle \right] + \frac{\eta}{2L}\|\nabla f_1(\mathbf{z}_0)\|_2^2 \\
&\geq f_1(\mathbf{z}_0) + \frac{\eta}{2L}\|\nabla f_1(\mathbf{z}_0)\|_2^2 + \min_{\mathbf{w}} \left\{ \frac{L}{2\eta}\|\mathbf{w}\|_2^2 + \langle \mathbf{w} - \mathbf{z}_0, \nabla f_1(\mathbf{z}_0)\rangle \right\} \\
&= f_1(\mathbf{z}_0) = f_1(\mathbf{w}_1).
\end{aligned}$$

where the inequality follows that by relaxing the minimization domain $\mathbf{w} \in \mathcal{W}$ to the whole space. We assume the inequality holds for t and aim to prove it for $t+1$. To this end, we define

$$\psi_t(\mathbf{w}) = \left[\frac{L}{2\eta}\|\mathbf{w}\|_2^2 + \sum_{\tau=1}^{t} f_\tau(\mathbf{z}_{\tau-1}) + \langle \mathbf{w} - \mathbf{z}_{\tau-1}, \nabla f_\tau(\mathbf{z}_{\tau-1})\rangle\right]$$
$$+ \frac{\eta}{2L}\sum_{\tau=0}^{t-1} \|\nabla f_{\tau+1}(\mathbf{z}_\tau) - \nabla f_\tau(\mathbf{z}_\tau)\|_2^2.$$

According to the updating procedure for $\mathbf{z}_t$ in step 6, we have $\mathbf{z}_t = \arg\min_{\mathbf{w}\in\mathcal{W}} \psi_t(\mathbf{w})$. Define $\phi_t = \psi_t(\mathbf{z}_t) = \min_{\mathbf{w}\in\mathcal{W}} \psi_t(\mathbf{w})$. Since $\psi_t(\mathbf{w})$ is a (L/η)-strongly convex function, we have

$$\begin{aligned}\psi_{t+1}(\mathbf{w}) - \psi_{t+1}(\mathbf{z}_t) &\geq \frac{L}{2\eta}\|\mathbf{w} - \mathbf{z}_t\|_2^2 + \langle \mathbf{w} - \mathbf{z}_t, \nabla\psi_{t+1}(\mathbf{z}_t)\rangle \\ &= \frac{L}{2\eta}\|\mathbf{w} - \mathbf{z}_t\|_2^2 + \langle \mathbf{w} - \mathbf{z}_t, \nabla\psi_t(\mathbf{z}_t) + \nabla f_{t+1}(\mathbf{z}_t)\rangle.\end{aligned}$$

Setting $\mathbf{w} = \mathbf{z}_{t+1} = \arg\min_{\mathbf{w}\in\mathcal{W}} \psi_{t+1}(\mathbf{w})$ in the above inequality results in

$$\begin{aligned}\psi_{t+1}(\mathbf{z}_{t+1}) - \psi_{t+1}(\mathbf{z}_t) &= \phi_{t+1} - (\phi_t + f_{t+1}(\mathbf{z}_t) + \frac{\eta}{2L}\|\nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\|_2^2) \\ &\geq \frac{L}{2\eta}\|\mathbf{z}_{t+1} - \mathbf{z}_t\|_2^2 + \langle \mathbf{z}_{t+1} - \mathbf{z}_t, \nabla\psi_t(\mathbf{z}_t) + \nabla f_{t+1}(\mathbf{z}_t)\rangle \qquad (5.26) \\ &\geq \frac{L}{2\eta}\|\mathbf{z}_{t+1} - \mathbf{z}_t\|_2^2 + \langle \mathbf{z}_{t+1} - \mathbf{z}_t, \nabla f_{t+1}(\mathbf{z}_t)\rangle,\end{aligned}$$

where the second inequality follows from the fact $\mathbf{z}_t = \arg\min_{\mathbf{w}\in\mathcal{W}} \psi_t(\mathbf{w})$, and therefore $(\mathbf{w} - \mathbf{z}_t)^\top \nabla\psi_t(\mathbf{z}_t) \geq 0, \forall \mathbf{w} \in \mathcal{W}$. Moving $f_{t+1}(\mathbf{z}_t)$ in the above inequality to the right hand side, we have

$$
\begin{aligned}
&\phi_{t+1} - \phi_t - \frac{\eta}{2L}\|\nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\|_2^2 \\
&\geq \frac{L}{2\eta}\|\mathbf{z}_{t+1} - \mathbf{z}_t\|_2^2 + \langle \mathbf{z}_{t+1} - \mathbf{z}_t, \nabla f_{t+1}(\mathbf{z}_t)\rangle + f_{t+1}(\mathbf{z}_t) \\
&\geq \min_{\mathbf{w}\in\mathcal{W}} \left\{ \frac{L}{2\eta}\|\mathbf{w} - \mathbf{z}_t\|_2^2 + \langle \mathbf{w} - \mathbf{z}_t, \nabla f_{t+1}(\mathbf{z}_t)\rangle + f_{t+1}(\mathbf{z}_t) \right\} \\
&= \min_{\mathbf{w}\in\mathcal{W}} \left\{ \underbrace{\frac{L}{2\eta}\|\mathbf{w} - \mathbf{z}_t\|_2^2 + \langle \mathbf{w} - \mathbf{z}_t, \nabla f_t(\mathbf{z}_t)\rangle}_{\rho(\mathbf{w})} + f_{t+1}(\mathbf{z}_t) + \underbrace{\langle \mathbf{w} - \mathbf{z}_t, \nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\rangle}_{r(\mathbf{w})} \right\}.
\end{aligned}
\tag{5.27}
$$

To bound the right hand side, we note that $\mathbf{w}_{t+1}$ is the minimizer of $\rho(\mathbf{w})$ by step 4 in Algorithm 3, and $\rho(\mathbf{w})$ is a L/η-strongly convex function, so we have

$$
\rho(\mathbf{w}) \geq \rho(\mathbf{w}_{t+1}) + \underbrace{\langle \mathbf{w} - \mathbf{w}_{t+1}, \nabla\rho(\mathbf{w}_{t+1})\rangle}_{\geq 0} + \frac{L}{2\eta}\|\mathbf{w} - \mathbf{w}_{t+1}\|_2^2 \geq \rho(\mathbf{w}_{t+1}) + \frac{L}{2\eta}\|\mathbf{w} - \mathbf{w}_{t+1}\|_2^2.
$$

Then we have

$$
\begin{aligned}
&\rho(\mathbf{w}) + f_{t+1}(\mathbf{z}_t) + r(\mathbf{w}) \geq \rho(\mathbf{w}_{t+1}) + f_{t+1}(\mathbf{z}_t) + \frac{L}{2\eta}\|\mathbf{w} - \mathbf{w}_{t+1}\|_2^2 + r(\mathbf{w}) \\
&= \underbrace{\frac{L}{2\eta}\|\mathbf{w}_{t+1} - \mathbf{z}_t\|_2^2 + \langle \mathbf{w}_{t+1} - \mathbf{z}_t, \nabla f_t(\mathbf{z}_t)\rangle}_{\rho(\mathbf{w}_{t+1})} + f_{t+1}(\mathbf{z}_t) + \frac{L}{2\eta}\|\mathbf{w} - \mathbf{w}_{t+1}\|_2^2 + r(\mathbf{w})
\end{aligned}
$$

Plugging above inequality into the inequality in (5.27), we have

$$\begin{aligned}
&\phi_{t+1} - \phi_t - \frac{\eta}{2L}\|\nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\|_2^2 \\
\geq & \frac{L}{2\eta}\|\mathbf{w}_{t+1} - \mathbf{z}_t\|_2^2 + \langle \mathbf{w}_{t+1} - \mathbf{z}_t, \nabla f_t(\mathbf{z}_t) + f_{t+1}(\mathbf{z}_t)\rangle \\
& + \min_{\mathbf{w}\in\mathcal{W}} \left\{ \frac{L}{2\eta}\|\mathbf{w} - \mathbf{w}_{t+1}\|_2^2 + \langle \mathbf{w} - \mathbf{z}_t, \nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\rangle \right\}
\end{aligned}$$

To continue the bounding, we proceed as follows

$$\begin{aligned}
&\phi_{t+1} - \phi_t - \frac{\eta}{2L}\|\nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\|_2^2 \\
\geq & \frac{L}{2\eta}\|\mathbf{w}_{t+1} - \mathbf{z}_t\|_2^2 + \langle \mathbf{w}_{t+1} - \mathbf{z}_t, \nabla f_t(\mathbf{z}_t)\rangle + f_{t+1}(\mathbf{z}_t) \\
& + \min_{\mathbf{w}\in\mathcal{W}} \left\{ \frac{L}{2\eta}\|\mathbf{w} - \mathbf{w}_{t+1}\|_2^2 + \langle \mathbf{w} - \mathbf{z}_t, \nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\rangle \right\} \\
= & \frac{L}{2\eta}\|\mathbf{w}_{t+1} - \mathbf{z}_t\|_2^2 + \langle \mathbf{w}_{t+1} - \mathbf{z}_t, \nabla f_{t+1}(\mathbf{z}_t)\rangle + f_{t+1}(\mathbf{z}_t) \\
& + \min_{\mathbf{w}\in\mathcal{W}} \left\{ \frac{L}{2\eta}\|\mathbf{w} - \mathbf{w}_{t+1}\|_2^2 + \langle \mathbf{w} - \mathbf{w}_{t+1}, \nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\rangle \right\} \\
\geq & \frac{L}{2\eta}\|\mathbf{w}_{t+1} - \mathbf{z}_t\|_2^2 + \langle \mathbf{w}_{t+1} - \mathbf{z}_t, \nabla f_{t+1}(\mathbf{z}_t)\rangle + f_{t+1}(\mathbf{z}_t) \\
& + \min_{\mathbf{w}} \left\{ \frac{L}{2\eta}\|\mathbf{w} - \mathbf{w}_{t+1}\|_2^2 + \langle \mathbf{w} - \mathbf{w}_{t+1}, \nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\rangle \right\} \\
= & \frac{L}{2\eta}\|\mathbf{w}_{t+1} - \mathbf{z}_t\|_2^2 + \langle \mathbf{w}_{t+1} - \mathbf{z}_t, \nabla f_{t+1}(\mathbf{z}_t)\rangle + f_{t+1}(\mathbf{z}_t) - \frac{\eta}{2L}\|\nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\|_2^2 \\
\geq & f_{t+1}(\mathbf{w}_{t+1}) - \frac{\eta}{2L}\|\nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\|_2^2,
\end{aligned}$$

where the first equality follows by writing $\langle \mathbf{w}_{t+1} - \mathbf{z}_t, \nabla f_t(\mathbf{z}_t)\rangle = \langle \mathbf{w}_{t+1} - \mathbf{z}_t, \nabla f_{t+1}(\mathbf{z}_t)\rangle - \langle \mathbf{w}_{t+1} - \mathbf{z}_t, \nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\rangle$ and combining with $\langle \mathbf{w} - \mathbf{z}_t, \nabla f_{t+1}(\mathbf{z}_t) - \nabla f_t(\mathbf{z}_t)\rangle$, and the last inequality follows from the smoothness condition of $f_{t+1}(\mathbf{w})$. Since by induction $\phi_t \geq \sum_{\tau=1}^{t} f_\tau(\mathbf{w}_\tau)$, we have $\phi_{t+1} \geq \sum_{\tau=1}^{t+1} f_\tau(\mathbf{w}_\tau)$. □

5.6.2 Proof of Theorem 5.5

To prove Theorem 5.5, we need the following lemma, which is the Lemma 3.1 in [117] stated in our notations.

Lemma 5.16 (**Lemma 3.1** [117])**.** *Let $\Phi(\mathbf{z})$ be a α-strongly convex function with respect to the norm $\|\cdot\|$, whose dual norm is denoted by $\|\cdot\|_*$, and $\mathsf{B}(\mathbf{w},\mathbf{z}) = \Phi(\mathbf{w}) - (\Phi(\mathbf{z}) + (\mathbf{w} - \mathbf{z})^\top \Phi'(\mathbf{z}))$ be the Bregman distance induced by function $\Phi(\mathbf{w})$. Let Z be a convex compact set, and $U \subseteq Z$ be convex and closed. Let $\mathbf{z} \in Z$, $\gamma > 0$, Consider the points,*

$$\mathbf{w} = \arg\min_{\mathbf{u}\in U} \gamma \mathbf{u}^\top \xi + \mathsf{B}(\mathbf{u},\mathbf{z}), \tag{5.28}$$

$$\mathbf{z}_+ = \arg\min_{\mathbf{u}\in U} \gamma \mathbf{u}^\top \zeta + \mathsf{B}(\mathbf{u},\mathbf{z}), \tag{5.29}$$

then for any $\mathbf{u} \in U$, we have

$$\gamma\zeta^\top(\mathbf{w}-\mathbf{u}) \leq \mathsf{B}(\mathbf{u},\mathbf{z}) - \mathsf{B}(\mathbf{u},\mathbf{z}_+) + \frac{\gamma^2}{\alpha}\|\xi-\zeta\|_*^2 - \frac{\alpha}{2}[\|\mathbf{w}-\mathbf{z}\|^2 + \|\mathbf{w}-\mathbf{z}_+\|^2].$$

In order not to have readers struggle with complex notations in [117] for the proof of Lemma 5.16, we present a detailed proof later in Appendix A.3 which is an adaption of the original proof to our notations.

Theorem 5.5 can be proved by using the above lemma, because the updates of $\mathbf{w}_t, \mathbf{z}_t$ can be written equivalently as (5.28). The proof below starts from (5.16) and bounds the summation of each term over $t = 1, \ldots, T$, respectively.

Proof of Theorem 5.5. First, we note that the two updates in step 4 and step 6 of Algorithm 4 fit in the Lemma 5.16 if we let $U = Z = \mathcal{W}$, $\mathbf{z} = \mathbf{z}_{t-1}$, $\mathbf{w} = \mathbf{w}_t$, $\mathbf{z}_+ = \mathbf{z}_t$, and $\Phi(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2$, which is 1-strongly convex function with respect to $\|\cdot\|_2$. Then $\mathsf{B}(\mathbf{u},\mathbf{z}) = \frac{1}{2}\|\mathbf{u}-\mathbf{z}\|_2^2$. As a result, the two updates for $\mathbf{w}_t, \mathbf{z}_t$ in Algorithm 4 are exactly the updates in (5.28) with $\mathbf{z} = \mathbf{z}_{t-1}, \gamma = \eta/L$, $\xi = \nabla f_{t-1}(\mathbf{z}_{t-1})$, and $\zeta = \nabla f_t(\mathbf{w}_t)$. Replacing these into (5.16), we have the following inequality for any $\mathbf{u} \in \mathcal{W}$,

$$\begin{aligned}\frac{\eta}{L}(\mathbf{w}_t - \mathbf{u})^\top \nabla f_t(\mathbf{w}_t) \le \frac{1}{2}\left(\|\mathbf{u}-\mathbf{z}_{t-1}\|_2^2 - \|\mathbf{u}-\mathbf{z}_t\|_2^2\right) \\ + \frac{\eta^2}{L^2}\|\nabla f_t(\mathbf{w}_t) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|_2^2 - \frac{1}{2}\left(\|\mathbf{w}_t - \mathbf{z}_{t-1}\|_2^2 + \|\mathbf{w}_t - \mathbf{z}_t\|_2^2\right)\end{aligned} \tag{5.30}$$

Then we have

$$\begin{aligned}&\frac{\eta}{L}(f_t(\mathbf{w}_t) - f_t(\mathbf{u})) \le \frac{\eta}{L}(\mathbf{w}_t - \mathbf{u})^\top \nabla f_t(\mathbf{w}_t) \le \frac{1}{2}\left(\|\mathbf{u}-\mathbf{z}_{t-1}\|_2^2 - \|\mathbf{u}-\mathbf{z}_t\|_2^2\right)\\ &+ \frac{2\eta^2}{L^2}\|\nabla f_t(\mathbf{w}_{t-1}) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|_2^2 + \frac{2\eta^2}{L^2}\|\nabla f_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_{t-1})\|_2^2\\ &- \frac{1}{2}\left(\|\mathbf{w}_t - \mathbf{z}_{t-1}\|_2^2 + \|\mathbf{w}_t - \mathbf{z}_t\|_2^2\right)\\ &\le \frac{1}{2}\left(\|\mathbf{u}-\mathbf{z}_{t-1}\|_2^2 - \|\mathbf{u}-\mathbf{z}_t\|_2^2\right) + \frac{2\eta^2}{L^2}\|\nabla f_t(\mathbf{w}_{t-1}) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|_2^2\\ &+ 2\eta^2\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 - \frac{1}{2}\left(\|\mathbf{w}_t - \mathbf{z}_{t-1}\|_2^2 + \|\mathbf{w}_t - \mathbf{z}_t\|_2^2\right),\end{aligned} \tag{5.31}$$

where the first inequality follows the convexity of $f_t(\mathbf{w})$, and the third inequality follows the smoothness of $f_t(\mathbf{w})$.

By taking the summation over $t = 1, \cdots, T$ with $\mathbf{z}^* = \arg\min_{\mathbf{u}\in\mathcal{W}} \sum_{t=1}^T f_t(\mathbf{u})$, and dividing both sides by η/L, we have

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1} f_t(\mathbf{w}) \le \frac{L}{2\eta} + \frac{2\eta}{L}\sum_{t=0}^{T-1} \|\nabla f_{t+1}(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t)\|_2^2 \\ + \sum_{t=1}^{T} 2\eta^2 \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 - \underbrace{\sum_{t=1}^{T} \frac{1}{2}\left(\|\mathbf{w}_t - \mathbf{z}_{t-1}\|_2^2 + \|\mathbf{w}_t - \mathbf{z}_t\|_2^2\right)}_{\triangleq B_T}$$

We can bound B_T as follows:

$$\begin{aligned} B_T &= \frac{1}{2}\sum_{t=1}^{T} \|\mathbf{w}_t - \mathbf{z}_{t-1}\|_2^2 + \frac{1}{2}\sum_{t=2}^{T+1} \|\mathbf{w}_{t-1} - \mathbf{z}_{t-1}\|_2^2 \\ &\ge \frac{1}{2}\sum_{t=2}^{T} \left(\|\mathbf{w}_t - \mathbf{z}_{t-1}\|_2^2 + \|\mathbf{w}_{t-1} - \mathbf{z}_{t-1}\|_2^2\right) \\ &\ge \frac{1}{4}\sum_{t=2}^{T} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 = \frac{1}{4}\sum_{t=1}^{T} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 \end{aligned}$$

where the last equality follows that $\mathbf{w}_1 = \mathbf{w}_0$. Plugging the above bound into (5.32), we have

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1} f_t(\mathbf{w}) \le \frac{L}{2\eta} + \frac{2\eta}{L}\sum_{t=0}^{T-1} \|\nabla f_{t+1}(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t)\|_2^2 \\ + \sum_{t=1}^{T} \left(2\eta^2 - \frac{1}{4}\right) \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2$$

We complete the proof by plugging the value of η. □

5.6.3 Proof of Corollary 5.12

We first have the key inequality in (5.19): for any $\mathbf{z} \in \mathcal{W}$

$$
\begin{aligned}
f_t(\mathbf{w}_t) - f_t(\mathbf{z}) \le\ & \mathsf{B}_t(\mathbf{z}, \mathbf{z}_{t-1}) - \mathsf{B}_t(\mathbf{z}, \mathbf{z}_t) - \beta\|\mathbf{w}_t - \mathbf{z}\|^2_{\mathbf{M}_t} \\
& + \|\nabla f_t(\mathbf{w}_t) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|^2_{\mathbf{H}_t^{-1}} - \frac{1}{2}\left[\|\mathbf{w}_t - \mathbf{z}_{t-1}\|^2_{\mathbf{H}_t} + \|\mathbf{w}_t - \mathbf{z}_t\|^2_{\mathbf{H}_t}\right].
\end{aligned}
$$

Taking summation over $t = 1, \ldots, T$, we have

$$
\begin{aligned}
\sum_{t=1}^T f_t(\mathbf{w}_t) - \sum_{t=1}^T f_t(\mathbf{z}) \le & \sum_{t=1}^T \underbrace{(\mathsf{B}_t(\mathbf{z}, \mathbf{z}_{t-1}) - \mathsf{B}_t(\mathbf{z}, \mathbf{z}_t))}_{\triangleq A_t} - \sum_{t=1}^T \underbrace{\beta\|\mathbf{w}_t - \mathbf{z}\|^2_{\mathbf{M}_t}}_{\triangleq C_t} \\
& + \sum_{t=1}^T \underbrace{\|\nabla f_t(\mathbf{w}_t) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|^2_{\mathbf{H}_t^{-1}}}_{\triangleq S_t} - \sum_{t=1}^T \underbrace{\frac{1}{2}\left[\|\mathbf{w}_t - \mathbf{z}_{t-1}\|^2_{\mathbf{H}_t} + \|\mathbf{w}_t - \mathbf{z}_t\|^2_{\mathbf{H}_t}\right]}_{\triangleq B_t}
\end{aligned}
$$

Next we bound each term individually. First,

$$
\sum_{t=1}^T A_t = \mathsf{B}_1(\mathbf{z}, \mathbf{z}_0) - \mathsf{B}_T(\mathbf{z}, \mathbf{z}_T) + \sum_{t=1}^{T-1} (\mathsf{B}_{t+1}(\mathbf{z}, \mathbf{z}_t) - \mathsf{B}_t(\mathbf{z}, \mathbf{z}_t)) \tag{5.32}
$$

Note that $\mathsf{B}_1(\mathbf{z}, \mathbf{z}_0) = \frac{1}{2}(1 + \beta G^2)\|\mathbf{z}\|_2^2 \le \frac{1}{2}(1 + \beta G^2)$ for any $\mathbf{z} \in \mathcal{W}$, and $\mathsf{B}_{t+1}(\mathbf{z}, \mathbf{z}_t) - \mathsf{B}_t(\mathbf{w}, \mathbf{z}_t) = \frac{\beta}{2}\|\mathbf{z} - \mathbf{z}_t\|^2_{\mathbf{H}_t}$, therefore

$$
\sum_{t=1}^T A_t \le \frac{1}{2}(1 + \beta G^2) + \sum_{t=1}^T \frac{\beta}{2}\|\mathbf{z} - \mathbf{z}_t\|^2_{\mathbf{H}_t} \tag{5.33}
$$

Then

$$\begin{aligned}\sum_{t=1}^{T}(A_t - C_t) &\leq \frac{1}{2}(1+\beta G^2) + \sum_{t=1}^{T}\left[\frac{\beta}{2}\|\mathbf{z}-\mathbf{z}_t\|_{\mathbf{M}_t}^2 - \beta\|\mathbf{w}_t - \mathbf{z}\|_{\mathbf{M}_t}^2\right] \\ &\leq \frac{1}{2}(1+\beta G^2) + \sum_{t=1}^{T}\left[\beta\|\mathbf{z}-\mathbf{w}_t\|_{\mathbf{M}_t}^2 + \beta\|\mathbf{w}_t-\mathbf{z}_t\|_{\mathbf{M}_t}^2 - \beta\|\mathbf{w}_t - \mathbf{z}\|_{\mathbf{M}_t}^2\right] \\ &\leq \frac{1}{2}(1+\beta G^2) + \sum_{t=1}^{T}\beta\|\mathbf{w}_t-\mathbf{z}_t\|_{\mathbf{M}_t}^2 \leq \frac{1}{2}(1+\beta G^2) + \sum_{t=1}^{T}\|\mathbf{w}_t-\mathbf{z}_t\|_{\mathbf{H}_t}^2\end{aligned} \tag{5.34}$$

Noting the updates in (5.18) and from inequality in (A.9) in the proof of Lemma 5.5 in Appendix A.3, we can get

$$\|\mathbf{w}_t - \mathbf{z}_t\|_{\mathbf{H}_t} \leq \|\nabla f_t(\mathbf{w}_t) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|_{\mathbf{H}_t^{-1}}$$

Next, we bound $\sum_{t=1}^{T} B_t$.

$$\begin{aligned}\sum_{t=1}^{T} B_t &= \frac{1}{2}\sum_{t=0}^{T-1}\|\mathbf{w}_{t+1}-\mathbf{z}_t\|_{\mathbf{H}_{t+1}}^2 + \frac{1}{2}\sum_{t=1}^{T}\|\mathbf{w}_t-\mathbf{z}_t\|_{\mathbf{H}_t}^2 \\ &\geq \frac{1}{2}\sum_{t=1}^{T-1}\|\mathbf{w}_{t+1}-\mathbf{z}_t\|_{\mathbf{H}_t}^2 + \frac{1}{2}\sum_{t=1}^{T-1}\|\mathbf{w}_t-\mathbf{z}_t\|_{\mathbf{H}_t}^2 \\ &\geq \frac{1}{4}\sum_{t=1}^{T-1}\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_{\mathbf{H}_t}^2 \geq \frac{1}{4}\sum_{t=0}^{T-1}\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2 - \frac{1}{4}\|\mathbf{w}_1-\mathbf{w}_0\|_2^2 \\ &\geq \frac{1}{4}\sum_{t=0}^{T-1}\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2,\end{aligned} \tag{5.35}$$

where the last inequality follows that $\mathbf{w}_0 = \mathbf{w}_1 = 0$. Therefore,

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{z}) \leq \frac{1}{2}(1+\beta G^2) + 2\sum_{t=1}^{T} \|\nabla f_t(\mathbf{w}_t) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|_{\mathbf{H}_t^{-1}} - \frac{1}{4}\sum_{t=0}^{T-1} \|\mathbf{w}_{t+1} - \mathbf{w}_t\|_2^2 \tag{5.36}$$

To proceed, we need the following lemma.

Lemma 5.17. *We have*

$$\sum_{t=1}^{T} \|\nabla f_t(\mathbf{w}_t) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|_{\mathbf{H}_t^{-1}} \leq \frac{4d}{\beta} \ln \left(1 + \frac{\beta}{4}\sum_{t=1}^{T} \|\nabla f_t(\mathbf{w}_t) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|_2^2\right) \tag{5.37}$$

Thus,

$$\begin{aligned}
&\sum_{t=1}^{T} \|\nabla f_t(\mathbf{w}_t) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|_{\mathbf{H}_t^{-1}} \leq \frac{4d}{\beta} \ln \left(1 + \frac{\beta}{4}\sum_{t=1}^{T} \|\nabla f_t(\mathbf{w}_t) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|_2^2\right)\\
&\leq \frac{4d}{\beta} \ln \left(1 + \frac{\beta}{4}\sum_{t=1}^{T} \|\nabla f_t(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_{t-1}) + \nabla f_t(\mathbf{w}_{t-1}) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|_2^2\right)\\
&\leq \frac{4d}{\beta} \ln \left(1 + \frac{\beta}{2}\sum_{t=1}^{T} L^2\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 + \frac{\beta}{2}\sum_{t=1}^{T} \|\nabla f_t(\mathbf{w}_{t-1}) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|_2^2\right)\\
&\leq \frac{4d}{\beta} \ln \left(1 + \frac{\beta}{2}\sum_{t=0}^{T-1} L^2\|\mathbf{w}_{t+1} - \mathbf{w}_t\|_2^2 + \frac{\beta}{2}\sum_{t=0}^{T-1} \|\nabla f_{t+1}(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t)\|_2^2\right)\\
&\leq \frac{4d}{\beta} \ln \left(1 + \frac{\beta}{2}\sum_{t=0}^{T-1} L^2\|\mathbf{w}_{t+1} - \mathbf{w}_t\|_2^2 + \frac{\beta}{2}\mathrm{EGV_{T,2}}\right)
\end{aligned} \tag{5.38}$$

Then,

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{z}) \leq \frac{1}{2}(1+\beta G^2) + \frac{8d}{\beta}\ln\left(1 + \frac{\beta}{2}\sum_{t=0}^{T-1} L^2\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2 + \frac{\beta}{2}\mathrm{EGV}_{\mathrm{T},2}\right) - \frac{1}{4}\sum_{t=0}^{T-1}\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2 \tag{5.39}$$

Without loss of generality we assume $8dL^2 \geq 1$. Next, let us consider two cases. In the first case, we assume $\beta\mathrm{EGV}_{T,2} \leq 16dL^2$. Then

$$\begin{aligned}
&\frac{8d}{\beta}\ln\left(1 + \frac{\beta}{2}\sum_{t=0}^{T-1} L^2\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2 + \frac{\beta}{2}\mathrm{EGV}_{\mathrm{T},2}\right)\\
&\leq \frac{8d}{\beta}\ln\left(1 + \frac{\beta}{2}\sum_{t=0}^{T-1} L^2\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2 + 8dL^2\right)\\
&\leq \frac{8d}{\beta}\ln\left(\frac{\beta}{2}\sum_{t=0}^{T-1} L^2\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2 + 16dL^2\right)\\
&= \frac{8d}{\beta}\left[\ln 16dL^2 + \ln\left(\frac{\frac{\beta}{2}\sum_{t=0}^{T-1} L^2\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2}{16dL^2} + 1\right)\right]\\
&\leq \frac{8d}{\beta}\ln 16dL^2 + \frac{8d}{\beta}\frac{\frac{\beta}{2}\sum_{t=0}^{T-1} L^2\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2}{16dL^2} = \frac{8d}{\beta}\ln 16dL^2 + \frac{\sum_{t=0}^{T-1}\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2}{4}
\end{aligned} \tag{5.40}$$

where the last inequality follows $\ln(1+x) \leq x$ for $x \geq 0$. Then we get

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{z}) \leq \frac{1}{2}(1+\beta G^2) + \frac{8d}{\beta}\ln 16dL^2 \tag{5.41}$$

In the second case, we assume $\beta\mathrm{EGV}_{T,2} \geq 16dL^2 \geq 2$, then we have

$$\begin{aligned}
&\frac{8d}{\beta}\ln\left(1+\frac{\beta}{2}\sum_{t=0}^{T-1}L^2\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2+\frac{\beta}{2}\mathrm{EGV}_{\mathrm{T},2}\right)\\
&\leq\frac{8d}{\beta}\ln\left(\frac{\beta}{2}\sum_{t=0}^{T-1}L^2\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2+\beta\mathrm{EGV}_{T,2}\right)\\
&\leq\frac{8d}{\beta}\left[\ln(\beta\mathrm{EGV}_{T,2})+\ln\left(\frac{\frac{\beta}{2}\sum_{t=0}^{T-1}L^2\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2}{\beta\mathrm{EGV}_{T,2}}+1\right)\right] \qquad (5.42)\\
&=\frac{8d}{\beta}\ln(\beta\mathrm{EGV}_{T,2})+\frac{8d}{\beta}\frac{\frac{\beta}{2}\sum_{t=0}^{T-1}L^2\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2}{\beta\mathrm{EGV}_{T,2}}\\
&=\frac{8d}{\beta}\ln(\beta\mathrm{EGV}_{T,2})+\frac{4dL^2\sum_{t=0}^{T-1}\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2}{\beta\mathrm{EGV}_{T,2}}\\
&\leq\frac{8d}{\beta}\ln(\beta\mathrm{EGV}_{T,2})+\frac{\sum_{t=0}^{T-1}\|\mathbf{w}_{t+1}-\mathbf{w}_t\|_2^2}{4}
\end{aligned}$$

where the last inequality follows $\beta\mathrm{EGV}_{T,2}\geq 16dL^2$. Then we get

$$\sum_{t=1}^{T}f_t(\mathbf{w}_t)-\sum_{t=1}^{T}f_t(\mathbf{z})\leq\frac{1}{2}(1+\beta G^2)+\frac{8d}{\beta}\ln(\beta\mathrm{EGV}_{T,2}) \tag{5.43}$$

Thus, we complete the proof by combining the two cases.

Next, we prove Lemma 5.17. We need the following lemma, which can be proved by using Lemma 6 [68] and noting that $|\mathbf{I}+\sum_{\tau=1}^{t}\mathbf{u}_\tau\mathbf{u}_\tau^\top|\leq(1+\sum_{t=1}^{T}\|\mathbf{u}_t\|_2^2)^d$, where $|\cdot|$ denotes the determinant of a matrix.

Lemma 5.18. *Let $\mathbf{u}_1,\mathbf{u}_2,\cdots,\mathbf{u}_T\in\mathbb{R}^d$ be a sequence of vectors. Let $\mathbf{V}_t=\mathbf{I}+\sum_{\tau=1}^{t}\mathbf{u}_\tau\mathbf{u}_\tau^\top$. Then,*

$$\sum_{t=1}^{T}\mathbf{u}_t^\top\mathbf{V}_t^{-1}\mathbf{u}_t\leq d\ln\left(1+\sum_{t=1}^{T}\|\mathbf{u}_t\|_2^2\right) \tag{5.44}$$

To prove Lemma 5.17, we let $\mathbf{v}_t=\nabla f_t(\mathbf{w}_t), t=1,\ldots,T$ and $\mathbf{v}_0=0$. Then $\mathbf{H}_t=$

$\mathbf{I} + \beta G^2 \mathbf{I} + \beta \sum_{\tau=0}^{t-1} \mathbf{v}_\tau \mathbf{v}_\tau^\top$. Note that we assume $\|\nabla f_t(\mathbf{w})\|_2 \leq G$, therefore

$$\begin{aligned} \mathbf{H}_t \succeq \mathbf{I} + \beta \sum_{\tau=1}^{t} \mathbf{v}_\tau \mathbf{v}_\tau^\top &\succeq \mathbf{I} + \frac{\beta}{2} \sum_{\tau=1}^{t} (\mathbf{v}_\tau \mathbf{v}_\tau^\top + \mathbf{v}_{\tau-1} \mathbf{v}_{\tau-1}^\top) \\ &\succeq \mathbf{I} + \frac{\beta}{4} \sum_{\tau=1}^{t} (\mathbf{v}_\tau - \mathbf{v}_{\tau-1})(\mathbf{v}_\tau - \mathbf{v}_{\tau-1})^\top = \mathbf{V}_t \end{aligned} \tag{5.45}$$

Let $\mathbf{u}_t = (\sqrt{\beta}/2)(\mathbf{v}_t - \mathbf{v}_{t-1})$, then $\mathbf{V}_t = \mathbf{I} + \sum_{\tau=1}^{t} \mathbf{u}_\tau \mathbf{u}_\tau^\top$. By applying the above lemma, we have

$$\frac{\beta}{4} \sum_{t=1}^{T} (\mathbf{v}_t - \mathbf{v}_{t-1})^\top \mathbf{V}_t^{-1} (\mathbf{v}_t - \mathbf{v}_{t-1}) \leq d \ln \left(1 + \frac{\beta}{4} \sum_{t=1}^{T} \|\mathbf{v}_t - \mathbf{v}_{t-1}\|_2^2 \right) \tag{5.46}$$

Thus,

$$\begin{aligned} \sum_{t=1}^{T} (\mathbf{v}_\tau - \mathbf{v}_{\tau-1})^\top \mathbf{H}_t^{-1} (\mathbf{v}_\tau - \mathbf{v}_{\tau-1}) &\leq \sum_{t=1}^{T} (\mathbf{v}_\tau - \mathbf{v}_{\tau-1})^\top \mathbf{V}_t^{-1} (\mathbf{v}_\tau - \mathbf{v}_{\tau-1}) \\ &\leq \frac{4d}{\beta} \ln \left(1 + \frac{\beta}{4} \sum_{t=1}^{T} \|\mathbf{v}_t - \mathbf{v}_{t-1}\|_2^2 \right) \end{aligned} \tag{5.47}$$

5.6.4 Proof of Theorem 5.13

Let $h_t(\mathbf{w}) = f_t(\mathbf{w}) + \langle \mathbf{g}_t - \nabla f_t(\mathbf{w}_t), \mathbf{w} \rangle$. It is easy seen that $\nabla h_t(\mathbf{w}_t) = \mathbf{g}_t$. Followed by Lemma 5.16, we have for any $\mathbf{z} \in (1-\alpha)\mathcal{W}$,

$$\begin{aligned} &\frac{\eta}{G} \nabla h_t(\mathbf{w}_t)^\top (\mathbf{w}_t - \mathbf{z}) \leq \frac{1}{2} \left(\|\mathbf{z} - \mathbf{z}_{t-1}\|_2^2 - \|\mathbf{z} - \mathbf{z}_t\|_2^2 \right) + \frac{\eta^2}{G^2} \|\mathbf{g}_t - \mathbf{g}_{t-1}\|_2^2 \\ &- \frac{1}{2} \left(\|\mathbf{w}_t - \mathbf{z}_{t-1}\|_2^2 + \|\mathbf{w}_t - \mathbf{z}_t\|_2^2 \right) \end{aligned} \tag{5.48}$$

Taking summation over $t = 1, \ldots, T$, we have,

$$
\begin{aligned}
\sum_{t=1}^{T} \frac{\eta}{G} \nabla h_t(\mathbf{w}_t)^\top (\mathbf{w}_t - \mathbf{z}) \leq & \frac{\|\mathbf{z} - \mathbf{z}_0\|_2^2}{2} + \sum_{t=1}^{T} \frac{\eta^2}{G^2} \|\mathbf{g}_t - \mathbf{g}_{t-1}\|_2^2 \\
& - \sum_{t=1}^{T} \frac{1}{2} \left(\|\mathbf{w}_t - \mathbf{z}_{t-1}\|_2^2 + \|\mathbf{w}_t - \mathbf{z}_t\|_2^2 \right) \\
\leq & \frac{\|\mathbf{z} - \mathbf{z}_0\|_2^2}{2} + \sum_{t=1}^{T} \frac{\eta^2}{G^2} \|\mathbf{g}_t - \mathbf{g}_{t-1}\|_2^2 - \sum_{t=1}^{T} \frac{1}{4} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 \\
\leq & \frac{1}{2} + \sum_{t=1}^{T} \frac{\eta^2}{G^2} \|\mathbf{g}_t - \mathbf{g}_{t-1}\|_2^2 - \sum_{t=1}^{T} \frac{1}{4} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 \\
\leq & \frac{1}{2} + \sum_{t=1}^{T} \frac{2\eta^2}{G^2} \|\mathbf{g}_t - \mathbf{g}_{t-1}\|_2^2 + \sum_{t=1}^{T} \frac{2\eta^2}{G^2} \|\mathbf{g}_t - \mathbf{g}_{t-1}\|_2^2 \\
& - \sum_{t=1}^{T} \frac{1}{4} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2 \\
\leq & \frac{1}{2} + \frac{2\eta^2}{\delta^2 G^2} \sum_{t=1}^{T} \left\| \sum_{i=1}^{d} (f_t(\mathbf{w}_t + \delta \mathbf{e}_i) - f_t(\mathbf{w}_{t-1} + \delta \mathbf{e}_i)) \mathbf{e}_i - (f_t(\mathbf{w}_t) - f_t(\mathbf{w}_{t-1})) \mathbf{e}_i \right\|_2^2 \\
& + \frac{2\eta^2}{\delta^2 G^2} \sum_{t=1}^{T} \left\| \sum_{i=1}^{d} (f_t(\mathbf{w}_{t-1} + \delta \mathbf{e}_i) - f_{t-1}(\mathbf{w}_{t-1} + \delta \mathbf{e}_i)) \mathbf{e}_i - (f_t(\mathbf{w}_{t-1}) - f_{t-1}(\mathbf{w}_{t-1})) \mathbf{e}_i \right\|_2^2 \\
& - \sum_{t=1}^{T} \frac{1}{4} \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_2^2,
\end{aligned}
\tag{5.49}
$$

where the second inequality follows (5.32). Next, we bound the middle two terms in right hand side of the above inequality.

$$\begin{aligned}
&\sum_{t=1}^{T}\left\|\sum_{i=1}^{d}(f_t(\mathbf{w}_t+\delta\mathbf{e}_i)-f_t(\mathbf{w}_{t-1}+\delta\mathbf{e}_i))\mathbf{e}_i-(f_t(\mathbf{w}_t)-f_t(\mathbf{w}_{t-1}))\mathbf{e}_i\right\|_2^2\\
&\leq\sum_{t=1}^{T}2d\sum_{i=1}^{d}\Big(|f_t(\mathbf{w}_t+\delta\mathbf{e}_i)-f_t(\mathbf{w}_{t-1}+\delta\mathbf{e}_i)|^2+|f_t(\mathbf{w}_t)-f_t(\mathbf{w}_{t-1})|^2\Big)\\
&\leq\sum_{t=1}^{T}4d^2G^2\|\mathbf{w}_t-\mathbf{w}_{t-1}\|_2^2,
\end{aligned}\tag{5.50}$$

and

$$\begin{aligned}
&\sum_{t=1}^{T}\left\|\sum_{i=1}^{d}(f_t(\mathbf{w}_{t-1}+\delta\mathbf{e}_i)-f_{t-1}(\mathbf{w}_{t-1}+\delta\mathbf{e}_i))\mathbf{e}_i-(f_t(\mathbf{w}_{t-1})-f_{t-1}(\mathbf{w}_{t-1}))\mathbf{e}_i\right\|_2^2\\
&\leq\sum_{t=1}^{T}2d\sum_{i=1}^{d}\Big(|f_t(\mathbf{w}_{t-1}+\delta\mathbf{e}_i)-f_{t-1}(\mathbf{w}_{t-1}+\delta\mathbf{e}_i)|^2+|f_t(\mathbf{w}_{t-1})-f_{t-1}(\mathbf{w}_{t-1})|^2\Big)\\
&\leq\sum_{t=1}^{T}4d^2\max_{\mathbf{w}\in\mathcal{W}}|f_t(\mathbf{w})-f_{t-1}(\mathbf{w})|^2.
\end{aligned}\tag{5.51}$$

Then we have

$$\begin{aligned}
\sum_{t=1}^{T}\frac{\eta}{G}\nabla h_t(\mathbf{w}_t)^\top(\mathbf{w}_t-\mathbf{z})&\leq\frac{1}{2}+\frac{8d^2\eta^2}{\delta^2}\sum_{t=1}^{T}\|\mathbf{w}_t-\mathbf{w}_{t-1}\|_2^2\\
&+\frac{8d^2\eta^2}{\delta^2G^2}\sum_{t=1}^{T}\max_{\mathbf{w}\in\mathcal{W}}|f_t(\mathbf{w})-f_{t-1}(\mathbf{w})|^2-\sum_{t=1}^{T}\frac{1}{4}\|\mathbf{w}_t-\mathbf{w}_{t-1}\|_2^2\\
&\leq\frac{1}{2}+\frac{8d^2\eta^2}{\delta^2G^2}\sum_{t=1}^{T}\max_{\mathbf{w}\in\mathcal{W}}|f_t(\mathbf{w})-f_{t-1}(\mathbf{w})|^2
\end{aligned}\tag{5.52}$$

where the last inequality follows that $\eta\leq\delta/(4\sqrt{2}d)$. Then by using the convexity of $h_t(\mathbf{w})$ and dividing both sides by η/G, we have

$$\sum_{t=1}^{T} h_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} h_t((1-\alpha)\mathbf{w}) \leq \frac{G}{2\eta} + \frac{8\eta d^2}{G\delta^2}\mathrm{EVAR}_T^c \leq \frac{4d}{\delta}\max\left(\sqrt{2}G, \sqrt{\mathrm{EVAR}_T^c}\right) \tag{5.53}$$

Following the the proof of Theorem 8 in [5], we have

$$\begin{aligned}
&\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}) \leq \sum_{t=1}^{T} h_t(\mathbf{w}_t) - \sum_{t=1}^{T} h_t(\mathbf{w}) + \sum_{t=1}^{T} f_t(\mathbf{w}_t) - h_t(\mathbf{w}_t) - f_t(\mathbf{w}) + h_t(\mathbf{w}) \\
&\leq \sum_{t=1}^{T} h_t(\mathbf{w}_t) - \sum_{t=1}^{T} h_t(\mathbf{w}) + \sum_{t=1}^{T} \langle \mathbf{g}_t - \nabla f_t(\mathbf{w}_t), \mathbf{w} - \mathbf{w}_t\rangle \\
&\leq \sum_{t=1}^{T} h_t(\mathbf{w}_t) - \sum_{t=1}^{T} h_t(\mathbf{w}) + \sqrt{d}L\delta T
\end{aligned} \tag{5.54}$$

where the last inequality follows from the following facts:

$$\begin{aligned}
&\|\mathbf{g}_t - \nabla f_t(\mathbf{w}_t)]\|_2 \leq \frac{\sqrt{d}L\delta}{2} \\
&\|\mathbf{w} - \mathbf{w}_t\| \leq 2
\end{aligned} \tag{5.55}$$

Then we have

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t((1-\alpha)\mathbf{w}) \leq \frac{4d}{\delta}\max\left(\sqrt{2}G, \sqrt{\mathrm{EVAR}_T^c}\right) + \sqrt{d}L\delta T \tag{5.56}$$

By the Lipschitz continuity of $f_t(\mathbf{w})$, we have

$$\sum_{t=1}^{T} f_t((1-\alpha)\mathbf{w}) \leq \sum_{t=1}^{T} f_t(\mathbf{w}) + G\alpha T \tag{5.57}$$

The we get

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq \frac{4d}{\delta} \max\left(\sqrt{2}G, \sqrt{\mathrm{EVAR}_T^c}\right) + \delta\sqrt{d}LT + \alpha GT \tag{5.58}$$

Plugging the stated values of δ and α completes the proof.

5.7 Summary

In this chapter, we proposed two novel algorithms for online convex optimization that bound the regret by the gradual variation of consecutive cost functions. The first algorithm is an improvement of the FTRL algorithm, and the second algorithm is based on the mirror prox method. Both algorithms maintain two sequence of solution points, a sequence of decision points and a sequence of searching points, and share the same order of regret bound up to a constant. The online mirror prox method only requires to keep tracking of a single gradient of each cost function, while the improved FTRL algorithm needs to evaluate the gradient of each cost function at two points and maintain a sum of up-to-date gradients of the cost functions.

We note that a very recent work Chiang et al. [40] extends the prox method into a two-point bandit setting and achieves a similar regret bound in expectation as that in the full setting, i.e., $O\left(d^2\sqrt{\mathrm{EGV}_{\mathrm{T},2}\ln\mathrm{T}}\right)$ for smooth functions and $O\left(d^2\ln(\mathrm{EGV}_{\mathrm{T},2}+\ln\mathrm{T})\right)$ for smooth and strongly convex cost functions, where $\mathrm{EGV}_{\mathrm{T},2}$ is the gradual variation defined on the gradients of the cost functions. We would like to make a thought-provoking comparison between our regret bound and their regret bound for online bandit convex optimization with smooth cost functions. First, the gradual variation in our bandit setting is defined on the values of the cost functions in contrast to that defined on the gradients of the cost functions. Second, we query the cost function d times in contrast to 2 times in their algorithms, and as a tradeoff our regret bound has a better dependence on the number of dimensions (i.e., $O(d)$) than that (i.e., $O(d^2)$) of their regret bound. Third, our regret bound has an annoying factor of $\sqrt{T}$ in comparison with $\sqrt{\ln T}$ in theirs. Therefore, some open problems are how to achieve a lower order of dependence on d than d^2 in the two-point bandit setting, and how to remove the factor of $\sqrt{T}$ while keeping a small order of dependence on d in our multi-point bandit setting; and studying the two different types of gradual variations for bandit settings is a future work as well.

5.8 Bibliographic Notes

As is well known, a wide range of literature deals with the online decision making problem and there exist a number of regret-minimizing algorithms that have the optimal regret bound. The first distribution-free framework for sequential decision making was proposed by Hannan [65] which was rediscovered in [86]. Blackwell in his seminal paper [23] generalized the Hannan's result and concerned the problem of playing a repeated game with a vector-valued payoff function and gave a precise necessary and sufficient condition for when a set is approachable. The most well-known and successful work is probably the Hedge algorithm [58], which was a direct generalization of Littlestone and Warmuth's Weighted Majority (WM) algorithm [99]. Another algorithm for online decision making problem is the Vovk's so-called aggregating strategies [147]. Other recent studies include the improved theoretical bounds and the parameter-free hedging algorithm [39] and adaptive Hedge [53] for decision-theoretic online learning. We refer readers to the [38] for an in-depth discussion of this subject.

As we already discussed in Chapter 2, over the past decade many algorithms have been proposed for online convex optimization, especially for online linear optimization. As the first seminal paper in online convex optimization, Zinkevich [158] proposed a gradient descent algorithm with a regret bound of $O(\sqrt{T})$. When cost functions are strongly convex, the regret bound of the online gradient descent algorithm is reduced to $O(\log T)$ with appropriately chosen step size [68]. Another common methodology for online convex optimization, especially for online linear optimization, is based on the framework of Follow the Leader (FTL). FTL chooses $\mathbf{w}_t$ by minimizing the cost incurred by $\mathbf{w}_t$ in all previous trials. Since the naive FTL algorithm fails to achieve sublinear regret in the worst case, many variants have been developed to fix the problem, including Follow The Perturbed Leader (FTPL) [86], Follow The Regularized Leader (FTRL) [3], and Follow The Approximate Leader (FTAL) [68]. Other methodologies for online convex optimization introduce a potential function (or link function) to map solutions between the space of primal variables and the space of dual variables, and carry out primal-dual update based on the potential function. The well-known Exponentiated Gradient (EG) algorithm [89] or Multiplicative Weights algorithm [99, 58] belong to this category. We note that these different algorithms are closely related. For example, in online linear optimization, the potential-based primal-dual algorithm is equivalent to FTRL algorithm.

Chapter 6

Gradual Variation for Composite Losses

This chapter continues our investigation and analysis of online learning methods which can lead to better regret bounds in gradually evolving environments. The results we have obtained in Chapter 5 rely on the assumption that the cost functions are smooth. Additionally, we showed that for general non-smooth functions when the only information presented to the learner is the first order information about the cost functions, it is impossible to obtain a regret bounded by gradual variation. However, in this chapter, we show that a gradual variation bound is achievable for a special class of non-smooth functions that is composed of a smooth component and a non-smooth component.

We consider two categories for the non-smooth component. In the first category, we assume that the non-smooth component is a fixed function and is relatively easy such that the composite gradient mapping can be solved without too much computational overhead compared to gradient mapping. A common example that falls into this category is to consider a non-smooth regularizer. For example, in addition to the basic domain $\mathcal{W}$, one would enforce the sparsity constraint on the decision vector $\mathbf{w}$, i.e., $\|\mathbf{w}\|_0 \leq k < d$, which is important in feature selection. However, the sparsity constraint $\|\mathbf{w}\|_0 \leq k$ is a non-convex function, and is usually implemented by adding a ℓ_1 regularizer $\lambda\|\mathbf{w}\|_1$ to the objective function, where

$\lambda > 0$ is a regularization parameter. Therefore, at each iteration the cost function is given by $f_t(\mathbf{w}) + \lambda\|\mathbf{w}\|_1$. To prove a regret bound by gradual variation for this type of non-smooth optimization, we first present a simplified version of the general online mirror prox method from Chapter 5 and show that it has the exactly same regret bound as stated in Chapter 5, and then extend the algorithm to the non-smooth optimization with a fixed non-smooth component.

In the second category, we assume that the non-smooth component can be written as an explicit maximization structure. In general, we consider a time-varying non-smooth component, present a primal-dual prox method, and prove a min-max regret bound by gradual variation. When the non-smooth components are equal across all trials, the usual regret is bounded by the min-max bound plus a variation in the non-smooth component. To see an application of min-max regret bound, we consider the problem of online classification with hinge loss and show that the number of mistakes can be bounded by a variation in sequential examples.

Before moving to the detailed analysis, it is worth mentioning that several pieces of works have proposed algorithms for optimizing the two types of non-smooth functions as described above to obtain an optimal convergence rate of $O(1/T)$ [122, 123]. Therefore, the existence of a regret bound by gradual variation for these two types of non-smooth optimization does not violate the contradictory argument in Section 5.3.

6.1 Composite Losses with a Fixed Non-smooth Component

6.1.1 A Simplified Online Mirror Prox Algorithm

In this subsection, we present a simplified version of online mirror prox (OMP) method algorithm proposed in Chapter 5, which is the foundation for us to develop the algorithm for non-smooth optimization.

The key trick is to replace the domain constraint $\mathbf{w} \in \mathcal{W}$ with a non-smooth function in the objective. Let $\delta_{\mathcal{W}}(\mathbf{w})$ denote the indicator function of the domain $\mathcal{W}$, i.e.,

$$\delta_{\mathcal{W}}(\mathbf{w}) = \begin{cases} 0, & \mathbf{w} \in \mathcal{W} \\ \infty, & \text{otherwise} \end{cases}$$

Then the proximal gradient mapping for updating $\mathbf{w}$ (step 4) in Algorihtm 5 is equivalent to

$$\mathbf{w}_t = \arg\min_{\mathbf{w}} \langle \mathbf{w}, \nabla f_{t-1}(\mathbf{w}_{t-1}) \rangle + \frac{L}{\eta} \mathsf{B}(\mathbf{w}, \mathbf{z}_{t-1}) + \delta_{\mathcal{W}}(\mathbf{w}).$$

By the first order optimality condition, there exists a sub-gradient $\mathbf{v}_t \in \partial \delta_{\mathcal{W}}(\mathbf{w}_t)$ such that

$$\nabla f_{t-1}(\mathbf{w}_{t-1}) + \frac{L}{\eta}(\nabla \Phi(\mathbf{w}_t) - \nabla \Phi(\mathbf{z}_{t-1})) + \mathbf{v}_t = 0. \tag{6.1}$$

Thus, $\mathbf{w}_t$ is equal to

$$\mathbf{w}_t = \arg\min_{\mathbf{w}} \langle \mathbf{w}, \nabla f_{t-1}(\mathbf{w}_{t-1}) + \mathbf{v}_t \rangle + \frac{L}{\eta} \mathsf{B}(\mathbf{w}, \mathbf{z}_{t-1}). \tag{6.2}$$

Then we can change the update for $\mathbf{z}_t$ to

$$\mathbf{z}_t = \arg\min_{\mathbf{w}} \langle \mathbf{w}, \nabla f_t(\mathbf{w}_t) + \mathbf{v}_t \rangle + \frac{L}{\eta} \mathsf{B}(\mathbf{w}, \mathbf{z}_{t-1}). \tag{6.3}$$

The key ingredient of above update compared to step 6 in Algorithm 5 is that we explicitly use the sub-gradient $\mathbf{v}_t$ that satisfies the optimality condition for $\mathbf{w}_t$ instead of solving a domain constrained optimization problem. The advantage of updating $\mathbf{z}_t$ by (6.3) is that we can easily compute $\mathbf{z}_t$ by the first order optimality condition, i.e.,

$$\nabla f_t(\mathbf{w}_t) + \mathbf{v}_t + \frac{L}{\eta}(\nabla \Phi(\mathbf{z}_t) - \nabla \Phi(\mathbf{z}_{t-1})) = 0. \tag{6.4}$$

Note that Eq. (6.1) indicates $\mathbf{v}_t = -\nabla f_{t-1}(\mathbf{w}_{t-1}) - \nabla \Phi(\mathbf{w}_t) + \nabla \Phi(\mathbf{z}_{t-1})$. By plugging this into (6.4), we reach to the following simplified update for $\mathbf{z}_t$,

$$\nabla \Phi(\mathbf{z}_t) = \nabla \Phi(\mathbf{w}_t) + \frac{\eta}{L}(\nabla f_{t-1}(\mathbf{w}_{t-1}) - \nabla f_t(\mathbf{w}_t)).$$

The simplified version of Algorithm 5 is presented in Algorithm 7.

Remark 6.1. *We make three remarks for Algorithm 7. First, the searching point $\mathbf{z}_t$ does not necessarily belong to the domain $\mathcal{W}$, which is usually not a problem given that the decision point $\mathbf{w}_t$ is always in $\mathcal{W}$. Nevertheless, the update can be followed by a projection step $\mathbf{z}_t = \min_{\mathbf{w} \in \mathcal{W}} \mathsf{B}(\mathbf{w}, \mathbf{z}_t')$ to ensure the searching point also stay in the domain $\mathcal{W}$, where we slightly abuse the notation $\mathbf{z}_t'$ in $\nabla \Phi(\mathbf{z}_t') = \nabla \Phi(\mathbf{w}_t) + \frac{\eta}{L}(\nabla f_{t-1}(\mathbf{w}_{t-1}) - \nabla f_t(\mathbf{w}_t))$.*

Algorithm 7 A Simplified General Online Mirror Prox Method

```
Input: $\eta > 0, \Phi(\mathbf{z})$
Initialize:: $\mathbf{z}_0 = \mathbf{w}_0 = \min_{\mathbf{z} \in \mathcal{W}} \Phi(\mathbf{z})$ and $f_0(\mathbf{w}) = 0$
for $t = 1, \ldots, T$ do
    Predict $\mathbf{w}_t$ by
        $$\mathbf{w}_t = \arg\min_{\mathbf{w} \in \mathcal{W}} \left\{ \langle \mathbf{w}, \nabla f_{t-1}(\mathbf{w}_{t-1}) \rangle + \frac{L}{\eta} \mathsf{B}(\mathbf{w}, \mathbf{z}_{t-1}) \right\}$$
    Receive cost function $f_t(\cdot)$ and incur loss $f_t(\mathbf{w}_t)$
    Update $\mathbf{z}_t$ by solving
        $$\nabla \Phi(\mathbf{z}_t) = \nabla \Phi(\mathbf{w}_t) + \frac{\eta}{L}(\nabla f_{t-1}(\mathbf{w}_{t-1}) - \nabla f_t(\mathbf{w}_t)).$$
end for
```

Second, the update in step 6 can be implemented by [38, Chapter 11]:

$$\mathbf{z}_t = \nabla \Phi^*(\nabla \Phi(\mathbf{w}_t) + \frac{\eta}{L}(\nabla f_{t-1}(\mathbf{w}_{t-1}) - \nabla f_t(\mathbf{w}_t))),$$

where $\Phi^*(\cdot)$ is the Legendre-Fenchel conjugate of $\Phi(\cdot)$ (see Appendix **??** *for definition). For example, when $\Phi(\mathbf{w}) = 1/2\|\mathbf{w}\|_2^2$, $\Phi^*(\mathbf{w}) = 1/2\|\mathbf{w}\|_2^2$ and the update for the searching point is given by*

$$\mathbf{z}_t = \mathbf{w}_t + (\ \eta/L)(\nabla f_{t-1}(\mathbf{w}_{t-1}) - \nabla f_t(\mathbf{w}_t));$$

when $\Phi(\mathbf{w}) = \sum_i w_i \ln w_i$, $\Phi^*(\mathbf{w}) = \log \left[\sum_i \exp(w_i)\right]$ and the update for the searching point can be computed by

$$[\mathbf{z}_t]_i \propto [\mathbf{w}_t]_i \exp\left(\eta/L[\nabla f_{t-1}(\mathbf{w}_{t-1}) - \nabla f_t(\mathbf{w}_t)]\right), \quad s.t. \quad \sum_i [\mathbf{z}_t]_i = 1.$$

Third, the key inequality in (5.16) for proving the regret bound still hold for $\zeta = \nabla f_t(\mathbf{w}_t) + \mathbf{v}_t$, $\xi = \nabla f_{t-1}(\mathbf{w}_{t-1}) + \mathbf{v}_t$ by noting the equivalence between the pairs (6.2, 6.3) and (5.28, 5.29), which is given below:

$$\frac{\eta}{L}\langle \nabla f_t(\mathbf{w}_t) + \mathbf{v}_t, \mathbf{w}_t - \mathbf{w}\rangle \leq \mathsf{B}(\mathbf{w}, \mathbf{z}_{t-1}) - \mathsf{B}(\mathbf{w}, \mathbf{z}_t)$$
$$+ \frac{\gamma^2}{\alpha}\|\nabla f_t(\mathbf{w}_t) - \nabla f_{t-1}(\mathbf{w}_{t-1})\|_*^2 - \frac{\alpha}{2}[\|\mathbf{w}_t - \mathbf{z}_{t-1}\|^2 + \|\mathbf{w} - \mathbf{z}_t\|^2], \forall \mathbf{w} \in \mathcal{W},$$

where $\mathbf{v}_t \in \partial\delta_{\mathcal{W}}(\mathbf{w}_t)$. As a result, we can apply the same analysis as in the proof of Theorem 5.5 to obtain the same regret bound in Theorem 5.6 for Algorithm 7. Note that the above inequality remains valid even if we take a projection step after the update for $\mathbf{z}_t'$ due to the generalized pythagorean inequality $\mathsf{B}(\mathbf{w}, \mathbf{z}_t) \leq \mathsf{B}(\mathbf{w}, \mathbf{z}_t'), \forall \mathbf{w} \in \mathcal{W}$ [38].

6.1.2 A Gradual Variation Bound for Online Non-Smooth Optimization

In spirit of Algorithm 7, we present an algorithm for online non-smooth optimization of functions $f_t(\mathbf{w}) = \widehat{f}_t(\mathbf{w}) + g(\mathbf{w})$ with a regret bound by gradual variation as:

$$\mathrm{EGV}_T = \sum_{t=0}^{T-1} \|\nabla f_{t+1}(\mathbf{w}_t) - \nabla f_t(\mathbf{w}_t)\|_*^2.$$

The trick is to solve the composite gradient mapping:

$$\mathbf{w}_t = \arg\min_{\mathbf{w}\in\mathcal{W}} \langle \mathbf{w}, \nabla f_{t-1}(\mathbf{w}_{t-1})\rangle + \frac{L}{\eta}\mathsf{B}(\mathbf{w}, \mathbf{z}_{t-1}) + g(\mathbf{w})$$

and update $\mathbf{z}_t$ by

$$\nabla\Phi(\mathbf{z}_t) = \nabla\Phi(\mathbf{w}_t) + \frac{\eta}{L}(\nabla f_{t-1}(\mathbf{w}_{t-1}) - \nabla f_t(\mathbf{w}_t)).$$

Algorithm 8 Online Mirror Prox Method with a Fixed Non-Smooth Component

```
1: Input: $\eta > 0, \Phi(\mathbf{z})$
2: Initialization: $\mathbf{z}_0 = \mathbf{w}_0 = \min_{\mathbf{z}\in\mathcal{W}} \Phi(\mathbf{z})$ and $f_0(\mathbf{w}) = 0$
3: for $t = 1, \ldots, T$ do
4:     Predict $\mathbf{w}_t$ by
           $$\mathbf{w}_t = \arg\min_{\mathbf{w}\in\mathcal{W}} \left\{ \langle \mathbf{w}, \nabla f_{t-1}(\mathbf{w}_{t-1})\rangle + \frac{L}{\eta}\mathsf{B}(\mathbf{w}, \mathbf{z}_{t-1}) + g(\mathbf{w}) \right\}$$
5:     Receive cost function $f_t(\cdot)$ and incur loss $f_t(\mathbf{w}_t)$
6:     Update $\mathbf{z}_t$ by
           $$\mathbf{z}'_t = \nabla\Phi^* \left( \nabla\Phi(\mathbf{w}_t) + \frac{\eta}{L}(\nabla f_{t-1}(\mathbf{w}_{t-1}) - \nabla f_t(\mathbf{w}_t)) \right)$$
       and $\mathbf{z}_t = \min_{\mathbf{w}\in\mathcal{W}} \mathsf{B}(\mathbf{w}, \mathbf{z}'_t)$
7: end for
```

Algorithm 8 shows the detailed steps and Corollay 6.2 states the regret bound, which can be proved similarly.

Corollary 6.2. *Let $f_t(\mathbf{w}) = \widehat{f}_t(\mathbf{w}) + g(\mathbf{w}), t = 1, \ldots, T$ be a sequence of convex functions where $\widehat{f}_t(\mathbf{w})$ are L-smooth continuous w.r.t $\|\cdot\|$ and $g(\mathbf{w})$ is a non-smooth function, $\Phi(\mathbf{z})$ be a α-strongly convex function w.r.t $\|\cdot\|$, and* EGV_T *be defined in (5.13). By setting $\eta = (1/2)\min\left\{\sqrt{\alpha}/\sqrt{2}, LR/\sqrt{\mathrm{EGV}_T}\right\}$, we have the following regret bound*

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \leq 2R\max\left(\sqrt{2}LR/\sqrt{\alpha}, \sqrt{\mathrm{EGV}_T}\right).$$

6.2 Composite Losses with an Explicit Max Structure

In previous subsection, we assume the composite gradient mapping with the non-smooth component can be efficiently solved. Here, we replace this assumption with an explicit max structure of the non-smooth component.

In what follows, we present a primal-dual prox method for such non-smooth cost functions and prove its regret bound. We consider a general setting, where the non-smooth functions $f_t(\mathbf{w})$ has the following structure:

$$f_t(\mathbf{w}) = \widehat{f}_t(\mathbf{w}) + \max_{\mathbf{u}\in\mathcal{Q}} \langle \mathbf{A}_t\mathbf{w}, \mathbf{u}\rangle - \widehat{\phi}_t(\mathbf{u}), \tag{6.5}$$

where $\widehat{f}_t(\mathbf{w})$ and $\widehat{\phi}_t(\mathbf{u})$ are L_1-smooth and L_2-smooth functions, respectively, and $\mathbf{A}_t \in \mathbb{R}^{m\times d}$ is a matrix used to characterize the non-smooth component of $f_t(\mathbf{w})$ with $-\widehat{\phi}_t(\mathbf{u})$ by maximization. Similarly, we define a dual cost function $\phi_t(\mathbf{u})$ as

$$\phi_t(\mathbf{u}) = -\widehat{\phi}_t(\mathbf{u}) + \min_{\mathbf{x}\in\mathcal{W}} \langle \mathbf{A}_t\mathbf{w}, \mathbf{u}\rangle + \widehat{f}_t(\mathbf{w}). \tag{6.6}$$

We refer to $\mathbf{w}$ as the primal variable and to $\mathbf{u}$ as the dual variable. To motivate the setup, let us consider online classification with hinge loss $\ell_t(\mathbf{w}) = \max(0, 1 - y_t\langle \mathbf{w}, \mathbf{x}_t\rangle)$, where we slightly abuse the notation $(\mathbf{x}_t, y_t)$ to denote the attribute and label pair received at trial t. It is straightforward to see that $\ell_t(\mathbf{w})$ is a non-smooth function and can be cast into the form in (6.5) by

$$\ell_t(\mathbf{w}) = \max_{\alpha\in[0,1]} \alpha(1 - y_t\mathbf{x}_t^\top\mathbf{w}) = \max_{\alpha\in[0,1]} -\alpha y_t\mathbf{x}_t^\top\mathbf{w} + \alpha.$$

To present the algorithm and analyze its regret bound, we introduce some notations. Let $F_t(\mathbf{w}, \mathbf{u}) = \widehat{f}_t(\mathbf{w}) + \langle \mathbf{A}_t\mathbf{w}, \mathbf{u}\rangle - \widehat{\phi}_t(\mathbf{u})$, $\Phi_1(\mathbf{w})$ be a α_1-strongly convex function defined on the primal variable $\mathbf{w}$ w.r.t a norm $\|\cdot\|_p$ and $\Phi_2(\mathbf{u})$ be a α_2-strongly convex function defined on the dual variable $\mathbf{u}$ w.r.t a norm $\|\cdot\|_q$. Correspondingly, let $\mathsf{B}_1(\mathbf{w}, \mathbf{z})$ and $\mathsf{B}_2(\mathbf{u}, \mathbf{v})$ denote the induced Bregman distance, respectively. We assume the domains $\mathcal{W}$, $\mathcal{Q}$ are bounded and matrices $\mathbf{A}_t$ have a bounded norm, i.e.,

$$\begin{aligned}
&\max_{\mathbf{w}\in\mathcal{W}} \|\mathbf{w}\|_p \leq R_1, \quad \max_{\mathbf{u}\in\mathcal{Q}} \|\mathbf{u}\|_q \leq R_2 \\
&\max_{\mathbf{w}\in\mathcal{W}} \Phi_1(\mathbf{w}) - \min_{\mathbf{w}\in\mathcal{W}} \Phi_1(\mathbf{w}) \leq M_1 \\
&\max_{\mathbf{u}\in\mathcal{Q}} \Phi_2(\mathbf{u}) - \min_{\mathbf{u}\in\mathcal{Q}} \Phi_2(\mathbf{u}) \leq M_2 \\
&\|\mathbf{A}_t\|_{p,q} = \max_{\|\mathbf{w}\|_p\leq 1, \|\mathbf{u}\|_q\leq 1} \mathbf{u}^\top \mathbf{A}_t \mathbf{w} \leq \sigma.
\end{aligned} \tag{6.7}$$

Let $\|\cdot\|_{p,*}$ and $\|\cdot\|_{q,*}$ denote the dual norms to $\|\cdot\|_p$ and $\|\cdot\|_q$, respectively. To prove a variational regret bound, we define a gradual variation as follows:

$$\begin{aligned}
\mathrm{EGV}_{T,p,q} = &\sum_{t=0}^{T-1} \|\nabla \widehat{f}_{t+1}(\mathbf{w}_t) - \nabla \widehat{f}_t(\mathbf{w}_t)\|_{p,*}^2 + (R_1^2 + R_2^2) \sum_{t=0}^{T-1} \|\mathbf{A}_t - \mathbf{A}_{t-1}\|_{p,q}^2 \\
&+ \sum_{t=0}^{T-1} \|\nabla \widehat{\phi}_{t+1}(\mathbf{u}_t) - \nabla \widehat{\phi}_t(\mathbf{u}_t)\|_{q,*}^2.
\end{aligned}$$

Given above notations, Algorithm 9 shows the detailed steps and Theorem 6.3 states a min-max bound.

Theorem 6.3. *Let $f_t(\mathbf{w}) = \widehat{f}_t(\mathbf{w}) + \max_{\mathbf{u}\in\mathcal{Q}}\langle \mathbf{A}_t\mathbf{w}, \mathbf{u}\rangle - \widehat{\phi}_t(\mathbf{u}), t = 1, \ldots, T$ be a sequence of non-smooth functions. Assume $\widehat{f}_t(\mathbf{w}), \widehat{\phi}(\mathbf{u})$ are $L = \max(L_1, L_2)$-smooth functions and the domain $\mathcal{W}, \mathcal{Q}$ and $\mathbf{A}_t$ satisfy the boundness condition as in (6.7). Let $\Phi(\mathbf{w})$ be a α_1-strongly convex function w.r.t the norm $\|\cdot\|_p$, $\Phi(\mathbf{u})$ be a α_2-strongly convex function w.r.t. the norm $\|\cdot\|_q$, and $\alpha = \min(\alpha_1, \alpha_2)$. By setting $\eta = \min\left(\sqrt{M_1 + M_2}/(2\sqrt{\mathrm{EGV}_{T,p,q}}), \sqrt{\alpha}/(4\sqrt{\sigma^2 + L^2})\right)$ in Algorithm 9, then we have*

Algorithm 9 Online Mirror Prox Method with an Explicit Max Structure

```
Input: $\eta > 0, \Phi_1(\mathbf{z}), \Phi_2(\mathbf{v})$
Initialization: $\mathbf{z}_0 = \mathbf{w}_0 = \min_{\mathbf{z}\in\mathcal{W}} \Phi_1(\mathbf{z})$, $\mathbf{v}_0 = \mathbf{u}_0 = \min_{\mathbf{v}\in\mathcal{Q}} \Phi_2(\mathbf{v})$ and $\widehat{f}_0(\mathbf{w}) = \widehat{\phi}_0(\mathbf{u}) = 0$
for $t = 1, \ldots, T$ do
    Update $\mathbf{u}_t$ by
        $\mathbf{u}_t = \arg\max_{\mathbf{u}\in\mathcal{Q}} \left\{ \langle \mathbf{u}, \mathbf{A}_{t-1}\mathbf{w}_{t-1} - \nabla\widehat{\phi}_{t-1}(\mathbf{u}_{t-1})\rangle - \frac{L_2}{\eta}\mathsf{B}_2(\mathbf{u}, \mathbf{v}_{t-1}) \right\}$
    Predict $\mathbf{w}_t$ by
        $\mathbf{w}_t = \arg\min_{\mathbf{w}\in\mathcal{W}} \left\{ \langle \mathbf{w}, \nabla\widehat{f}_{t-1}(\mathbf{w}_{t-1}) + \mathbf{A}_{t-1}\mathbf{A}_{t-1}^{\top}\mathbf{u}_{t-1}\rangle + \frac{L_1}{\eta}\mathsf{B}_1(\mathbf{w}, \mathbf{z}_{t-1}) \right\}$
    Receive cost function $f_t(\cdot)$ and incur loss $f_t(\mathbf{w}_t)$
    Update $\mathbf{v}_t$ by
        $\mathbf{v}_t = \arg\max_{\mathbf{u}\in\mathcal{Q}} \left\{ \langle \mathbf{u}, \mathbf{A}_t\mathbf{w}_t - \nabla\widehat{\phi}_t(\mathbf{u}_t)\rangle - \frac{L_2}{\eta}\mathsf{B}_2(\mathbf{u}, \mathbf{v}_{t-1}) \right\}$
    Update $\mathbf{z}_t$ by
        $\mathbf{z}_t = \arg\min_{\mathbf{w}\in\mathcal{W}} \left\{ \langle \mathbf{w}, \nabla\widehat{f}_t(\mathbf{w}_t) + \mathbf{A}_t^{\top}\mathbf{u}_t\rangle + \frac{L_1}{\eta}\mathsf{B}_1(\mathbf{w}, \mathbf{z}_{t-1}) \right\}$
end for
```

$$
\begin{aligned}
&\max_{\mathbf{u}\in\mathcal{Q}} \sum_{t=1}^{T} F_t(\mathbf{w}_t, \mathbf{u}) - \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} F_t(\mathbf{w}, \mathbf{u}_t) \\
&\leq 4\sqrt{M_1 + M_2}\max\left(2\sqrt{\frac{(M_1 + M_2)(\sigma^2 + L^2)}{\alpha}}, \sqrt{\mathrm{EGV}_{T,p,q}}\right).
\end{aligned}
$$

To facilitate understanding, we break the proof into several lemmas. The following lemma is by analogy with Lemma 5.16.

Lemma 6.4. *Let $\theta = \begin{pmatrix}\mathbf{w}\\ \mathbf{u}\end{pmatrix}$ denote a single vector with a norm $\|\theta\| = \sqrt{\|\mathbf{w}\|_p^2 + \|\mathbf{u}\|_q^2}$ and a dual norm $\|\theta\|_* = \sqrt{\|\mathbf{w}\|_{p*}^2 + \|\mathbf{u}\|_{q*}^2}$. Let $\Phi(\theta) = \Phi_1(\mathbf{w}) + \Phi_2(\mathbf{u})$, $\mathsf{B}(\theta, \zeta) = \mathsf{B}_1(\mathbf{w}, \mathbf{u}) + \mathsf{B}_2(\mathbf{z}, \mathbf{v})$. Then*

$$
\begin{aligned}
&\eta \begin{pmatrix} \nabla_{\mathbf{w}} F_t(\mathbf{w}_t, \mathbf{u}_t) \\ -\nabla_{\mathbf{u}} F_t(\mathbf{w}_t, \mathbf{u}_t) \end{pmatrix}^{\top} \begin{pmatrix} \mathbf{w}_t - \mathbf{w} \\ \mathbf{u}_t - \mathbf{u} \end{pmatrix} \leq \mathsf{B}(\theta, \zeta_{t-1}) - \mathsf{B}(\theta, \zeta_t) \\
&+ \eta^2 \left(\|\nabla_{\mathbf{w}} F_t(\mathbf{w}_t, \mathbf{u}_t) - \nabla_{\mathbf{w}} F_{t-1}(\mathbf{w}_{t-1}, \mathbf{u}_{t-1})\|_{p,*}^2 \right) \\
&+ \eta^2 \left(\|\nabla_{\mathbf{u}} F_t(\mathbf{w}_t, \mathbf{u}_t) - \nabla_{\mathbf{u}} F_{t-1}(\mathbf{w}_{t-1}, \mathbf{u}_{t-1})\|_{q,*}^2 \right) \\
&- \frac{\alpha}{2} \left(\|\mathbf{w}_t - \mathbf{z}_t\|_p^2 + \|\mathbf{u}_t - \mathbf{v}_t\|_q^2 + \|\mathbf{w}_t - \mathbf{z}_{t-1}\|_p^2 + \|\mathbf{u}_t - \mathbf{v}_{t-1}\|_q^2 \right).
\end{aligned}
$$

Proof. The updates of $(\mathbf{w}_t, \mathbf{u}_t)$ in Algorithm 9 can be seen as applying the updates in Lemma 5.16 with $\theta_t = \begin{pmatrix} \mathbf{w}_t \\ \mathbf{u}_t \end{pmatrix}$ in place of $\mathbf{w}$, $\zeta_t = \begin{pmatrix} \mathbf{z}_t \\ \mathbf{v}_t \end{pmatrix}$ in place of $\mathbf{z}_+$, $\zeta_{t-1} = \begin{pmatrix} \mathbf{z}_{t-1} \\ \mathbf{v}_{t-1} \end{pmatrix}$ in place of $\mathbf{z}$. Note that $\Phi(\theta)$ is a $\alpha = \min(\alpha_1, \alpha_2)$-strongly convex function with respect to the norm $\|\theta\|$. Then applying the results in Lemma 5.16 we can complete the proof. □

Applying the convexity of $F_t(\mathbf{w}, \mathbf{u})$ in terms of $\mathbf{w}$ and the concavity of $F_t(\mathbf{w}, \mathbf{u})$ in terms of $\mathbf{u}$ to the result in Lemma 6.4, we have

$$
\begin{aligned}
&\eta \left(F_t(\mathbf{w}_t, \mathbf{u}) - F_t(\mathbf{w}, \mathbf{u}_t) \right) \\
&\leq \mathsf{B}_1(\mathbf{w}, \mathbf{z}_{t-1}) - \mathsf{B}_1(\mathbf{w}, \mathbf{z}_t) + \mathsf{B}_2(\mathbf{u}, \mathbf{v}_{t-1}) - \mathsf{B}_2(\mathbf{u}, \mathbf{v}_t) \\
&\quad + \eta^2 \|\nabla \widehat{f}_t(\mathbf{w}_t) - \nabla \widehat{f}_{t-1}(\mathbf{w}_{t-1}) + \mathbf{A}_t^{\top} \mathbf{u}_t - \mathbf{A}_{t-1}^{\top} \mathbf{u}_{t-1}\|_{p,*}^2 \\
&\quad + \eta^2 \|\nabla \widehat{\phi}_t(\mathbf{u}_t) - \nabla \widehat{\phi}_{t-1}(\mathbf{u}_{t-1}) + \mathbf{A}_t \mathbf{w}_t - \mathbf{A}_{t-1} \mathbf{w}_{t-1}\|_{q,*}^2 \\
&\quad - \frac{\alpha}{2} \left(\|\mathbf{w}_t - \mathbf{z}_{t-1}\|_p^2 + \|\mathbf{w}_t - \mathbf{z}_t\|_p^2 \right) - \frac{\alpha}{2} \left(\|\mathbf{u}_t - \mathbf{v}_{t-1}\|_q^2 + \|\mathbf{u}_t - \mathbf{v}_t\|_q^2 \right) \qquad (6.8) \\
&\leq \mathsf{B}_1(\mathbf{w}, \mathbf{z}_{t-1}) - \mathsf{B}_1(\mathbf{w}, \mathbf{z}_t) + \mathsf{B}_2(\mathbf{u}, \mathbf{v}_{t-1}) - \mathsf{B}_2(\mathbf{u}, \mathbf{v}_t) \\
&\quad + 2\eta^2 \|\nabla \widehat{f}_t(\mathbf{w}_t) - \nabla \widehat{f}_{t-1}(\mathbf{w}_{t-1})\|_{p,*}^2 + 2\eta^2 \|\mathbf{A}_t^{\top} \mathbf{u}_t - \mathbf{A}_{t-1}^{\top} \mathbf{u}_{t-1}\|_{p,*}^2 \\
&\quad + 2\eta^2 \|\nabla \widehat{\phi}_t(\mathbf{u}_t) - \nabla \widehat{\phi}_{t-1}(\mathbf{u}_{t-1})\|_{q,*}^2 + 2\eta^2 \|\mathbf{A}_t \mathbf{w}_t - \mathbf{A}_{t-1} \mathbf{w}_{t-1}\|_{q,*}^2 \\
&\quad - \frac{\alpha}{2} \left(\|\mathbf{w}_t - \mathbf{z}_{t-1}\|_p^2 + \|\mathbf{w}_t - \mathbf{z}_t\|_p^2 \right) - \frac{\alpha}{2} \left(\|\mathbf{u}_t - \mathbf{v}_{t-1}\|_q^2 + \|\mathbf{u}_t - \mathbf{v}_t\|_q^2 \right)
\end{aligned}
$$

The following lemma provides tools for proceeding the bound.

Lemma 6.5.

$$\|\nabla \widehat{f}_t(\mathbf{w}_t) - \nabla \widehat{f}_{t-1}(\mathbf{w}_{t-1})\|_{p,*}^2 \leq 2\|\nabla \widehat{f}_t(\mathbf{w}_t) - \nabla \widehat{f}_{t-1}(\mathbf{w}_t)\|_{p,*}^2 + 2L^2\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_p^2$$

$$\|\nabla \widehat{\phi}_t(\mathbf{u}_t) - \nabla \widehat{\phi}_{t-1}(\mathbf{u}_{t-1})\|_{q,*}^2 \leq 2\|\nabla \widehat{\phi}_t(\mathbf{u}_t) - \nabla \widehat{\phi}_{t-1}(\mathbf{u}_t)\|_{q,*}^2 + 2L^2\|\mathbf{u}_t - \mathbf{u}_{t-1}\|_q^2$$

$$\|\mathbf{A}_t\mathbf{w}_t - \mathbf{A}_{t-1}\mathbf{w}_{t-1}\|_{q,*}^2 \leq 2R_1^2\|\mathbf{A}_t - \mathbf{A}_{t-1}\|_{p,q}^2 + 2\sigma^2\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_p^2$$

$$\|\mathbf{A}_t^\top\mathbf{u}_t - A_{t-1}^\top\mathbf{u}_{t-1}\|_{p,*}^2 \leq 2R_2^2\|\mathbf{A}_t - \mathbf{A}_{t-1}\|_{p,q}^2 + 2\sigma^2\|\mathbf{u}_t - \mathbf{u}_{t-1}\|_q^2$$

Proof. We prove the first and the third inequalities. Another two inequalities can be proved similarly.

$$\begin{aligned}
&\|\nabla \widehat{f}_t(\mathbf{w}_t) - \nabla \widehat{f}_{t-1}(\mathbf{w}_{t-1})\|_{p,*}^2 \\
&\leq 2\|\nabla \widehat{f}_t(\mathbf{w}_t) - \nabla \widehat{f}_{t-1}(\mathbf{w}_t)\|_{p,*}^2 + 2\|\nabla \widehat{f}_{t-1}(\mathbf{w}_t) - \nabla \widehat{f}_{t-1}(\mathbf{w}_{t-1})\|_{p,*}^2 \\
&\leq 2\|\nabla \widehat{f}_t(\mathbf{w}_t) - \nabla \widehat{f}_{t-1}(\mathbf{w}_t)\|_{p,*}^2 + 2L^2\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_p^2
\end{aligned}$$

where we use the smoothness of $\widehat{f}_t(\mathbf{w})$.

$$\begin{aligned}
\|\mathbf{A}_t\mathbf{w}_t - \mathbf{A}_{t-1}\mathbf{w}_{t-1}\|_{q,*}^2 &\leq 2\|(\mathbf{A}_t - \mathbf{A}_{t-1})\mathbf{w}_t\|_{q,*}^2 + 2\|\mathbf{A}_{t-1}(\mathbf{w}_t - \mathbf{w}_{t-1})\|_p^2 \\
&\leq 2R_1^2\|\mathbf{A}_t - \mathbf{A}_{t-1}\|_{p,q}^2 + 2\sigma^2\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_p^2
\end{aligned}$$

□

Lemma 6.6. *For any* $\mathbf{w} \in \mathcal{W}$ *and* $\mathbf{u} \in \mathcal{Q}$*, we have*

$$\begin{aligned}
&\eta \left(F_t(\mathbf{w}_t, \mathbf{u}) - F_t(\mathbf{w}, \mathbf{u}_t) \right) \\
&\leq \mathsf{B}_1(\mathbf{w}, \mathbf{z}_{t-1}) - \mathsf{B}_1(\mathbf{w}, \mathbf{z}_t) + \mathsf{B}_2(\mathbf{u}, \mathbf{v}_{t-1}) - \mathsf{B}_2(\mathbf{u}, \mathbf{v}_t) \\
&\quad + 4\eta^2 \Bigg(\|\nabla \widehat{f}_t(\mathbf{w}_t) - \nabla \widehat{f}_{t-1}(\mathbf{w}_t)\|_{p,*}^2 + \|\nabla \widehat{\phi}_t(\mathbf{u}_t) - \nabla \widehat{\phi}_{t-1}(\mathbf{u}_t)\|_{q,*}^2 \\
&\qquad\qquad + (R_1^2 + R_2^2)\|\mathbf{A}_t - \mathbf{A}_{t-1}\|_{p,q}^2 \Bigg) \\
&\quad + 4\eta^2\sigma^2 \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_p^2 + 4\eta^2 L^2 \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_p^2 - \frac{\alpha}{2}(\|\mathbf{w}_t - \mathbf{z}_t\|_p^2 + \|\mathbf{w}_t - \mathbf{z}_{t-1}\|_p^2) \\
&\quad + 4\eta^2\sigma^2 \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_p^2 + 4\eta^2 L^2 \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_q^2 - \frac{\alpha}{2}(\|\mathbf{u}_t - \mathbf{v}_t\|_q^2 + \|\mathbf{u}_t - \mathbf{v}_{t-1}\|_q^2).
\end{aligned}$$

Proof. The lemma can be proved by combining the results in Lemma 6.5 and the inequality in (6.8).

$$\begin{aligned}
&\eta \left(F_t(\mathbf{w}_t, \mathbf{u}) - F_t(\mathbf{w}, \mathbf{u}_t) \right) \\
&\leq \mathsf{B}_1(\mathbf{w}, \mathbf{z}_{t-1}) - \mathsf{B}_1(\mathbf{w}, \mathbf{z}_t) + \mathsf{B}_2(\mathbf{u}, \mathbf{v}_{t-1}) - \mathsf{B}_2(\mathbf{u}, \mathbf{v}_t) \\
&\quad + 2\eta^2 \|\nabla \widehat{f}_t(\mathbf{w}_t) - \nabla \widehat{f}_{t-1}(\mathbf{w}_{t-1})\|_{p,*}^2 + 2\eta^2 \|\mathbf{A}_t^\top \mathbf{u}_t - \mathbf{A}_{t-1}^\top \mathbf{u}_{t-1}\|_{p,*}^2 \\
&\quad + 2\eta^2 \|\nabla \widehat{\phi}_t(\mathbf{u}_t) - \nabla \widehat{\phi}_{t-1}(\mathbf{u}_{t-1})\|_{q,*}^2 + 2\eta^2 \|\mathbf{A}_t \mathbf{w}_t - \mathbf{A}_{t-1}\mathbf{w}_{t-1}\|_{q,*}^2 \\
&\quad - \frac{\alpha}{2}\left(\|\mathbf{w}_t - \mathbf{z}_{t-1}\|_p^2 + \|\mathbf{w}_t - \mathbf{z}_t\|_p^2 \right) - \frac{\alpha}{2}\left(\|\mathbf{u}_t - \mathbf{v}_{t-1}\|_q^2 + \|\mathbf{u}_t - \mathbf{v}_t\|_q^2 \right) \\
&\leq \mathsf{B}_1(\mathbf{w}, \mathbf{z}_{t-1}) - \mathsf{B}_1(\mathbf{w}, \mathbf{z}_t) + \mathsf{B}_2(\mathbf{u}, \mathbf{v}_{t-1}) - \mathsf{B}_2(\mathbf{u}, \mathbf{v}_t) \\
&\quad + 4\eta^2 \|\nabla \widehat{f}_t(\mathbf{w}_t) - \nabla \widehat{f}_{t-1}(\mathbf{w}_t)\|_{p,*}^2 + 4\eta^2 L^2 \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_p^2 + 4\eta^2\sigma^2 \|\mathbf{w}_t - \mathbf{w}_{t-1}\|_p^2 \\
&\quad + 4\eta^2 R_1^2 \|\mathbf{A}_t - \mathbf{A}_{t-1}\|_{p,q}^2 + 4\eta^2 R_2^2 \|\mathbf{A}_t - \mathbf{A}_{t-1}\|_{p,q}^2 \\
&\quad + 4\eta^2 \|\nabla \widehat{\phi}_t(\mathbf{u}_t) - \nabla \widehat{\phi}_{t-1}(\mathbf{u}_t)\|_{q,*}^2 + 4\eta^2 L^2 \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_q^2 + 4\eta^2\sigma^2 \|\mathbf{u}_t - \mathbf{u}_{t-1}\|_q^2 \\
&\quad - \frac{\alpha}{2}\left(\|\mathbf{w}_t - \mathbf{z}_{t-1}\|_p^2 + \|\mathbf{w}_t - \mathbf{z}_t\|_p^2 \right) - \frac{\alpha}{2}\left(\|\mathbf{u}_t - \mathbf{v}_{t-1}\|_q^2 + \|\mathbf{u}_t - \mathbf{v}_t\|_q^2 \right)
\end{aligned}$$

$$
\begin{aligned}
&\leq \mathsf{B}_1(\mathbf{w}, \mathbf{z}_{t-1}) - \mathsf{B}_1(\mathbf{w}, \mathbf{z}_t) + \mathsf{B}_2(\mathbf{u}, \mathbf{v}_{t-1}) - \mathsf{B}_2(\mathbf{u}, \mathbf{v}_t) \\
&\quad + 4\eta^2 \Bigg(\|\nabla \widehat{f}_t(\mathbf{w}_t) - \nabla \widehat{f}_{t-1}(\mathbf{w}_t)\|_{p,*}^2 + \|\nabla \widehat{\phi}_t(\mathbf{u}_t) - \nabla \widehat{\phi}_{t-1}(\mathbf{u}_t)\|_{q,*}^2 \\
&\qquad\quad + (R_1^2 + R_2^2)\|\mathbf{A}_t - \mathbf{A}_{t-1}\|_{p,q}^2 \Bigg) \\
&\quad + 4\eta^2\sigma^2\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_p^2 + 4\eta^2 L^2\|\mathbf{w}_t - \mathbf{w}_{t-1}\|_p^2 - \frac{\alpha}{2}(\|\mathbf{w}_t - \mathbf{z}_t\|_p^2 + \|\mathbf{w}_t - \mathbf{z}_{t-1}\|_p^2) \\
&\quad + 4\eta^2\sigma^2\|\mathbf{u}_t - \mathbf{u}_{t-1}\|_p^2 + 4\eta^2 L^2\|\mathbf{u}_t - \mathbf{u}_{t-1}\|_q^2 - \frac{\alpha}{2}(\|\mathbf{u}_t - \mathbf{v}_t\|_q^2 + \|\mathbf{u}_t - \mathbf{v}_{t-1}\|_q^2).
\end{aligned}
$$

□

Proof of Theorem 6.3. Taking summation of the inequalities in Lemma 6.6 over $t = 1, \ldots, T$, applying the inequality in (5.32) twice and using $\eta \leq \sqrt{\alpha}/(4\sqrt{\sigma^2 + L^2})$, we have

$$
\begin{aligned}
\sum_{t=1}^T F_t(\mathbf{w}_t, \mathbf{u}) - \sum_{t=1}^T F_t(\mathbf{w}, \mathbf{u}_t) &\leq 4\eta \mathrm{EGV}_{T,p,q} + \frac{M_1 + M_2}{\eta} \\
&= 4\sqrt{M_1 + M_2} \max\left(2\sqrt{\frac{(M_1 + M_2)(\sigma^2 + L^2)}{\alpha}}, \sqrt{\mathrm{EGV}_{T,p,q}} \right).
\end{aligned}
$$

We complete the proof by using

$$
\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^T F_t(\mathbf{w}_t, \mathbf{u})
$$

and

$$
\mathbf{u}^* = \arg\max_{\mathbf{u} \in \mathcal{Q}} \sum_{t=1}^T F_t(\mathbf{w}, \mathbf{u}_t)
$$

. □

As an immediate result of Theorem 6.3, the following Corollary states a regret bound for non-smooth optimization with a fixed non-smooth component that can be written as a max structure, i.e., $f_t(\mathbf{w}) = \widehat{f}_t(\mathbf{w}) + [g(\mathbf{w}) = \max_{\mathbf{u}\in\mathcal{Q}}\langle \mathbf{A}\mathbf{w}, \mathbf{u}\rangle - \widehat{\phi}(\mathbf{u})]$.

Corollary 6.7. *Let $f_t(\mathbf{w}) = \widehat{f}_t(\mathbf{w}) + g(\mathbf{w}), t = 1, \ldots, T$ be a sequence of non-smooth functions, where $g(\mathbf{w}) = \max_{\mathbf{u}\in\mathcal{Q}}\langle \mathbf{A}\mathbf{w}, \mathbf{u}\rangle - \widehat{\phi}(\mathbf{u})$, and the gradual variation* EGV_T *be defined in (5.13) w.r.t the dual norm $\|\cdot\|_{p,*}$. Assume $\widehat{f}_t(\mathbf{w})$ are L-smooth functions w.r.t $\|\cdot\|$, the domain $\mathcal{W}, \mathcal{Q}$ and* $\mathbf{A}$ *satisfy the boundness condition as in (6.7). If we set $\eta = \min\left(\sqrt{M_1 + M_2}/(2\sqrt{\mathrm{EGV}_T}), \sqrt{\alpha}/(4\sqrt{\sigma^2 + L^2})\right)$ in Algorithm 9, then we have the following regret bound*

$$\begin{aligned}&\sum_{t=1}^T f_t(\mathbf{w}_t) - \min_{\mathbf{w}\in\mathcal{W}}\sum_{t=1}^T f_t(\mathbf{w})\\ &\leq 4\sqrt{M_1 + M_2}\max\left(2\sqrt{\frac{(M_1 + M_2)(\sigma^2 + L^2)}{\alpha}}, \sqrt{\mathrm{EGV}_T}\right) + V(g, \mathbf{w}_{1:T}),\end{aligned}$$

where $\widehat{\mathbf{w}}_T = \sum_{t=1}^T \mathbf{w}_t/T$ and $V(g, \mathbf{w}_{1:T}) = \sum_{t=1}^T |g(\mathbf{w}_t) - g(\widehat{\mathbf{w}}_T)|$ measures the variation in the non-smooth component.

Proof. In the case of fixed non-smooth component, the gradual variation defined in (6.8) reduces the one defined in (5.13) w.r.t the dual norm $\|\cdot\|_{p,*}$. By using the bound in (6.9) and noting that $f_t(\mathbf{w}) = \max_{\mathbf{u}\in\mathcal{Q}} F_t(\mathbf{w}, \mathbf{u}) \geq F_t(\mathbf{w}, \mathbf{u}_t)$, we have

$$\begin{aligned}\sum_{t=1}^T \left(\widehat{f}_t(\mathbf{w}_t) + \langle \mathbf{A}\mathbf{w}_t, \mathbf{u}\rangle - \widehat{\phi}(\mathbf{u})\right) &\leq \sum_{t=1}^T f_t(\mathbf{w}) +\\ &4\sqrt{M_1 + M_2}\max\left(2\sqrt{\frac{(M_1 + M_2)(\sigma^2 + L^2)}{\alpha}}, \sqrt{\mathrm{EGV}_T}\right).\end{aligned}$$

Therefore

$$\sum_{t=1}^{T}\left(\widehat{f}_t(\mathbf{w}_t)+g(\widehat{\mathbf{w}}_T)\right)\leq\sum_{t=1}^{T}f_t(\mathbf{w})+$$
$$4\sqrt{M_1+M_2}\max\left(2\sqrt{\frac{(M_1+M_2)(\sigma^2+L^2)}{\alpha}},\sqrt{\mathrm{EGV}_T}\right).$$

We complete the proof by complementing $\widehat{f}_t(\mathbf{w}_t)$ with $g(\mathbf{w}_t)$ to obtain $f_t(\mathbf{w}_t)$ and moving the additional term $\sum_{t=1}^{T}(g(\widehat{\mathbf{w}}_T)-g(\mathbf{w}_t))$ to the right hand side . □

Remark 6.8. *Note that the regret bound in Corollary 6.7 has an additional term $V(g,\mathbf{w}_{1:T})$ compared to the regret bound in Corollary 6.2, which constitutes a tradeoff between the reduced computational cost in solving a composite gradient mapping.*

To see an application of Theorem 6.3 to an online non-smooth optimization with time-varying non-smooth components, let us consider the example of online classification with hinge loss. At each trial, upon receiving an example $\mathbf{x}_t$, we need to make a prediction based on the current model $\mathbf{w}_t$, i.e., $\widehat{y}_t=\langle\mathbf{w}_t,\mathbf{x}_t\rangle$, then we receive the true label of $\mathbf{x}_t$ denoted by $y_t\in\{+1,-1\}$. The goal is to minimize the total number of mistakes across the time line $M_T=\sum_{t=1}^{T}\mathbb{I}[\widehat{y}_t y_t\leq 0]$. Here we are interested in a scenario that the data sequence $(\mathbf{x}_t,y_t),t=1,\ldots,T$ has a small gradual variation in terms of $y_t\mathbf{x}_t$. To obtain such a gradual variation based mistake bound, we can apply Algorithm 9. For the purpose of deriving the mistake bound, we need to make a small change to Algorithm 9. At the beginning of each trial, we first make a prediction $\widehat{y}_t=\langle\mathbf{w}_t,\mathbf{x}_t\rangle$, and if we make a mistake $\mathbb{I}[\widehat{y}_t y_t\leq 0]$ the we proceed to update the auxiliary primal-dual pair $(\mathbf{w}_t',\beta_t)$ similar to $(\mathbf{z}_t,\mathbf{v}_t)$ in Algorithm 9 and the primal-dual pair $(\mathbf{w}_{t+1},\alpha_{t+1})$ similar to $(\mathbf{w}_{t+1},\mathbf{u}_{t+1})$ in Algorithm 9, which are given explicitly as follows:

$$\beta_t = \prod_{[0,1]} \left(\beta_{t-1} + \eta(1 - \mathbf{w}_t^\top y_t \mathbf{x}_t)\right), \quad \mathbf{w}'_t = \prod_{\|\mathbf{w}\|_2 \le R} (\mathbf{w}'_{t-1} + \eta \alpha_t y_t \mathbf{x}_t)$$

$$\alpha_{t+1} = \prod_{[0,1]} \left(\beta_t + \eta(1 - \mathbf{w}_t^\top y_t \mathbf{x}_t)\right), \quad \mathbf{w}_{t+1} = \prod_{\|\mathbf{w}\|_2 \le R} (\mathbf{w}'_t + \eta \alpha_t y_t \mathbf{x}_t).$$

Without loss of generality, we let $(\mathbf{x}_t, y_t), t = 1, \ldots, M_T$ denote the examples that are predicted incorrectly. The function $F_t(\cdot, \cdot)$ is written as $F_t(\mathbf{w}, \alpha) = \alpha(1 - y_t \langle \mathbf{w}, \mathbf{x}_t \rangle)$. Then for a total sequence of T examples, we have the following bound by assuming $\|\mathbf{w}_t\|_2 \le 1$ and $\eta \le 1/2\sqrt{2}$

$$\sum_{t=1}^{M_T} F_t(\mathbf{w}_t, \alpha) \le \sum_{t=1}^{M_T} \ell(y_t \mathbf{w}_t^\top \mathbf{x}_t) + \eta \sum_{t=0}^{M_T - 1} (R^2 + 1)\|y_{t+1}\mathbf{x}_{t+1} - y_t \mathbf{x}_t\|_2^2 + \frac{R^2 + \alpha^2}{2\eta}.$$

Since $y_t \mathbf{w}_t^\top \mathbf{x}_t$ is less than 0 for the incorrectly predicted examples, if we set $\alpha = 1$ in the above inequality, we have

$$\begin{aligned} M_T &\le \sum_{t=1}^{M_T} \ell(y_t \mathbf{w}_t^\top \mathbf{x}_t) + \eta \sum_{t=0}^{M_T - 1} (R^2 + 1)\|y_{t+1}\mathbf{x}_{t+1} - y_t \mathbf{x}_t\|_2^2 + \frac{R^2 + 1}{2\eta} \\ &\le \sum_{t=1}^{M_T} \ell(y_t \mathbf{w}_t^\top \mathbf{x}_t) + \sqrt{2}(R^2 + 1) \max(2, \sqrt{\mathrm{EGV}_{T,2}}). \end{aligned}$$

which results in a gradual variational mistake bound, where $\sqrt{\mathrm{EVG}_{T,2}}$ measures the gradual variation in the incorrectly predicted examples. To end the discussion, we note that one may find applications of a small gradual variation of $y_t \mathbf{x}_t$ in time series classification. For instance, if $\mathbf{x}_t$ represent some medical measurements of a person and y_t indicates whether the person observes a disease, since the health conditions usually change slowly then it is expected that the gradual variation of $y_t\mathbf{x}_t$ is small. Similarly, if $\mathbf{x}_t$ are some sensor measurements of an equipment and y_t indicates whether the equipment fails or not, we would also observe a small gradual variation of the sequence $y_t\mathbf{x}_t$ during a time period.

6.3 Summary

In this chapter we developed a simplified online mirror prox method using a composite gradient mapping for non-smooth optimization with a fixed non-smooth component and a primal-dual prox method for non-smooth optimization with the non-smooth component written as a max structure. Despite the impossibility result in Chapter 5 which demonstrated that smoothness of loss functions in necessary to obtain gradual variation bounds, we showed that a simplified version of online mirror prix method is able to attain regret bounded by gradual variation for loss functions with a smooth component and two types of mentioned non-smooth components.

Chapter 7

Mixed Optimization for Smooth Losses

In this part of the thesis, we consider stochastic convex optimization problem and show that leveraging the smoothness of functions allows us to devise stochastic optimization algorithms that enjoy faster convergence rate.

The focus of this chapter is on stochastic smooth optimization. The motivation for exploiting smoothness in stochastic optimization stems from the observation that the optimal convergence rate for stochastic optimization of smooth functions is $O(1/\sqrt{T})$, which is same as stochastic optimization of Lipschitz continuous convex functions. This is in contrast to optimizing smooth functions using full gradients, which yields a convergence rate of $O(1/T^2)$. Therefore, it is of great interest to exploit smoothness in stochastic setting as well. In particular, we are interested in designing an efficient algorithm that is in the same spirit of the stochastic gradient descent method, but can effectively leverage the smoothness of the loss function to achieve a significantly faster convergence rate.

We introduce a new setup for optimizing convex functions, termed as **mixed optimization**, which allows to access both a stochastic oracle and a full gradient oracle to take advantages of their individual merits. Our goal is to significantly improve the convergence rate of stochastic optimization of smooth functions by having an additional small number of accesses to the full gradient oracle. We show that, with an $O(\ln T)$ calls to the full gradient oracle and an $O(T)$ calls to the stochastic oracle, the proposed mixed optimization algorithm is able to achieve an optimization error of $O(1/T)$. The key insight underlying the mixed optimization paradigm is that by infrequent use of full gradients at specific points we are able to progressively reduce the variance of stochastic gradients which leads to faster convergence rates.

The rest of this chapter is organized as follows. In Section 7.1 we motivate the problem. Section 7.2 describes the MixedGrad algorithm, discusses the main intuition behind it, and states the main result on its convergence rate. The proof of convergence rate is given in Section 7.3 and the omitted proofs are deferred to Section 7.4. Section 7.5 concludes the paper and discusses few open questions. Finally, Section 7.6 briefly reviews the literature on deterministic and stochastic optimization.

7.1 Motivation

As it has been shown in Chapter 2, many practical machine learning algorithms follow the framework of empirical risk minimization, which often can be cast into the following generic optimization problem:

$$\min_{\mathbf{w}\in\mathcal{W}} \mathcal{F}(\mathbf{w}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w}), \tag{7.1}$$

where n is the number of training examples, $f_i(\mathbf{w})$ encodes the loss function related to the ith training example $(\mathbf{x}_i, y_i)$, and $\mathcal{W}$ is a bounded convex domain that is introduced to regularize the solution $\mathbf{w} \in \mathcal{W}$ (i.e., the smaller the size of $\mathcal{W}$, the stronger the regularization is). In this chapter, we focus on the learning problems for which the loss function $f_i(\mathbf{w})$ is smooth. Examples of smooth loss functions include least square with $f_i(\mathbf{w}) = (y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle)^2$ and logistic regression with $f_i(\mathbf{w}) = \log(1 + \exp(-y_i \langle \mathbf{w}, \mathbf{x}_i \rangle))$. Since the regularization is enforced through the restricted domain $\mathcal{W}$, we did not introduce a ℓ_2 regularizer $\lambda \|\mathbf{w}\|^2/2$ into the optimization problem and as a result, we do not assume the loss function to be strongly convex. We note that a small ℓ_2 regularizer does NOT improve the convergence rate of stochastic optimization. More specifically, the convergence rate for stochastically optimizing a ℓ_2 regularized loss function remains as $O(1/\sqrt{T})$ when $\lambda = O(1/\sqrt{T})$ [72], a scenario that is often encountered in real-world applications.

A preliminary approach for solving the optimization problem in (7.1) is the batch gradient descent (GD) algorithm [121]. It starts with some initial point, and iteratively updates the solution using the equation $\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}_t - \eta \nabla \mathcal{F}(\mathbf{w}_t))$, where $\Pi_{\mathcal{W}}(\cdot)$ is the orthogonal projection onto the convex domain $\mathcal{W}$. It has been shown that for smooth objective functions, the convergence rate of standard GD is $O(1/T)$ [121], and can be improved to $O(1/T^2)$ by an accelerated GD algorithm [120, 121, 123]. The main shortcoming of GD method is its high cost in computing the full gradient $\nabla \mathcal{F}(\mathbf{w}_t)$ when the number of training examples is large, i.e., it requires $O(n)$ gradient computations per iteration. Stochastic gradient descent (SGD) alleviates this limitation of GD by sampling one (or a small set of) examples and computing a stochastic (sub)gradient at each iteration based on the sampled examples [29, 115, 134]. Since the computational cost of SGD per iteration is independent of the size of the data (i.e., n), it is usually appealing for large-scale learning and optimization.

While SGD enjoys a high computational efficiency per iteration, it suffers from a slow convergence rate for optimizing smooth functions. It has been shown in [119] that the effect of the stochastic noise cannot be decreased with a better rate than $O(1/\sqrt{T})$ which is significantly worse than GD that uses the full gradients for updating the solutions and this limitation is also valid when the target function is smooth. In addition for general Lipschitz continuous convex functions, SGD exhibits the same convergence rate as that for the smooth functions, implying that smoothness of the loss function is essentially not very useful and can not be exploited in stochastic optimization. The slow convergence rate for stochastically optimizing smooth loss functions is mostly due to the variance in stochastic gradients: unlike the full gradient case where the norm of a gradient approaches to zero when the solution is approaching to the optimal solution, in stochastic optimization, the norm of a stochastic gradient is constant even when the solution is close to the optimal solution. It is the variance in stochastic gradients that makes the convergence rate $O(1/\sqrt{T})$ unimprovable for stochastic smooth optimization [119, 4].

In this chapter, we are interested in designing an efficient algorithm that is in the same spirit of SGD but can effectively leverage the smoothness of the loss function to achieve a significantly faster convergence rate. To this end, we consider a new setup for optimization that allows us to interplay between stochastic and deterministic gradient descent methods. In particular, we assume that the optimization algorithm has an access to two oracles:

- A stochastic oracle $\mathcal{O}_s$ that returns the loss function $f_i(\mathbf{w})$ based on the sampled training example $(\mathbf{x}_i, y_i)$ [1], and
- A full gradient oracle $\mathcal{O}_f$ that returns the gradient $\nabla \mathcal{F}(\mathbf{w})$ for any given solution $\mathbf{w} \in \mathcal{W}$.

We refer to this new setting as *mixed optimization* in order to distinguish it from both stochastic and full gradient optimization models. Obviously, the challenging issue in this

[1] We note that the stochastic oracle assumed in our study is slightly stronger than the stochastic gradient oracle as it returns the sampled function instead of the stochastic gradient.

regard is to minimize the number of full gradients to be as minimum as possible while having the same number of stochastic gradient accesses. The key question we examined in this chapter is:

"Is it possible to improve the convergence rate for stochastic optimization of smooth functions by having a small number of calls to the full gradient oracle $\mathcal{O}_f$?"

In this chapter we give an affirmative answer to this question. In particular, we show that with an additional $O(\ln T)$ accesses to the full gradient oracle $\mathcal{O}_f$, the proposed algorithm, referred to as MixedGrad, can improve the convergence rate for stochastic optimization of smooth functions to $O(1/T)$, the same rate for stochastically optimizing a strongly convex function [72, 125, 137]. The MixedGrad algorithm builds off on multi-stage methods [72] and operates in epochs, but involves novel ingredients so as to obtain an $O(1/T)$ rate for smooth losses. In particular, we form a sequence of strongly convex objective functions to be optimized at each epoch and decrease the amount of regularization and shrink the domain as the algorithm proceeds. The full gradient oracle $\mathcal{O}_f$ is only called at the beginning of each epoch. In this chapter our focus is only smooth functions and in the coming chapters, we show that the it is further possible to develop faster optimization schemes when the functions is both smooth and strongly convex by making the number of full gradients independent of condition number.

7.2 The MixedGrad Algorithm

In stochastic first-order optimization setting, instead of having direct access to $\mathcal{F}(\mathbf{w})$, we only have access to a stochastic gradient oracle, which given a solution $\mathbf{w} \in \mathcal{W}$, returns the gradient $\nabla f_i(\mathbf{w})$ where i is sampled uniformly at random from $\{1, 2, \cdots, n\}$. The goal of stochastic optimization to use a bounded number T of oracle calls, and compute some $\bar{\mathbf{w}} \in \mathcal{W}$ such that the optimization error, $\mathcal{F}(\bar{\mathbf{w}}) - \mathcal{F}(\mathbf{w}^*)$, is as small as possible.

In the mixed optimization model considered in this study, we first relax the stochastic oracle $\mathcal{O}_s$ by assuming that it will return a randomly sampled loss function $f_i(\mathbf{w})$, instead of the gradient $\nabla f_i(\mathbf{w})$ for a given solution $\mathbf{w}$ may be applied to achieve $O(1/T)$ convergence. Second, we assume that the learner also has an access to the full gradient oracle $\mathcal{O}_f$. Our goal is to significantly improve the convergence rate of stochastic gradient descent (SGD) by making a small number of calls to the full gradient oracle $\mathcal{O}_f$. In particular, we show that by having only $O(\log T)$ accesses to the full gradient oracle and $O(T)$ accesses to the stochastic oracle, we can tolerate the noise in stochastic gradients and attain an $O(1/T)$ convergence rate for optimizing smooth functions.

We now turn to describe the proposed mixed optimization algorithm and state its convergence rate. The detailed steps of MixedGrad algorithm are shown in Algorithm 10. It follows the epoch gradient descent algorithm proposed in [72] for stochastically minimizing strongly convex functions and divides the optimization process into m epochs, but involves novel ingredients so as to obtain an $O(1/T)$ convergence rate. The key idea is to introduce a ℓ_2 regularizer into the objective function to make it strongly convex, and gradually reduce the amount of regularization over the epochs. We also shrink the domain as the algorithm proceeds. We note that reducing the amount of regularization over time is closely-related to the classic proximal-point algorithms. Throughout the chapter, we will use the subscript for the index of each epoch, and the superscript for the index of iterations within each epoch. Below, we describe the key idea behind the MixedGrad algorithm.

Let $\bar{\mathbf{w}}_k$ be the solution obtained before the kth epoch, which is initialized to be $\mathbf{0}$ for the first epoch. Instead of searching for $\mathbf{w}_*$ at the kth epoch, our goal is to find $\mathbf{w}_* - \bar{\mathbf{w}}_k$,

Algorithm 10 MixedGrad Algorithm

```
Input:
  • step size $\eta_1$
  • domain size $\Delta_1$
  • the number of iterations $T_1$ for the first epoch
  • the number of epochs $m$
  • regularization parameter $\lambda_1$
  • shrinking parameter $\gamma > 1$
Initialize: $\bar{\mathbf{w}}_1 = \mathbf{0}$
for $k = 1, \ldots, m$ do
  Construct the domain $\mathcal{W}_k = \{\mathbf{w} : \mathbf{w} + \mathbf{w}_k \in \mathcal{W}, \|\mathbf{w}\| \leq \Delta_k\}$
  Call the full gradient oracle $\mathcal{O}_f$ for $\nabla \mathcal{F}(\bar{\mathbf{w}}_k)$
  Compute $\mathbf{g}_k = \lambda_k \bar{\mathbf{w}}_k + \nabla \mathcal{F}(\bar{\mathbf{w}}_k) = \lambda_k \bar{\mathbf{w}}_k + \frac{1}{n}\sum_{i=1}^n \nabla f_i(\bar{\mathbf{w}}_k)$
  Initialize $\mathbf{w}_k^1 = \mathbf{0}$
  for $t = 1, \ldots, T_k$ do
    Call stochastic oracle $\mathcal{O}_s$ to return a randomly selected loss function $f_{i_k^t}(\mathbf{w})$
    Compute the stochastic gradient as $\hat{\mathbf{g}}_k^t = \mathbf{g}_k + \nabla f_{i_k^t}(\mathbf{w}_k^t + \bar{\mathbf{w}}_k) - \nabla f_{i_k^t}(\bar{\mathbf{w}}_k)$
    Update the solution by
      $$\mathbf{w}_k^{t+1} = \arg\max_{\mathbf{w} \in \mathcal{W}_k} \eta_k \langle \mathbf{w} - \mathbf{w}_k^t, \hat{\mathbf{g}}_k^t + \lambda_k \mathbf{w}_k^t \rangle + \frac{1}{2}\|\mathbf{w} - \mathbf{w}_k^t\|^2$$
  end for
  Set $\widetilde{\mathbf{w}}_{k+1} = \frac{1}{T+1}\sum_{t=1}^{T+1} \mathbf{w}_k^t$ and $\bar{\mathbf{w}}_{k+1} = \bar{\mathbf{w}}_k + \widetilde{\mathbf{w}}_{k+1}$
  Set $\Delta_{k+1} = \Delta_k/\gamma$, $\lambda_{k+1} = \lambda_k/\gamma$, $\eta_{k+1} = \eta_k/\gamma$, and $T_{k+1} = \gamma^2 T_k$
end for
Return $\bar{\mathbf{w}}_{m+1}$
```

resulting in the following optimization problem for the kth epoch

$$\min_{\substack{\mathbf{w} + \mathbf{w}_k \in \mathcal{W} \\ \|\mathbf{w}\| \leq \Delta_k}} \quad \frac{\lambda_k}{2}\|\mathbf{w} + \bar{\mathbf{w}}_k\|^2 + \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w} + \bar{\mathbf{w}}_k), \tag{7.2}$$

where Δ_k specifies the domain size of $\mathbf{w}$ and λ_k is the regularization parameter introduced at the kth epoch. By introducing the ℓ_2 regularizer, the objective function in (7.2) becomes strongly convex, making it possible to exploit the technique for stochastic optimization of strongly convex function in order to improve the convergence rate. The domain size Δ_k and the regularization parameter λ_k are initialized to be $\Delta_1 > 0$ and $\lambda_1 > 0$, respectively, and are reduced by a constant factor $\gamma > 1$ every epoch, i.e., $\Delta_k = \Delta_1/\gamma^{k-1}$ and $\lambda_k = \lambda_1/\gamma^{k-1}$. By removing the constant term $\lambda_k\|\bar{\mathbf{w}}_k\|^2/2$ from the objective function in (7.2), we obtain the following optimization problem for the kth epoch

$$\min_{\mathbf{w}\in\mathcal{W}_k} \left[\mathcal{F}_k(\mathbf{w}) = \frac{\lambda_k}{2}\|\mathbf{w}\|^2 + \lambda_k\langle\mathbf{w}, \bar{\mathbf{w}}_k\rangle + \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w}+\bar{\mathbf{w}}_k)\right], \tag{7.3}$$

where $\mathcal{W}_k = \{\mathbf{w} : \mathbf{w}+\mathbf{w}_k \in \mathcal{W},\ \|\mathbf{w}\| \leq \Delta_k\}$. We rewrite the objective function $\mathcal{F}_k(\mathbf{w})$ as

$$\begin{aligned}
\mathcal{F}_k(\mathbf{w}) &= \frac{\lambda_k}{2}\|\mathbf{w}\|^2 + \lambda_k\langle\mathbf{w}, \bar{\mathbf{w}}_k\rangle + \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w}+\bar{\mathbf{w}}_k) \\
&= \frac{\lambda_k}{2}\|\mathbf{w}\|^2 + \left\langle \mathbf{w}, \lambda_k\bar{\mathbf{w}}_k + \frac{1}{n}\sum_{i=1}^n \nabla f_i(\bar{\mathbf{w}}_k)\right\rangle + \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w}+\bar{\mathbf{w}}_k) - \langle\mathbf{w}, \nabla f_i(\bar{\mathbf{w}}_k)\rangle \\
&= \frac{\lambda_k}{2}\|\mathbf{w}\|^2 + \langle\mathbf{w}, \mathbf{g}_k\rangle + \frac{1}{n}\sum_{i=1}^n \widehat{f}_i^k(\mathbf{w}) \qquad (7.4)
\end{aligned}$$

where

$$\mathbf{g}_k = \lambda_k\bar{\mathbf{w}}_k + \frac{1}{n}\sum_{i=1}^n \nabla f_i(\bar{\mathbf{w}}_k) \text{ and } \widehat{f}_i^k(\mathbf{w}) = f_i(\mathbf{w}+\bar{\mathbf{w}}_k) - \langle\mathbf{w}, \nabla f_i(\bar{\mathbf{w}}_k)\rangle.$$

The main reason for using $\widehat{f}_i^k(\mathbf{w})$ instead of $f_i(\mathbf{w})$ is to tolerate the variance in the stochastic gradients. To see this, from the smoothness assumption of $f_i(\mathbf{w})$ we obtain the following inequality for the norm of $\widehat{f}_i^k(\mathbf{w})$ as:

$$\left\|\nabla\widehat{f}_i^k(\mathbf{w})\right\| = \|\nabla f_i(\mathbf{w}+\bar{\mathbf{w}}_k) - \nabla f_i(\bar{\mathbf{w}}_k)\| \leq \beta\|\mathbf{w}\|.$$

As a result, since $\|\mathbf{w}\| \leq \Delta_k$ and Δ_k shrinks over epochs, then $\|\mathbf{w}\|$ will approach to zero over epochs and consequentially $\|\nabla \widehat{f}_i^k(\mathbf{w})\|$ approaches to zero, which allows us to effectively control the variance in stochastic gradients, a key to improving the convergence of stochastic optimization for smooth functions to $O(1/T)$.

Using $\mathcal{F}_k(\mathbf{w})$ in (7.4), at the tth iteration of the kth epoch, we call the stochastic oracle $\mathcal{O}_s$ to randomly select a loss function $f_{i_t^k}(\mathbf{w})$ and update the solution by following the standard paradigm of SGD by

$$\begin{aligned}
\mathbf{w}_k^{t+1} &= \Pi_{\mathbf{w}\in\mathcal{W}_k}\left(\mathbf{w}_k^t - \eta_k(\lambda_k \mathbf{w}_k^t + \mathbf{g}_k + \nabla \widehat{f}^k_{i_k^t}(\mathbf{w}_k^t))\right) \\
&= \Pi_{\mathbf{w}\in\mathcal{W}_k}\left(\mathbf{w}_k^t - \eta_k(\lambda_k \mathbf{w}_k^t + \mathbf{g}_k + \nabla f_{i_k^t}(\mathbf{w}_k^t + \bar{\mathbf{w}}_k) - \nabla f_{i_k^t}(\bar{\mathbf{w}}_k))\right), \qquad (7.5)
\end{aligned}$$

where $\Pi_{\mathbf{w}\in\mathcal{W}_k}(\cdot)$ projects the solution $\mathbf{w}$ into the domain $\mathcal{W}_k$ that shrinks over epochs.

At the end of each epoch, we compute the average solution $\widetilde{\mathbf{w}}_k$, and update the solution from $\bar{\mathbf{w}}_k$ to $\bar{\mathbf{w}}_{k+1} = \bar{\mathbf{w}}_k + \widetilde{\mathbf{w}}_k$. Similar to the epoch gradient descent algorithm [72], we increase the number of iterations by a constant γ^2 for every epoch, i.e. $T_k = T_1\gamma^{2(k-1)}$.

In order to perform stochastic gradient updating given in (7.5), we need to compute vector $\mathbf{g}_k$ at the beginning of the kth epoch, which requires an access to the full gradient oracle $\mathcal{O}_f$. It is easy to count that the number of accesses to the full gradient oracle $\mathcal{O}_f$ is m, and the number of accesses to the stochastic oracle $\mathcal{O}_s$ is

$$T = T_1 \sum_{i=1}^{m} \gamma^{2(i-1)} = \frac{\gamma^{2m} - 1}{\gamma^2 - 1} T_1.$$

Thus, if the total number of accesses to the stochastic gradient oracle is T, the number of access to the full gradient oracle required by MixedGrad algorithm is $O(\ln T)$, consistent with our goal of making a small number of calls to the full gradient oracle.

The theorem below shows that for smooth objective functions, by having $O(\ln T)$ access to the full gradient oracle $\mathcal{O}_f$ and $O(T)$ access to the stochastic oracle $\mathcal{O}_s$, by running MixedGrad algorithm, we achieve an optimization error of $O(1/T)$.

Theorem 7.1. *Let $\delta \leq e^{-9/2}$ be the failure probability. Set $\gamma = 2$, $\lambda_1 = 16\beta$ and*

$$T_1 = 300 \ln \frac{m}{\delta}, \quad \eta_1 = \frac{1}{2\beta\sqrt{3T_1}}, \quad \textit{and} \;\; \Delta_1 = R.$$

Define $T = T_1 \left(2^{2m} - 1\right)/3$. Let $\bar{\mathbf{w}}_{m+1}$ be the solution returned by MixedGrad method in Algorithm 10 after m epochs with $m = O(\ln T)$ calls to the full gradient oracle $\mathcal{O}_f$ and T calls to the stochastic oracle $\mathcal{O}_s$. Then, with a probability $1 - 2\delta$, we have

$$\mathcal{F}(\bar{\mathbf{w}}_{m+1}) - \min_{\mathbf{w} \in \mathcal{W}} \mathcal{F}(\mathbf{w}) \leq \frac{80\beta R^2}{2^{2m-2}} = O\left(\frac{\beta}{T}\right).$$

7.3 Analysis of Convergence Rate

Now we turn to proving the main theorem. The proof will be given in a series of lemmas and theorems where the proof of few are given in Section 7.4. The proof of main theorem is based on induction. To this end, let $\widehat{\mathbf{w}}_*^k$ be the optimal solution that minimizes $\mathcal{F}_k(\mathbf{w})$ defined in (7.3). The key to our analysis is show that when $\|\widehat{\mathbf{w}}_*^k\| \leq \Delta_k$, with a high probability, it holds that $\|\widehat{\mathbf{w}}_*^{k+1}\| \leq \Delta_k/\gamma$, where $\widehat{\mathbf{w}}_*^{k+1}$ is the optimal solution that minimizes $\mathcal{F}_{k+1}(\mathbf{w})$, as revealed by the following theorem.

Theorem 7.2. *Let $\widehat{\mathbf{w}}_*^k$ and $\widehat{\mathbf{w}}_*^{k+1}$ be the optimal solutions that minimize $\mathcal{F}_k(\mathbf{w})$ and $\mathcal{F}_{k+1}(\mathbf{w})$,* *respectively, and $\widetilde{\mathbf{w}}_{k+1}$ be the average solution obtained at the end of kth epoch of MixedGrad algorithm. Suppose $\|\widehat{\mathbf{w}}_*^k\| \leq \Delta_k$. By setting the step size $\eta_k = 1/\left(2\beta\sqrt{3T_k}\right)$, we have, with a probability $1-2\delta$,*

$$\|\widehat{\mathbf{w}}_*^{k+1}\| \leq \frac{\Delta_k}{\gamma} \ \ \textit{and} \ \ \mathcal{F}_k(\widetilde{\mathbf{w}}_{k+1}) - \min_{\mathbf{w}} \mathcal{F}_k(\mathbf{w}) \leq \frac{\lambda_k \Delta_k^2}{2\gamma^4}$$

provided that $\delta \leq e^{-9/2}$ and

$$T_k \geq \frac{300\gamma^8\beta^2}{\lambda_k^2} \ln\frac{1}{\delta}.$$

Taking this statement as given for the moment, we proceed with the proof of Theorem 7.1, returning later to establish the claim stated in Theorem 7.2.

Proof of Theorem 7.1. It is easy to check that for the first epoch, using the fact $\mathcal{W} \in \mathbb{B}_R$, we have

$$\|\mathbf{w}_*^1\| = \|\mathbf{w}_*\| \leq R := \Delta_1.$$

Let $\mathbf{w}_*^m$ be the optimal solution that minimizes $\mathcal{F}_m(\mathbf{w})$ and let $\widehat{\mathbf{w}}_*^{m+1}$ be the optimal solution obtained in the last epoch. Using Theorem 7.1, with a probability $1-2m\delta$, we have

$$\|\widehat{\mathbf{w}}_*^m\| \leq \frac{\Delta_1}{\gamma^{m-1}}, \quad \mathcal{F}_m(\widetilde{\mathbf{w}}_{m+1}) - \mathcal{F}_m(\widehat{\mathbf{w}}_*^m) \leq \frac{\lambda_m \Delta_m^2}{2\gamma^4} = \frac{\lambda_1\Delta_1^2}{2\gamma^{3m+1}}$$

Hence,

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^n f_i(\bar{\mathbf{w}}_{m+1}) &\leq \mathcal{F}_m(\widehat{\mathbf{w}}_*^m) + \frac{\lambda_1\Delta_1^2}{2\gamma^{3m+1}} - \frac{\lambda_1}{\gamma^{m-1}}\langle\widetilde{\mathbf{w}}_{m+1}, \bar{\mathbf{w}}_m\rangle \\
&\leq \mathcal{F}_m(\widehat{\mathbf{w}}_*^m) + \frac{\lambda_1\Delta_1^2}{2\gamma^{3m+1}} + \frac{\lambda_1\|\bar{\mathbf{w}}_m\|\Delta_1}{\gamma^{2m-2}}
\end{aligned}$$

where the last step uses the fact $\|\widehat{\mathbf{w}}_*^{m+1}\| \leq \Delta_m = \Delta_1 \gamma^{1-m}$. Since

$$\|\bar{\mathbf{w}}_m\| \leq \sum_{i=1}^m |\widetilde{\mathbf{w}}_i| \leq \sum_{i=1}^m \Delta_i \leq \frac{\gamma \Delta_1}{\gamma - 1} \leq 2\Delta_1$$

where in the last step holds under the condition $\gamma \geq 2$. By combining above inequalities, we obtain

$$\frac{1}{n}\sum_{i=1}^n f_i(\bar{\mathbf{w}}_{m+1}) \leq \mathcal{F}_m(\widehat{\mathbf{w}}_*^m) + \frac{\lambda_1 \Delta_1^2}{2\gamma^{3m+1}} + \frac{2\lambda_1\Delta_1^2}{\gamma^{2m-2}}.$$

Our final goal is to relate $\mathcal{F}_m(\mathbf{w})$ to $\min_{\mathbf{w}} \mathcal{L}(\mathbf{w})$. Since $\widehat{\mathbf{w}}_*^m$ minimizes $\mathcal{F}_m(\mathbf{w})$, for any $\mathbf{w}_* \in \arg\min \mathcal{L}(\mathbf{w})$, we have

$$\mathcal{F}_m(\mathbf{w}_*^m) \leq \mathcal{F}_m(\mathbf{w}_*) = \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w}_*) + \frac{\lambda_1}{2\gamma^{m-1}}\left(\|\mathbf{w}_* - \bar{\mathbf{w}}_m\|^2 + 2\langle \mathbf{w}_* - \bar{\mathbf{w}}_m, \bar{\mathbf{w}}_m\rangle\right). \quad (7.6)$$

Thus, the key to bound $|\mathcal{F}(\mathbf{w}_*^m) - \mathcal{F}(\mathbf{w}_*)|$ is to bound $\|\mathbf{w}_* - \bar{\mathbf{w}}_m\|$. To this end, after the first m epoches, we run Algorithm 10 with $f$ull gradients. Let $\bar{\mathbf{w}}_{m+1}, \bar{\mathbf{w}}_{m+2}, \ldots$ be the sequence of solutions generated by Algorithm 10 after the first m epochs. For this sequence of solutions, Theorem 7.2 will hold deterministically as we deploy the full gradient for updating, i.e., $\|\widetilde{\mathbf{w}}_k\| \leq \Delta_k$ for any $k \geq m+1$. Since we reduce λ_k exponentially, λ_k will approach to zero and therefore the sequence $\{\bar{\mathbf{w}}_k\}_{k=m+1}^\infty$ will converge to $\mathbf{w}_*$, one of the optimal solutions that minimize $\mathcal{L}(\mathbf{w})$. Since $\mathbf{w}_*$ is the limit of sequence $\{\bar{\mathbf{w}}_k\}_{k=m+1}^\infty$ and $\|\bar{\mathbf{w}}_k\| \leq \Delta_k$ for any $k \geq m+1$, we have

$$\|\mathbf{w}_* - \bar{\mathbf{w}}_m\| \leq \sum_{i=m+1}^\infty |\widetilde{\mathbf{w}}_i| \leq \sum_{k=m+1}^\infty \Delta_k \leq \frac{\Delta_1}{\gamma^m(1-\gamma^{-1})} \leq \frac{2\Delta_1}{\gamma^m}$$

where the last step follows from the condition $\gamma \geq 2$. Thus,

$$\begin{aligned}
\mathcal{F}_m(\mathbf{w}_*^m) &\leq \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w}_*) + \frac{\lambda_1}{2\gamma^{m-1}}\left(\frac{4\Delta_1^2}{\gamma^{2m}} + \frac{8\Delta_1^2}{\gamma^m}\right) \\
&= \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w}_*) + \frac{2\lambda_1\Delta_1^2}{\gamma^{2m-1}}\left(2 + \gamma^{-m}\right) \leq \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w}_*) + \frac{5\lambda_1\Delta_1^2}{\gamma^{2m-1}} \qquad (7.7)
\end{aligned}$$

By combining the bounds in (7.6) and (7.7), we have, with a probability $1 - 2m\delta$,

$$\frac{1}{n}\sum_{i=1}^n f_i(\bar{\mathbf{w}}_{m+1}) - \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w}_*) \leq \frac{5\lambda_1\Delta_1^2}{\gamma^{2m-2}} = O(1/T)$$

where

$$T = T_1\sum_{k=0}^{m-1}\gamma^{2k} = \frac{T_1\left(\gamma^{2m} - 1\right)}{\gamma^2 - 1} \leq \frac{T_1}{3}\gamma^{2m}.$$

We complete the proof by plugging in the stated values for γ, λ_1 and Δ_1. □

We turn now to proving the Theorem 7.2. For the convenience of discussion, we drop the subscript k for epoch just to simplify our notation. Let $\lambda = \lambda_k$, $T = T_k$, $\Delta = \Delta_k$, $\mathbf{g} = \mathbf{g}_k$. Let $\bar{\mathbf{w}} = \bar{\mathbf{w}}_k$ be the solution obtained before the start of the epoch k, and let $\bar{\mathbf{w}}' = \bar{\mathbf{w}}_{k+1}$ be the solution obtained after running through the kth epoch. We denote by $\mathcal{F}(\mathbf{w})$ and $\mathcal{F}'(\mathbf{w})$ the objective functions $\mathcal{F}_k(\mathbf{w})$ and $\mathcal{F}_{k+1}(\mathbf{w})$. They are given by

$$\begin{aligned}
\mathcal{F}(\mathbf{w}) &= \frac{\lambda}{2}\|\mathbf{w}\|^2 + \lambda\langle\mathbf{w}, \bar{\mathbf{w}}\rangle + \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w} + \bar{\mathbf{w}}) \qquad (7.8) \\
\mathcal{F}'(\mathbf{w}) &= \frac{\lambda}{2\gamma}\|\mathbf{w}\|^2 + \frac{\lambda}{\gamma}\langle\mathbf{w}, \bar{\mathbf{w}}'\rangle + \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{w} + \bar{\mathbf{w}}') \qquad (7.9)
\end{aligned}$$

Let $\widehat{\mathbf{w}}_* = \widehat{\mathbf{w}}_*^k$ and $\widehat{\mathbf{w}}_*' = \widehat{\mathbf{w}}_*^{k+1}$ be the optimal solutions that minimize $\mathcal{F}(\mathbf{w})$ and $\mathcal{F}'(\mathbf{w})$ over the domain $\mathcal{W}_k$ and $\mathcal{W}_{k+1}$, respectively. Under the assumption that $\|\widehat{\mathbf{w}}_*\| \leq \Delta$, our goal is to show

$$\|\widehat{\mathbf{w}}'_*\| \leq \frac{\Delta}{\gamma}, \quad \mathcal{F}(\bar{\mathbf{w}}') - \mathcal{F}(\widehat{\mathbf{w}}_*) \leq \frac{\lambda\Delta^2}{2\gamma^4}$$

The following lemma bounds $\mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\widehat{\mathbf{w}}_*)$ where the proof is deferred to Section 7.4.

Lemma 7.3.

$$\begin{aligned}\mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\widehat{\mathbf{w}}_*) \leq & \frac{\|\mathbf{w}_t - \widehat{\mathbf{w}}_*\|^2}{2\eta} - \frac{\|\mathbf{w}_{t+1} - \widehat{\mathbf{w}}_*\|^2}{2\eta} \\ & + \frac{\eta}{2}\left\|\nabla\widehat{f}_{i_t}(\mathbf{w}_t) + \lambda\mathbf{w}_t\right\|^2 + \langle \mathbf{g}, \mathbf{w}_t - \mathbf{w}_{t+1}\rangle \\ & + \left\langle \nabla\widehat{\mathcal{F}}(\widehat{\mathbf{w}}_*) - \nabla\widehat{f}_{i_t}(\widehat{\mathbf{w}}_*), \mathbf{w}_t - \widehat{\mathbf{w}}_*\right\rangle \\ & + \left\langle -\nabla\widehat{f}_{i_t}(\mathbf{w}_t) + \nabla\widehat{f}_{i_t}(\widehat{\mathbf{w}}_*) - \nabla\widehat{\mathcal{F}}(\widehat{\mathbf{w}}_*) + \nabla\widehat{\mathcal{F}}(\mathbf{w}_t), \mathbf{w}_t - \widehat{\mathbf{w}}_*\right\rangle\end{aligned}$$

By adding the inequality in Lemma 7.7 over all iterations, using the fact $\bar{\mathbf{w}}_1 = \mathbf{0}$, we have

$$\begin{aligned}\sum_{t=1}^{T}\mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\widehat{\mathbf{w}}_*) \leq & \frac{\|\widehat{\mathbf{w}}_*\|^2}{2\eta} - \frac{\|\mathbf{w}_{T+1} - \widehat{\mathbf{w}}_*\|^2}{2\eta} - \langle \mathbf{g}, \mathbf{w}_{T+1}\rangle \\ & + \frac{\eta}{2}\underbrace{\sum_{t=1}^{T}\|\nabla\widehat{f}_{i_t}(\mathbf{w}_t) + \lambda\mathbf{w}_t\|^2}_{\triangleq A_T} + \underbrace{\sum_{t=1}^{T}\langle\nabla\widehat{\mathcal{F}}(\widehat{\mathbf{w}}_*) - \nabla\widehat{f}_{i_t}(\widehat{\mathbf{w}}_*), \mathbf{w}_t - \widehat{\mathbf{w}}_*\rangle}_{\triangleq B_T} \\ & + \underbrace{\sum_{t=1}^{T}\left\langle -\nabla\widehat{f}_{i_t}(\mathbf{w}_t) + \nabla\widehat{f}_{i_t}(\widehat{\mathbf{w}}_*) - \nabla\widehat{\mathcal{F}}(\widehat{\mathbf{w}}_*) + \nabla\widehat{\mathcal{F}}(\mathbf{w}_t), \mathbf{w}_t - \widehat{\mathbf{w}}_*\right\rangle}_{\triangleq C_T}.\end{aligned}$$

Since $\mathbf{g} = \nabla\mathcal{F}(\mathbf{0})$ and

$$\mathcal{F}(\mathbf{w}_{T+1}) - \mathcal{F}(\mathbf{0}) \leq \langle\nabla\mathcal{F}(\mathbf{0}), \mathbf{w}_{T+1}\rangle + \frac{\beta}{2}\|\mathbf{w}_{T+1}\|^2 = \langle\mathbf{g}, \mathbf{w}_{T+1}\rangle + \frac{\beta}{2}\|\mathbf{w}_{T+1}\|^2$$

using the fact $\mathcal{F}(\mathbf{0}) \leq \mathcal{F}(\mathbf{w}_*) + \frac{\beta}{2}\|\mathbf{w}_*\|^2$ and $\max(\|\mathbf{w}_*\|, \|\mathbf{w}_{T+1}\|) \leq \Delta$, we have

$$-\langle \mathbf{g}, \mathbf{w}_{T+1} \rangle \leq \mathcal{F}(\mathbf{0}) - \mathcal{F}(\mathbf{w}_{T+1}) + \frac{\beta}{2}\Delta^2 \leq \beta\Delta^2 - (\mathcal{F}(\mathbf{w}_{T+1}) - \mathcal{F}(\widehat{\mathbf{w}}_*))$$

and therefore

$$\sum_{t=1}^{T+1} \mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\widehat{\mathbf{w}}_*) \leq \Delta^2 \left(\frac{1}{2\eta} + \beta \right) + \frac{\eta}{2} A_T + B_T + C_T. \tag{7.10}$$

The following lemmas bound A_T, B_T and C_T.

Lemma 7.4. *For A_T defined above we have $A_T \leq 6\beta^2\Delta^2 T$.*

The following lemma upper bounds B_T and C_T. The proof is based on the Bernstein's inequality for martingales and is given later .

Lemma 7.5. *With a probability $1 - 2\delta$, we have*

$$B_T \leq \beta\Delta^2 \left(\ln\frac{1}{\delta} + \sqrt{2T \ln\frac{1}{\delta}} \right) \quad \textit{and} \quad C_T \leq 2\beta\Delta^2 \left(\ln\frac{1}{\delta} + \sqrt{2T \ln\frac{1}{\delta}} \right).$$

Using Lemmas 7.8 and 7.9, by substituting the uppers bounds for A_T, B_T, and C_T in (7.10), with a probability $1 - 2\delta$, we obtain

$$\sum_{t=1}^{T+1} \mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\widehat{\mathbf{w}}_*) \leq \Delta^2 \left(\frac{1}{2\eta} + \beta + 6\beta^2\eta T + 3\beta \ln\frac{1}{\delta} + 3\beta\sqrt{2T \ln\frac{1}{\delta}} \right)$$

By choosing $\eta = 1/[2\beta\sqrt{3T}]$, we have

$$\sum_{t=1}^{T+1} \mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\widehat{\mathbf{w}}_*) \leq \Delta^2 \left(2\beta\sqrt{3T} + \beta + 3\beta \ln\frac{1}{\delta} + 3\beta\sqrt{2T \ln\frac{1}{\delta}} \right)$$

and using the fact $\widetilde{\mathbf{w}} = \sum_{i=1}^{T+1} \mathbf{w}_t/(T+1)$, we have

$$\mathcal{F}(\widetilde{\mathbf{w}}) - \mathcal{F}(\widehat{\mathbf{w}}_*) \leq \Delta^2 \frac{5\beta\sqrt{3\ln[1/\delta]}}{\sqrt{T+1}}, \ \text{ and } \ \widehat{\Delta}^2 = \|\widetilde{\mathbf{w}} - \widehat{\mathbf{w}}_*\|^2 \leq \Delta^2 \frac{5\beta\sqrt{3\ln[1/\delta]}}{\lambda\sqrt{T+1}}.$$

Thus, when $T \geq [300\gamma^8\beta^2 \ln \frac{1}{\delta}]/\lambda^2$, we have, with a probability $1 - 2\delta$,

$$\widehat{\Delta}^2 \leq \frac{\Delta^2}{\gamma^4}, \quad \text{and} \quad |\mathcal{F}(\widetilde{\mathbf{w}}) - \mathcal{F}(\widehat{\mathbf{w}}_*)| \leq \frac{\lambda}{2\gamma^4}\Delta^2. \tag{7.11}$$

The next lemma relates $\|\widehat{\mathbf{w}}'_*\|$ to $\|\widetilde{\mathbf{w}} - \widehat{\mathbf{w}}_*\|$.

Lemma 7.6. *We have* $\|\widehat{\mathbf{w}}'_*\| \leq \gamma\|\widetilde{\mathbf{w}} - \widehat{\mathbf{w}}_*\|$.

Combining the bound in (7.11) with Lemma 7.10, we have $\|\widehat{\mathbf{w}}'_*\| \leq \Delta/\gamma$.

7.4 Proofs of Convergence Rate

7.4.1 Proof of Lemma 7.7

Before proving the lemmas we recall the definition of $\mathcal{F}(\mathbf{w})$, $\mathcal{F}'(\mathbf{w})$, $\mathbf{g}$, and $\widehat{f}_i(\mathbf{w})$ as:

$$\begin{aligned}
\mathcal{F}(\mathbf{w}) &= \frac{\lambda}{2}\|\mathbf{w}\|^2 + \lambda\langle \mathbf{w}, \bar{\mathbf{w}}\rangle + \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w} + \bar{\mathbf{w}}), \\
\mathcal{F}'(\mathbf{w}) &= \frac{\lambda}{2\gamma}\|\mathbf{w}\|^2 + \frac{\lambda}{\gamma}\langle \mathbf{w}, \bar{\mathbf{w}}'\rangle + \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w} + \bar{\mathbf{w}}'), \\
\mathbf{g} &= \lambda\bar{\mathbf{w}} + \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(\bar{\mathbf{w}}), \\
\widehat{f}_i(\mathbf{w}) &= f_i(\mathbf{w} + \bar{\mathbf{w}}) - \langle \mathbf{w}, \nabla f_i(\bar{\mathbf{w}})\rangle.
\end{aligned}$$

We also recall that $\widehat{\mathbf{w}}_*$ and $\widehat{\mathbf{w}}'_*$ are the optimal solutions that minimize $\mathcal{F}(\mathbf{w})$ and $\mathcal{F}'(\mathbf{w})$ over the domain $\mathcal{W}_k$ and $\mathcal{W}_{k+1}$, respectively.

Lemma 7.7.

$$\begin{aligned}
\mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\widehat{\mathbf{w}}_*) \leq & \quad \tfrac{\|\mathbf{w}_t - \widehat{\mathbf{w}}_*\|^2}{2\eta} - \tfrac{\|\mathbf{w}_{t+1} - \widehat{\mathbf{w}}_*\|^2}{2\eta} + \tfrac{\eta}{2}\left\|\nabla \widehat{f}_{i_t}(\mathbf{w}_t) + \lambda \mathbf{w}_t\right\|^2 + \langle \mathbf{g}, \mathbf{w}_t - \mathbf{w}_{t+1}\rangle \\
+ & \quad \left\langle \nabla \widehat{f}_{i_t}(\widehat{\mathbf{w}}_*) - \nabla \widehat{\mathcal{F}}(\widehat{\mathbf{w}}_*), \mathbf{w}_t - \widehat{\mathbf{w}}_* \right\rangle \\
+ & \quad \left\langle -\nabla \widehat{f}_{i_t}(\mathbf{w}_t) + \nabla \widehat{f}_{i_t}(\widehat{\mathbf{w}}_*) - \nabla \widehat{\mathcal{F}}(\widehat{\mathbf{w}}_*) + \nabla \widehat{\mathcal{F}}(\mathbf{w}_t), \mathbf{w}_t - \widehat{\mathbf{w}}_* \right\rangle
\end{aligned}$$

Proof. For each iteration t in the kth epoch, from the strong convexity of $\mathcal{F}(\mathbf{w})$ we have

$$\begin{aligned}
& \mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\widehat{\mathbf{w}}_*) \leq \langle \nabla \mathcal{F}(\mathbf{w}_t), \mathbf{w}_t - \widehat{\mathbf{w}}_* \rangle - \frac{\lambda}{2}\|\mathbf{w}_t - \widehat{\mathbf{w}}_*\|^2 \\
& \quad = \ \langle \mathbf{g} + \nabla \widehat{f}_{i_t}(\mathbf{w}_t) + \lambda \mathbf{w}_t, \mathbf{w}_t - \widehat{\mathbf{w}}_* \rangle + \left\langle -\nabla \widehat{f}_{i_t}(\mathbf{w}_t) + \nabla \widehat{\mathcal{F}}(\mathbf{w}_t), \mathbf{w}_t - \widehat{\mathbf{w}}_* \right\rangle - \frac{\lambda}{2}\|\mathbf{w}_t - \widehat{\mathbf{w}}_*\|^2,
\end{aligned}$$

where $\widehat{\mathcal{F}}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \widehat{f}_i(\mathbf{w})$. We now try to upper bound the first term in the right hand side. Since

$$\begin{aligned}
& \langle \mathbf{g} + \nabla \widehat{f}_{i_t}(\mathbf{w}_t) + \lambda \mathbf{w}_t, \mathbf{w}_t - \widehat{\mathbf{w}}_* \rangle \\
= \ & \langle \mathbf{g} + \nabla \widehat{f}_{i_t}(\mathbf{w}_t) + \lambda \mathbf{w}_t, \mathbf{w}_t - \widehat{\mathbf{w}}_* \rangle - \frac{\|\mathbf{w}_t - \widehat{\mathbf{w}}_*\|^2}{2\eta} + \frac{\|\mathbf{w}_t - \widehat{\mathbf{w}}_*\|^2}{2\eta} \\
\leq \ & \langle \mathbf{g} + \nabla \widehat{f}_{i_t}(\mathbf{w}_t) + \lambda \mathbf{w}_t, \mathbf{w}_t - \mathbf{w}_{t+1} \rangle - \frac{\|\mathbf{w}_t - \mathbf{w}_{t+1}\|^2}{2\eta} - \frac{\|\mathbf{w}_{t+1} - \widehat{\mathbf{w}}_*\|^2}{2\eta} + \frac{\|\mathbf{w}_t - \widehat{\mathbf{w}}_*\|^2}{2\eta} \\
\leq \ & \langle \mathbf{g}, \mathbf{w}_t - \mathbf{w}_{t+1} \rangle - \frac{\|\mathbf{w}_{t+1} - \widehat{\mathbf{w}}_*\|^2}{2\eta} + \frac{\|\mathbf{w}_t - \widehat{\mathbf{w}}_*\|^2}{2\eta} \\
+ \ & \max_{\mathbf{w}} \left[\langle \nabla \widehat{f}_{i_t}(\mathbf{w}_t) + \lambda \mathbf{w}_t, \mathbf{w}_t - \mathbf{w} \rangle - \frac{\|\mathbf{w}_t - \mathbf{w}\|^2}{2\eta} \right] \\
= \ & \langle \mathbf{g}, \mathbf{w}_t - \mathbf{w}_{t+1} \rangle - \frac{\|\mathbf{w}_{t+1} - \widehat{\mathbf{w}}_*\|^2}{2\eta} + \frac{\|\mathbf{w}_t - \widehat{\mathbf{w}}_*\|^2}{2\eta} + \frac{\eta}{2}\|\nabla \widehat{f}_{i_t}(\mathbf{w}_t) + \lambda \mathbf{w}_t\|^2
\end{aligned}$$

where the first inequality follows from the fact that $\mathbf{w}_{t+1}$ in the minimizer of the following optimization problem:

$$\mathbf{w}_{t+1} = \mathop{\arg\min}_{\mathbf{w}\in\mathcal{W}\cap\|\mathbf{w}-\bar{\mathbf{w}}\|\leq\Delta} \langle \mathbf{g} + \nabla\widehat{f}_{i_t}(\mathbf{w}_t) + \lambda\mathbf{w}_t, \mathbf{w}-\mathbf{w}_t\rangle + \frac{\|\mathbf{w}-\mathbf{w}_t\|^2}{2\eta}.$$

Therefore, we obtain

$$\begin{aligned}
&\mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\widehat{\mathbf{w}}_*) \\
&\leq \frac{\|\mathbf{w}_t - \widehat{\mathbf{w}}_*\|^2}{2\eta} - \frac{\|\mathbf{w}_{t+1} - \widehat{\mathbf{w}}_*\|^2}{2\eta} - \frac{\lambda}{2}\|\mathbf{w}_t - \widehat{\mathbf{w}}_*\|^2 \\
&\quad + \langle \mathbf{g}, \mathbf{w}_t - \mathbf{w}_{t+1}\rangle + \frac{\eta}{2}\left\|\nabla\widehat{f}_{i_t}(\mathbf{w}_t) + \lambda\mathbf{w}_t\right\|^2 + \left\langle \nabla\widehat{\mathcal{F}}(\widehat{\mathbf{w}}_*) - \nabla\widehat{f}_{i_t}(\widehat{\mathbf{w}}_*), \mathbf{w}_t - \widehat{\mathbf{w}}_*\right\rangle \\
&\quad + \left\langle -\nabla\widehat{f}_{i_t}(\mathbf{w}_t) + \nabla\widehat{f}_{i_t}(\widehat{\mathbf{w}}_*) - \nabla\widehat{\mathcal{F}}(\widehat{\mathbf{w}}_*) + \nabla\widehat{\mathcal{F}}(\mathbf{w}_t), \mathbf{w}_t - \widehat{\mathbf{w}}_*\right\rangle,
\end{aligned}$$

as desired. □

We now turn to prove the upper bound on A_T.

Lemma 7.8.

$$A_T \leq 6\beta^2\Delta^2 T$$

Proof. We bound A_T as

$$\begin{aligned}
A_T &= \sum_{t=1}^{T} \|\nabla\widehat{f}_{i_t}(\mathbf{w}_t) + \lambda\mathbf{w}_t\|^2 \\
&\leq \sum_{t=1}^{T} 2\|\nabla\widehat{f}_{i_t}(\mathbf{w}_t)\|^2 + 2\lambda^2\|\mathbf{w}_t\|^2 \\
&\leq \sum_{t=1}^{T} 2\lambda^2\Delta^2 + 2\|\nabla\widehat{f}_{i_t}(\mathbf{w}_t) - \nabla\widehat{f}_{i_t}(\widehat{\mathbf{w}}_*) + \nabla\widehat{f}_{i_t}(\widehat{\mathbf{w}}_*)\|^2 \\
&\leq 6\beta^2\Delta^2 T
\end{aligned}$$

where the second inequality follows $(a+b)^2 \leq 2(a^2+b^2)$ and the last inequality follows from the smoothness assumption. □

Lemma 7.9. *With a probability* $1-2\delta$*, we have*

$$B_T \leq \beta\Delta^2\left(\ln\frac{1}{\delta}+\sqrt{2T\ln\frac{1}{\delta}}\right) \quad \textit{and} \;\; C_T \leq 2\beta\Delta^2\left(\ln\frac{1}{\delta}+\sqrt{2T\ln\frac{1}{\delta}}\right)$$

The proof is based on the Berstein inequality for martingales stated in Theorem A.25. Equipped with this concentration inequality, we are now in a position to upper bound B_T and C_T as follows.

Proof of Lemma 7.9. Denote $X_t = \langle\nabla\widehat{f}_{i_t}(\widehat{\mathbf{w}}_*) - \nabla\widehat{\mathcal{F}}(\widehat{\mathbf{w}}_*), \mathbf{w}_t - \widehat{\mathbf{w}}_*\rangle$. We have that the conditional expectation of X_t, given randomness in previous rounds, is $\mathbb{E}_{t-1}[X_t] = 0$. We now apply Theorem A.25 to the sum of martingale differences. In particular, we have, with a probability $1-e^{-t}$,

$$B_T \leq \frac{\sqrt{2}}{3}Kt + \sqrt{2\Sigma t}$$

where

$$\begin{aligned} K &= \max_{1\leq t\leq T}\langle\nabla\widehat{f}_{i_t}(\widehat{\mathbf{w}}_*) - \nabla\widehat{\mathcal{F}}(\widehat{\mathbf{w}}_*), \mathbf{w}_t - \widehat{\mathbf{w}}_*\rangle \leq 2\beta\Delta^2 \\ \Sigma &= \sum_{t=1}^{T}\mathbb{E}_t\left[|\langle\nabla\widehat{f}_{i_t}(\widehat{\mathbf{w}}_*) - \nabla\widehat{\mathcal{F}}(\widehat{\mathbf{w}}_*), \mathbf{w}_t - \widehat{\mathbf{w}}_*\rangle|^2\right] \leq \beta^2\Delta^4 T \end{aligned}$$

Hence, with a probability $1-\delta$, we have

$$B_T \leq \beta\Delta^2\left(\ln\frac{1}{\delta}+\sqrt{2T\ln\frac{1}{\delta}}\right)$$

Similar, for C_T, we have, with a probability $1-\delta$,

$$C_T \leq 2\beta\Delta^2\left(\ln\frac{1}{\delta} + \sqrt{2T\ln\frac{1}{\delta}}\right)$$

□

Lemma 7.10. $\|\widehat{\mathbf{w}}'_*\| \leq \gamma\|\widetilde{\mathbf{w}} - \widehat{\mathbf{w}}_*\|$.

Proof. We rewrite $\mathcal{F}(\mathbf{w})$ as

$$\begin{aligned}
\mathcal{F}(\mathbf{w}) &= \frac{\lambda}{2}\|\mathbf{w}\|^2 + \lambda\langle\mathbf{w},\bar{\mathbf{w}}\rangle + \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w}+\bar{\mathbf{w}}) \\
&= \frac{\lambda}{2}\|\mathbf{w}-\widetilde{\mathbf{w}}+\widetilde{\mathbf{w}}\|^2 + \lambda\langle\mathbf{w}-\widetilde{\mathbf{w}}+\widetilde{\mathbf{w}},\bar{\mathbf{w}}\rangle + \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w}-\widetilde{\mathbf{w}}+\bar{\mathbf{w}}')
\end{aligned}$$

Define $\mathbf{z} = \mathbf{w} - \widetilde{\mathbf{w}}$. We have

$$\begin{aligned}
\mathcal{F}(\mathbf{w}) &= \frac{\lambda}{2}\|\mathbf{z}+\widetilde{\mathbf{w}}\|^2 + \lambda\langle\mathbf{z},\bar{\mathbf{w}}\rangle + \lambda\langle\widetilde{\mathbf{w}},\bar{\mathbf{w}}\rangle + \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{z}+\bar{\mathbf{w}}') \\
&= \frac{\lambda}{2}\|\mathbf{z}\|^2 + \lambda\langle\mathbf{z},\bar{\mathbf{w}}'\rangle + \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{z}+\bar{\mathbf{w}}') + \frac{\lambda}{2}\|\widetilde{\mathbf{w}}\|^2 + \lambda\langle\widetilde{\mathbf{w}},\bar{\mathbf{w}}\rangle \\
&= \widetilde{\mathcal{F}}(\mathbf{z}) + \frac{\lambda}{2}\|\widetilde{\mathbf{w}}\|^2 + \lambda\langle\widetilde{\mathbf{w}},\bar{\mathbf{w}}\rangle
\end{aligned}$$

where

$$\widetilde{\mathcal{F}}(\mathbf{z}) = \frac{\lambda}{2}\|\mathbf{z}\|^2 + \lambda\langle\mathbf{z},\bar{\mathbf{w}}'\rangle + \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{z}+\bar{\mathbf{w}}')$$

Define $\widetilde{\mathbf{w}}_* = \widehat{\mathbf{w}}_* - \widetilde{\mathbf{w}}$. Evidently, $\widetilde{\mathbf{w}}_*$ minimizes $\widetilde{\mathcal{F}}(\mathbf{w})$. The only difference between $\widetilde{\mathcal{F}}(\mathbf{w})$ and $F'(\mathbf{w})$ is that they use different modulus of strong convexity λ. Thus, following [153], we have

$$\|\widetilde{\mathbf{w}}_* - \widehat{\mathbf{w}}'_*\| \leq \frac{1-\gamma^{-1}}{\gamma^{-1}}\|\widetilde{\mathbf{w}}_*\| \leq (\gamma - 1)\|\widetilde{\mathbf{w}}_*\|$$

Hence,

$$\|\widehat{\mathbf{w}}'_*\| \leq \gamma\|\widetilde{\mathbf{w}}_*\| = \gamma\|\widehat{\mathbf{w}}_* - \widetilde{\mathbf{w}}\|$$

which completes the proofs. □

7.5 Summary

We presented a new paradigm for optimization, termed as mixed optimization, that aims to improve the convergence rate of stochastic optimization by making a small number of calls to the full gradient oracle. We proposed the MixedGrad algorithm and showed that it is able to achieve an $O(1/T)$ convergence rate by accessing stochastic and full gradient oracles for $\mathcal{O}(T)$ and $\mathcal{O}(\log T)$ times, respectively. We showed that the MixedGrad algorithm is able to exploit the $s$moothness of the function, which is believed to be not very useful in stochastic optimization. The key insight behind the MixedGrad algorithm is to use infrequent full gradients to progressively reduce the variance of stochastic gradients as the optimization proceeds.

There are few directions that are worthy of investigation. First, it would be interesting to examine the optimality of our algorithm, namely if it is possible to achieve a better convergence rate for stochastic optimization of smooth functions using $O(\ln T)$ accesses to the full gradient oracle. Furthermore, to alleviate the computational cost caused by $O(\log T)$ accesses to the full gradient oracle, it would be interesting to empirically evaluate the proposed algorithm in a distributed framework by distributing the individual functions among processors to parallelize the full gradient computation at the beginning of each epoch which requires $O(\log T)$ communications between the processors in total. Lastly, it is very interesting to check whether an $O(1/T^2)$ rate could be achieved by an accelerated method in the mixed optimization scenario, and whether linear convergence rates could be achieved in the strongly-convex case.

7.6 Bibliographic Notes

Deterministic Smooth Optimization. The convergence rate of gradient based methods usually depends on the analytical properties of the objective function to be optimized. When the objective function is strongly convex and smooth, it is well known that a simple GD method can achieve a linear convergence rate [33]. For a non-smooth Lipschitz-continuous function, the optimal rate for the first order method is only $O(1/\sqrt{T})$ [121]. Although $O(1/\sqrt{T})$ rate is not improvable in general, several recent studies are able to improve this rate to $O(1/T)$ by exploiting the special structure of the objective function [123, 122]. In the full gradient based convex optimization, smoothness is a highly desirable property. It has been shown that a simple GD achieves a convergence rate of $O(1/T)$ when the objective function is smooth, which is further can be improved to $O(1/T^2)$ by using the accelerated gradient methods [120, 123, 121].

Stochastic Smooth Optimization. Unlike the optimization methods based on full gradients, the smoothness assumption was not exploited by most stochastic optimization methods. In fact, it was shown in [119] that the $O(1/\sqrt{T})$ convergence rate for stochastic optimization cannot be improved even when the objective function is smooth. This classical result is further confirmed by the recent studies of composite bounds for the first order optimization methods [17, 96]. The smoothness of the objective function is exploited extensively in mini-batch stochastic optimization [44, 47], where the goal is not to improve the convergence rate but to reduce the variance in stochastic gradients and consequentially the number of times for updating the solutions [154]. We finally note that the smoothness assumption coupled with the strong convexity of function is beneficial in stochastic setting and yields a geometric convergence in $e$xpectation using Stochastic Average Gradient (SAG) and Stochastic Dual Coordinate Ascent (SDCA) algorithms proposed in [128] and [135], respectively.

Finally, we would like to distinguish mixed optimization from hybrid methods that use growing sample-sizes as optimization method proceeds to gradually transform the iterates into the full gradient method [60], which makes the iterations to be dependent to the sample size n as opposed to SGD. In contrast, MixedGrad is as an alternation of deterministic and stochastic gradient steps, with different of frequencies for each type of steps. Our result for mixed optimization is useful for the scenario when the full gradient of the objective function can be computed relatively efficient although it is still significantly more expensive than computing a stochastic gradient. An example of such a scenario is distributed computing where the computation of full gradients can be speeded up by having it run in parallel on many machines with each machine containing a relatively small subset of the entire training data. Of course, the latency due to the communication between machines will result in an additional cost for computing the full gradient in a distributed fashion.

Chapter 8

Mixed Optimization for Smooth and Strongly Convex Losses

In the preceding chapter, we presented a new paradigm for stochastic optimization that allowed us to leverage the smoothness of objective function to devise faster algorithms. In this chapter of thesis, we continue our study of efficient optimization algorithms in mixed optimization regime and show that we may leverage the smoothness assumption of loss functions to devise algorithms with iteration complexities that are independent of condition number in accessing the full gradient oracle. To motivate the setting considered in this chapter, consider the optimization of smooth and strongly convex functions where the optimal iteration complexity of the gradient-based algorithm is $O(\sqrt{\kappa}\log 1/\epsilon)$, where κ is the condition number of the objective function to be optimized. In the case that the optimization problem is ill-conditioned, we need to evaluate a larger number of full gradients, which could be computationally expensive despite the linear convergence rate of the algorithm in terms of the target accuracy ϵ.

In this chapter, we propose to reduce the number of full gradients required by allowing the algorithm to access the stochastic gradients of the objective function. To this end, we present an algorithm named Epoch Mixed Gradient Descent (EMGD) that is able to utilize two kinds of gradients similar to Chapter 7. As same as MixedGrad a distinctive step in

EMGD is the mixed gradient descent, where we use a combination of the full gradient and the stochastic gradient to update the intermediate solutions. By performing a fixed number of mixed gradient descents, we are able to improve the sub-optimality of the solution by a constant factor, and thus achieve a linear convergence rate. Theoretical analysis shows that EMGD is able to find an ϵ-optimal solution by computing $O(\log 1/\epsilon)$ full gradients and $O(\kappa^2 \log 1/\epsilon)$ stochastic gradients. We also provide experimental evidence complementing our theoretical results for classification problem on few medium-sized data sets.

8.1 Introduction

The optimal iteration complexities for some popular optimization methods considering different combinations of characteristics of the objective function are shown in Table 8.1. We observe that when the objective function is smooth (and strongly convex), the convergence rate for full gradient descent is much faster than that for stochastic gradient descent. On the other hand, the evaluation of a stochastic gradient is usually significantly more efficient than that for a full gradient. Thus, replacing full gradients with stochastic gradients essentially trades the number of iterations with a low computational cost per iteration.

In this chapter, we consider the case when the objective function is both smooth and strongly convex, where the *optimal* iteration complexity is $O(\sqrt{\kappa} \log \frac{1}{\epsilon})$ if the optimization method is first order and has access to the full gradients. For the optimization problems that are ill-conditioned, the condition number κ can be very large, leading to many evaluations of full gradients, an operation that is computationally expensive for large data sets. To reduce the computational cost, we are interested in the possibility of making the number of full gradients required independent from κ. Although the $O(\sqrt{\kappa} \log \frac{1}{\epsilon})$ rate is in general not

	Lipschitz continuous	Smooth	Smooth & Strongly Convex
Full Gradient	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{L}{\sqrt{\epsilon}}\right)$	$O\left(\sqrt{\kappa}\log\frac{1}{\epsilon}\right)$
Stochastic Gradient	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{1}{\epsilon^2}\right)$	$O\left(\frac{1}{\lambda\epsilon}\right)$

Table 8.1: The optimal iteration complexity of convex optimization. L and λ are the moduli of smoothness and strongly convexity, respectively. $\kappa = L/\lambda$ is the conditional number.

improvable for any first order method, we bypass this difficulty by allowing the algorithm to have access to both full and stochastic gradients. Our objective is to reduce the iteration complexity from $O(\sqrt{\kappa}\log\frac{1}{\epsilon})$ to $O(\log\frac{1}{\epsilon})$ by replacing most of the evaluations of full gradients with the evaluations of stochastic gradients. Under the assumption that stochastic gradients can be computed efficiently, this tradeoff could lead to a significant improvement in computational efficiency.

We propose an efficient algorithm, dubbed Epoch Mixed Gradient Descent (EMGD), which fits the mixed optimization regime introduced in Chapter 7. The proposed EMGD algorithm divides the optimization process into a sequence of different epochs, an idea that is borrowed from the epoch gradient descent [72]. In each epoch, the proposed algorithm performs *mixed gradient descent* by evaluating *one* full gradient and $O(\kappa^2)$ stochastic gradients. It achieves a constant reduction in the optimization error for every epoch, leading to a linear convergence rate. Our analysis shows that EMGD is able to find an ϵ-optimal solution by computing $O(\log\frac{1}{\epsilon})$ full gradients and $O(\kappa^2\log\frac{1}{\epsilon})$ stochastic gradients. In other words, with the help of stochastic gradients, the number of full gradients required is reduced from $O(\sqrt{\kappa}\log\frac{1}{\epsilon})$ to $O(\log\frac{1}{\epsilon})$, independent from the condition number.

8.2 The Epoch Mixed Gradient Descent Algorithm

First, we recall from Chapter 2 that we wish to solve the following optimization problem

$$\min_{\mathbf{w}\in\mathcal{W}} \mathcal{F}(\mathbf{w}) \quad \text{for} \quad \mathcal{F}(\mathbf{w}) = \mathbb{E}[f(\mathbf{w},\xi)] = \int_{\Xi} f(\mathbf{w},\xi)dP(\xi), \tag{8.1}$$

where $\mathcal{W}$ is a convex domain, and $f(\mathbf{w},\xi)$ is a convex function with respect to the first argument. An special setting which is more appropriate for machine learning tasks is the case when the objective function can be written as a sum of finite number of convex functions, i.e.,

$$\mathcal{F}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{w},\xi_i). \tag{8.2}$$

For learning problems such as classification and regression, each individual function in the summand can be considered as the prediction loss on the ith training example $\xi_i = (\mathbf{x}_i, y_i)$ for a fixed loss function, i.e., $f(\mathbf{w},\xi_i) = \ell(\mathbf{w};(\mathbf{x}_i,y_i))$. For simplicity of exposition, we absorb the randomness in the individual functions and use $f_i(\mathbf{w})$ to denote the loss on random sample ξ_i instead of $f(\mathbf{w},\xi_i)$. We note that although the formulation in (8.2) seems attractive from a practical point of view, but the proposed algorithm is general enough to solve any stochastic optimization problem formulated in (8.1). As a result, in the remainder of this chapter we base the randomness on sampling the individual functions according to the unknown distribution defined over functions.

Similar to the setting introduced in Chapter 7, we assume there exist two oracles.

1. The first one is a gradient oracle $\mathcal{O}_g$, which for a given input point $\mathbf{w}$ returns the

gradient $\nabla\mathcal{F}(\mathbf{w})$, that is,

$$\mathcal{O}_g(\mathbf{w}) = \nabla\mathcal{F}(\mathbf{w}).$$

2. The second one is a function oracle $\mathcal{O}_f$, each call of which returns a random function $f(\mathbf{w})$, such that

$$\mathcal{F}(\mathbf{w}) = \mathbb{E}_f[f(\mathbf{w})], \forall \mathbf{w} \in \mathcal{W},$$

and $f(\mathbf{w})$ has Lipschitz continuous gradients with constant L, that is,

$$\|\nabla f(\mathbf{w}) - \nabla f(\mathbf{w}')\| \leq L\|\mathbf{w} - \mathbf{w}'\|, \ \forall \mathbf{w}, \mathbf{w}' \in \mathcal{W}. \tag{8.3}$$

Although we do not define a stochastic gradient oracle directly, the function oracle $\mathcal{O}_f$ allows us to evaluate the stochastic gradient of $\mathcal{F}(\mathbf{w})$ at any point $\mathbf{w} \in \mathcal{W}$.

Notice that the assumption about the function oracle $\mathcal{O}_f$ implies that the objective function $\mathcal{F}(\cdot)$ is also L-smooth. To see this, since $\nabla\mathcal{F}(\mathbf{w}) = \mathbb{E}_f[f(\mathbf{w})]$, by Jensen's inequality we have:

$$\|\nabla\mathcal{F}(\mathbf{w}) - \nabla\mathcal{F}(\mathbf{w}')\| \leq \mathbb{E}_f\|\nabla f(\mathbf{w}) - \nabla f(\mathbf{w}')\| \leq L\|\mathbf{w} - \mathbf{w}'\|, \ \forall \mathbf{w}, \mathbf{w}' \in \mathcal{W}. \tag{8.4}$$

Besides, we further assume $\mathcal{F}(\cdot)$ is λ-strongly convex, that is,

$$\|\nabla\mathcal{F}(\mathbf{w}) - \nabla\mathcal{F}(\mathbf{w}')\| \geq \lambda\|\mathbf{w} - \mathbf{w}'\|, \ \forall \mathbf{w}, \mathbf{w}' \in \mathcal{W}. \tag{8.5}$$

From (8.4) and (8.5) it is straightforward to see that $L \geq \lambda$. The condition number κ is defined as the ratio between these two parameters, i.e., $\kappa = L/\lambda \geq 1$.

The detailed steps of the proposed Epoch Mixed Gradient Descent (EMGD) are shown in Algorithm 11, where we use the superscript for the index of epochs, and the subscript for the index of iterations in each epoch. Similar to the MixedGrad algorithm, we divided the optimization process into a sequence of epochs (step 4 to step 11). While in the MixedGrad algorithm the size of epochs increases exponentially and the full gradient oracle is called at the beginning of each epoch, the size of epochs and the number of access to the two types of oracles in EMGD is fixed.

At the beginning of each epoch, we initialize the solution $\mathbf{w}_1^k$ to be the average solution $\bar{\mathbf{w}}^k$ obtained from the last epoch, and then call the gradient oracle $\mathcal{O}_g$ to obtain $\nabla\mathcal{F}(\bar{\mathbf{w}}^k)$. At each iteration t of epoch k, we call the function oracle $\mathcal{O}_f$ to obtain a random function $f_t^k(\mathbf{w})$ and define the *mixed gradient* at the current solution $\mathbf{w}_t^k$ as

$$\tilde{\mathbf{g}}_t^k = \nabla\mathcal{F}(\bar{\mathbf{w}}^k) + \nabla f_t^k(\mathbf{w}_t^k) - \nabla f_t^k(\bar{\mathbf{w}}^k),$$

which involves both the full gradient and the stochastic gradient. The mixed gradient can be divided into two parts: the deterministic part $\nabla\mathcal{F}(\bar{\mathbf{w}}^k)$ and the stochastic part $\nabla f_t^k(\mathbf{w}_t^k) - \nabla f_t^k(\bar{\mathbf{w}}^k)$. Due to the smoothness property of $f_t^k(\cdot)$, the norm of the stochastic part is well bounded, which facilitates the convergence analysis.

Based on the mixed gradient, we update $\mathbf{w}_t^k$ by a gradient mapping over a shrinking domain (i.e., $\mathcal{W} \cap \|\mathbf{w} - \bar{\mathbf{w}}^k\| \leq \Delta^k$) in step 9. Since the updating is similar to the standard gradient descent except for the domain constraint, we refer to it as mixed gradient descent for short. At the end of the iterations for epoch k, we compute the average value of $T+1$ solutions, instead of T solutions, and update the domain size by reducing a factor of $\sqrt{2}$.

The following theorem shows the convergence rate of the proposed algorithm.

Algorithm 11 Epoch Mixed Gradient Descent (EMGD) Algorithm

```
Input:
  • step size $\eta$
  • the initial domain size $\Delta^1$
  • the number of iterations $T$ per epoch
  • the number of epochs $m$
Initialize: $\bar{\mathbf{w}}^1 = \mathbf{0}$
for $k = 1, \dots, m$ do
    Set $\mathbf{w}_1^k = \bar{\mathbf{w}}^k$
    Call the gradient oracle $\mathcal{O}_g$ to obtain $\nabla \mathcal{F}(\bar{\mathbf{w}}^k)$
    for $t = 1, \dots, T$ do
        Call the function oracle $\mathcal{O}_f$ to obtain a random function $f_t^k(\cdot)$
        Compute the mixed gradient as
            $$\tilde{\mathbf{g}}_t^k = \nabla \mathcal{F}(\bar{\mathbf{w}}^k) + \nabla f_t^k(\mathbf{w}_t^k) - \nabla f_t^k(\bar{\mathbf{w}}^k)$$
        Update the solution by
            $$\mathbf{w}_{t+1}^k = \arg \min_{\mathbf{w} \in \mathcal{W} \cap \|\mathbf{w} - \bar{\mathbf{w}}^k\| \leq \Delta^k} \eta \langle \mathbf{w} - \mathbf{w}_t^k, \tilde{\mathbf{g}}_t^k \rangle + \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t^k\|^2$$
    end for
    Set $\bar{\mathbf{w}}^{k+1} = \frac{1}{T+1}\sum_{t=1}^{T+1} \mathbf{w}_t^k$ and $\Delta^{k+1} = \Delta^k/\sqrt{2}$
end for
Return $\bar{\mathbf{w}}^{m+1}$
```

Theorem 8.1. *Assume*

$$\delta \leq e^{-1/2}, \; T \geq \frac{1152L^2}{\lambda^2} \ln \frac{1}{\delta}, \; \textit{and } \Delta^1 \geq \max\left(\sqrt{\frac{2}{\lambda}(\mathcal{F}(\mathbf{0}) - \mathcal{F}(\mathbf{w}_*))}, \|\mathbf{w}_*\|\right). \tag{8.6}$$

Set $\eta = 1/[L\sqrt{T}]$. Let $\bar{\mathbf{w}}^{m+1}$ be the solution returned by Algorithm 11 after m epoches that has m access to oracle $\mathcal{O}_g$ and mT access to oracle $\mathcal{O}_f$. Then, with a probability at least $1 - m\delta$, we have

$$\mathcal{F}(\bar{\mathbf{w}}^{m+1}) - \mathcal{F}(\mathbf{w}_*) \leq \frac{\lambda[\Delta^1]^2}{2^{m+1}}, \; \textit{and } \|\bar{\mathbf{w}}^{m+1} - \mathbf{w}_*\|^2 \leq \frac{[\Delta^1]^2}{2^m}.$$

Theorem 8.1 immediately implies that EMGD is able to achieve an ϵ optimization error by computing $O(\log \frac{1}{\epsilon})$ full gradients and $O(\kappa^2 \log \frac{1}{\epsilon})$ stochastic gradients.

8.3 Analysis of Convergence Rate

The proof of Theorem 8.1 is based on induction. From the assumption about Δ^1 in (8.6), we have

$$\mathcal{F}(\bar{\mathbf{w}}^1) - \mathcal{F}(\mathbf{w}_*) \leq \frac{\lambda[\Delta^1]^2}{2}, \text{ and } \|\bar{\mathbf{w}}^1 - \mathbf{w}_*\|^2 \leq [\Delta^1]^2,$$

which means Theorem 8.1 is true for $m = 0$. Suppose Theorem 8.1 is true for $m = k$. That is, with a probability at least $1 - k\delta$, we have

$$\mathcal{F}(\bar{\mathbf{w}}^{k+1}) - \mathcal{F}(\mathbf{w}_*) \leq \frac{\lambda[\Delta^1]^2}{2^{k+1}}, \text{ and } \|\bar{\mathbf{w}}^{k+1} - \mathbf{w}_*\|^2 \leq \frac{[\Delta^1]^2}{2^k}.$$

Our goal is to show that after running the k+1-th epoch, with a probability at least $1-(k+1)\delta$, we have

$$\mathcal{F}(\bar{\mathbf{w}}^{k+2}) - \mathcal{F}(\mathbf{w}_*) \leq \frac{\lambda[\Delta^1]^2}{2^{k+2}}, \text{ and } \|\bar{\mathbf{w}}^{k+2} - \mathbf{w}_*\|^2 \leq \frac{[\Delta^1]^2}{2^{k+1}}.$$

For the simplicity of presentation, we drop the index k for epoch. Let $\bar{\mathbf{w}}$ be the solution obtained from the epoch k. Given the condition

$$\mathcal{F}(\bar{\mathbf{w}}) - \mathcal{F}(\mathbf{w}_*) \leq \frac{\lambda}{2}\Delta^2, \text{ and } \|\bar{\mathbf{w}} - \mathbf{w}_*\|^2 \leq \Delta^2, \tag{8.7}$$

we will show that that after running the T iterations in one epoch, the new solution, denoted by $\widehat{\mathbf{w}}$, satisfies

$$\mathcal{F}(\widehat{\mathbf{w}}) - \mathcal{F}(\mathbf{w}_*) \leq \frac{\lambda}{4}\Delta^2, \text{ and } \|\widehat{\mathbf{w}} - \bar{\mathbf{w}}\|^2 \leq \frac{1}{2}\Delta^2, \tag{8.8}$$

with a probability at least $1 - \delta$.

In the proof, we frequently use the following property of strongly convex function (see Appendix **??**).

Lemma 8.2. *Let $\mathcal{F}(\mathbf{w})$ be a λ-strongly convex function over the domain $\mathcal{W}$, and $\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathcal{W}} \mathcal{F}(\mathbf{w})$. Then, for any $\mathbf{w} \in \mathcal{W}$, we have*

$$\mathcal{F}(\mathbf{w}) - \mathcal{F}(\mathbf{w}_*) \geq \frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}_*\|^2. \tag{8.9}$$

Define

$$\mathbf{g} = \nabla\mathcal{F}(\bar{\mathbf{w}}), \ \widehat{\mathcal{F}}(\mathbf{w}) = \mathcal{F}(\mathbf{w}) - \langle \mathbf{w}, \mathbf{g}\rangle, \text{ and } g_t(\mathbf{w}) = f_t(\mathbf{w}) - \langle \mathbf{w}, \nabla f_t(\bar{\mathbf{w}})\rangle. \tag{8.10}$$

The objective function can be rewritten as

$$\mathcal{F}(\mathbf{w}) = \langle \mathbf{w}, \mathbf{g}\rangle + \widehat{\mathcal{F}}(\mathbf{w}). \tag{8.11}$$

And the mixed gradient can be rewritten as

$$\tilde{\mathbf{g}}^k = \mathbf{g} + \nabla g_t(\mathbf{w}_t).$$

Then, the updating rule given in Algorithm 11 becomes

$$\mathbf{w}_{t+1} = \arg \min_{\mathbf{w} \in \mathcal{W} \cap \|\mathbf{w} - \bar{\mathbf{w}}\| \leq \Delta} \eta \langle \mathbf{w} - \mathbf{w}_t, \mathbf{g} + \nabla g_t(\mathbf{w}_t) \rangle + \frac{1}{2}\|\mathbf{w} - \mathbf{w}_t\|^2. \tag{8.12}$$

For each iteration t in the current epoch, we have

$$\begin{aligned}
& \mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\mathbf{w}_*) \\
\overset{(8.5)}{\leq} & \langle \nabla \mathcal{F}(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_* \rangle - \frac{\lambda}{2}\|\mathbf{w}_t - \mathbf{w}_*\|^2 \\
\overset{(8.11)}{=} & \langle \mathbf{g} + \nabla g_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_* \rangle + \left\langle \nabla \widehat{\mathcal{F}}(\mathbf{w}_t) - \nabla g_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_* \right\rangle - \frac{\lambda}{2}\|\mathbf{w}_t - \mathbf{w}_*\|^2,
\end{aligned} \tag{8.13}$$

and

$$\begin{aligned}
& \langle \mathbf{g} + \nabla g_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_* \rangle \\
= & \langle \mathbf{g} + \nabla g_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_* \rangle - \frac{\|\mathbf{w}_t - \mathbf{w}_*\|^2}{2\eta} + \frac{\|\mathbf{w}_t - \mathbf{w}_*\|^2}{2\eta} \\
\overset{(8.12),\,(8.13)}{\leq} & \langle \mathbf{g} + \nabla g_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_{t+1} \rangle - \frac{\|\mathbf{w}_t - \mathbf{w}_{t+1}\|^2}{2\eta} - \frac{\|\mathbf{w}_{t+1} - \mathbf{w}_*\|^2}{2\eta} + \frac{\|\mathbf{w}_t - \mathbf{w}_*\|^2}{2\eta} \\
\leq & \langle \mathbf{g}, \mathbf{w}_t - \mathbf{w}_{t+1} \rangle + \frac{\|\mathbf{w}_t - \mathbf{w}_*\|^2}{2\eta} - \frac{\|\mathbf{w}_{t+1} - \mathbf{w}_*\|^2}{2\eta} \\
& + \max_{\mathbf{w}} \left(\langle \nabla g_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w} \rangle - \frac{\|\mathbf{w}_t - \mathbf{w}\|^2}{2\eta} \right) \\
= & \langle \mathbf{g}, \mathbf{w}_t - \mathbf{w}_{t+1} \rangle + \frac{\|\mathbf{w}_t - \mathbf{w}_*\|^2}{2\eta} - \frac{\|\mathbf{w}_{t+1} - \mathbf{w}_*\|^2}{2\eta} + \frac{\eta}{2}\|\nabla g_t(\mathbf{w}_t)\|^2.
\end{aligned} \tag{8.14}$$

Combining (8.13) and (8.14), we have

$$\begin{aligned}
& \mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\mathbf{w}_*) \\
\leq & \frac{\|\mathbf{w}_t - \mathbf{w}_*\|^2}{2\eta} - \frac{\|\mathbf{w}_{t+1} - \mathbf{w}_*\|^2}{2\eta} - \frac{\lambda}{2}\|\mathbf{w}_t - \mathbf{w}_*\|^2 \\
& + \langle \mathbf{g}, \mathbf{w}_t - \mathbf{w}_{t+1} \rangle + \frac{\eta}{2}\|\nabla g_t(\mathbf{w}_t)\|^2 + \left\langle \nabla \widehat{\mathcal{F}}(\mathbf{w}_t) - \nabla g_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_* \right\rangle.
\end{aligned}$$

By adding the inequalities of all iterations, we have

$$\begin{aligned}
&\sum_{t=1}^{T} \mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\mathbf{w}_*) \\
\leq & \frac{\|\bar{\mathbf{w}} - \mathbf{w}_*\|^2}{2\eta} - \frac{\|\mathbf{w}_{T+1} - \mathbf{w}_*\|^2}{2\eta} - \frac{\lambda}{2}\sum_{t=1}^{T} \|\mathbf{w}_t - \mathbf{w}_*\|^2 + \langle \mathbf{g}, \bar{\mathbf{w}} - \mathbf{w}_{T+1} \rangle \\
& + \frac{\eta}{2} \underbrace{\sum_{t=1}^{T} \|\nabla g_t(\mathbf{w}_t)\|^2}_{\triangleq A_T} + \underbrace{\sum_{t=1}^{T} \langle \nabla \widehat{\mathcal{F}}(\mathbf{w}_t) - \nabla g_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_* \rangle}_{\triangleq B_T}.
\end{aligned} \tag{8.15}$$

Since $\mathcal{F}(\cdot)$ is L-smooth, we have

$$\mathcal{F}(\mathbf{w}_{T+1}) - \mathcal{F}(\bar{\mathbf{w}}) \leq \langle \nabla \mathcal{F}(\bar{\mathbf{w}}), \mathbf{w}_{T+1} - \bar{\mathbf{w}} \rangle + \frac{L}{2}\|\bar{\mathbf{w}} - \mathbf{w}_{T+1}\|^2,$$

which implies

$$\begin{aligned}
&\langle \mathbf{g}, \bar{\mathbf{w}} - \mathbf{w}_{T+1} \rangle \\
\leq & \mathcal{F}(\bar{\mathbf{w}}) - \mathcal{F}(\mathbf{w}_{T+1}) + \frac{L}{2}\Delta^2 \\
\overset{(8.7)}{\leq} & \mathcal{F}(\mathbf{w}_*) - \mathcal{F}(\mathbf{w}_{T+1}) + \frac{\lambda}{2}\Delta^2 + \frac{L}{2}\Delta^2 \\
\leq & \mathcal{F}(\mathbf{w}_*) - \mathcal{F}(\mathbf{w}_{T+1}) + L\Delta^2.
\end{aligned} \tag{8.16}$$

From (8.15) and (8.16), we have

$$\sum_{t=1}^{T+1} \mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\mathbf{w}_*) \leq \Delta^2 \left(\frac{1}{2\eta} + L \right) + \frac{\eta}{2} A_T + B_T. \tag{8.17}$$

Next, we consider how to bound A_T and B_T. The upper bound of A_T is given by

$$A_T = \sum_{t=1}^{T} \|\nabla g_t(\mathbf{w}_t)\|^2 = \sum_{t=1}^{T} \|\nabla f_t(\mathbf{w}_t) - \nabla f_t(\bar{\mathbf{w}})\|^2 \overset{(8.3)}{\leq} L^2 \sum_{t=1}^{T} \|\mathbf{w}_t - \bar{\mathbf{w}}\|^2 \leq TL^2\Delta^2. \tag{8.18}$$

To bound B_T, we need the Hoeffding-Azuma inequality which is stated in Theorem A.20 for completeness. Define

$$V_t = \langle \nabla \widehat{\mathcal{F}}(\mathbf{w}_t) - \nabla g_t(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_* \rangle, \ t = 1, \ldots, T.$$

Recall the definition of $\widehat{\mathcal{F}}(\mathbf{w})$ and $g_t(\mathbf{w})$ in (8.10). Based on our assumption about the function oracle $\mathcal{O}_f$, it is straightforward to check that $V_1, \ldots$ is a martingale difference with respect to $g_1, \ldots$. The value of V_t can be bounded by

$$\begin{aligned}
|V_t| \quad & \leq \quad \left\| \nabla \widehat{\mathcal{F}}(\mathbf{w}_t) - \nabla g_t(\mathbf{w}_t) \right\| \|\mathbf{w}_t - \mathbf{w}_*\| \\
& \leq \quad 2\Delta \left(\|\nabla \mathcal{F}(\mathbf{w}_t) - \nabla \mathcal{F}(\bar{\mathbf{w}})\| + \|\nabla f_t(\mathbf{w}_t) - \nabla f_t(\bar{\mathbf{w}})\| \right) \\
& \overset{(8.3),\,(8.4)}{\leq} \quad 4L\Delta \|\mathbf{w}_t - \bar{\mathbf{w}}\| \leq 4L\Delta^2.
\end{aligned}$$

Following Theorem A.20, with a probability at least $1 - \delta$, we have

$$B_T \leq 4L\Delta^2 \sqrt{2T \ln \frac{1}{\delta}}. \tag{8.19}$$

By adding the inequalities in (8.17), (8.18) and (8.19) together, with a probability at least $1 - \delta$, we have

$$\sum_{t=1}^{T+1} \mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\mathbf{w}_*) \leq \Delta^2 \left(\frac{1}{2\eta} + L + \frac{\eta T L^2}{2} + 4L\sqrt{2T \ln \frac{1}{\delta}} \right).$$

By choosing $\eta = 1/[L\sqrt{T}]$, we have

$$\sum_{t=1}^{T+1} \mathcal{F}(\mathbf{w}_t) - \mathcal{F}(\mathbf{w}_*) \leq L\Delta^2 \left(\sqrt{T} + 1 + 4\sqrt{2T \log \frac{1}{\delta}}\right) \leq 6L\Delta^2\sqrt{2T \ln \frac{1}{\delta}}.$$

and therefore

$$\mathcal{F}(\widehat{\mathbf{w}}) - \mathcal{F}(\mathbf{w}_*) \leq \Delta^2 \frac{6L\sqrt{2 \ln 1/\delta}}{\sqrt{T+1}}, \text{ and } \|\widehat{\mathbf{w}} - \mathbf{w}_*\|^2 \overset{(\text{A.2})}{\leq} \Delta^2 \frac{12L\sqrt{2\ln 1/\delta}}{\lambda\sqrt{T+1}}.$$

Thus, when

$$T \geq \frac{1152L^2}{\lambda^2} \ln \frac{1}{\delta},$$

with a probability at least $1 - \delta$, we have

$$\mathcal{F}(\widehat{\mathbf{w}}) - \mathcal{F}(\mathbf{w}_*) \leq \frac{\lambda}{4}\Delta^2, \text{ and } \|\widehat{\mathbf{w}} - \mathbf{w}_*\|^2 \leq \frac{1}{2}\Delta^2.$$

8.4 Experiments

In this section, we provide experimental evidence complementing our theoretical results. In particular we consider solving the regularized logistic regression problem formulated as:

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w}) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

where

$$f_i(\mathbf{w}) = \log\left(1 + \exp(-y_i\langle \mathbf{w}, \mathbf{x}_i\rangle)\right), \text{ and } \nabla f_i(\mathbf{w}) = \frac{-y_i}{1 + \exp(y_i\langle \mathbf{w}, \mathbf{x}_i\rangle)}\mathbf{x}_i.$$

We compare the EMGD algorithm to the SGD method. SGD start with solution $\mathbf{w}_1 = \mathbf{0}$ and at each iteration samples a random function indexed by i_k uniformly at random over all n available functions and updates the solution by:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \frac{1}{\lambda t}\left(\nabla f_{i_k}(\mathbf{w}_t) + \lambda \mathbf{w}_t\right) = \left(1 - \frac{1}{t}\right)\mathbf{w}_t - \frac{1}{\lambda t}\nabla f_{i_k}(\mathbf{w}_t).$$

The variance of SGD method at each iteration can be computed as:

$$\left\|\nabla f_{i_k}(\mathbf{w}_t) + \lambda \mathbf{w}_t - \left(\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{w}_t) + \lambda \mathbf{w}_t\right)\right\|^2 = \mathbb{E}_{i_k}\left\|\nabla f_{i_k}(\mathbf{w}_t) - \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{w}_t)\right\|^2$$
$$= \mathbb{E}_{i_k}\left\|\nabla f_{i_k}(\mathbf{w}_t)\right\|^2 - \left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{w}_t)\right\|^2 = \frac{1}{n}\left\|\nabla f_i(\mathbf{w}_t)\right\|^2 - \left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{w}_t)\right\|^2$$

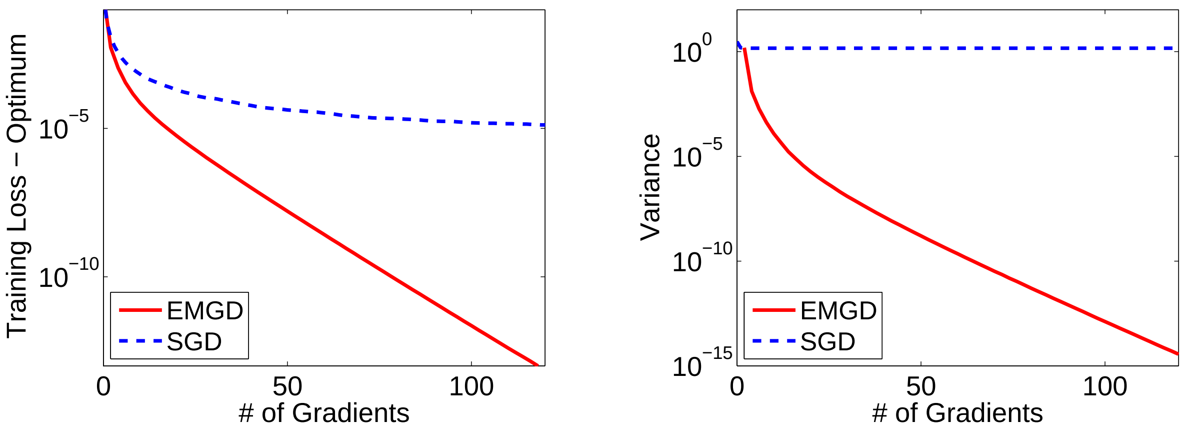
By noting the gradient at each iteration as

$$\left(\nabla f_{i_k}(\mathbf{w}_t) + \lambda \mathbf{w}_t\right) - \left(\nabla f_{i_k}(\bar{\mathbf{w}}) + \lambda \bar{\mathbf{w}}\right) + \left(\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\bar{\mathbf{w}}) - \lambda \bar{\mathbf{w}}\right) - \left(\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{w}_t) + \lambda \mathbf{w}_t\right),$$

the variance of the mixed gradient in EMGD is given by

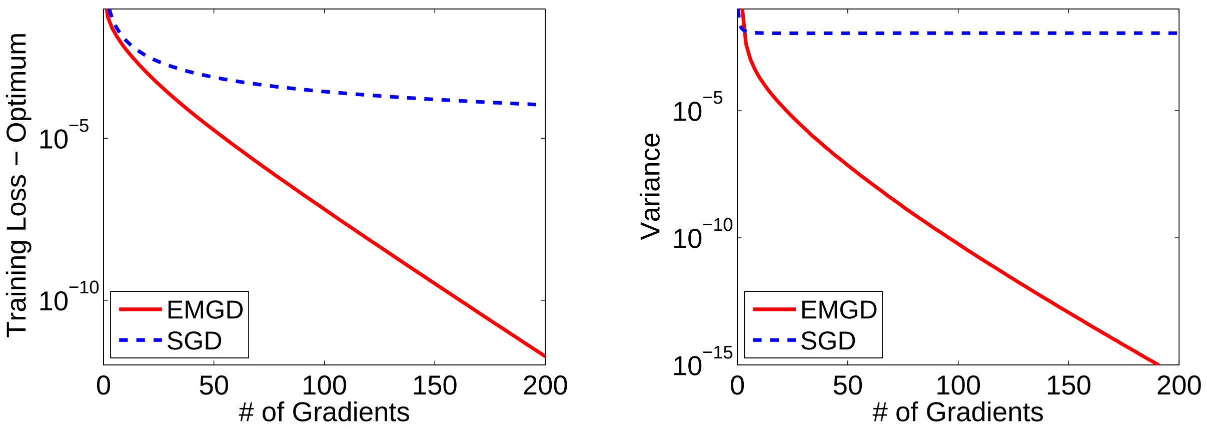
$$\mathbb{E}_{i_k}\left\|\nabla f_{i_k}(\mathbf{w}_t) - \nabla f_{i_k}(\bar{\mathbf{w}}) - \left(\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\bar{\mathbf{w}}) - \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{w}_t)\right)\right\|^2$$
$$= \frac{1}{n}\left\|\nabla f_i(\mathbf{w}_t) - \nabla f_i(\bar{\mathbf{w}})\right\|^2 - \left\|\frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\mathbf{w}_t) - \frac{1}{n}\sum_{i=1}^{n}\nabla f_i(\bar{\mathbf{w}})\right\|^2.$$

We run both algorithms on two well-known Adult and RCV1 data sets. The data sets used in this experiment were sourced from the University of California Irvine (UCI) Machine Learning Repository [11] and are referred to as the Adult and RCV1 data sets. The Adult data set contain information on individuals such as age, level of education and current



(a) The training error (b) The variance of the (stochastic or mixed) gradient

Table 8.2: Experimental results on the Adult data set. $\lambda = 1e{-}3$ and $\eta = 0.01$ in EMGD.

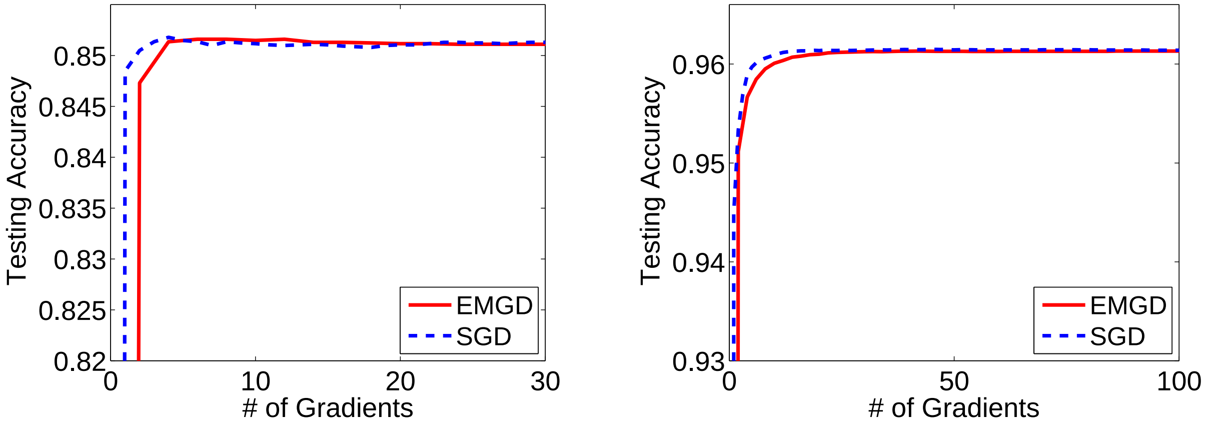


(a) The training error (b) The variance of the (stochastic or mixed) gradient

Table 8.3: Experimental results on the RCV1 data set. $\lambda = 1e{-}5$ and $\eta = 1$ in EMGD.

employment type. The Adult data set contains over 30,000 records of census information taken in 1994 from many diverse demographics. The RCV1 is an archive of over 800,000 manually categorized newswire stories recently made available by Reuters, Ltd. for research purposes. For EMGD, we set the number of stochastic gradient in each epoch to n, which is the number of training data.

The results of both SGD and EMGD algorithms on the Adult data set are provided in Figure 8.2. Figure 8.1a shows the difference between the current objective value and the



(a) The Adult data set

(b) The RCV1 data set

Table 8.4: The testing accuracy on RCV1 and Adult data sets.

optimum (which is obtained by running a batch algorithm for a long time) versus

$$\text{The number of full gradients} + \frac{\text{The number of stochastic gradients}}{n}.$$

Figure 8.1b shows the variances of the SGD and EMGD, respectively. It can be inferred from the results that the variance of SGD is almost same even when the algorithm approaches the optimal solution. As can be seen, the EMGD method is able to reduce the training error and the variance exponentially. We obtain similar results on the RCV1 data set, which are shown in Figure 8.3.

We also examine the testing accuracy of SGD and EMGD, which are depicted in Figure 8.4. As can be seen, in terms of testing, EMGD is slightly worse than SGD at the beginning. This behavior is expected since EMGD needs one full gradient to initialize.

8.5 Discussion

Compared to the optimization algorithm that only relies on full gradients [121], the number of full gradients needed in EMGD is $O(\log \frac{1}{\epsilon})$ instead of $O(\sqrt{\kappa} \log \frac{1}{\epsilon})$. Compared to the optimization algorithms that only relies on stochastic gradients [81, 72, 125], EMGD is more efficient since it achieves a linear convergence rate.

The proposed EMGD algorithm can also be applied to the special optimization problem considered in [128, 135], where $\mathcal{F}(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} f_i(\mathbf{w})$. To make quantitative comparisons, let's assume the full gradient is n times more expensive to compute than the stochastic gradient. Table 8.5 lists the computational complexity of the algorithms that enjoy linear convergence. As can be seen, the computational complexity of EMGD is lower than Nesterov's algorithm [121] as long as the condition number $\kappa \leq n^{2/3}$, the complexity of SAG [128] is lower than Nesterov's algorithm if $\kappa \leq n/8$, and the complexity of SDCA [135] is lower than Nesterov's algorithm if $\kappa \leq n^2$.[1] The complexity of EMGD is on the same order as SAG and SDCA when $\kappa \leq n^{1/2}$, but higher in other cases. Thus, in terms of computational cost, EMGD may not be the best one, but it has advantages in other aspects.

1. Unlike SAG and SDCA that only work for unconstrained optimization problem, the proposed algorithm works for both constrained and unconstrained optimization problems, provided the constrained problem in Step 9 can be solved efficiently.

[1] In learning problems, we usually face a regularized optimization problem $\min_{\mathbf{w} \in \mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i; \langle \mathbf{w}, \mathbf{x}_i \rangle) + \frac{\tau}{2} \|\mathbf{w}\|^2$, where $\ell(\cdot;\cdot)$ is some loss fixed function. When the norm of the data is bounded, the smoothness parameter L can be treated as a constant. The strong convexity parameter λ is lower bounded by τ. As a result, as long as $\tau > \Omega(n^{-2/3})$, which is a reasonable scenario [149], we have $\kappa < O(n^{2/3})$, indicating that our proposed EMGD algorithm can be applied.

Nesterov's algorithm [121]	EMGD	SAG ($n \geq 8\kappa$) [128]	SDCA [135]
$O\left(\sqrt{\kappa}n \log \frac{1}{\epsilon}\right)$	$O\left((n+\kappa^2) \log \frac{1}{\epsilon}\right)$	$O\left(n \log \frac{1}{\epsilon}\right)$	$O\left((n+\kappa) \log \frac{1}{\epsilon}\right)$

Table 8.5: The computational complexity for minimizing $(1/n)\sum_{i=1}^{n} f_i(\mathbf{w})$

2. Unlike the SAG and SDCA that require an $\Omega(n)$ storage space, the proposed algorithm only requires the storage space of $\Omega(d)$, where d is the dimension of $\mathbf{w}$.

3. The only step in Algorithm 11 that has dependence on n is step 5 for computing the gradient $\nabla \mathcal{F}(\bar{\mathbf{w}}^k)$. By utilizing distributed computing, the running time of this step can be reduced to $O(n/k)$, where k is the number of computers, and the convergence rate remains the same. For SAG and SDCA , it is unclear whether they can reduce the running time without affecting the convergence rate.

4. The linear convergence of SAG and SDCA only holds in expectation, whereas the linear convergence of EMGD holds with a high probability, which is much stronger.

8.6 Summary

In this chapter, we considered how to reduce the number of full gradients needed for smooth and strongly convex optimization problems. Under the assumption that both the gradient and the stochastic gradient are available, the EMGD algorithm, with the help of stochastic gradients, is are able to reduce the number of gradients needed from $O(\sqrt{\kappa} \log \frac{1}{\epsilon})$ to $O(\log \frac{1}{\epsilon})$. In the case that the objective function is in the form of (8.1), i.e., a sum of n smooth functions, EMGD has lower computational cost than the full gradient method [121], if the condition number $\kappa \leq n^{2/3}$. We validated our theoretical results on the convergence of EMGD algorithm and in particular its ability in reducing the variance during the optimization by some experimental results on two data sets. We note that although EMGD enjoys many nice properties, it is unclear whether it is the optimal algorithm when two kinds of gradients are available. Finally, we provided experimental evidence complementing our theoretical results for classification problem on few standard data sets.

8.7 Bibliographic Notes

During the last three decades, there have been significant advances in convex optimization [119, 121, 34]. In this section, we provide a brief review of the first order optimization methods.

We first discuss deterministic optimization, where the gradient of the objective function is available. For the general convex and Lipschitz continuous optimization problem, the iteration complexity of gradient (subgradient) descent is $O(\frac{1}{\epsilon^2})$, which is optimal up to constant factors [119]. When the objective function is convex and smooth, the optimal optimization scheme is the accelerated gradient descent developed by Nesterov, whose iteration complexity is $O(\frac{L}{\sqrt{\epsilon}})$ [120, 123]. With slight modifications, the accelerated gradient descent algorithm can also be applied to optimize the smooth and strongly convex objective function, whose iteration complexity is $O(\sqrt{\kappa}\log\frac{1}{\epsilon})$ and is in general not improvable [121, 124]. The objective of our work is to reduce the number of access to the full gradients by exploiting the availability of stochastic gradients.

In stochastic optimization, we have the access to the stochastic gradient, which is an unbiased estimate of the full gradient [115]. Similar to the case in deterministic optimization, if the objective function is convex and Lipschitz continuous, stochastic gradient (sub-gradient) descent is the optimal algorithm and the iteration complexity is also $O(\frac{1}{\epsilon^2})$ [119, 115]. When the objective function is strongly convex, the algorithms proposed in very recent works [81, 72, 125] achieve the optimal $O(\frac{1}{\lambda\epsilon})$ iteration complexity [4]. Since the convergence rate of stochastic optimization is dominated by the randomness in the gradient [94, 62], smoothness usually does not lead to a faster convergence rate for stochastic optimization.

From the above discussion, we observe that the iteration complexity in stochastic optimization is polynomial in $\frac{1}{\epsilon}$, making it is difficult to find high-precision solutions. However, when the objective function is strongly convex and can be written as a sum of a finite number of functions, i.e.,

$$\mathcal{F}(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{w}),$$

where each $f_i(\mathbf{w})$ is smooth, the iteration complexity of some specific algorithms may exhibit a logarithmic dependence on $\frac{1}{\epsilon}$, i.e., a linear convergence rate. The first such algorithm is the stochastic average gradient (SAG) [128], whose iteration complexity is $O(n \log \frac{1}{\epsilon})$, provided $n \geq 8\kappa$. The second one is the stochastic dual coordinate ascent (SDCA) [135], whose iteration complexity is $O((n+\kappa)\log\frac{1}{\epsilon})$.

Chapter 9

Efficient Optimization with Bounded Projections

In this chapter we aim at developing more efficient optimization methods by reducing the number of projection steps. An examination of optimization methods for both online and convex optimization problems introduced in Chapter 2 reveals that most of them require projecting the updated solution at *each* iteration to ensure that the obtained solution stays within the feasible domain. For complex domains (e.g., positive semidefinite cone), the projection step can be computationally expensive, making first order optimization methods unattractive for large-scale optimization problems. The broad question in this chapter of the thesis is the extent to which it is possible to reduce the number of projection steps in stochastic and online optimization algorithms.

In stochastic setting, we address this limitation by developing novel stochastic optimization algorithms that do not need intermediate projections. Instead, only one projection at the last iteration is needed to obtain a feasible solution in the given domain. Our theoretical analysis shows that with a high probability, the proposed algorithms achieve an $O(1/\sqrt{T})$ convergence rate for general convex functions, and an $O(\ln T/T)$ rate for strongly convex functions under mild conditions about the domain and the objective function. The key insight underlying the proposed method for strongly convex objectives is that by smoothing the objective function and leveraging it in the optimization, we are able to skip the intermediate projections. This is in contrast to other parts of the thesis where we explicitly leveraged the smoothness assumption. To the best of our knowledge, these are the first projection-free stochastic optimization methods.

In online setting, to tackle the computational challenge arising from the projection steps, we consider an alternative online learning problem as follows. Instead of requiring that each solution obeys the constraints which define the convex domain, we only require the constraints to be satisfied in a long run. Then, the online learning problem becomes a task to find a sequence of solutions under the long term constraints. In other words, instead of solving the projection on each round, we allow the learner to make decisions at some rounds which do not belong to the constraint set, but the overall sequence of chosen decisions must obey the constraints at the end by a vanishing rate. We refer to this problem as online learning with soft constraints. By turning the problem into an online convex-concave optimization problem, we propose an efficient algorithm which achieves an $O(\sqrt{T})$ regret bound and an $O(T^{3/4})$ bound on the violation of the constraints. Then, we modify the algorithm in order to guarantee that the constraints are exactly satisfied in the long run. This gain is achieved at the price of getting an $O(T^{3/4})$ regret bound.

9.1 Setup and Motivation

In stochastic setting, we consider the following convex optimization problem:

$$\min_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}), \tag{9.1}$$

where $\mathcal{W}$ is a bounded convex domain. We assume that $\mathcal{W}$ can be characterized by an inequality constraint and without loss of generality is bounded by the unit ball, i.e.,

$$\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^d : g(\mathbf{w}) \leq 0\} \subseteq \mathbb{B} = \{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\|_2 \leq 1\}, \tag{9.2}$$

where $g(\mathbf{w})$ is a convex constraint function. We assume that $\mathcal{W}$ has a non-empty interior, i.e., there exists $\mathbf{w}$ such that $g(\mathbf{w}) < 0$ and the optimal solution $\mathbf{w}_*$ to (9.1) is in the interior of the unit ball $\mathbb{B}$, i.e., $\|\mathbf{w}_*\|_2 < 1$. Note that when a domain is characterized by multiple convex constraint functions, say $g_i(\mathbf{w}) \leq 0, i = 1, \ldots, m$, we can summarize them into one constraint $g(\mathbf{w}) \leq 0$, by defining it as $g(\mathbf{w}) = \max_{1 \leq i \leq m} g_i(\mathbf{w})$.

To solve the optimization problem in (9.1), we assume that the only information available to the algorithm is through a stochastic oracle that provides unbiased estimates of the gradient of $f(\mathbf{w})$. More precisely, let $\xi_1, \ldots, \xi_T$ be a sequence of independently and identically distributed (i.i.d) random variables sampled from an unknown distribution. At each iteration t, given solution $\mathbf{w}_t$, the oracle returns $\mathbf{g}(\mathbf{w}_t; \xi_t)$, an unbiased estimate of the true gradient $\nabla f(\mathbf{w}_t)$, i.e., $\mathbb{E}_{\xi_t}[\mathbf{g}(\mathbf{w}_t, \xi_t)] = \nabla f(\mathbf{w}_t)$. The goal of the learner is to find an approximate optimal solution by making T calls to this oracle.

Recall from Chapter 2 that to find a solution within the domain $\mathcal{W}$ which optimizes the given objective function $f(\mathbf{w})$, SGD computes an unbiased estimate of the gradient of $f(\mathbf{w})$, and updates the solution by moving it in the opposite direction of the estimated gradient. To ensure that the solution stays within the domain $\mathcal{W}$, SGD has to project the updated solution back into the $\mathcal{W}$ at *every iteration*. More precisely, SGD method produces a sequence of solutions by following updating:

$$\mathbf{w}_{t+1} = \Pi_{\mathcal{W}}(\mathbf{w}_t - \eta_t \mathbf{g}(\mathbf{w}_t, \xi_t)), \tag{9.3}$$

where η_t is the step size at iteration t, $\Pi_{\mathcal{W}}(\cdot)$ is a projection operator that projects $\mathbf{w}$ into the domain $\mathcal{W}$, and $\mathbf{g}(\mathbf{w}, \xi_t)$ is an unbiased stochastic gradient of $f(\mathbf{w})$, for which we further assume bounded gradient variance as

$$\mathbb{E}_{\xi_t}[\exp(\|\mathbf{g}(\mathbf{w}, \xi_t) - \nabla f(\mathbf{w})\|_2^2/\sigma^2)] \leq \exp(1). \tag{9.4}$$

For general convex optimization, stochastic gradient descent methods can obtain an $O(1/\sqrt{T})$ convergence rate in expectation or in a high probability provided (9.4) holds [115]. Although a large number of iterations is usually needed to obtain a solution of desirable accuracy, the lightweight computation per iteration makes SGD attractive for many large-scale learning problems.

The SGD method is computationally efficient only when the projection $\Pi_{\mathcal{W}}(\cdot)$ can be carried out efficiently. Although efficient algorithms have been developed for projecting solutions into special domains (e.g., simplex and ℓ_1 ball [51, 100]); for complex domains, such as a positive semidefinite (PSD) cone in metric learning and bounded trace norm matrices in matrix completion (more examples of complex domains can be found in [73] and [77]), the projection step requires solving an expensive convex optimization, leading to a high computational cost per iteration and consequently making SGD unappealing for large-scale optimization problems over such domains. For instance, projecting a matrix into a PSD cone requires computing the full eigen-decomposition of the matrix, whose complexity is cubic in the size of the matrix.

The central theme of this chapter in stochastic setting is to develop a SGD based method that does not require projection at each iteration. This problem was first addressed in a very recent work [73], where the authors extended Frank-Wolfe algorithm [56] for online learning. But, one main shortcoming of the algorithm proposed in [73] is that it has a slower convergence rate (i.e., $O(T^{-1/3})$) than a standard SGD algorithm (i.e., $O(T^{-1/2})$). In this work, we demonstrate that a properly modified SGD algorithm can achieve the optimal convergence rate of $O(T^{-1/2})$ using only **ONE** projection for general stochastic convex optimization problem.

We further develop an SGD based algorithm for strongly convex optimization that achieves a convergence rate of $O(\ln T/T)$, which is only a logarithmic factor worse than the optimal rate [72]. The key idea of both algorithms is to appropriately penalize the intermediate solutions when they are outside the domain. With an appropriate design of penalization mechanism, the average solution $\widehat{\mathbf{w}}_T$ obtained by the SGD after T iterations will be very close to the domain $\mathcal{W}$, even without intermediate projections. As a result, the final feasible solution $\widetilde{\mathbf{w}}_T$ can be obtained by projecting $\widehat{\mathbf{w}}_T$ into the domain $\mathcal{W}$, the only projection that is needed for the entire algorithm. We note that our approach is very different from the previous efforts in developing projection free convex optimization algorithms (see [73, 78, 77] and references therein), where the key idea is to develop appropriate *updating* procedures to restore the feasibility of solutions at every iteration.

9.2 Stochastic Frank-Wolfe Algorithm

Before presenting the proposed algorithms, here we investigate and analyze a greedy algorithm, which is a slight modification of the Frank-Wolfe (FW) method, when applied to stochastic optimization problem. As discussed in Chapter 2, the FW or conditional gradient algorithm is a feasible direction method and at each iteration, it finds the best feasible direction (with respect to the linear approximation of the function) and updates the next solution as a convex combination of the current solution and chosen feasible direction. This algorithm is revisited in [77] for deterministic *smooth* convex function over a convex domain aiming to devise an efficient algorithm without projections. As discussed before, the algorithm in [77] generates a sequence of solutions via the following steps:

$$\begin{aligned} \mathbf{u}_t &= \arg\max_{\mathbf{u}\in\mathcal{W}} \langle \mathbf{u}, -\nabla f(\mathbf{w}_t)\rangle \\ \mathbf{w}_{t+1} &= (1-\eta_t)\mathbf{w}_t + \eta_t \mathbf{u}_t \end{aligned} \tag{9.5}$$

The bulk of computation in FW algorithm is step (9.5) and the algorithm is an attractive method whenever step (9.5) can be achieved with low cost complexity, for otherwise, it is not practically interesting. This is true for some special domains, such as polyhedron, simplex and ℓ_1 ball where the linear optimization problem can be efficiently solved. However, this assumption does not hold for general complex domains (e.g. general PSD cones). This limitation makes the FW algorithm computationally unattractive for general complex domains since it simply translates the complexity of quadratic optimization for projection into a linear optimization problem over the same domain. We note that the FW algorithm is also attractive because of its sparsity merits which is not the focus of this chapter and we only consider the computational efficiency of optimization methods.

A trivial modification to extend the FW algorithm to stochastic gradients is to replace the true gradient $\nabla f(\mathbf{w}_t)$ with a stochastic gradient $\mathbf{g}(\mathbf{w}_t, \xi_t)$ in (9.5). Next, we sketch an analysis of stochastic greedy algorithm and discuss the problem caused by using unbiased estimate of the gradient instead of true gradient in updating solutions.

A key inequality in the convergence analysis of stochastic greedy algorithm is

$$f(\mathbf{w}_{t+1}) \leq f(\mathbf{w}_t) - \eta_t \underbrace{\max_{\mathbf{w}\in\mathcal{W}} \langle \mathbf{w}_t - \mathbf{w}, \nabla f(\mathbf{w}_t)\rangle}_{\triangleq\, g(\mathbf{w}_t, \nabla f(\mathbf{w}_t))} + \eta_t^2 L, \tag{9.6}$$

where $f(\mathbf{w})$ is assumed to be L-smooth function, $g(\mathbf{w}, \nabla f(\mathbf{w}))) = \max_{\mathbf{u}\in\mathcal{W}} \langle \mathbf{w} - \mathbf{u}, \nabla f(\mathbf{w})\rangle$ is the duality gap. Using $g(\mathbf{w}_t, \nabla f(\mathbf{w}_t)) \geq f(\mathbf{w}_t) - f(\mathbf{w}_*)$, we can obtain an $O(1/T)$ convergence bound for the greedy algorithm with $\eta_t = 2/(t+1)$ by induction (e.g., see Theorem 2.21 in Chapter 2). However, when using the stochastic gradient, $\mathbf{u}_t = \arg\max_{\mathbf{u}\in\mathcal{W}} \langle \mathbf{u}, -\mathbf{g}(\mathbf{w}_t, \xi_t)\rangle$, and the key inequality becomes as:

$$\begin{aligned} f(\mathbf{w}_{t+1}) &\leq f(\mathbf{w}_t) - \eta_t \langle \mathbf{w}_t - \mathbf{u}_t, \nabla f(\mathbf{w}_t)\rangle + \eta_t^2 L \\ &\leq f(\mathbf{w}_t) - \eta_t g(\mathbf{w}_t, \nabla f(\mathbf{w}_t)) + \eta_t^2 L \\ &\quad + \underbrace{\eta_t(g(\mathbf{w}_t, \nabla f(\mathbf{w}_t)) - g(\mathbf{w}_t, \mathbf{g}(\mathbf{w}_t, \xi_t))) + \eta_t \langle \mathbf{w}_t - \mathbf{u}_t, \mathbf{g}(\mathbf{w}_t, \xi_t) - \nabla f(\mathbf{w}_t)\rangle}_{\triangleq\, \zeta_t} \end{aligned}$$

where the top line recovers the inequality in (9.6). However, the problem is that the quantity ζ_t in the second line of above inequality is not a martingale sequence, i.e., $\mathbb{E}_{\xi_t|\xi_{[t-1]}}[\zeta_t] \neq 0$. We can take a conservative analysis to bound $\zeta_t \leq C\eta_t \|\nabla f(\mathbf{w}_t) - \mathbf{g}(\mathbf{w}_t, \xi_t)\|_* \triangleq C\eta_t\delta_t$, where C is a constant. Assuming $\|\nabla f(\mathbf{w}_t) - \mathbf{g}(\mathbf{w}_t, \xi_t)\|_* \leq \sigma$, we could have, with a probably $1-\epsilon$, $\eta_t\delta_t \leq \eta_t(1 + \ln(1/\epsilon))\sigma$ by Markov inequality in Lemma A.18. As a result, we obtain the following recursive inequality

$$f(\mathbf{w}_{t+1}) - f(\mathbf{w}_*) \leq (1 - \eta_t)(f(\mathbf{w}_t) - f(\mathbf{w}_*)) + \eta_t^2 L + \eta_t C(1 + \ln(1/\epsilon))\sigma$$

where the last term in the above inequality makes the convergence analysis more involved. Whether it is possible to design a sequence of step sizes η_t to have a vanishing bound for $f(\mathbf{w}_t) - f(\mathbf{w}_*)$ is unclear for us and remains an open problem, which is beyond the scope of this chapter.

9.3 Stochastic Optimization with Single Projection

We now turn to extending the SGD method to the setting where only one projection is allowed to perform for the entire sequence of updating. The main idea is to incorporate the constraint function $g(\mathbf{w})$ into the objective function to penalize the intermediate solutions that are outside the domain. The result of the penalization is that, although the average solution obtained by SGD may not be feasible, it should be very close to the boundary of the domain. A projection is performed at the end of the iterations to restore the feasibility of the average solution. Before proceeding, we recall few definitions from Appendix **??** about convex analysis.

Definition 9.1. *A function $f(\mathbf{w})$ is a Lipschitz continuous function with constant G w.r.t a norm $\|\cdot\|$, if*

$$|f(\mathbf{w}_1) - f(\mathbf{w}_2)| \leq G\|\mathbf{w}_1 - \mathbf{w}_2\|, \forall \mathbf{w}_1, \mathbf{w}_2 \in \mathbb{B}. \tag{9.7}$$

In particular, a convex function $f(\mathbf{w})$ with a bounded (sub)gradient $\|\partial f(\mathbf{w})\|_* \leq G$ is

Algorithm 12 SGD with ONE Projection by Primal Dual Updating (SGD-PD)

```
Input: a sequence of step sizes $\{\eta_t\}$, and a parameter $\gamma > 0$
Initialize:: $\mathbf{w}_1 = \mathbf{0}$ and $\lambda_1 = 0$
for $t = 1, 2, \ldots, T$ do
    Compute $\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t(\mathbf{g}(\mathbf{w}_t, \xi_t) + \lambda_t \nabla g(\mathbf{w}_t))$
    Update $\mathbf{w}_{t+1} = \mathbf{w}'_{t+1} / \max(\|\mathbf{w}'_{t+1}\|_2, 1)$,
    Update $\lambda_{t+1} = [(1 - \gamma\eta_t)\lambda_t + \eta_t g(\mathbf{w}_t)]_+$
end for
Output: $\widetilde{\mathbf{w}}_T = \Pi_{\mathcal{W}}(\widehat{\mathbf{w}}_T)$, where $\widehat{\mathbf{w}}_T = \sum_{t=1}^T \mathbf{w}_t / T$.
```

G-Lipschitz continuous, where $\|\cdot\|_*$ is the dual norm to $\|\cdot\|$.

Definition 9.2. *A convex function $f(\mathbf{w})$ is β-strongly convex w.r.t a norm $\|\cdot\|$ if there exists a constant $\beta > 0$ (often called the modulus of strong convexity) such that, for any $\alpha \in [0, 1]$, it holds:*

$$f(\alpha\mathbf{w}_1 + (1-\alpha)\mathbf{w}_2) \leq \alpha f(\mathbf{w}_1) + (1-\alpha) f(\mathbf{w}_2) - \frac{1}{2}\alpha(1-\alpha)\beta\|\mathbf{w}_1 - \mathbf{w}_2\|^2, \forall \mathbf{w}_1, \mathbf{w}_2 \in \mathbb{B}.$$

When $f(\mathbf{w})$ is differentiable, the strong convexity is equivalent to

$$f(\mathbf{w}_1) \geq f(\mathbf{w}_2) + \langle \nabla f(\mathbf{w}_2), \mathbf{w}_1 - \mathbf{w}_2 \rangle + \frac{\beta}{2}\|\mathbf{w}_1 - \mathbf{w}_2\|^2, \forall \mathbf{w}_1, \mathbf{w}_2 \in \mathbb{B}.$$

In the sequel, we use the standard Euclidean norm to define Lipschitz and strongly convex functions.

The key ingredient of proposed algorithms is to replace the projection step with the gradient computation of the constraint function defining the domain $\mathcal{W}$, which is significantly cheaper than projection step. As an example, when a solution is restricted to a PSD cone, i.e., $X \succeq 0$ where X is a symmetric matrix, the corresponding inequality constraint is $g(X) = \lambda_{\max}(-X) \leq 0$, where $\lambda_{\max}(X)$ computes the largest eigenvalue of X and is a convex function. In this case, $\nabla g(X)$ only requires computing the minimum eigenvector of a matrix, which is cheaper than a full eigenspectrum computation required at each iteration of the standard SGD algorithm to restore feasibility.

Below, we state a few assumptions about $f(\mathbf{w})$ and $g(\mathbf{w})$ often made in stochastic optimization as:

Assumption 9.3. *We assume that:*

$$\begin{aligned}
&\mathbf{A1} \qquad \|\nabla f(\mathbf{w})\|_2 \le G_1, \quad \|\nabla g(\mathbf{w})\|_2 \le G_2, \quad |g(\mathbf{w})| \le C_2, \quad \forall \mathbf{w} \in \mathbb{B}, \\
&\mathbf{A2} \qquad \mathbb{E}_{\xi_t}[\exp(\|\mathbf{g}(\mathbf{w}, \xi_t) - \nabla f(\mathbf{w})\|_2^2/\sigma^2)] \le \exp(1), \quad \forall \mathbf{w} \in \mathbb{B}.
\end{aligned} \tag{9.8}$$

We also make the following mild assumption about the boundary of the convex domain $\mathcal{W}$ as:

Assumption 9.4. *We assume that:*

$$\mathbf{A3} \qquad \textit{there exists a constant } \rho > 0 \textit{ such that } \min_{g(\mathbf{w})=0} \|\nabla g(\mathbf{w})\|_2 \ge \rho. \tag{9.9}$$

Remark 9.5. *The purpose of introducing assumption **A3** is to ensure that the optimal dual variable for the constrained optimization problem in (9.1) is well bounded from the above, a key factor for our analysis. To see this, we write the problem in (9.1) into a convex-concave optimization problem:*

$$\min_{\mathbf{w} \in \mathbb{B}} \max_{\lambda \ge 0} f(\mathbf{w}) + \lambda g(\mathbf{w}).$$

Let $(\mathbf{w}_*, \lambda_*)$ be the optimal solution to the above convex-concave optimization problem. Since we assume $g(\mathbf{w})$ is strictly feasible, $\mathbf{w}_*$ is also an optimal solution to (9.1) due to the strong duality theorem [33]. Using the first order optimality condition, we have $\nabla f(\mathbf{w}_*) =$

$-\lambda_* \nabla g(\mathbf{w}_*)$. *Hence,* $\lambda_* = 0$ *when* $g(\mathbf{w}_*) < 0$*, and* $\lambda_* = \|\nabla f(\mathbf{w}_*)\|_2 / \|\nabla g(\mathbf{w}_*)\|_2$ *when* $g(\mathbf{w}_*) = 0$*. Under assumption* ***A3****, we have* $\lambda_* \in [0, G_1/\rho]$.

We note that, from a practical point of view, it is straightforward to verify that for many domains including PSD cone and Polytope, the gradient of the constraint function is lower bounded on the boundary and therefore assumption **A3** does not limit the applicability of the proposed algorithms for stochastic optimization. For the example of $g(X) = \lambda_{\max}(-X)$, the assumption **A3** implies $\min_{g(X)=0} \|\nabla g(X)\|_{\mathrm{F}} = \|\mathbf{u}\mathbf{u}^\top\|_{\mathrm{F}} = 1$, where $\mathbf{u}$ is an orthonomal vector representing the corresponding eigenvector of the matrix X whose minimum eigenvalue is zero.

We propose two different ways of incorporating the constraint function into the objective function, which result in two algorithms, one for general convex and the other for strongly convex functions.

9.3.1 General Convex Functions

To incorporate the constraint function $g(\mathbf{w})$, we introduce a regularized Lagrangian function,

$$L(\mathbf{w}, \lambda) = f(\mathbf{w}) + \lambda g(\mathbf{w}) - \frac{\gamma}{2}\lambda^2, \quad \lambda \geq 0.$$

The summation of the first two terms in $L(\mathbf{w}, \lambda)$ corresponds to the Lagrangian function in dual analysis and λ corresponds to a Lagrangian multiplier. A regularization term $-(\gamma/2)\lambda^2$ is introduced in $L(\mathbf{w}, \lambda)$ to prevent λ from being too large. Instead of solving the constrained optimization problem in (9.1), we try to solve the following convex-concave optimization problem

$$\min_{\mathbf{w}\in\mathbb{B}} \max_{\lambda\geq 0} L(\mathbf{w},\lambda). \tag{9.10}$$

The proposed algorithm for stochastically optimizing the problem in (9.10) is summarized in Algorithm 12. It differs from the existing stochastic gradient descent methods in that it updates both the primal variable $\mathbf{w}$ (steps 4 and 5) and the dual variable λ (step 6), which shares the same step sizes. We note that the parameter ρ is not employed in the implementation of Algorithm 12 and is only required for the theoretical analysis. It is noticeable that a similar primal-dual updating will be explored later in this chapter to avoid projection in online learning. We note that in online setting the algorithm and analysis only lead to a bound for the regret and the violation of the constraints in a long run, which does not necessarily guarantee the feasibility of final solution. Also our proof techniques differ from [115], where the convergence rate is obtained for the saddle point; however our goal is to attain bound on the convergence of the primal feasible solution.

Remark 9.6. *The convex-concave optimization problem in (9.10) is equivalent to the following minimization problem:*

$$\min_{\mathbf{w}\in\mathbb{B}} \; f(\mathbf{w}) + \frac{[g(\mathbf{w})]_+^2}{2\gamma}, \tag{9.11}$$

where $[z]_+$ outputs z if $z > 0$ and zero otherwise. It thus may seem tempting to directly optimize the penalized function $f(\mathbf{w}) + [g(\mathbf{w})]_+^2/(2\gamma)$ using the standard SGD method, which unfortunately does not yield a regret of $O(\sqrt{T})$. This is because, in order to obtain a regret of $O(\sqrt{T})$, we need to set $\gamma = \Omega(\sqrt{T})$, which unfortunately will lead to a blowup of the *gradients and consequently a poor regret bound. Using a primal-dual updating schema allows us to adjust the penalization term more carefully to obtain an $O(1/\sqrt{T})$ convergence rate.*

Theorem 9.7. *For any general convex function $f(\mathbf{w})$, if we set $\eta_t = \gamma/(2G_2^2), t = 1, \cdots, T$, and $\gamma = G_2^2/\sqrt{(G_1^2 + C_2^2 + (1 + \ln(2/\delta))\sigma^2)T}$ in Algorithm 12, under assumptions **A1-A3**, we have, with a probability at least $1 - \delta$,*

$$f(\widetilde{\mathbf{w}}_T) \leq \min_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}) + O\left(\frac{1}{\sqrt{T}}\right),$$

where $O(\cdot)$ suppresses polynomial factors that depend on $\ln(2/\delta), G_1, G_2, C_2, \rho$, and σ.

9.3.2 Strongly Convex Functions with Smoothing

We first emphasize that it is difficult to extend Algorithm 12 to achieve an $O(\ln T/T)$ convergence rate for strongly convex optimization. This is because although the function $-L(\mathbf{w}, \lambda)$ is strongly convex in λ, its modulus for strong convexity is γ, which is too small to obtain an $O(\ln T)$ regret bound.

To achieve a faster convergence rate for strongly convex optimization, we change assumptions **A1** and **A2** as follows.

Assumption 9.8.

$$\textbf{A4} \quad \|\mathbf{g}(\mathbf{w}, \xi_t)\|_2 \leq G_1, \quad \|\nabla g(\mathbf{w})\|_2 \leq G_2, \quad \forall \mathbf{w} \in \mathbb{B},$$

where we slightly abuse the same notation G_1.

Note that **A1** only requires that $\|\nabla f(\mathbf{w})\|_2$ is bounded and **A2** assumes a mild condition on the stochastic gradient. In contrast, for strongly convex optimization we need to assume a bound on the stochastic gradient $\|\mathbf{g}(\mathbf{w}, \xi_t)\|_2$. Although assumption **A4** is stronger than assumptions **A1** and **A2**, however, it is always possible to bound the stochastic gradient for machine learning problems where $f(\mathbf{w})$ usually consists of a summation of loss functions on training examples, and the stochastic gradient is computed by sampling over the training examples. Given the bound on $\|\mathbf{g}(\mathbf{w}, \xi_t)\|_2$, we can easily have $\|\nabla f(\mathbf{w})\|_2 = \|\mathbb{E}\mathbf{g}(\mathbf{w}, \xi_t)\|_2 \leq \mathbb{E}\|\mathbf{g}(\mathbf{w}, \xi_t)\|_2 \leq G_1$, which is used to set an input parameter $\lambda_0 > G_1/\rho$ to the algorithm. According to the discussion in the last subsection, we know that the optimal dual variable λ_* is upper bounded by G_1/ρ, and consequently is upper bounded by λ_0.

Similar to the last approach, we write the optimization problem (9.1) into an equivalent convex-concave optimization problem:

$$\min_{g(\mathbf{w}) \leq 0} f(\mathbf{w}) = \min_{\mathbf{w} \in \mathbb{B}} \max_{0 \leq \lambda \leq \lambda_0} f(\mathbf{w}) + \lambda g(\mathbf{w}) = \min_{\mathbf{w} \in \mathbb{B}} f(\mathbf{w}) + \lambda_0 [g(\mathbf{w})]_+.$$

To avoid unnecessary complication due to the subgradient of $[\cdot]_+$, following [123], we introduce a smoothing term $H(\lambda/\lambda_0)$, where $H(p) = -p \ln p - (1-p)\ln(1-p)$ is the entropy function, into the Lagrangian function, leading to the optimization problem $\min_{\mathbf{w} \in \mathbb{B}} F(\mathbf{w})$, where $F(\mathbf{w})$ is defined as

$$F(\mathbf{w}) = f(\mathbf{w}) + \max_{0 \leq \lambda \leq \lambda_0} \lambda g(\mathbf{w}) + \gamma H(\lambda/\lambda_0) = f(\mathbf{w}) + \gamma \ln\left(1 + \exp\left(\frac{\lambda_0 g(\mathbf{w})}{\gamma}\right)\right),$$

where $\gamma > 0$ is a parameter whose value will be determined later. Given the smoothed objective function $F(\mathbf{w})$, we find the optimal solution by applying SGD to minimize $F(\mathbf{w})$,

Algorithm 13 SGD with ONE Projection by a Smoothing Technique (SGD-ST)

```
Input: a sequence of step sizes $\{\eta_t\}$, $\lambda_0$, and $\gamma$
Initialize: $\mathbf{w}_1 = \mathbf{0}$.
for $t = 1, \ldots, T$ do
    Compute
        $\mathbf{w}'_{t+1} = \mathbf{w}_t - \eta_t \left( \mathbf{g}(\mathbf{w}_t, \xi_t) + \frac{\exp(\lambda_0 g(\mathbf{w}_t)/\gamma)}{1 + \exp(\lambda_0 g(\mathbf{w}_t)/\gamma)} \lambda_0 \nabla g(\mathbf{w}_t) \right)$
    Update $\mathbf{w}_{t+1} = \mathbf{w}'_{t+1} / \max(\|\mathbf{w}'_{t+1}\|_2, 1)$
end for
Return: $\widetilde{\mathbf{w}}_T = \Pi_{\mathcal{W}}(\widehat{\mathbf{w}}_T)$, where $\widehat{\mathbf{w}}_T = \sum_{t=1}^{T} \mathbf{w}_t / T$.
```

where the gradient of $F(\mathbf{w})$ is computed by

$$\nabla F(\mathbf{w}) = \nabla f(\mathbf{w}) + \frac{\exp(\lambda_0 g(\mathbf{w})/\gamma)}{1 + \exp(\lambda_0 g(\mathbf{w})/\gamma)} \lambda_0 \nabla g(\mathbf{w}). \tag{9.12}$$

Algorithm 13 gives the detailed steps. Unlike Algorithm 12, only the primal variable $\mathbf{w}$ is updated in each iteration using the stochastic gradient computed in (9.12). The following theorem shows that Algorithm 13 achieves an $O(\ln T/T)$ convergence rate if the cost functions are strongly convex.

Theorem 9.9. *For any β-strongly convex function $f(\mathbf{w})$, if we set $\eta_t = 1/(2\beta t), t = 1, \ldots, T$, $\gamma = \ln T/T$, and $\lambda_0 > G_1/\rho$ in Algorithm 13, under assumptions **A3** and **A4**, we have with a probability at least $1 - \delta$,*

$$f(\widetilde{\mathbf{w}}_T) \leq \min_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w}) + O\left(\frac{\ln T}{T} \right),$$

where $O(\cdot)$ suppresses polynomial factors that depend on $\ln(1/\delta)$, $1/\beta, G_1, G_2, \rho$, and λ_0.

It is well known that the optimal convergence rate of SGD for strongly convex optimization is $O(1/T)$ [72] which has been proven to be tight in stochastic optimization setting [4].

According to Theorem 9.9, Algorithm 13 achieves an almost optimal convergence rate except for the factor of $\ln T$. It is worth mentioning that although it is not explicitly given in Theorem 9.9, the detailed expression for the convergence rate of Algorithm 13 exhibits a tradeoff in setting λ_0 (more can be found in the proof of Theorem 9.9). Finally, under assumptions **A1**-**A3**, Algorithm 13 can achieve an $O(1/\sqrt{T})$ convergence rate for general convex functions, similar to Algorithm 12.

9.4 Analysis of Convergence Rate

We here present the proofs of main theorems. The omitted proofs of some results are deferred to Section 9.6.

9.4.1 Convergence Rate for General Convex Functions

To pave the path for the proof of of Theorem 9.7, we present a series of lemmas. The lemma below states two key inequalities for the proof, which follows the standard analysis of gradient descent.

Lemma 9.10. *Under the bounded assumptions in (9.8) and (9.8), for any* $\mathbf{w} \in \mathbb{B}$ *and* $\lambda > 0$*, we have*

$$\begin{aligned}\langle \mathbf{w}_t - \mathbf{w}, \nabla_{\mathbf{w}} L(\mathbf{w}_t, \lambda_t)\rangle \leq & \frac{1}{2\eta_t}\left(\|\mathbf{w}-\mathbf{w}_t\|_2^2 - \|\mathbf{w}-\mathbf{w}_{t+1}\|_2^2\right) + 2\eta_t G_1^2 + \eta_t G_2^2 \lambda_t^2 \\ & + 2\eta_t \underbrace{\|\mathbf{g}(\mathbf{w}_t, \xi_t) - \nabla f(\mathbf{w}_t)\|_2^2}_{\equiv \Delta_t} + \underbrace{\langle \mathbf{w}-\mathbf{w}_t, \mathbf{g}(\mathbf{w}_t, \xi_t) - \nabla f(\mathbf{w}_t)\rangle}_{\equiv \zeta_t(\mathbf{w})},\end{aligned}$$

$$(\lambda - \lambda_t)\nabla_\lambda L(\mathbf{w}_t, \lambda_t) \leq \frac{1}{2\eta_t}\left(|\lambda - \lambda_t|^2 - |\lambda - \lambda_{t+1}|^2\right) + 2\eta_t C_2^2.$$

An immediate result of Lemma 9.10 is the following which states a regret-type bound.

Lemma 9.11. *For any general convex function $f(\mathbf{w})$, if we set $\eta_t = \gamma/(2G_2^2), t = 1, \cdots, T$, we have*

$$\sum_{t=1}^{T}(f(\mathbf{w}_t) - f(\mathbf{w}_*)) + \frac{[\sum_{t=1}^{T} g(\mathbf{w}_t)]_+^2}{2(\gamma T + 2G_2^2/\gamma)} \leq \frac{G_2^2}{\gamma} + \frac{(G_1^2 + C_2^2)}{G_2^2}\gamma T + \frac{\gamma}{G_2^2}\sum_{t=1}^{T}\Delta_t + \sum_{t=1}^{T}\zeta_t(\mathbf{w}_*), \tag{9.13}$$

where $\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w})$.

Proof. (Proof of Theorem 9.7) First, by martingale inequality (e.g., Lemma 4 in [94]), with a probability $1 - \delta/2$, we have $\sum_{t=1}^{T}\zeta_t(\mathbf{w}_*) \leq 2\sigma\sqrt{3\ln(2/\delta)}\sqrt{T}$. By Markov's inequality (Lemma A.18 in Appendix A.2), with a probability $1 - \delta/2$, we have $\sum_{t=1}^{T}\Delta_t \leq (1 + \ln(2/\delta))\sigma^2 T$. Substituting these inequalities into Lemma 9.11, plugging the stated value of γ, and using $O(\cdot)$ notation for ease of exposition, we have with a probability $1 - \delta$

$$\sum_{t=1}^{T}(f(\mathbf{w}_t) - f(\mathbf{w}_*)) + \frac{1}{C\sqrt{T}}\Big[\sum_{t=1}^{T} g(\mathbf{w}_t)\Big]_+^2 \leq O(\sqrt{T}),$$

where $C = 2G_2(1/\sqrt{G_1^2 + C_2^2 + (1 + \ln(2/\delta))\sigma^2} + 2\sqrt{G_1^2 + C_2^2 + (1 + \ln(2/\delta))\sigma^2})$ and $O(\cdot)$ suppresses polynomial factors that depend on $\ln(2/\delta), G_1, G_2, C_2, \sigma$.

Recalling the definition of $\widehat{\mathbf{w}}_T = \sum_{t=1}^{T}\mathbf{w}_t/T$ and using the convexity of $f(\mathbf{w})$ and $g(\mathbf{w})$, we have

$$f(\widehat{\mathbf{w}}_T) - f(\mathbf{w}_*) + \frac{\sqrt{T}}{C}[g(\widehat{\mathbf{w}}_T)]_+^2 \leq O\left(\frac{1}{\sqrt{T}}\right). \tag{9.14}$$

Assume $g(\widehat{\mathbf{w}}_T) > 0$, otherwise $\widetilde{\mathbf{w}}_T = \widehat{\mathbf{w}}_T$ and we easily have $f(\widetilde{\mathbf{w}}_T) \leq \min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w}) + O(1/\sqrt{T})$. Since $\widetilde{\mathbf{w}}_T$ is the projection of $\widehat{\mathbf{w}}_T$ into $\mathcal{W}$, i.e., $\widetilde{\mathbf{w}}_T = \arg\min_{g(\mathbf{w})\leq 0} \|\mathbf{w} - \widehat{\mathbf{w}}_T\|_2^2$, then by first order optimality condition, there exists a positive constant $s > 0$ such that

$$g(\widetilde{\mathbf{w}}_T) = 0, \text{ and } \widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T = s\nabla g(\widehat{\mathbf{w}}_T)$$

which indicates that $\widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T$ is in the same direction to $\nabla g(\tilde{\mathbf{w}}_T)$. Hence,

$$\begin{aligned} g(\widehat{\mathbf{w}}_T) = g(\widehat{\mathbf{w}}_T) - g(\widetilde{\mathbf{w}}_T) &\geq (\widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T)^\top \nabla g(\widetilde{\mathbf{w}}_T) \\ &= \|\widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T\|_2 \|\nabla g(\widetilde{\mathbf{w}}_T)\|_2 \\ &\geq \rho\|\widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T\|_2, \end{aligned} \tag{9.15}$$

where the last inequality follows the definition of $\min_{g(\mathbf{w})=0} \|\nabla g(\mathbf{w})\|_2 \geq \rho$. Additionally, we have

$$f(\mathbf{w}_*) - f(\widehat{\mathbf{w}}_T) \leq f(\mathbf{w}_*) - f(\widetilde{\mathbf{w}}_T) + f(\widetilde{\mathbf{w}}_T) - f(\widehat{\mathbf{w}}_T) \leq G_1\|\widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T\|_2, \tag{9.16}$$

due to $f(\mathbf{w}_*) \leq f(\widetilde{\mathbf{w}}_T)$ and Lipschitz continuity of $f(\mathbf{w})$. Combining inequalities (9.14), (9.15), and (9.16) yields

$$\frac{\rho^2}{C}\sqrt{T}\|\widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T\|_2^2 \leq O(1/\sqrt{T}) + G_1\|\widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T\|_2.$$

By simple algebra, we have $\|\widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T\|_2 \leq \frac{G_1 C}{\rho^2\sqrt{T}} + O\left(\sqrt{\frac{C}{\rho^2 T}}\right)$ Therefore

$$\begin{aligned} f(\widetilde{\mathbf{w}}_T) &\leq f(\widetilde{\mathbf{w}}_T) - f(\widehat{\mathbf{w}}_T) + f(\widehat{\mathbf{w}}_T) \\ &\leq G_1\|\widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T\|_2 + f(\mathbf{w}_*) + O\left(\frac{1}{\sqrt{T}}\right) \\ &\leq f(\mathbf{w}_*) + O\left(\frac{1}{\sqrt{T}}\right), \end{aligned} \tag{9.17}$$

where we use the inequality in (9.14) to bound $f(\widehat{\mathbf{w}}_T)$ by $f(\mathbf{w}_*)$ and absorb the dependence on ρ, G_1, C into the $O(\cdot)$ notation. □

Remark 9.12. *From the proof of Theorem 9.7, we can see that the key inequalities are (9.14), (9.15), and (9.16). In particular, the regret-type bound in (9.14) depends on the algorithm. If we only update the primal variable using the penalized objective in (9.11), whose gradient depends on $1/\gamma$, it will cause a blowup in the regret bound with $(1/\gamma + \gamma T + T/\gamma)$, which leads to a non-convergent bound.*

9.4.2 Convergence Rate for Strongly Convex Functions

Our proof of Theorem 9.9 for the convergence rate of Algorithm 13 when applied to strongly convex functions starts with the following lemma by analogy of Lemma 9.11.

Lemma 9.13. *For any β-strongly convex function $f(\mathbf{w})$, if we set $\eta_t = 1/(2\beta t)$, we have*

$$\sum_{t=1}^{T}(F(\mathbf{w}) - F(\mathbf{w}_*)) \leq \frac{(G_1^2 + \lambda_0^2 G_2^2)(1 + \ln T)}{2\beta} + \sum_{t=1}^{T}\zeta_t(\mathbf{w}_*) - \frac{\beta}{4}\sum_{t=1}^{T}\|\mathbf{w}_* - \mathbf{w}_t\|_2^2$$

where $\mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w})$.

In order to prove Theorem 9.9, we need the following result for an improved martingale inequality.

Lemma 9.14. *For any fixed* $\mathbf{w} \in \mathbb{B}$*, define* $D_T = \sum_{t=1}^T \|\mathbf{w}_t - \mathbf{w}\|_2^2$*,* $\Lambda_T = \sum_{t=1}^T \zeta_t(\mathbf{w})$*, and* $m = \lceil \log_2 T \rceil$*. We have*

$$\Pr\left(D_T \leq \frac{4}{T}\right) + \Pr\left(\Lambda_T \leq 4G_1\sqrt{D_T \ln \frac{m}{\delta}} + 4G_1 \ln \frac{m}{\delta}\right) \geq 1 - \delta.$$

Proof of Theorem 9.9. We substitute the bound in Lemma 9.14 into the inequality in Lemma 9.13 with $\mathbf{w} = \mathbf{w}_*$. We consider two cases. In the first case, we assume $D_T \leq 4/T$. As a result, we have

$$\sum_{t=1}^T \zeta_t(\mathbf{w}_*) = \sum_{t=1}^T (\nabla f(\mathbf{w}_t) - \mathbf{g}(\mathbf{w}_t, \xi_t))^\top (\mathbf{w}_* - \mathbf{w}_t) \leq 2G_1\sqrt{TD_T} \leq 4G_1,$$

which together with the inequality in Lemma 9.13 leads to the bound

$$\sum_{t=1}^T (F(\mathbf{w}_t) - F(\mathbf{w}_*)) \leq 4G_1 + \frac{(G_1^2 + \lambda_0^2 G_2^2)(1 + \ln T)}{2\beta}.$$

In the second case, we assume

$$\sum_{t=1}^T \zeta_t(\mathbf{w}_*) \leq 4G_1\sqrt{D_T \ln \frac{m}{\delta}} + 4G_1 \ln \frac{m}{\delta} \leq \frac{\beta}{4} D_T + \left(\frac{16G_1^2}{\beta} + 4G_1\right) \ln \frac{m}{\delta},$$

where the last step uses the fact $2\sqrt{ab} \leq a^2 + b^2$. We thus have

$$\sum_{t=1}^T (F(\mathbf{w}_t) - F(\mathbf{w}_*)) \leq \left(\frac{16G_1^2}{\beta} + 4G_1\right) \ln \frac{m}{\delta} + \frac{(G_1^2 + \lambda_0^2 G_2^2)(1 + \ln T)}{2\beta}.$$

Combing the results of the two cases, we have, with a probability $1 - \delta$,

$$\sum_{t=1}^{T}(F(\mathbf{w}_t) - F(\mathbf{w}_*)) \leq \underbrace{\left(\frac{16G_1^2}{\beta} + 4G_1\right)\ln\frac{m}{\delta} + 4G_1 + \frac{(G_1^2 + \lambda_0^2 G_2^2)(1 + \ln T)}{2\beta}}_{O(\ln T)}. \qquad (9.18)$$

By convexity of $F(\mathbf{w})$, we have $F(\widehat{\mathbf{w}}_T) \leq F(\mathbf{w}_*) + O(\ln T/T)$. Noting that $\mathbf{w}_* \in \mathcal{W}$, $g(\mathbf{w}_*) \leq 0$, we have $F(\mathbf{w}_*) \leq f(\mathbf{w}_*) + \gamma \ln 2$. On the other hand,

$$F(\widehat{\mathbf{w}}_T) = f(\widehat{\mathbf{w}}_T) + \gamma \ln\left(1 + \exp\left(\frac{\lambda_0 g(\widehat{\mathbf{w}}_T)}{\gamma}\right)\right) \geq f(\widehat{\mathbf{w}}_T) + \max(0, \lambda_0 g(\widehat{\mathbf{w}}_T)).$$

Therefore, with the value of $\gamma = \ln T/T$, we have

$$\begin{aligned} f(\widehat{\mathbf{w}}_T) &\leq f(\mathbf{w}_*) + O\left(\frac{\ln T}{T}\right), \\ f(\widehat{\mathbf{w}}_T) + \lambda_0 g(\widehat{\mathbf{w}}_T) &\leq f(\mathbf{w}_*) + O\left(\frac{\ln T}{T}\right). \end{aligned} \qquad (9.19)$$

Applying the inequalities (9.15) and (9.16) to (9.19), and noting that $\gamma = \ln T/T$, we have

$$\lambda_0 \rho \|\widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T\|_2 \leq G_1 \|\widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T\|_2 + O\left(\frac{\ln T}{T}\right).$$

For $\lambda_0 > G_1/\rho$, we have $\|\widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T\|_2 \leq (1/(\lambda_0\rho - G_1))O(\ln T/T)$. Therefore

$$f(\widetilde{\mathbf{w}}_T) \leq f(\widetilde{\mathbf{w}}_T) - f(\widehat{\mathbf{w}}_T) + f(\widehat{\mathbf{w}}_T) \leq G_1\|\widehat{\mathbf{w}}_T - \widetilde{\mathbf{w}}_T\|_2 + f(\mathbf{w}_*) + O\left(\frac{\ln T}{T}\right) \leq f(\mathbf{w}_*) + O\left(\frac{\ln T}{T}\right),$$

where in the second inequality we use inequality (9.19). □

9.5 Online Optimization with Soft Constraints

We now turn to making online learning algorithms more efficient by excluding the projection steps. To this end, we consider an alternative online learning problem in which, instead of requiring that each solution obeys the constraints, we only require the constraints, which define the convex domain $\mathcal{W}$, to be satisfied in a long run. Then, the online learning problem becomes a task to find a sequence of solutions under the long term constraints. We present and analyze an online gradient descent based algorithm for online convex optimization with soft constraints.

To facilitate our analysis, similar to the stochastic setting described in Section 9.1, we assume that the domain $\mathcal{W}$ can be written as an intersection of a finite number of convex constraints, that is,

$$\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^d : g_i(\mathbf{w}) \leq 0, i \in [m]\},$$

where $g_i(\mathbf{w}), i \in [m]$, are Lipschitz continuous functions. We assume that $\mathcal{W}$ is a bounded domain, that is, there exist constants $R > 0$ and $r < 1$ such that $\mathcal{W} \subseteq R\mathbb{B}$ and $r\mathbb{B} \subseteq \mathcal{W}$ where $\mathbb{B}$ denotes the unit ℓ_2 ball centered at the origin.

We focus on the problem of online convex optimization as introduced in Chapter 2, in which the goal is to achieve a low regret with respect to a fixed decision on a sequence of adversarially chosen cost functions. The difference between the setting considered here and the general online convex optimization is that, in our setting, instead of requiring $\mathbf{w}_t \in \mathcal{W}$, or equivalently $g_i(\mathbf{w}_t) \leq 0, i \in [m]$, for all $t \in [T]$, we only require the constraints to be satisfied in the long run, namely $\sum_{t=1}^{T} g_i(\mathbf{w}_t) \leq 0, i \in [m]$. Then, the problem becomes to find a sequence of solutions $\mathbf{w}_t, t \in [T]$ that minimizes the regret, under the long term constraints $\sum_{t=1}^{T} g_i(\mathbf{w}_t) \leq 0, i \in [m]$. Formally, we would like to solve the following optimization problem online,

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_T \in \mathbb{B}} \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{\mathbf{w} \in \mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w}) \;\; \text{s.t.} \;\; \sum_{t=1}^{T} g_i(\mathbf{w}_t) \leq 0 \;,\; i \in [m]. \tag{9.20}$$

For simplicity, we will focus on a finite-horizon setting where the number of rounds T is known in advance. This condition can be relaxed under certain conditions, using standard techniques (see, e.g., [38]). Note that in (9.20), (i) the solutions come from the ball $\mathbb{B} \supseteq \mathcal{W}$ instead of $\mathcal{W}$ and (ii) the constraint functions are fixed and are given in advance.

9.5.1 An Impossibility Theorem

Before we state our formulation and algorithms, let us review a few alternative techniques that do not need explicit projection. A straightforward approach is to introduce an appropriate self-concordant barrier function for the given convex set $\mathcal{W}$ and add it to the objective function such that the barrier diverges at the boundary of the set. Then we can interpret the resulting optimization problem, on the modified objective functions, as an unconstrained minimization problem that can be solved without projection steps. Following the analysis in [3], with an appropriately designed procedure for updating solutions, we could guarantee a regret bound of $O(\sqrt{T})$ without the violation of constraints. A similar idea is used in [2] for online bandit learning and in [114] for a random walk approach for regret minimization which, in fact, translates the issue of projection into the difficulty of sampling. Even for linear Lipschitz cost functions, the random walk approach requires sampling from a Gaussian distribution with covariance given by the Hessian of the self-concordant barrier of the convex set $\mathcal{W}$ that has the same time complexity as inverting a matrix. The main limitation with these approaches is that they require computing the Hessian matrix of the objective function in order to guarantee that the updated solution stays within the given domain $\mathcal{W}$. This limitation makes it computationally unattractive when dealing with high dimensional data. In addition, except for well known cases, it is often unclear how to efficiently construct a self-concordant barrier function for a general convex domain.

An alternative approach for online convex optimization with long term constraints is to introduce a penalty term in the loss function that penalizes the violation of constraints. More specifically, we can define a new loss function $\hat{f}_t(\mathbf{w})$ as

$$\hat{f}_t(\mathbf{w}) = f_t(\mathbf{w}) + \delta \sum_{i=1}^{m} [g_i(\mathbf{w})]_+, \tag{9.21}$$

where $[z]_+ = \max(0, 1-z)$ and $\delta > 0$ is a fixed positive constant used to penalize the violation of constraints. We then run the standard OGD algorithm from Chapter 2 to minimize the modified loss function $\hat{f}_t(\mathbf{w})$. The following theorem shows that this simple strategy fails to achieve sub-linear bound for both regret and the long term violation of constraints at the same time.

Theorem 9.15. *Given $\delta > 0$, there always exists a sequence of loss functions $f_1, f_2, \cdots, f_T$ and a constraint function $g(\mathbf{w})$ such that either $\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \min_{g(\mathbf{w}) \leq 0} \sum_{t=1}^{T} f_t(\mathbf{w}) = O(T)$ or $\sum_{t=1}^{T} [g(\mathbf{w}_t)]_+ = O(T)$ holds, where $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_T$ is the sequence of solutions generated by the OGD algorithm that minimizes the modified loss functions given in (9.21).*

Proof. We first show that when $\delta < 1$, there exists a loss function and a constraint function such that the violation of constraint is linear in T. To see this, we set $f_t(\mathbf{w}) = \mathbf{w}^\top \mathbf{w}, t \in [T]$ and $g(\mathbf{w}) = 1 - \mathbf{w}^\top \mathbf{w}$. Assume we start with an infeasible solution, that is, $g(\mathbf{w}_1) > 0$ or $\mathbf{w}_1^\top \mathbf{w} < 1$. Given the solution $\mathbf{w}_t$ obtained at tth trial, using the standard gradient descent approach, we have $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta(1 - \delta)\mathbf{w}$. Hence, if $\mathbf{w}_t^\top \mathbf{w} < 1$, since we have

$\mathbf{w}_{t+1}^\top\mathbf{w} < \mathbf{w}_t^\top\mathbf{w} < 1$, if we start with an infeasible solution, all the solutions obtained over the trails will violate the constraint $g(\mathbf{w}) \leq 0$, leading to a linear number of violation of constraints. Based on this analysis, we assume $\delta > 1$ in the analysis below.

Given a strongly convex loss function $f(\mathbf{w})$ with modulus γ, we consider a constrained optimization problem given by

$$\min_{g(\mathbf{w})\leq 0} f(\mathbf{w}),$$

which is equivalent to the following unconstrained optimization problem

$$\min_{\mathbf{w}} f(\mathbf{w}) + \lambda[g(\mathbf{w})]_+,$$

where $\lambda \geq 0$ is the Lagrangian multiplier. Since we can always scale $f(\mathbf{w})$ to make $\lambda \leq 1/2$, it is safe to assume $\lambda \leq 1/2 < \delta$. Let $\mathbf{w}_*$ and $\mathbf{w}_a$ be the optimal solutions to the constrained optimization problems $\arg\min_{g(\mathbf{w})\leq 0} f(\mathbf{w})$ and $\arg\min\limits_{\mathbf{w}} f(\mathbf{w}) + \delta[g(\mathbf{w})]_+$, respectively. We choose $f(\mathbf{w})$ such that $\|\nabla f(\mathbf{w}_*)\| > 0$, which leads to $\mathbf{w}_a \neq \mathbf{w}_*$. This holds because according to the first order optimality condition, we have

$$\nabla f(\mathbf{w}_*) = -\lambda \nabla g(\mathbf{w}_*),\ \nabla f(\mathbf{w}_a) = -\delta \nabla g(\mathbf{w}_*),$$

and therefore $\nabla f(\mathbf{w}_*) \neq \nabla f(\mathbf{w}_a)$ when $\lambda < \delta$. Define $\Delta = f(\mathbf{w}_a) - f(\mathbf{w}_*)$. Since $\Delta \geq \gamma\|\mathbf{w}_a - \mathbf{w}_*\|^2/2$ due to the strong convexity of $f(\mathbf{w})$, we have $\Delta > 0$.

Let $\{\mathbf{w}_t\}_{t=1}^T$ be the sequence of solutions generated by the OGD algorithm that minimizes the modified loss function $f(\mathbf{w}) + \delta[g(\mathbf{w})]_+$. We have

$$\begin{aligned}
& \sum_{t=1}^{T} f(\mathbf{w}_t) + \delta[g(\mathbf{w}_t)]_+ \geq T \min_{\mathbf{w}} f(\mathbf{w}) + \delta[g(\mathbf{w})]_+ \\
= \quad & T(f(\mathbf{w}_a) + \delta[g(\mathbf{w}_a)]_+) \geq T(f(\mathbf{w}_a) + \lambda[g(\mathbf{w}_a)]_+) \\
= \quad & T(f(\mathbf{w}_*) + \lambda[g(\mathbf{w}_*)]_+) + T(f(\mathbf{w}_a) + \lambda[g(\mathbf{w}_a)]_+ - f(\mathbf{w}_*) - \lambda[g(\mathbf{w}_*)]) \\
\geq \quad & T \min_{g(\mathbf{w}) \leq 0} f(\mathbf{w}) + T\Delta.
\end{aligned}$$

As a result, we have

$$\sum_{t=1}^{T} f(\mathbf{w}_t) + \delta[g(\mathbf{w}_t)]_+ - \min_{g(\mathbf{w}) \leq 0} f(\mathbf{w}) = O(T),$$

implying that either the regret $\sum_{t=1}^{T} f(\mathbf{w}_t) - Tf(\mathbf{w}_*)$ or the violation of the constraints $\sum_{t=1}^{T} [g(\mathbf{w})]_+$ is linear in T. □

To better understand the performance of penalty based approach, here we analyze the performance of the OGD in solving the online optimization problem in (9.20). The algorithm is analyzed using the following lemma from [158] (see also Theorem 2.8 in Chapter 2).

Lemma 9.16. *Let $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_T \in \mathcal{W}$ be the sequence of solutions obtained by applying OGD on the sequence of bounded convex functions $f_1, f_2, \ldots, f_T$. Then, for any solution $\mathbf{w}_* \in \mathcal{W}$ we have*

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}_*) \leq \frac{R^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\nabla f_t(\mathbf{w}_t)\|^2.$$

We apply OGD to functions $\hat{f}_t(\mathbf{w}),\ t \in [T]$ defined in (9.21), that is, instead of updating the solution based on the gradient of $f_t(\mathbf{w})$, we update the solution by the gradient of $\hat{f}_t(\mathbf{w})$.

Using Lemma 9.16, by expanding the functions $\hat{f}_t(\mathbf{w})$ based on (9.21) and considering the fact that $\sum_{i=1}^{m}[g_i(\mathbf{w}_*)]_+^2 = 0$, we get

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}_*) + \frac{\delta}{2}\sum_{t=1}^{T}\sum_{i=1}^{m}[g_i(\mathbf{w})]_+^2 \leq \frac{R^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\nabla\hat{f}_t(\mathbf{w}_t)\|^2. \tag{9.22}$$

From the definition of $\hat{f}_t(\mathbf{w})$, the norm of the gradient $\nabla\hat{f}_t(\mathbf{w}_t)$ is bounded as follows

$$\|\nabla\hat{f}_t(\mathbf{w})\|^2 = \|\nabla f_t(\mathbf{w}) + \delta\sum_{i=1}^{m}[g_i(\mathbf{w})]_+\nabla g_i(\mathbf{w})\|^2 \leq 2G^2(1 + m\delta^2 D^2), \tag{9.23}$$

where the inequality holds because $(a_1 + a_2)^2 \leq 2(a_1^2 + a_2^2)$. By substituting (9.23) into the (9.22) we have:

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}_*) + \frac{\delta}{2}\sum_{t=1}^{T}\sum_{i=1}^{m}[g_i(\mathbf{w}_t)]_+^2 \leq \frac{R^2}{2\eta} + \eta G^2(1 + m\delta^2 D^2)T. \tag{9.24}$$

Since $[\cdot]_+^2$ is a convex function, from Jensen's inequality and following the fact that $\sum_{t=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{w}_*) \geq -FT$, we have:

$$\frac{\delta}{2T}\sum_{i=1}^{m}\left[\sum_{t=1}^{T} g_i(\mathbf{w}_t)\right]_+^2 \leq \frac{\delta}{2}\sum_{i=1}^{m}\sum_{t=1}^{T}[g_i(\mathbf{w}_t)]_+^2 \leq \frac{R^2}{2\eta} + \eta G^2(1 + m\delta^2 D^2)T + FT.$$

By minimizing the right hand side of (9.24) with respect to η, we get the regret bound as

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}_*) \leq RG\sqrt{2(1 + m\delta^2 D^2)T} = O(\delta\sqrt{T}) \tag{9.25}$$

and the bound for the violation of constraints as

$$\sum_{t=1}^{T} g_i(\mathbf{w}_t) \leq \sqrt{\left(\frac{R^2}{2\eta} + \eta G^2(1+m\delta^2 D^2)T + FT\right)\frac{2T}{\delta}} = O(T^{1/4}\delta^{1/2} + T\delta^{-1/2}). \quad (9.26)$$

By examining the bounds obtained in (9.25) and (9.26), it turns out that in order to recover $O(\sqrt{T})$ regret bound, we need to set δ to be a constant, leading to $O(T)$ bound for the violation of constraints in the long run, which is not satisfactory at all. The analysis shows that in order to obtain $O(\sqrt{T})$ regret bound, linear bound on the long term violation of the constraints is unavoidable. The main reason for the failure of using modified loss function in (9.21) is that the weight constant δ is fixed and independent from the sequence of solutions obtained so far. In the next subsection, we present an online convex-concave formulation for online convex optimization with long term constraints, which explicitly addresses the limitation of (9.21) by automatically adjusting the weight constant based on the violation of the solutions obtained so far.

As mentioned before, our general strategy is to turn online convex optimization with long term constraints into a convex-concave optimization problem. Instead of generating a sequence of solutions that satisfies the long term constraints, we first consider an online optimization strategy that allows the violation of constraints on some rounds in a controlled way. We then modify the online optimization strategy to obtain a sequence of solutions that obeys the long term constraints. Although the online convex optimization with long term constraints is clearly easier than the standard online convex optimization problem, it is straightforward to see that optimal regret bound for online optimization with long term constraints should be on the order of $O(\sqrt{T})$, no better than the standard online convex optimization problem.

9.5.2 An Efficient Algorithm with $O(\sqrt{T})$ Regret Bound and $O(T^{3/4})$ Bound on the Violation of Constraints

The intuition behind our approach stems from the observation that the constrained optimization problem $\min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})$ is equivalent to the following convex-concave optimization problem

$$\min_{\mathbf{w}\in\mathbb{B}} \max_{\lambda\in\mathbb{R}_+^m} \sum_{t=1}^{T} f_t(\mathbf{w}) + \sum_{i=1}^{m} \lambda_i g_i(\mathbf{w}), \tag{9.27}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)^\top$ is the vector of Lagrangian multipliers associated with the constraints $g_i(\cdot), i = 1, \ldots, m$ and belongs to the nonnegative orthant $\mathbb{R}_+^m$. To solve the online convex-concave optimization problem, we extend the gradient based approach for variational inequality [116] to (9.27). To this end, we consider the following regularized convex-concave function as

$$\mathcal{L}_t(\mathbf{w}, \boldsymbol{\lambda}) = f_t(\mathbf{w}) + \sum_{i=1}^{m} \left\{ \lambda_i g_i(\mathbf{w}) - \frac{\delta\eta}{2}\lambda_i^2 \right\}, \tag{9.28}$$

where $\delta > 0$ is a constant whose value will be decided by the analysis. Note that in (9.28), we introduce a regularizer $\delta\eta\lambda_i^2/2$ to prevent λ_i from being too large. This is because, when λ_i is large, we may encounter a large gradient for $\mathbf{w}$ because of $\nabla_{\mathbf{w}}\mathcal{L}_t(\mathbf{w}, \boldsymbol{\lambda}) \propto \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{w})$, leading to unstable solutions and a poor regret bound. Although we can achieve the same goal by restricting λ_i to a bounded domain, using the quadratic regularizer makes it convenient for our analysis.

Algorithm 14 shows the detailed steps of the proposed algorithm. Unlike standard online convex optimization algorithms that only update $\mathbf{w}$, Algorithm 14 updates both $\mathbf{w}$ and $\boldsymbol{\lambda}$. In addition, unlike the modified loss function in (9.21) where the weights for constraints $\{g_i(\mathbf{w}) \leq 0\}_{i=1}^m$ are fixed, Algorithm 14 automatically adjusts the weights $\{\lambda_i\}_{i=1}^m$ based on $\{g_i(\mathbf{w})\}_{i=1}^m$, the violation of constraints, as the game proceeds. It is this property that allows Algorithm 14 to achieve sub-linear bound for both regret and the violation of constraints.

To analyze Algorithm 14, we first state the following lemma, the key to the main theorem on the regret bound and the violation of constraints.

Lemma 9.17. *Let $\mathcal{L}_t(\cdot,\cdot)$ be the function defined in (9.28) which is convex in its first argument and concave in its second argument. Then for any $(\mathbf{w},\boldsymbol{\lambda}) \in \mathbb{B} \times \mathbb{R}_+^m$ we have*

$$\begin{aligned}\mathcal{L}_t(\mathbf{w}_t,\boldsymbol{\lambda}) - \mathcal{L}_t(\mathbf{w},\boldsymbol{\lambda}_t) \leq \frac{1}{2\eta}(\|\mathbf{w}-\mathbf{w}_t\|^2 + \|\boldsymbol{\lambda}-\boldsymbol{\lambda}_t\|^2 - \|\mathbf{w}-\mathbf{w}_{t+1}\|^2 - \|\boldsymbol{\lambda}-\boldsymbol{\lambda}_{t+1}\|^2) \\ + \frac{\eta}{2}(\|\nabla_{\mathbf{w}}\mathcal{L}_t(\mathbf{w}_t,\boldsymbol{\lambda}_t)\|^2 + \|\nabla_{\boldsymbol{\lambda}}\mathcal{L}_t(\mathbf{w}_t,\boldsymbol{\lambda}_t)\|^2).\end{aligned}$$

Proof. Following the analysis of [158], convexity of $\mathcal{L}_t(\cdot,\boldsymbol{\lambda})$ implies that

$$\mathcal{L}_t(\mathbf{w}_t,\boldsymbol{\lambda}_t) - \mathcal{L}_t(\mathbf{w},\boldsymbol{\lambda}_t) \leq \langle \mathbf{w}_t - \mathbf{w}, \nabla_{\mathbf{w}}\mathcal{L}_t(\mathbf{w}_t,\boldsymbol{\lambda}_t)\rangle \tag{9.29}$$

and by concavity of $\mathcal{L}_t(\mathbf{w},\cdot)$ we have

$$\mathcal{L}_t(\mathbf{w}_t,\boldsymbol{\lambda}) - \mathcal{L}_t(\mathbf{w}_t,\boldsymbol{\lambda}_t) \leq \langle \boldsymbol{\lambda} - \boldsymbol{\lambda}_t, \nabla_{\boldsymbol{\lambda}}\mathcal{L}_t(\mathbf{w}_t,\boldsymbol{\lambda}_t)\rangle. \tag{9.30}$$

Combining the inequalities (9.29) and (9.30) results in

$$\mathcal{L}_t(\mathbf{w}_t,\boldsymbol{\lambda}) - \mathcal{L}_t(\mathbf{w},\boldsymbol{\lambda}_t) \leq \langle \mathbf{w}_t - \mathbf{w}, \nabla_{\mathbf{w}}\mathcal{L}_t(\mathbf{w}_t,\boldsymbol{\lambda}_t)\rangle - \langle \boldsymbol{\lambda} - \boldsymbol{\lambda}_t, \nabla_{\boldsymbol{\lambda}}\mathcal{L}_t(\mathbf{w}_t,\boldsymbol{\lambda}_t)\rangle. \tag{9.31}$$

Algorithm 14 Online Gradient Descent with Soft Constraints

```
Input:
  • constraints $g_i(\mathbf{w}) \leq 0, i \in [m]$
  • step size $\eta$
  • constant $\delta > 0$
Initialize: $\mathbf{w}_1 = \mathbf{0}$ and $\boldsymbol{\lambda}_1 = \mathbf{0}$
for $t = 1, 2, \ldots, T$ do
    Submit solution $\mathbf{w}_t$
    Receive the convex function $f_t(\cdot)$ and suffer loss $f_t(\mathbf{w}_t)$
    Compute the gradients as:
        $\nabla_{\mathbf{w}} \mathcal{L}_t(\mathbf{w}_t, \boldsymbol{\lambda}_t) = \nabla f_t(\mathbf{w}_t) + \sum_{i=1}^{m} \lambda_t^i \nabla g_i(\mathbf{w}_t)$
        $\nabla_{\lambda_i} \mathcal{L}_t(\mathbf{w}_t, \boldsymbol{\lambda}_t) = g_i(\mathbf{w}_t) - \eta\delta\lambda_t^i$
    Update $\mathbf{w}_t$ and $\boldsymbol{\lambda}_t$ by:
        $\mathbf{w}_{t+1} = \Pi_{\mathbb{B}}\left(\mathbf{w}_t - \eta \nabla_{\mathbf{w}} \mathcal{L}_t(\mathbf{w}_t, \boldsymbol{\lambda}_t)\right)$
        $\boldsymbol{\lambda}_{t+1} = \Pi_{[0,+\infty)^m}(\boldsymbol{\lambda}_t + \eta \nabla_{\boldsymbol{\lambda}} \mathcal{L}_t(\mathbf{w}_t, \boldsymbol{\lambda}_t))$
end for
```

Using the update rule for $\mathbf{w}_{t+1}$ in terms of $\mathbf{w}_t$ and expanding, we get

$$\|\mathbf{w} - \mathbf{w}_{t+1}\|^2 \leq \|\mathbf{w} - \mathbf{w}_t\|^2 - 2\eta\langle \mathbf{w}_t - \mathbf{w}, \nabla_{\mathbf{w}} \mathcal{L}_t(\mathbf{w}_t, \boldsymbol{\lambda}_t)\rangle + \eta^2 \|\nabla_{\mathbf{w}} \mathcal{L}_t(\mathbf{w}_t, \boldsymbol{\lambda}_t)\|^2, \quad (9.32)$$

where the first inequality follows from the nonexpansive property of the projection operation (see Lemma A.5). Expanding the inequality for $\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_{t+1}\|^2$ in terms of $\boldsymbol{\lambda}_t$ and plugging back into the (9.31) with (9.32) establishes the desired inequality. □

Proposition 9.18. *Let $\mathbf{w}_t$ and $\boldsymbol{\lambda}_t, t \in [T]$ be the sequence of solutions obtained by Algorithm 14. Then for any $\mathbf{w} \in \mathbb{B}$ and $\boldsymbol{\lambda} \in \mathbb{R}_+^m$, we have*

$$\begin{aligned}&\sum_{t=1}^{T} \mathcal{L}_t(\mathbf{w}_t, \boldsymbol{\lambda}) - \mathcal{L}_t(\mathbf{w}, \boldsymbol{\lambda}_t)\\ &\leq \frac{R^2 + \|\boldsymbol{\lambda}\|^2}{2\eta} + \frac{\eta T}{2}\left((m+1)G^2 + 2mD^2\right) + \frac{\eta}{2}\left((m+1)G^2 + 2m\delta^2\eta^2\right)\sum_{t=1}^{T}\|\boldsymbol{\lambda}_t\|^2.\end{aligned}$$

Proof. We first bound the gradient terms in the right hand side of Lemma 9.17. Using the inequality $(a_1 + a_2 + \ldots, a_n)^2 \leq n(a_1^2 + a_2^2 + \ldots + a_n^2)$, we have $\|\nabla_{\mathbf{w}}\mathcal{L}_t(\mathbf{w}_t, \boldsymbol{\lambda}_t)\|^2 \leq (m+1)G^2\left(1 + \|\boldsymbol{\lambda}_t\|^2\right)$ and $\|\nabla_{\boldsymbol{\lambda}}\mathcal{L}_t(\mathbf{w}_t, \boldsymbol{\lambda}_t)\|^2 \leq 2m(D^2 + \delta^2\eta^2\|\boldsymbol{\lambda}_t\|^2)$. In Lemma 9.17, by adding the inequalities of all iterations, and using the fact $\|\mathbf{w}\| \leq R$ we complete the proof. □

The following theorem bounds the regret and the violation of the constraints in the long run for Algorithm 14.

Theorem 9.19. *Define $a = R\sqrt{(m+1)G^2 + 2mD^2}$. Set $\eta = R^2/[a\sqrt{T}]$. Assume T is large enough such that $2\sqrt{2}\eta(m+1) \leq 1$. Choose δ such that $\delta \geq (m+1)G^2 + 2m\delta^2\eta^2$. Let $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_T$ be the sequence of solutions obtained by Algorithm 14. Then for the optimal solution $\mathbf{w}_* = \min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^{T} f_t(\mathbf{w})$ we have*

$$\begin{aligned}&\sum_{t=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{w}_*) \leq a\sqrt{T} = O(T^{1/2}), \text{ and}\\ &\sum_{t=1}^{T} g_i(\mathbf{w}_t) \leq \sqrt{2\left(FT + a\sqrt{T}\right)\sqrt{T}\left(\frac{\delta R^2}{a} + \frac{ma}{R^2}\right)} = O(T^{3/4}).\end{aligned}$$

Proof. We begin by expanding (9.33) using (9.28) and rearranging the terms to get

$$
\begin{aligned}
&\sum_{t=1}^{T}[f_t(\mathbf{w}_t)-f_t(\mathbf{w})]+\sum_{i=1}^{m}\left\{\lambda_i\sum_{t=1}^{T}g_i(\mathbf{w}_t)-\sum_{t=1}^{T}\lambda_t^i g_i(\mathbf{w})\right\}-\frac{\delta\eta T}{2}\|\boldsymbol{\lambda}\|^2\\
&\leq -\frac{\delta\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{\lambda}_t\|^2+\frac{R^2+\|\boldsymbol{\lambda}\|^2}{2\eta}+\frac{\eta T}{2}\Big((m+1)G^2+2mD^2\Big)\\
&\quad+\frac{\eta}{2}\Big((m+1)G^2+2m\delta^2\eta^2\Big)\sum_{t=1}^{T}\|\boldsymbol{\lambda}_t\|^2.
\end{aligned}
$$

Since $\delta \geq (m+1)G^2+2m\delta^2\eta^2$, we can drop the $\|\boldsymbol{\lambda}_t\|^2$ terms from both sides of the above inequality and obtain

$$
\begin{aligned}
&\sum_{t=1}^{T}[f_t(\mathbf{w}_t)-f_t(\mathbf{w})]+\sum_{i=1}^{m}\left\{\lambda_i\sum_{t=1}^{T}g_i(\mathbf{w}_t)-\left(\frac{\delta\eta T}{2}+\frac{m}{2\eta}\right)\lambda_i^2\right\}\\
&\qquad\leq \sum_{i=1}^{m}\sum_{t=1}^{T}\lambda_t^i g_i(\mathbf{w})+\frac{R^2}{2\eta}+\frac{\eta T}{2}\Big((m+1)G^2+2mD^2)\Big).
\end{aligned}
$$

The left hand side of above inequality consists of two terms. The first term basically measures the difference between the cumulative loss of the Algorithm 14 and the optimal solution and the second term includes the constraint functions with corresponding Lagrangian multipliers which will be used to bound the long term violation of the constraints. By taking maximization for $\boldsymbol{\lambda}$ over the range $(0,+\infty)$, we get

$$
\begin{aligned}
&\sum_{t=1}^{T}[f_t(\mathbf{w}_t)-f_t(\mathbf{w})]+\sum_{i=1}^{m}\left\{\frac{\Big[\sum_{t=1}^{T}g_i(\mathbf{w}_t)\Big]_+^2}{2(\delta\eta T+m/\eta)}-\sum_{t=1}^{T}\lambda_t^i g_i(\mathbf{w})\right\}\\
&\qquad\leq \frac{R^2}{2\eta}+\frac{\eta T}{2}\Big((m+1)G^2+2mD^2)\Big).
\end{aligned}
$$

Since $\mathbf{w}_* \in \mathcal{W}$, we have $g_i(\mathbf{w}_*) \leq 0, i \in [m]$, and the resulting inequality becomes

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{w}_*) + \sum_{i=1}^{m} \frac{\left[\sum_{t=1}^{T} g_i(\mathbf{w}_t)\right]_+^2}{2(\delta\eta T + m/\eta)} \leq \frac{R^2}{2\eta} + \frac{\eta T}{2}\Big((m+1)G^2 + 2mD^2\Big).$$

The statement of the first part of the theorem follows by using the expression for η. The second part is proved by substituting the regret bound by its lower bound as $\sum_{t=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{w}_*) \geq -FT$. □

Remark 9.20. *We observe that the introduction of quadratic regularizer $\delta\eta\|\boldsymbol{\lambda}\|^2/2$ allows us to turn the expression $\lambda_i \sum_{t=1}^{T} g_i(\mathbf{w}_t)$ into $\left[\sum_{t=1}^{T} g_i(\mathbf{w}_t)\right]_+^2$, leading to the bound for the violation of the constraints. In addition, the quadratic regularizer defined in terms of $\boldsymbol{\lambda}$ allows us to work with unbounded $\boldsymbol{\lambda}$ because it cancels the contribution of the $\|\boldsymbol{\lambda}_t\|$ terms from the loss function and the bound on the gradients $\|\nabla_{\mathbf{w}}\mathcal{L}_t(\mathbf{w}, \boldsymbol{\lambda})\|$. Note that the constraint for δ mentioned in Theorem 9.19 is equivalent to*

$$\frac{2}{1/(m+1) + \sqrt{(m+1)^{-2} - 8G^2\eta^2}} \leq \delta \leq \frac{1/(m+1) + \sqrt{(m+1)^{-2} - 8G^2\eta^2}}{4\eta^2}, \quad (9.33)$$

from which, when T is large enough (i.e., η is small enough), we can simply set $\delta = 2(m+1)G^2$ that will obey the constraint in (9.33).

By investigating Lemma 9.17, it turns out that the boundedness of the gradients is essential to obtain bounds for Algorithm 14 in Theorem 9.19. Although, at each iteration, $\boldsymbol{\lambda}_t$ is projected onto the $\mathbb{R}_+^m$, since $\mathcal{W}$ is a compact set and functions $f_t(\mathbf{w})$ and $g_i(\mathbf{w}), i \in [m]$ are convex, the boundedness of the functions implies that the gradients are bounded [22, Proposition 4.2.3].

9.5.3 An Efficient Algorithm with $O(T^{3/4})$ Regret Bound and without Violation of Constraints

In this subsection we generalize Algorithm 14 such that the constrained are satisfied in a long run. To create a sequence of solutions $\{\mathbf{w}_t, t \in [T]\}$ that satisfies the long term constraints $\sum_{t=1}^{T} g_i(\mathbf{w}_t) \leq 0, i \in [m]$, we make two modifications to Algorithm 14. First, instead of handling all of the m constraints, we consider a single constraint defined as $g(\mathbf{w}) = \max_{i \in [m]} g_i(\mathbf{w})$. Apparently, by achieving zero violation for the constraint $g(\mathbf{w}) \leq 0$, it is guaranteed that all of the constraints $g_i(\cdot), i \in [m]$ are also satisfied in the long term. Furthermore, we change Algorithm 14 by modifying the definition of $\mathcal{L}_t(\cdot, \cdot)$ as

$$\mathcal{L}_t(\mathbf{w}, \lambda) = f_t(\mathbf{w}) + \lambda(g(\mathbf{w}) + \gamma) - \frac{\eta\delta}{2}\lambda^2, \tag{9.34}$$

where $\gamma > 0$ will be decided later. This modification is equivalent to considering the constraint $g(\mathbf{w}) \leq -\gamma$, a tighter constraint than $g(\mathbf{w}) \leq 0$. The main idea behind this modification is that by using a tighter constraint in our algorithm, the resulting sequence of solutions will satisfy the long term constraint $\sum_{t=1}^{T} g(\mathbf{w}_t) \leq 0$, even though the tighter constraint is violated in many trials.

Before proceeding, we state a fact about the Lipschitz continuity of the function $g(\mathbf{w})$ in the following proposition.

Proposition 9.21. *Assume that functions $g_i(\cdot), i \in [m]$ are Lipschitz continuous with constant G. Then, function $g(\mathbf{w}) = \max_{i \in [m]} g_i(\mathbf{w})$ is Lipschitz continuous with constant G, that is,*

$$|g(\mathbf{w}) - g(\mathbf{w}')| \leq G\|\mathbf{w} - \mathbf{w}'\| \;\; \textit{for any } \mathbf{w} \in \mathbb{B} \textit{ and } \mathbf{w}' \in \mathbb{B}.$$

Proof. See Appendix **??**. □

To obtain a zero bound on the violation of constraints in the long run, we make the following assumption about the constraint function $g(\mathbf{w})$.

Assumption 9.22. *Let $\mathcal{W}' \subseteq \mathcal{W}$ be the convex set defined as $\mathcal{W}' = \{\mathbf{w} \in \mathbb{R}^d : g(\mathbf{w}) + \gamma \leq 0\}$ where $\gamma \geq 0$. We assume that the norm of the gradient of the constraint function $g(\mathbf{w})$ is lower bounded at the boundary of $\mathcal{W}'$, that is,*

$$\textbf{A5} \qquad \min_{g(\mathbf{w})+\gamma=0} \|\nabla g(\mathbf{w})\| \geq \sigma.$$

A direct consequence of assumption **A5** is that by reducing the domain $\mathcal{W}$ to $\mathcal{W}'$, the optimal value of the constrained optimization problem $\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w})$ does not change much, as revealed by the following theorem.

Theorem 9.23. *Let $\mathbf{w}_*$ and $\mathbf{w}_\gamma$ be the optimal solutions to the constrained optimization problems defined as $\min_{g(\mathbf{w})\leq 0} f(\mathbf{w})$ and $\min_{g(\mathbf{w})\leq -\gamma} f(\mathbf{w})$, respectively, where $f(\mathbf{w}) = \sum_{t=1}^{T} f_t(\mathbf{w})$ and $\gamma \geq 0$. We have*

$$|f(\mathbf{w}_*) - f(\mathbf{w}_\gamma)| \leq \frac{G}{\sigma}\gamma T.$$

Proof. We note that the optimization problem $\min_{g(\mathbf{w})\leq -\gamma} f(\mathbf{w}) = \min_{g(\mathbf{w})\leq -\gamma} \sum_{t=1}^{T} f_t(\mathbf{w})$, can also be written in the minimax form as

$$f(\mathbf{w}_\gamma) = \min_{\mathbf{w}\in\mathbb{B}} \max_{\lambda\in\mathbb{R}_+} \sum_{t=1}^{T} f_t(\mathbf{w}) + \lambda(g(\mathbf{w}) + \gamma), \tag{9.35}$$

where we use the fact that $\mathcal{W}' \subseteq \mathcal{W} \subseteq \mathbb{B}$. We denote by $\mathbf{w}_\gamma$ and λ_γ the optimal solutions to (9.35). We have

$$\begin{aligned} f(\mathbf{w}_\gamma) &= \min_{\mathbf{w}\in\mathbb{B}} \max_{\lambda\in\mathbb{R}_+} \sum_{t=1}^T f_t(\mathbf{w}) + \lambda(g(\mathbf{w}) + \gamma) \\ &= \min_{\mathbf{w}\in\mathbb{B}} \sum_{t=1}^T f_t(\mathbf{w}) + \lambda_\gamma(g(\mathbf{w}) + \gamma) \\ &\leq \sum_{t=1}^T f_t(\mathbf{w}_*) + \lambda_\gamma(g(\mathbf{w}_*) + \gamma) \leq \sum_{t=1}^T f_t(\mathbf{w}_*) + \lambda_\gamma\gamma, \end{aligned}$$

where the second equality follows the definition of the $\mathbf{w}_\gamma$ and the last inequality is due to the optimality of $\mathbf{w}_*$, that is, $g(\mathbf{w}_*) \leq 0$.

To bound $|f(\mathbf{w}_\gamma) - f(\mathbf{w}_*)|$, we need to bound λ_γ. Since $\mathbf{w}_\gamma$ is the minimizer of (9.35), from the optimality condition we have

$$-\sum_{t=1}^T \nabla f_t(\mathbf{w}_\gamma) = \lambda_\gamma \nabla g(\mathbf{w}_\gamma). \tag{9.36}$$

By setting $\mathbf{v} = -\sum_{t=1}^T \nabla f_t(\mathbf{w}_\gamma)$, we can simplify (9.36) as $\lambda_\gamma \nabla g(\mathbf{w}_\gamma) = \mathbf{v}$. From the KKT optimality condition [34], if $g(\mathbf{w}_\gamma) + \gamma < 0$ then we have $\lambda_\gamma = 0$; otherwise according to Assumption 9.22 we can bound λ_γ by

$$\lambda_\gamma \leq \frac{\|\mathbf{v}\|}{\|\nabla g(\mathbf{w}_\gamma)\|} \leq \frac{GT}{\sigma}.$$

We complete the proof by applying the fact $f(\mathbf{w}_*) \leq f(\mathbf{w}_\gamma) \leq f(\mathbf{w}_*) + \lambda_\gamma\gamma$. □

As indicated by Theorem 9.23, when γ is small, we expect the difference between two optimal values $f(\mathbf{w}_*)$ and $f(\mathbf{w}_\gamma)$ to be small. Using the result from Theorem 9.23, in the following theorem, we show that by running Algorithm 14 on the modified convex-concave functions defined in (9.34), we are able to obtain an $O(T^{3/4})$ regret bound and zero bound on the violation of constraints in the long run.

Theorem 9.24. *Set $a = 2R/\sqrt{2G^2 + 3(D^2 + b^2)}$, $\eta = R^2/[a\sqrt{T}]$, and $\delta = 4G^2$. Let $\mathbf{w}_t, t \in [T]$ be the sequence of solutions obtained by Algorithm 14 with functions defined in (9.34) with $\gamma = bT^{-1/4}$ and $b = 2\sqrt{F(\delta R^2 a^{-1} + aR^{-2})}$. Let $\mathbf{w}_*$ be the optimal solution to $\min_{\mathbf{w}\in\mathcal{W}} \sum_{t=1}^T f_t(\mathbf{w})$. With sufficiently large T, that is, $FT \geq a\sqrt{T}$, and under Assumption 9.22, we have $\mathbf{w}_t, t \in [T]$ satisfy the global constraint $\sum_{t=1}^T g(\mathbf{w}_t) \leq 0$ and the regret is bounded by*

$$\text{Regret}_T = \sum_{t=1}^T f_t(\mathbf{w}_t) - f_t(\mathbf{w}_*) \leq a\sqrt{T} + \frac{b}{\sigma} G T^{3/4} = O(T^{3/4}).$$

Proof. Let $\mathbf{w}_\gamma$ be the optimal solution to $\min_{g(\mathbf{w})\leq -\gamma} \sum_{t=1}^T f_t(\mathbf{w})$. Similar to the proof of Theorem 9.19 when applied to functions in (9.34) we have

$$\begin{aligned}
&\sum_{t=1}^T f_t(\mathbf{w}_t) - \sum_{t=1}^T f_t(\mathbf{w}) + \lambda \sum_{t=1}^T (g(\mathbf{w}_t) + \gamma) - \left(\sum_{t=1}^T \lambda_t\right)(g(\mathbf{w}) + \gamma) - \frac{\delta\eta T}{2}\lambda^2 \\
&\leq \; -\frac{\delta\eta}{2}\sum_{t=1}^T \lambda_t^2 + \frac{R^2 + \lambda^2}{2\eta} + \frac{\eta T}{2}\left(2G^2 + 3(D^2 + \gamma^2)\right) + \frac{\eta}{2}\left(2G^2 + 3\delta^2\eta^2\right)\sum_{t=1}^T \lambda_t^2.
\end{aligned}$$

By setting $\delta \geq 2G^2 + 3\delta^2\eta^2$ which is satisfied by $\delta = 4G^2$, we cancel the terms including λ_t from the right hand side of above inequality. By maximizing for λ over the range $(0, +\infty)$ and noting that $\gamma \leq b$, for the optimal solution $\mathbf{w}_\gamma$, we have

$$\sum_{t=1}^{T} \left[f_t(\mathbf{w}_t) - f_t(\mathbf{w}_\gamma)\right] + \frac{\left[\sum_{t=1}^{T} g(\mathbf{w}_t) + \gamma T\right]_+^2}{2(\delta\eta T + 1/\eta)} \leq \frac{R^2}{2\eta} + \frac{\eta T}{2}\left(2G^2 + 3(D^2 + b^2)\right),$$

which, by optimizing for η and applying the lower bound for the regret as $\sum_{t=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{w}_\gamma) \geq -FT$, yields the following inequalities

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{w}_\gamma) \leq a\sqrt{T} \tag{9.37}$$

and

$$\sum_{t=1}^{T} g(\mathbf{w}_t) \leq \sqrt{2\left(FT + a\sqrt{T}\right)\sqrt{T}\left(\frac{\delta R^2}{a} + \frac{a}{R^2}\right)} - \gamma T, \tag{9.38}$$

for the regret and the violation of the constraint, respectively. Combining (9.37) with the result of Theorem 9.23 results in $\sum_{t=1}^{T} f_t(\mathbf{w}_\gamma) \leq \sum_{t=1}^{T} f_t(\mathbf{w}_*) + a\sqrt{T} + (G/\sigma)\gamma T$. By choosing $\gamma = bT^{-1/4}$ we attain the desired regret bound as

$$\sum_{t=1}^{T} f_t(\mathbf{w}_t) - f_t(\mathbf{w}_*) \leq a\sqrt{T} + \frac{bG}{\sigma}T^{3/4} = O(T^{3/4}).$$

To obtain the bound on the violation of constraints, we note that in (9.38), when T is sufficiently large, that is, $FT \geq a\sqrt{T}$, we have $\sum_{t=1}^{T} g(\mathbf{w}_t) \leq 2\sqrt{F(\delta R^2 a^{-1} + aR^{-2})}T^{3/4} - bT^{3/4}$. Choosing $b = 2\sqrt{F(\delta R^2 a^{-1} + aR^{-2})}T^{3/4}$ guarantees the zero bound on the violation of constraints as claimed. □

9.6 Proofs of Convergence Rates

In this section we provide the proof of the main lemmas omitted from the analysis of convergence rate.

9.6.1 Proof of Lemma 9.10

Following the standard analysis of gradient descent methods, we have for any $\mathbf{w} \in \mathbb{B}$,

$$\begin{aligned}
&\|\mathbf{w}_{t+1} - \mathbf{w}\|_2^2 - \|\mathbf{w}_t - \mathbf{w}\|_2^2 \leq \|\mathbf{w}'_{t+1} - \mathbf{w}\|_2^2 - \|\mathbf{w}_t - \mathbf{w}\|_2^2 \\
&= \|\mathbf{w}_t - \eta_t(\mathbf{g}(\mathbf{w}_t, \xi_t) + \lambda_t \nabla g(\mathbf{w}_t)) - \mathbf{w}\|_2^2 - \|\mathbf{w}_t - \mathbf{w}\|_2^2 \\
&\leq \eta_t^2 \|\mathbf{g}(\mathbf{w}_t, \xi_t) + \lambda_t \nabla g(\mathbf{w}_t)\|_2^2 - 2\eta_t(\mathbf{w}_t - \mathbf{w})^\top(\mathbf{g}(\mathbf{w}_t, \xi_t) + \lambda_t \nabla g(\mathbf{w}_t)) \\
&\leq \eta_t^2 \|\mathbf{g}(\mathbf{w}_t, \xi_t) + \lambda_t \nabla g(\mathbf{w}_t)\|_2^2 \\
&\quad - 2\eta_t(\mathbf{w}_t - \mathbf{w})^\top \underbrace{(\nabla f(\mathbf{w}_t) + \lambda_t \nabla g(\mathbf{w}_t))}_{\equiv \nabla_{\mathbf{w}} L(\mathbf{w}_t, \lambda_t)} + 2\eta_t \underbrace{(\mathbf{w} - \mathbf{w}_t)^\top(\mathbf{g}(\mathbf{w}_t, \xi_t) - \nabla f(\mathbf{w}_t))}_{\equiv \zeta_t(\mathbf{w})},
\end{aligned}$$

Then we have

$$\begin{aligned}
&(\mathbf{w}_t - \mathbf{w})^\top \nabla_{\mathbf{w}} L(\mathbf{w}_t, \lambda_t) \\
&\leq \frac{1}{2\eta_t}\left(\|\mathbf{w}_t - \mathbf{w}\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{w}\|_2^2\right) + \frac{\eta_t}{2}\|\mathbf{g}(\mathbf{w}_t, \xi_t) + \lambda_t \nabla g(\mathbf{w}_t)\|_2^2 + \zeta_t(\mathbf{w}) \\
&\leq \frac{1}{2\eta_t}\left(\|\mathbf{w}_t - \mathbf{w}\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{w}\|_2^2\right) + \eta_t\|\mathbf{g}(\mathbf{w}_t, \xi_t)\|_2^2 + \eta_t \lambda_t^2 \|\nabla g(\mathbf{w}_t)\|_2^2 + \zeta_t(\mathbf{w}) \\
&\leq \frac{1}{2\eta_t}\left(\|\mathbf{w}_t - \mathbf{w}\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{w}\|_2^2\right) \\
&\quad + 2\eta_t \underbrace{\|\mathbf{g}(\mathbf{w}_t, \xi_t) - \nabla f(\mathbf{w}_t)\|_2^2}_{\equiv \Delta_t} + 2\eta_t \|\nabla f(\mathbf{w}_t)\|_2^2 + \eta_t \lambda_t^2 \|\nabla g(\mathbf{w}_t)\|_2^2 + \zeta_t(\mathbf{w})
\end{aligned}$$

By using the bound on $\|\nabla f(\mathbf{w}_t)\|_2$ and $\|\nabla g(\mathbf{w}_t)\|_2$, we obtain the first inequality in Lemma 9.10. To prove the second inequality, we follow the same analysis, i.e.,

$$\begin{aligned}
&|\lambda_{t+1} - \lambda|^2 - |\lambda_t - \lambda|^2 \leq |\lambda_t + \eta_t(g(\mathbf{w}_t) - \gamma\lambda_t)|^2 - |\lambda_t - \lambda|^2 \\
&\leq \eta_t^2 |g(\mathbf{w}_t) - \gamma\lambda_t|^2 + 2\eta_t(\lambda_t - \lambda)\underbrace{(g(\mathbf{w}_t) - \gamma\lambda_t)}_{\equiv\nabla_\lambda L(\mathbf{w}_t,\lambda_t)}.
\end{aligned}$$

Then we have

$$(\lambda - \lambda_t)\nabla_\lambda L(\mathbf{w}_t, \lambda_t) \leq \frac{1}{2\eta_t}\left(|\lambda_t - \lambda|^2 - |\lambda_{t+1} - \lambda|^2\right) + \frac{\eta_t}{2}|g(\mathbf{w}_t) - \gamma\lambda_t|^2.$$

By induction, it is straightforward to show that $\lambda_t \leq C_2/\gamma$, which yields the second inequality in Lemma 9.10, i.e.,

$$(\lambda - \lambda_t)\nabla_\lambda L(\mathbf{w}_t, \lambda_t) \leq \frac{1}{2\eta_t}\left(|\lambda_t - \lambda|^2 - |\lambda_{t+1} - \lambda|^2\right) + 2\eta_t C_2^2.$$

9.6.2 Proof of Lemma 9.11

Since $L_t(\mathbf{w}, \lambda)$ is convex in $\mathbf{w}$ and concave in λ, we have the following inequalities

$$\begin{aligned}
&L(\mathbf{w}, \lambda_t) - L(\mathbf{w}_t, \lambda_t) \geq (\mathbf{w} - \mathbf{w}_t)^\top \nabla_{\mathbf{w}} L(\mathbf{w}_t, \lambda_t), \\
&L(\mathbf{w}_t, \lambda) - L(\mathbf{w}_t, \lambda_t) \leq (\lambda - \lambda_t)\nabla_\lambda L(\mathbf{w}_t, \lambda_t).
\end{aligned}$$

Using the inequalities in Lemma 9.10, we have

$$
\begin{aligned}
L(\mathbf{w}_t, \lambda_t) - L(\mathbf{w}, \lambda_t) &\leq \frac{1}{2\eta_t}\left(\|\mathbf{w} - \mathbf{w}_t\|_2^2 - \|\mathbf{w} - \mathbf{w}_{t+1}\|_2^2\right) + 2\eta_t G_1^2 + \eta_t G_2^2 \lambda_t^2 + 2\eta_t \Delta_t + \zeta_t(\mathbf{w}),\\
L(\mathbf{w}_t, \lambda) - L(\mathbf{w}_t, \lambda_t) &\leq \frac{1}{2\eta_t}\left(|\lambda - \lambda_t|^2 - |\lambda - \lambda_{t+1}|^2\right) + 2\eta_t C_2^2,
\end{aligned}
$$

where $\zeta_t(\mathbf{w}) = \langle \mathbf{w} - \mathbf{w}_t, \mathbf{g}(\mathbf{w}_t, \xi_t) - \nabla f(\mathbf{w}_t)\rangle$ as abbreviated before. Since $\eta_1 = \cdots = \eta_T$, denoted by η, by taking summation of above two inequalities over $t = 1, \cdots, T$, we get

$$
\sum_{t=1}^T L(\mathbf{w}_t, \lambda) - L(\mathbf{w}, \lambda_t) \leq \frac{\|\mathbf{w}\|_2^2}{2\eta} + \frac{\lambda^2}{2\eta} + 2\eta T(G_1^2 + C_2^2) + \sum_t \eta G_2^2 \lambda_t^2 + 2\eta \sum_{t=1}^T \Delta_t + \sum_{t=1}^T \zeta_t(\mathbf{w}).
$$

By plugging the expression of $L(\mathbf{w}, \lambda)$, and due to $\|\mathbf{w}\|_2 \leq 1$, we have

$$
\begin{aligned}
&\sum_{t=1}^T (f(\mathbf{w}_t) - f(\mathbf{w})) + \lambda \sum_{t=1}^T g(\mathbf{w}_t) - \left(\frac{\gamma T}{2} + \frac{1}{2\eta}\right)\lambda^2\\
&\leq \frac{1}{2\eta} + 2\eta T(G_1^2 + C_2^2) + \sum_t (\eta G_2^2 - \gamma/2)\lambda_t^2 + \sum_t \lambda_t g(\mathbf{w}) + 2\eta \sum_{t=1}^T \Delta_t + \sum_{t=1}^T \zeta_t(\mathbf{w}).
\end{aligned}
$$

Let $\mathbf{w} = \mathbf{w}_* = \arg\min_{\mathbf{w} \in \mathcal{W}} f(\mathbf{w})$. By taking minimization over $\lambda \geq 0$ on left hand side and considering $\eta = \gamma/(2G_2^2)$, we have

$$
\sum_{t=1}^T (f(\mathbf{w}_t) - f(\mathbf{w}_*)) + \frac{[\sum_{t=1}^T g(\mathbf{w}_t)]_+^2}{2(\gamma T + 2G_2^2/\gamma)} \leq \frac{G_2^2}{\gamma} + \frac{(G_1^2 + C_2^2)}{G_2^2}\gamma T + \frac{\gamma}{G_2^2}\sum_{t=1}^T \Delta_t + \sum_{t=1}^T \zeta_t(\mathbf{w}_*)
$$

9.6.3 Proof of Lemma 9.13

Since $F(\mathbf{w})$ is strongly convex in $\mathbf{w}$, we have

$$F(\mathbf{w}) - F(\mathbf{w}_t) \geq \langle \mathbf{w} - \mathbf{w}_t, \nabla F(\mathbf{w}_t)\rangle + \frac{\beta}{2}\|\mathbf{w} - \mathbf{w}_t\|_2^2.$$

Following the same analysis as in Lemma 9.10, we have

$$\begin{aligned}
(\mathbf{w}_t - \mathbf{w})^\top \nabla F(\mathbf{w}_t) \leq & \frac{1}{2\eta_t}\left(\|\mathbf{w} - \mathbf{w}_t\|_2^2 - \|\mathbf{w} - \mathbf{w}_{t+1}\|_2^2\right) + \frac{\eta_t}{2}\|\mathbf{g}(\mathbf{w}_t, \xi_t) + p(\mathbf{w}_t)\lambda_0 \nabla g(\mathbf{w}_t)\|_2^2 \\
& + \zeta_t(\mathbf{w}) - \frac{\beta}{2}\|\mathbf{w} - \mathbf{w}_t\|_2^2 \\
\leq & \frac{1}{2\eta_t}\left(\|\mathbf{w} - \mathbf{w}_t\|_2^2 - \|\mathbf{w} - \mathbf{w}_{t+1}\|_2^2\right) \\
& + \eta_t G_1^2 + \eta_t \lambda_0^2 G_2^2 + \zeta_t(\mathbf{w}) - \frac{\beta}{2}\|\mathbf{w} - \mathbf{w}_t\|_2^2,
\end{aligned}$$

where

$$p(\mathbf{w}) = \frac{\exp\left(\lambda_0 g(\mathbf{w})/\gamma\right)}{1 + \exp\left(\lambda_0 g(\mathbf{w})/\gamma\right)}.$$

Taking summation of above inequality over $t = 1, \cdots, T$ gives

$$\begin{aligned}
\sum_{t=1}^{T} F(\mathbf{w}_t) - F(\mathbf{w}) \leq & \sum_{t=1}^{T} \frac{1}{2}\left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} - \frac{\beta}{2}\right)\|\mathbf{w} - \mathbf{w}_t\|_2^2 \\
& + \sum_{t=1}^{T} \eta_t (G_1^2 + \lambda_0^2 G_2^2) + \sum_{t=1}^{T} \zeta_t(\mathbf{w}) - \frac{\beta}{4}\sum_{t=1}^{T}\|\mathbf{w} - \mathbf{w}_t\|_2^2.
\end{aligned}$$

Since $\eta_t = 1/(2\beta t)$, we have

$$\sum_{t=1}^{T}(F(\mathbf{w}_t) - F(\mathbf{w})) \leq \frac{(G_1^2 + \lambda_0^2 G_2^2)(1+\ln T)}{2\beta} + \sum_{t=1}^{T}\zeta_t(\mathbf{w}) - \frac{\beta}{4}\sum_{t=1}^{T}\|\mathbf{w} - \mathbf{w}_t\|_2^2$$

We complete the proof by letting $\mathbf{w} = \mathbf{w}_* = \arg\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w})$.

9.6.4 Proof of Lemma 9.14

The proof is based on the Berstein's inequality for martingales (see Theorem A.25). To do so, define the martingale difference $X_t = \langle \mathbf{w} - \mathbf{w}_t, \nabla f(\mathbf{w}_t) - \mathbf{g}(\mathbf{w}_t, \xi_t)\rangle$ and martingale $\Lambda_T = \sum_{t=1}^{T} X_t$. Define the conditional variance Σ_T^2 as

$$\Sigma_T^2 = \sum_{t=1}^{T}\mathbb{E}_{\xi_t}\left[X_t^2\right] \leq 4G_1^2\sum_{t=1}^{T}\|\mathbf{w}_t - \mathbf{w}\|_2^2 = 4G_1^2 D_T.$$

Define $K = 4G_1$. We have

$$\begin{aligned}
&\Pr\left(\Lambda_T \geq 2\sqrt{4G_1^2 D_T\tau} + \sqrt{2}K\tau/3\right)\\
&= \Pr\left(\Lambda_T \geq 2\sqrt{4G_1^2 D_T\tau} + \sqrt{2}K\tau/3, \Sigma_T^2 \leq 4G_1^2 D_T\right)\\
&= \Pr\left(\Lambda_T \geq 2\sqrt{4G_1^2 D_T\tau} + \sqrt{2}K\tau/3, \Sigma_T^2 \leq 4G_1^2 D_T, D_T \leq \frac{4}{T}\right)\\
&+ \sum_{i=1}^{m}\Pr\left(\Lambda_T \geq 2\sqrt{4G_1^2 D_T\tau} + \sqrt{2}K\tau/3, \Sigma_T^2 \leq 4G_1^2 D_T, \frac{4}{T}2^{i-1} < D_T \leq \frac{4}{T}2^i\right)\\
&\leq \Pr\left(D_T \leq \frac{4}{T}\right) + \sum_{i=1}^{m}\Pr\left(\Lambda_T \geq \sqrt{2\times 4G_1^2\frac{4}{T}2^i\tau} + \sqrt{2}K\tau/3, \Sigma_T^2 \leq 4G_1^2\frac{4}{T}2^i\right)\\
&\leq \Pr\left(D_T \leq \frac{4}{T}\right) + me^{-\tau}.
\end{aligned}$$

where we use the fact $\|\mathbf{w}_t - \mathbf{w}\|_2^2 \leq 4$ for any $\mathbf{w} \in \mathbb{B}$, and the last step follows the Bernstein inequality for martingales. We complete the proof by setting $\tau = \ln(m/\delta)$.

9.7 Summary

In this chapter, we made a progress towards making the SGD method efficient by proposing a framework in which it is possible to exclude the projection steps from the SGD algorithm. We have proposed two novel algorithms to overcome the computational bottleneck of the projection step in applying SGD to optimization problems with complex domains. We showed using novel theoretical analysis that the proposed algorithms can achieve an $O(1/\sqrt{T})$ convergence rate for general convex functions and an $O(\ln T/T)$ rate for strongly convex functions with a overwhelming probability which are known to be optimal (up to a logarithmic factor) for stochastic optimization.

We have also addressed the problem of online convex optimization with constraints, where we only need the constraints to be satisfied in the long run. In addition to the regret bound which is the main tool in analyzing the performance of general online convex optimization algorithms, we defined the bound on the violation of constraints in the long term which measures the cumulative violation of the solutions from the constraints for all rounds. Our setting is applied to solving online convex optimization without projecting the solutions onto the complex convex domain at each iteration, which may be computationally inefficient for complex domains. Our strategy is to turn the problem into an online convex-concave optimization problem and apply online gradient descent algorithm to solve it.

9.8 Bibliographic Notes

Generally, the computational complexity of the projection step in SGD has seldom been taken into account in the literature. Here, we briefly review the previous works on projection free convex optimization, which is closely related to the theme of this study. For some specific domains, efficient algorithms have been developed to circumvent the high computational cost caused by projection step at each iteration of gradient descent methods. The main idea is to select an appropriate direction to take from the current solution such that the next solution is guaranteed to stay within the domain. Clarkson [42] proposed a sparse greedy approximation algorithm for convex optimization over a simplex domain, which is a generalization of an old algorithm by Frank and Wolfe [56] (a.k.a conditional gradient descent [21]). Zhang [155] introduced a similar sequential greedy approximation algorithm for certain convex optimization problems over a domain given by a convex hull. Hazan [67] devised an algorithm for approximately maximizing a concave function over a trace norm bounded PSD cone, which only needs to compute the maximum eigenvalue and the corresponding eigenvector of a symmetric matrix. Ying et al. [151] formulated the distance metric learning problems into eigenvalue maximization and proposed an algorithm similar to [67].

Recently, Jaggi [77] put these ideas into a general framework for convex optimization with a general convex domain. Instead of projecting the intermediate solution into a complex convex domain, Jaggi's algorithm solves a linearized problem over the same domain. He showed that Clark's algorithm , Zhang's algorithm and Hazan's algorithm discussed above are special cases of his general algorithm for special domains. It is important to note that all these algorithms are designed for batch optimization, not for stochastic optimization, which is the focus of this chapter.

The proposed stochastic optimization methods with only one projection are closely related to the online Frank-Wolfe algorithm proposed in [73]. It is a projection free online learning algorithm, built on the the assumption that it is possible to efficiently minimize a linear function over the complex domain. Indeed, the FW algorithm replaces the projection into the convex domain with a linear programming over the same domain which makes sense if solving the linear programming is cheaper than the projection. One main shortcoming of the OFW algorithm is that its convergence rate for general stochastic optimization is $O(T^{-1/3})$, significantly slower than that of a standard stochastic gradient descent algorithm (i.e., $O(T^{-1/2})$). It achieves a convergence rate of $O(T^{-1/2})$ only when the objective function is smooth, which unfortunately does not hold for many machine learning problems where either a non-smooth regularizer or a non-smooth loss function is used. Another limitation of OFW is that it assumes a linear optimization problem over the domain $\mathcal{W}$ can be solved efficiently. Although this assumption holds for some specific domains as discussed in [73], but in many settings of practical interest, this may not be true. The proposed algorithms address the two limitations explicitly. In particular, we show that how two seemingly different modifications of the SGD can be used to avoid performing expensive projections with similar convergency rates as the original SGD method.

The proposed online optimization with soft constraints setup is reminiscent of regret minimization with side constraints or constrained regret minimization addressed in [110], motivated by applications in wireless communication. In regret minimization with side constraints, beyond minimizing regret, the learner has some side constraints that need to be satisfied on average for all rounds. Unlike our setting, in learning with side constraints, the set $\mathcal{W}$ is controlled by adversary and can vary arbitrarily from trial to trial. It has been shown that if the convex set is affected by both decisions and loss functions, the minimax optimal regret is generally unattainable online [111].

APPENDIX

Appendix A

Technical Background

A.1 Convex Analysis

In this appendix, we introduce the basic definitions and results of convex analysis [121, 34, 28] needed for the analysis of the algorithms in this thesis. We begin by formalizing a few topological definitions we used throughout the thesis followed by few definitions and results from convexity.

Definition A.1 (Euclidean Ball)**.** *A Euclidean ball with radius r centered at point $\mathbf{w}_0$ is $\mathbb{B}(\mathbf{w}_0, r) = \{\mathbf{w} \in \mathbb{R}^d;\ \|\mathbf{w} - \mathbf{w}_0\| \leq r\}$. An ℓ_p ball is also defined equivalently but now the distance is defined by the ℓ_p norm as $\|\mathbf{w}\|_p = (\sum_{i=1}^d |w_i|^p)^{1/p}$.*

Definition A.2 (Boundary and Interior)**.** *For a given set $\mathcal{W}$, we say that $\mathbf{w}$ is on the boundary of $\mathcal{W}$, denoted by $\partial\mathcal{W}$, if for some small $\epsilon > 0$, the ball centered at $\mathbf{w}$ with radius ϵ covers both inside and outside of the set, i.e., $\mathbb{B}(\mathbf{w}_0, \epsilon) \cap \mathcal{W} \neq \emptyset$ and $\mathbb{B}(\mathbf{w}_0, \epsilon) \cap \bar{\mathcal{W}} \neq \emptyset$, where $\bar{\mathcal{W}}$ is the complement of set $\mathcal{W}$. We say that a point $\mathbf{w}$ is in the interior of set $\mathcal{W}$ if $\exists \epsilon > 0$ s.t. $\mathbb{B}(\mathbf{w}_0, \epsilon) \subset \mathcal{W}$.*

Definition A.3 (Convex Set)**.** *A set $\mathcal{W}$ in a vector space is convex if for any two vectors $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$, the line segment connecting two points is contained in $\mathcal{W}$ as well. In other fords, for any $\lambda \in [0, 1]$, we have that $\lambda\mathbf{w} + (1 - \lambda)\mathbf{w}' \in \mathcal{W}$.*

Intuitively, a set is convex if its surface has no "dips". The intersection of an arbitrary family of convex sets is obviously convex.

Definition A.4 (Projection onto Convex Sets). *Let $\mathcal{W} \subseteq \mathbb{R}^d$ be a closed convex set and let $\mathbf{w} \in \mathbb{R}^d$ be a point. The projection of $\mathbf{w}$ onto $\mathcal{W}$ is denoted by $\Pi_{\mathcal{W}}(\mathbf{w})$ and defined by*

$$\Pi_{\mathcal{W}}(\mathbf{w}) = \arg \min_{\mathbf{w}' \in \mathcal{W}} \|\mathbf{w} - \mathbf{w}'\|.$$

In particular, for every $\mathbf{w} \in \mathbb{R}^d$ there exists a point $\mathbf{w}_*$ which satisfies this and it is unique.

Here are few examples of projections. For the positive semidefinite (PSD) cone, i.e., $\mathcal{S}_+^d = \{\mathbf{X} \in \mathbb{R}^{d\times d} : \mathbf{X} \succeq 0\}$, the projection requires a full eigendecomposition of the input matrix. Precisely, let $\mathbf{W} \in \mathcal{S}_+^d$ has the eigendecomposition $\mathbf{W} = \mathbf{U}^\top \Sigma \mathbf{U}$ where $\Sigma = \text{diag}(\lambda_1, \lambda_2, \cdots, \lambda_d)$ and $\mathbf{U}$ is an orthogonal matrix. Then, $\Pi_{\mathcal{S}_+^d}(\mathbf{W}) = \mathbf{U}^\top \Sigma_+ \mathbf{U}$ where $\Sigma_+ = \text{diag}([\lambda_1]_+, [\lambda_2]_+, \cdots, [\lambda_d]_+)$. For Hyperplane $\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^d : \langle \mathbf{a}, \mathbf{x}\rangle = b\}, \mathbf{a} \neq \mathbf{0}$, the projection is equivalent to $\Pi_{\mathcal{H}}(\mathbf{w}) = \mathbf{w} - \frac{\langle \mathbf{a}, \mathbf{w}\rangle - b}{\|\mathbf{a}\|^2}\mathbf{a}$. For Affine set $\mathcal{A} = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{A}\mathbf{x} = \mathbf{b}\}$, the projection can be obtained by $\Pi_{\mathcal{A}}(\mathbf{w}) = \mathbf{w} - \mathbf{A}^\top \left(\mathbf{A}\mathbf{A}^{-1}\right)(\mathbf{A}\mathbf{w} - \mathbf{b})$. For Polyhedron set, $\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ there is no analytical solution and the projection $\Pi_{\mathcal{P}}(\mathbf{w})$ is an offline optimization by itself where the corresponding optimization problem $\arg\min_{\mathbf{w}' \in \mathcal{P}} \|\mathbf{w}' - \mathbf{w}\|$ must be solved approximately.

Lemma A.5 (Non-expensiveness of Projection). *Let $\mathcal{W} \subseteq \mathbb{R}^d$ be a convex set and consider the projection operation defined as above. Then,*

$$\|\Pi_{\mathcal{W}}(\mathbf{w}) - \Pi_{\mathcal{W}}(\mathbf{w}')\| \leq \|\mathbf{w} - \mathbf{w}'\|, \forall \mathbf{w}, \mathbf{w}' \in \mathbb{R}^d$$

Definition A.6 (Convex Function). *A function $f : \mathcal{W} \mapsto \mathbb{R}$ is said to be convex if $\mathcal{W}$ is* *convex and for every* $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$ *and* $\alpha \in [0, 1]$*,*

$$f(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') \leq \lambda f(\mathbf{w}) + (1-\lambda) f(\mathbf{w}').$$

A continuously differentiable function is convex if $f(\mathbf{w}) \geq f(\mathbf{w}') + \langle \nabla f(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle$ *for all* $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$*. If* f *is non-smooth then this inequality holds for any sub-gradient* $\mathbf{g} \in \partial f(\mathbf{w}')$*.*

Definition A.7 (Subgradient)**.** *A subgradient of a convex function* $f : \mathbb{R}^d \mapsto \mathbb{R}$ *at some point* $\mathbf{w}$ *is any vector* $\mathbf{g} \in \mathbb{R}^d$ *that achieves the same lower bound as the tangent line to function* f *at point* $\mathbf{w}$*, i.e.,*

$$f(\mathbf{w}') \geq f(\mathbf{w}) + \langle \mathbf{g}, \mathbf{w}' - \mathbf{w} \rangle, \; \forall \mathbf{w}' \in \mathbb{R}^d$$

The subgradient $\mathbf{g}$ always exists for convex functions on the relative interior of their domain. Furthermore, if f is differentiable at $\mathbf{w}$, then there is a unique subgradient $\mathbf{g} = \nabla f(\mathbf{w})$. Note that subgradients need not exist for non-convex functions.

Definition A.8 (Subdifferential)**.** *The subdifferential of a convex function* $f : \mathbb{R}^d \mapsto \mathbb{R}$ *at some point* $\mathbf{w}$ *is the set of all subgradients of* f *at* $\mathbf{w}$ *, i.e.,*

$$\partial f(\mathbf{w}) = \{\mathbf{g} \in \mathbb{R}^d : \forall \mathbf{w}', \; f(\mathbf{w}') \geq f(\mathbf{w}) + \langle \mathbf{g}, \mathbf{w}' - \mathbf{w} \rangle\}$$

An important property of subdifferential set of a function ∂f is that it is closed and convex set, even for non-convex functions which is straightforward to verify by following the definition. Moreover, the subdifferential is also always nonempty for convex functions. This is a consequence of the supporting hyperplane theorem, which states that at any point on the boundary of a convex set, there exists at least one supporting hyperplane. Since the epigraph of a convex function is a convex set, we can apply the supporting hyperplane theorem to the set of points $(\mathbf{w}; f(\mathbf{w}))$, which are exactly the boundary points of the epigraph. The subdifferential of a differentiable function contains only single element which is the gradient of function at that point, i.e., $\partial f(\mathbf{w}) = \{\nabla f(\mathbf{w})\}$.

We are now prepared to introduce the concept of Lipschitz continuity, designed to measure change of function values versus change in the independent variable for a general function $f(\cdot)$.

Definition A.9 (Lipschitzness)**.** *A function $f : \mathcal{W} \mapsto \mathbb{R}$ is ρ-Lipschitz over the set $\mathcal{W}$ if for every $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$ we have that $|f(\mathbf{w}) - f(\mathbf{w}')| \leq \rho||\mathbf{w} - \mathbf{w}'||$.*

The following lemma shows a result on the Lipschitz continuity of a function which is defined as the maximum of other Lipschitz continuous functions.

Lemma A.10. *Assume that functions $f_i : \mathcal{W} \mapsto \mathbb{R}, i \in [m]$ are Lipschitz continuous with constant ρ. Then, function $f(\mathbf{w}) = \max_{i\in[m]} f_i(\mathbf{w})$ is Lipschitz continuous with constant ρ, that is,*

$$|f(\mathbf{w}) - f(\mathbf{w}')| \leq \rho\|\mathbf{w} - \mathbf{w}'\| \text{ for any } \mathbf{w}, \mathbf{w}' \in \mathcal{W}.$$

Proof. First, we rewrite $f(\mathbf{w}) = \max_{i\in[m]} f_i(\mathbf{w})$ as $f(\mathbf{w}) = \max_{\alpha\in\Delta_m} \sum_{i=1}^m \alpha_i f_i(\mathbf{w})$ where Δ_m is the m-simplex, that is, $\Delta_m = \{\alpha \in \mathbb{R}_+^m; \sum_{i=1}^m \alpha_i = 1\}$. Then, we have

$$\begin{aligned}
|f(\mathbf{w}) - f(\mathbf{w}')| &= \left|\max_{\alpha\in\Delta_m} \sum_{i=1}^m \alpha_i f_i(\mathbf{w}) - \max_{\alpha\in\Delta_m} \sum_{i=1}^m \alpha_i f_i(\mathbf{w}')\right| \\
&\leq \max_{\alpha\in\Delta_m} \left|\sum_{i=1}^m \alpha_i f_i(\mathbf{w}) - \sum_{i=1}^m \alpha_i f_i(\mathbf{w}')\right| \\
&\leq \max_{\alpha\in\Delta_m} \sum_{i=1}^m \alpha_i \left|f_i(\mathbf{w}) - f_i(\mathbf{w}')\right| \leq \rho\|\mathbf{w} - \mathbf{w}'\|,
\end{aligned}$$

where the last inequality follows from the Lipschitz continuity of $f_i(\mathbf{w}), i \in [m]$. □

Definition A.11 (Smoothness). *A differentiable function $f : \mathcal{W} \mapsto \mathbb{R}$ is said to be β-smooth with respect to a norm $\|\cdot\|$, if it holds that*

$$f(\mathbf{w}) \leq f(\mathbf{w}') + \langle \nabla f(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle + \frac{\beta}{2}\|\mathbf{w} - \mathbf{w}'\|^2, \ \forall\, \mathbf{w}, \mathbf{w}' \in \mathcal{W}. \tag{A.1}$$

The following result is an important property of smooth functions which has been utilized in the proof of few results in the thesis.

Lemma A.12 (Self-bounding Property). *For any β-smooth non-negative function $f : \mathbb{R} \to \mathbb{R}$, we have $|f'(w)| \leq \sqrt{4\beta f(w)}$*

As a simple proof, first from the smoothness assumption, by setting $w_1 = w_2 - \frac{1}{\beta} f'(w_2)$ in (A.1) and rearranging the terms we obtain $f(w_2) - f(w_1) \geq \frac{1}{2\beta}|f'(w_2)|^2$. On the other hand, from the convexity of loss function we have $f(w_1) \geq f'(w_2) + \langle f'(w_1), w_1 - w_2 \rangle$. Combining these inequalities and considering the fact that the function is non-negative gives the desired inequality.

Definition A.13 (Strong Convexity). *A function $f(\mathbf{w})$ is said to be α-strongly convex w.r.t a norm $\|\cdot\|$, if there exists a constant $\alpha > 0$ (often called the modulus of strong convexity) such that, for any $\lambda \in [0, 1]$ and for all $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$, it holds that*

$$f(\lambda \mathbf{w} + (1-\lambda)\mathbf{w}') \leq \alpha f(\mathbf{w}) + (1-\lambda) f(\mathbf{w}') - \frac{1}{2}\lambda(1-\lambda)\alpha\|\mathbf{w} - \mathbf{w}'\|^2.$$

If $f(\mathbf{w})$ is twice differentiable, then an equivalent definition of strong convexity is $\nabla^2 f(\mathbf{w}) \succeq \alpha \mathbf{I}$ which indicates that the smallest eigenvalue of the Hessian of $f(\mathbf{w})$ is uniformly lower bounded by α everywhere. When $f(\mathbf{w})$ is differentiable, the strong convexity is equivalent to

$$f(\mathbf{w}) \geq f(\mathbf{w}') + \langle \nabla f(\mathbf{w}'), \mathbf{w} - \mathbf{w}' \rangle + \frac{\alpha}{2}\|\mathbf{w} - \mathbf{w}'\|^2, \; \forall \, \mathbf{w}, \mathbf{w}' \in \mathcal{W}.$$

An important property of strongly convex functions that we used in the proof of few results in the following:

Lemma A.14. *Let $f(\mathbf{w})$ be a α-strongly convex function over the domain $\mathcal{W}$, and $\mathbf{w}^* = \arg\min_{\mathbf{w}\in\mathcal{W}} f(\mathbf{w})$. Then, for any $\mathbf{w} \in \mathcal{W}$, we have*

$$f(\mathbf{w}) - f(\mathbf{w}^*) \geq \frac{\alpha}{2}\|\mathbf{w} - \mathbf{w}^*\|^2. \tag{A.2}$$

Definition A.15 (Dual Norm)**.** *Let $\|\cdot\|$ be any norm on $\mathbb{R}^d$. Its dual norm denoted by $\|\cdot\|_*$ is defined by*

$$\|\mathbf{w}'\|_* = \sup_{\mathbf{w}\in\mathbb{R}^d} \langle \mathbf{w}', \mathbf{w} \rangle - \|\mathbf{w}\|$$

An equivalent definition is $\|\mathbf{w}'\|_* = \sup_{\mathbf{w}\in\mathbb{R}^d}\{\langle \mathbf{w}, \mathbf{w}' \rangle | \|\mathbf{w}\| \leq 1\}$. If $p, q \in [1, \infty]$ satisfy $1/p + 1/q = 1$, then the ℓ_p and ℓ_q norms are dual to each other.

Definition A.16 (Fenchel Conjugate [1])**.** *The Fenchel conjugate of a function $f : \mathcal{W} \mapsto \mathbb{R}$ is defined as*

$$f^*(\mathbf{v}) = \sup_{\mathbf{w}\in\mathcal{W}} \langle \mathbf{w}, \mathbf{v} \rangle - f(\mathbf{w}).$$

An immediate result of above definition is the Fenchel-Young inequality stating that for any $\mathbf{w}$ and $\mathbf{v}$ we have that $\langle \mathbf{w}, \mathbf{v} \rangle \leq f(\mathbf{w}) + f^*(\mathbf{v})$.

Definition A.17 (Bregman Divergence)**.** *Let $\Phi : \mathcal{W} \to \mathbb{R}$ be a continuously-differentiable*

[1] Also known as the convex conjugate or Legendre-Fenchel transformation

real-valued and strictly convex function defined on a closed convex set $\mathcal{W}$. The Bregman divergence between $\mathbf{w}$ *and* $\mathbf{w}'$ *is the difference at* $\mathbf{w}$ *between* f *and a linear approximation around* $\mathbf{w}'$

$$\mathsf{B}_{\Phi}(\mathbf{w}, \mathbf{w}') = \Phi(\mathbf{w}) - \Phi(\mathbf{w}') - \langle \mathbf{w} - \mathbf{w}', \nabla\Phi(\mathbf{w}') \rangle.$$

For example the squared Euclidean distance $\mathsf{B}(\mathbf{w}, \mathbf{w}') = \|\mathbf{w} - \mathbf{w}'\|^2$ is the canonical example of a Bregman distance, generated by the convex function $\Phi(\mathbf{w}) = \|\mathbf{w}\|^2$. The entropy function $\Phi(\mathbf{p}) = \sum_i p_i \log p_i - \sum p_i$ gives rises to the generalized Kullback-Leibler divergence as $\mathsf{B}_{\mathrm{KL}}(\mathbf{p}, \mathbf{q}) = \sum p_i \log \frac{p_i}{q_i} - \sum p_i + \sum q_i$.

A.2 Concentration Inequalities

This appendix is a quick excursion into concentration of measure. We consider some important concentration results used in the proofs given in this thesis. Concentration inequalities give probability bounds for a random variable to be concentrated around its mean (e.g., see [49] , [30] and [31] for a through discussion and derivation of these inequalities).

Lemma A.18 (Markov's Inequality)**.** *For a non-negative random variable X and $t > 0$,*

$$\mathbb{P}[X \geq t] \leq \frac{\mathbb{E}[X]}{t}$$

The upside of Markov's inequality is that it does not need almost no assumptions about the random variable, but the downside is that it only gives very weak bounds.

Lemma A.19 (Hoeffding's Lemma)**.** *Let X be any real-valued random variable with expected* *value* $\mathbb{E}[X] = 0$ *and such that* $X \in [a, b]$ *almost surely. Then, for all* $\lambda \in \mathbb{R}$*,*

$$\mathbb{E}\left[e^{\lambda X}\right] \leq \exp\left(\frac{\lambda^2(b-a)^2}{8}\right).$$

Theorem A.20 (Hoeffding's Inequality)**.** *Let* $X_1, \cdots, X_n$ *be a sequence of i.i.d random variables. Assume* $X_i \in [a_i, b_i]$ *and let* $S = X_1 + X_2 + \cdots + X_n$*. Then,*

$$\mathbb{P}(S - \mathbb{E}[S] \geq t) \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right),$$
$$\mathbb{P}(|S - \mathbb{E}[S]| \geq t) \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

The following result extends Hoeffding's inequality to more general functions $f(x_1, x_2, \cdots, x_n)$.

Theorem A.21 (McDiarmid's Inequality)**.** *Let* $X_1, \cdots, X_n$ *be independent real-valued random variables. Suppose that*

$$\sup_{x_1, x_2, \cdots, x_n, x_i'} \left|f(x_1, \cdots, x_{i-1}, x_i, x_{i+1}, \cdots, x_n) - f(x_1, \cdots, x_{i-1}, x_i', x_{i+1}, \cdots, x_n)\right| \leq c_i,$$

for $i = 1, 2, \cdots, n$*. Then,*

$$\mathbb{P}\left(|f(X_1, X_2, \cdots, X_n) - \mathbb{E}\left[f(X_1, X_2, \cdots, X_n)\right]| \geq t\right) \leq 2\exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right)$$

Hoeffding's inequality does not use any information about the random variables except the fact that they are bounded. If the variance of $X_i, i \in [n]$ is small, then we can get a sharper inequality from Bennett's inequality and its simplified version in Bernstein's Inequality.

Theorem A.22 (Bennett's Inequality)**.** *Let $X_1, \cdots, X_n$ be a sequence of i.i.d random variables. Assume $\mathbb{E}[X_i] = 0$, $\mathbb{E}[X_i^2] = \sigma^2$, and $|X_i| \leq M, i \in [n]$. Then,*

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i \geq t\right) \leq \exp\left(\frac{-n\sigma^2}{M^2}\phi\left(\frac{tM}{n\sigma^2}\right)\right),$$

where $\phi(x) = (1+x)\log(1+x) - x$.

Theorem A.23 (Bernstein's Inequality)**.** *Let $X_1, \ldots, X_n$ be independent zero-mean $\mathbb{E}[X_i] = 0, i \in [n]$ random variables. Suppose that $|X_i| \leq M$ almost surely, for all $i \in [n]$. Then, for all positive t,*

$$\mathbb{P}\left(\sum_{i=1}^{n} X_i > t\right) \leq \exp\left(-\frac{t^2}{2\left(\sum \mathbb{E}\left[X_j^2\right] + \frac{1}{3}Mt\right)}\right).$$

The next result is an extension of Hoeffding's inequality to martingales with zero-mean and bounded increments which is due to [10]. Before stating the Hoeffding-Azuma's inequality, we need few definitions. A sequence of random variables $(X_i)_{i\in\mathbb{N}}$ on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$ is a martingale difference sequence with respect to a filtration $(\mathcal{F}_i)_{i\in\mathbb{N}}$ if and only if, or all $i \geq 1$, each random variable X_i is $\mathcal{F}_i$-measurable and satisfies,

$$\mathbb{E}[X_i|\mathcal{F}_{i-1}] = 0.$$

Theorem A.24 (Hoeffding-Azuma Inequality)**.** *Let $X_1, X_2, \ldots$ be a martingale difference sequence with respect to a filtration $\mathcal{F} = (\mathcal{F}_i)_{i\in\mathbb{N}}$. Assume that for all $i \geq 1$, there exists a*

$\mathcal{F}_{i-1}$-measurable random variable A_i and a non-negative constant c_i such that $X_i \in [A_i, A_i + c_i]$. Then, the martingale S_n defined by $S_n = \sum_{i=1}^n X_i$, satisfies for any $t > 0$,

$$\mathbb{P}[S_n > t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right).$$

The Hoeffding-Azuma's inequality indicates that S_n is sub-gaussian with variance $\sum_{i=1}^n c_i^2/4$, i.e.,

$$\mathbb{P}\left[\max_{1\leq i\leq n} S_i \geq t\right] \leq \exp\left(\frac{-2t^2}{\sum_{i=1}^n c_i^2}\right).$$

The following inequality which is due to Freedman [57] extends Bernstein's result to the case of discrete-time martingales with bounded jumps (a.k.a Freedman's inequality). This result demonstrates that a martingale exhibits normal-type concentration near its mean value on a scale determined by the predictable quadratic variation of the sequence.

Theorem A.25 (Bernstein's Inequality for Martingales)**.** *Let $X_1, \ldots, X_n$ be a bounded martingale difference sequence with respect to the filtration $\mathcal{F} = (\mathcal{F}_i)_{1\leq i\leq n}$ and with $|X_i| \leq M$. Let*

$$S_i = \sum_{j=1}^i X_j$$

be the associated martingale. Denote the sum of the conditional variances by

$$\Sigma_n^2 = \sum_{i=1}^n \mathbb{E}\left[X_i^2|\mathcal{F}_{i-1}\right],$$

Then for all constants t, $\nu > 0$,

$$\mathbb{P}\left[\max_{i=1,\ldots,n} S_i > t \text{ and } \Sigma_n^2 \leq \nu\right] \leq \exp\left(-\frac{t^2}{2(\nu + Mt/3)}\right),$$

and therefore,

$$\mathbb{P}\left[\max_{i=1,\ldots,n} S_i > \sqrt{2\nu t} + \frac{\sqrt{2}}{3}Mt \text{ and } \Sigma_n^2 \leq \nu\right] \leq e^{-t}.$$

A.3 Miscellanea

A.3.1 Extra-gradient Lemma

Lemma A.26. *Let $\Phi(\mathbf{z})$ be a α-strongly convex function with respect to the norm $\|\cdot\|$, whose dual norm is denoted by $\|\cdot\|_*$, and $\mathsf{B}(\mathbf{w},\mathbf{z}) = \Phi(\mathbf{w}) - (\Phi(\mathbf{z}) + (\mathbf{w}-\mathbf{z})^\top \Phi'(\mathbf{z}))$ be the Bregman distance induced by function $\Phi(\mathbf{w})$. Let Z be a convex compact set, and $U \subseteq Z$ be convex and closed. Let $\mathbf{z} \in Z$, $\gamma > 0$, Consider the points,*

$$\mathbf{w} = \arg\min_{\mathbf{u}\in U} \gamma \mathbf{u}^\top \xi + \mathsf{B}(\mathbf{u},\mathbf{z}), \tag{A.3}$$

$$\mathbf{z}_+ = \arg\min_{\mathbf{u}\in U} \gamma \mathbf{u}^\top \zeta + \mathsf{B}(\mathbf{u},\mathbf{z}), \tag{A.4}$$

then for any $\mathbf{u} \in U$, we have

$$\gamma\zeta^\top(\mathbf{w}-\mathbf{u}) \leq \mathsf{B}(\mathbf{u},\mathbf{z}) - \mathsf{B}(\mathbf{u},\mathbf{z}_+) + \frac{\gamma^2}{\alpha}\|\xi-\zeta\|_*^2 - \frac{\alpha}{2}[\|\mathbf{w}-\mathbf{z}\|^2 + \|\mathbf{w}-\mathbf{z}_+\|^2].$$

Proof. By using the definition of Bregman distance $\mathsf{B}(\mathbf{u},\mathbf{z})$, we can write two updates in (A.3) as

$$
\begin{aligned}
\mathbf{w} &= \arg\min_{\mathbf{u}\in U} \mathbf{u}^\top(\gamma\xi - \Phi'(\mathbf{z})) + \Phi(\mathbf{u}),\\
\mathbf{z}_+ &= \arg\min_{\mathbf{u}\in U} \mathbf{u}^\top(\gamma\zeta - \Phi'(\mathbf{z})) + \Phi(\mathbf{u}),
\end{aligned}
\tag{A.5}
$$

by the first oder optimality condition, we have

$$
\begin{aligned}
&(\mathbf{u}-\mathbf{w})^\top(\gamma\xi - \Phi'(\mathbf{z}) + \Phi'(\mathbf{w})) \geq 0, \forall \mathbf{u}\in U,\\
&(\mathbf{u}-\mathbf{z}_+)^\top(\gamma\zeta - \Phi'(\mathbf{z}) + \Phi'(\mathbf{z}_+)) \geq 0, \forall \mathbf{u}\in U
\end{aligned}
\tag{A.6}
$$

Applying (A.6) with $\mathbf{u} = \mathbf{z}_+$ and $\mathbf{u} = \mathbf{w}$, we get

$$
\begin{aligned}
&\gamma(\mathbf{w}-\mathbf{z}_+)^\top\xi \leq (\Phi'(\mathbf{z}) - \Phi'(\mathbf{w}))^\top(\mathbf{w}-\mathbf{z}_+),\\
&\gamma(\mathbf{z}_+-\mathbf{w})^\top\zeta \leq (\Phi'(\mathbf{z}) - \Phi'(\mathbf{z}_+))^\top(\mathbf{z}_+-\mathbf{w}).
\end{aligned}
\tag{A.7}
$$

Summing up the two inequalities, we have

$$
\gamma(\mathbf{w}-\mathbf{z}_+)^\top(\xi-\zeta) \leq (\Phi'(\mathbf{z}_+) - \Phi'(\mathbf{w}))^\top(\mathbf{w}-\mathbf{z}_+).
\tag{A.8}
$$

Then

$$
\begin{aligned}
\gamma\|\xi-\zeta\|_*\|\mathbf{w}-\mathbf{z}_+\| &\geq -\gamma(\mathbf{w}-\mathbf{z}_+)^\top(\xi-\zeta) \geq (\Phi'(\mathbf{z}_+) - \Phi'(\mathbf{w}))^\top(\mathbf{z}_+-\mathbf{w})\\
&\geq \alpha\|\mathbf{z}_+-\mathbf{w}\|^2.
\end{aligned}
\tag{A.9}
$$

where in the last inequality, we use the strong convexity of $\Phi(\mathbf{w})$.

$$
\begin{aligned}
\mathsf{B}(\mathbf{u},\mathbf{z}) - \mathsf{B}(\mathbf{u},\mathbf{z}_+) &= \Phi(\mathbf{z}_+) - \Phi(\mathbf{z}) + (\mathbf{u}-\mathbf{z}_+)^\top \Phi'(\mathbf{z}_+) - (\mathbf{u}-\mathbf{z})^\top \Phi'(\mathbf{z}) \\
&= \Phi(\mathbf{z}_+) - \Phi(\mathbf{z}) + (\mathbf{u}-\mathbf{z}_+)^\top \Phi'(\mathbf{z}_+) - (\mathbf{u}-\mathbf{z}_+)^\top \Phi'(\mathbf{z}) - (\mathbf{z}_+-\mathbf{z})^\top \Phi'(\mathbf{z}) \\
&= \Phi(\mathbf{z}_+) - \Phi(\mathbf{z}) - (\mathbf{z}_+-\mathbf{z})^\top \Phi'(\mathbf{z}) + (\mathbf{u}-\mathbf{z}_+)^\top (\Phi'(\mathbf{z}_+) - \Phi'(\mathbf{z})) \\
&= \Phi(\mathbf{z}_+) - \Phi(\mathbf{z}) - (\mathbf{z}_+-\mathbf{z})^\top \Phi'(\mathbf{z}) + (\mathbf{u}-\mathbf{z}_+)^\top (\gamma\zeta + \Phi'(\mathbf{z}_+) - \Phi'(\mathbf{z})) - (\mathbf{u}-\mathbf{z}_+)^\top \gamma\zeta \\
&\geq \Phi(\mathbf{z}_+) - \Phi(\mathbf{z}) - (\mathbf{z}_+-\mathbf{z})^\top \Phi'(\mathbf{z}) - (\mathbf{u}-\mathbf{z}_+)^\top \gamma\zeta \\
&= \underbrace{\Phi(\mathbf{z}_+) - \Phi(\mathbf{z}) - (\mathbf{z}_+-\mathbf{z})^\top \Phi'(\mathbf{z}) - (\mathbf{w}-\mathbf{z}_+)^\top \gamma\zeta}_{\epsilon} + (\mathbf{w}-\mathbf{u})^\top \gamma\zeta,
\end{aligned}
\tag{A.10}
$$

where the inequality follows from (A.6). We proceed by bounding ϵ as:

$$
\begin{aligned}
\epsilon =& \Phi(\mathbf{z}_+) - \Phi(\mathbf{z}) - (\mathbf{z}_+-\mathbf{z})^\top \Phi'(\mathbf{z}) - (\mathbf{w}-\mathbf{z}_+)^\top \gamma\zeta \\
=& \Phi(\mathbf{z}_+) - \Phi(\mathbf{z}) - (\mathbf{z}_+-\mathbf{z})^\top \Phi'(\mathbf{z}) - (\mathbf{w}-\mathbf{z}_+)^\top \gamma(\zeta-\xi) - (\mathbf{w}-\mathbf{z}_+)^\top \gamma\xi \\
=& \Phi(\mathbf{z}_+) - \Phi(\mathbf{z}) - (\mathbf{z}_+-\mathbf{z})^\top \Phi'(\mathbf{z}) - (\mathbf{w}-\mathbf{z}_+)^\top \gamma(\zeta-\xi) \\
& + (\mathbf{z}_+-\mathbf{w})^\top (\gamma\xi - \Phi'(\mathbf{z}) + \Phi'(\mathbf{w})) - (\mathbf{z}_+-\mathbf{w})^\top (\Phi'(\mathbf{w}) - \Phi'(\mathbf{z})) \\
\geq& \Phi(\mathbf{z}_+) - \Phi(\mathbf{z}) - (\mathbf{z}_+-\mathbf{z})^\top \Phi'(\mathbf{z}) - (\mathbf{w}-\mathbf{z}_+)^\top \gamma(\zeta-\xi) - (\mathbf{z}_+-\mathbf{w})^\top (\Phi'(\mathbf{w}) - \Phi'(\mathbf{z})) \\
=& \Phi(\mathbf{z}_+) - \Phi(\mathbf{z}) - (\mathbf{w}-\mathbf{z})^\top \Phi'(\mathbf{z}) - (\mathbf{w}-\mathbf{z}_+)^\top \gamma(\zeta-\xi) - (\mathbf{z}_+-\mathbf{w})^\top \Phi'(\mathbf{w}) \\
=& \Big[\Phi(\mathbf{z}_+) - \Phi(\mathbf{w}) - (\mathbf{z}_+-\mathbf{w})^\top \Phi'(\mathbf{w})\Big] + \Big[\Phi(\mathbf{w}) - \Phi(\mathbf{z}) - (\mathbf{w}-\mathbf{z})^\top \Phi'(\mathbf{z})\Big] \\
& - (\mathbf{w}-\mathbf{z}_+)^\top \gamma(\zeta-\xi) \\
\geq& \frac{\alpha}{2}\|\mathbf{w}-\mathbf{z}_+\|^2 + \frac{\alpha}{2}\|\mathbf{w}-\mathbf{z}\|^2 - \gamma\|\mathbf{w}-\mathbf{z}_+\|\|\zeta-\xi\|_* \\
\geq& \frac{\alpha}{2}\{\|\mathbf{w}-\mathbf{z}_+\|^2 + \|\mathbf{w}-\mathbf{z}\|^2\} - \frac{\gamma^2}{\alpha}\|\zeta-\xi\|_*^2,
\end{aligned}
\tag{A.11}
$$

where the first inequality follows from (A.6), the second inequality follows from the strong convexity of $\Phi(\mathbf{w})$, and the last inequality follows from (A.10). Combining the above results, we have

$$\gamma(\mathbf{w}-\mathbf{u})^\top\zeta \leq \mathsf{B}(\mathbf{u},\mathbf{z}) - \mathsf{B}(\mathbf{u},\mathbf{z}_+) + \frac{\gamma^2}{\alpha}\|\zeta-\xi\|_*^2 - \frac{\alpha}{2}\{\|\mathbf{w}-\mathbf{z}_+\|^2 + \|\mathbf{w}-\mathbf{z}\|^2\}. \tag{A.12}$$

□

A.3.2 Stochastic Mirror Descent for Smooth Losses

In this subsection we provide a tight analysis of stochastic mirror decent algorithm for smooth losses. We will show that the convergence rate of the mirror descent algorithm [119] for stochastic optimization of both non-smooth and smooth functions is dominated by an $O(1/\sqrt{T})$ convergence rate. We consider the nonlinear projected subgradient formulation of the mirror descent from [17] which iteratively updates the solution by:

$$\mathbf{w}_{t+1} = \arg\min_{\mathbf{w}\in\mathcal{W}} \left\{\langle \mathbf{w}, \mathbf{g}_t\rangle + \frac{1}{\eta_t}\mathsf{B}_\Phi(\mathbf{w},\mathbf{w}_t)\right\}, \tag{A.13}$$

where $\mathbf{g}_t$ is a (sub)gradient of $f(\mathbf{w})$ at point $\mathbf{w}_t$ and $\mathsf{B}_\Phi(\cdot,\cdot)$ is the Bregmen diveregnce defined with strongly convex function $\Phi(\cdot)$. Define the average solution as $\widehat{\mathbf{w}}_T = \sum_{t=1}^T \mathbf{w}_t/T$ and let $\mathbf{g}(\mathbf{w},\xi)$ denote a stochastic gradient of function at point $\mathbf{w}$, i.e., $\mathbb{E}[\mathbf{g}(\mathbf{w},\xi)] = \nabla f(\mathbf{w})$. We state few standard assumptions we make in our analysis about the stochastic optimization problem.

A1. (Lipschitz Continuity) $\mathbb{E}[\|\mathbf{g}(\mathbf{w},\xi)\|_*^2] \leq G^2$

A2. (Bounded Variance) $\mathbb{E}[\|\nabla f(\mathbf{w}) - \mathbf{g}(\mathbf{w}, \xi)\|] \leq \sigma^2$.

A3. (Smoothness) The function f has L Lipschitz continuous gradient.

A4. (Compactness) $\mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}) \leq R^2$.

The proof is extensively uses the following key inequalities:

(1) Optimality condition

$$\langle \mathbf{w} - \mathbf{w}_{t+1}, \mathbf{g}_t + \frac{1}{\eta_t}\nabla\Phi(\mathbf{w}_{t+1}) - \frac{1}{\eta_t}\nabla\Phi(\mathbf{w}_t)\rangle \geq 0$$

(2) Generalized inequality

$$\mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_t) - \mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_{t+1}) - \mathsf{B}_\Phi(\mathbf{w}_{t+1}, \mathbf{w}_t) = \langle \mathbf{w}_* - \mathbf{w}_{t+1}, \nabla\Phi(\mathbf{w}_{t+1}) - \nabla\Phi(\mathbf{w}_t)\rangle$$

(3) Fenchel-Yang inequality applied to the conjugate pair $\frac{1}{2}\|\cdot\|^2$ and $\frac{1}{2}\|\cdot\|_*^2$ yields

$$\langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_{t+1}\rangle \leq \frac{\eta_t}{2}\|\mathbf{g}_t\|_*^2 + \frac{1}{2\eta_t}\|\mathbf{w}_t - \mathbf{w}_{t+1}\|^2$$

(4)

$$\mathsf{B}_\Phi(\mathbf{w}, \mathbf{y}) \geq \frac{1}{2}\|\mathbf{w} - \mathbf{y}\|^2$$

(5) From smoothness assumption we have

$$f(\mathbf{w}') - f(\mathbf{w}) - \langle \nabla f(\mathbf{w}), \mathbf{w}' - \mathbf{w}\rangle \leq \frac{L}{2}\|\mathbf{w}' - \mathbf{w}\|^2$$

The analysis of the convergence rate of basic mirror descent algorithm follows standard techniques in convex optimization and for completeness we provide a proof here following [17].

Lemma A.27. *We have*

$$f(\mathbf{w}_t) - f(\mathbf{w}_*) \leq \frac{\eta_t}{2}||\mathbf{g}_t||_*^2 + \frac{1}{\eta_t}\left[\mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_t) - \mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_{t+1})\right]$$

Proof. From the convexity of $f(\mathbf{w})$ and by setting $\zeta_t = \nabla f(\mathbf{w}_t) - \mathbf{g}_t$ we have

$$\begin{aligned}
& f(\mathbf{w}_t) - f(\mathbf{w}_*) \\
& \leq \langle \nabla f(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_* \rangle \\
& \leq \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_* \rangle + \langle \nabla f(\mathbf{w}_t) - \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_* \rangle \\
& \overset{(1)}{\leq} \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_* \rangle + \langle \mathbf{w}_* - \mathbf{w}_{t+1}, \mathbf{g}_t + \frac{1}{\eta_t}\nabla\Phi(\mathbf{w}_{t+1}) - \frac{1}{\eta_t}\nabla\Phi(\mathbf{w}_t) \rangle + \langle \zeta_t, \mathbf{w}_t - \mathbf{w}_* \rangle \\
& = \langle \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_{t+1} \rangle + \frac{1}{\eta_t}\langle \mathbf{w}_* - \mathbf{w}_{t+1}, \nabla\Phi(\mathbf{w}_{t+1}) - \nabla\Phi(\mathbf{w}_t) \rangle + \langle \zeta_t, \mathbf{w}_t - \mathbf{w}_* \rangle \\
& \overset{(2)}{\leq} \frac{\eta_t}{2}\|\mathbf{g}_t\|_*^2 + \frac{1}{2\eta_t}\|\mathbf{w}_t - \mathbf{w}_{t+1}\|^2 + \frac{1}{\eta_t}[\mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_t) - \mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_{t+1}) \\
& - \mathsf{B}_\Phi(\mathbf{w}_{t+1}, \mathbf{w}_t)] + \langle \zeta_t, \mathbf{w}_t - \mathbf{w}_* \rangle \\
& \overset{(3)}{\leq} \frac{\eta_t}{2}\|\mathbf{g}_t\|_*^2 + \frac{1}{\eta_t}[\mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_t) - \mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_{t+1})] + \langle \zeta_t, \mathbf{w}_t - \mathbf{w}_* \rangle
\end{aligned}$$

□

Theorem A.28. *By setting leaning rate as $\eta_t = \frac{R}{G\sqrt{t}}$ we have the following rate for the convergence of stochastic mirror descent for non-smooth objectives:*

$$\mathbb{E}[f(\widehat{\mathbf{w}}_T)] - f(\mathbf{w}_*) \leq O(\frac{RG}{\sqrt{T}})$$

Proof. Summing up Lemma 1 for all iterations we get

$$\begin{aligned}
\sum_{t=1}^{T} f(\mathbf{w}_t) - f(\mathbf{w}_*) &\leq \frac{1}{\eta_1}\mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_1) + \sum_{t=2}^{T} \mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_t)(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) \\
&+ \sum_{t=1}^{T} \frac{\eta_t}{2}\|\mathbf{g}_t\|_*^2 + \sum_{t=1}^{T} \langle \zeta_t, \mathbf{w}_t - \mathbf{w}_* \rangle \\
&\leq \frac{R^2}{\eta_T} + \frac{G^2}{2}\sum_{t=1}^{T} \eta_t \\
&= O(RG\sqrt{T}),
\end{aligned}$$

where the second inequality follows since each $\mathbf{w}_t$ is a deterministic function of $\xi_1, \cdots, \xi_{t-1}$, so $\mathbb{E}[\langle \nabla f(\mathbf{w}_t) - \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_* \rangle | \xi_1, \cdots, \xi_{t-1}] = 0$ □

We generalize the proof for smooth functions.

Theorem A.29. *By setting $\eta_t = \frac{1}{L+\beta_t}$ where $\beta_t = (\sigma/R)\sqrt{t}$, the following holds for the convergence rate of stochastic mirror descent for smooth functions:*

$$\mathbb{E}[f(\widehat{\mathbf{w}}_T)] - f(\mathbf{w}_*) \leq O(\frac{LR^2}{T} + \frac{\sigma R}{\sqrt{T}})$$

Proof.

$$\begin{aligned}
f(\mathbf{w}_t) - f(\mathbf{w}_*) &\leq \langle \nabla f(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_* \rangle \\
&\leq \langle \nabla f(\mathbf{w}_t), \mathbf{w}_t - \mathbf{w}_{t+1} \rangle + \langle \nabla f(\mathbf{w}_t), \mathbf{w}_{t+1} - \mathbf{w}_* \rangle \\
&\overset{(5)}{\leq} f(\mathbf{w}_t) - f(\mathbf{w}_{t+1}) + \frac{L}{2}\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2 + \langle \nabla f(\mathbf{w}_t), \mathbf{w}_{t+1} - \mathbf{w}_* \rangle
\end{aligned}$$

By rearranging the terms we proceed as follows:

$$
\begin{aligned}
& f(\mathbf{w}_{t+1}) - f(\mathbf{w}_*) \\
& \leq \langle \nabla f(\mathbf{w}_t), \mathbf{w}_{t+1} - \mathbf{w}_* \rangle + \frac{L}{2}\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2 \\
& \leq \langle \mathbf{g}_t, \mathbf{w}_{t+1} - \mathbf{w}_* \rangle + \frac{L}{2}\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2 + \langle \zeta_t, \mathbf{w}_{t+1} - \mathbf{w}_* \rangle \\
& \overset{(1)}{\leq} \frac{1}{\eta_t}\langle \mathbf{w}_* - \mathbf{w}_{t+1}, \nabla\Phi(\mathbf{w}_{t+1}) - \nabla\Phi(\mathbf{w}_t)\rangle + \frac{L}{2}\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2 + \langle \zeta_t, \mathbf{w}_{t+1} - \mathbf{w}_* \rangle \\
& \overset{(2)}{\leq} \frac{1}{\eta_t}[\mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_t) - \mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_{t+1}) - \mathsf{B}_\Phi(\mathbf{w}_t, \mathbf{w}_{t+1})] + \frac{L}{2}\|\mathbf{w}_{t+1} - \mathbf{w}_t\|^2 + \langle \zeta_t, \mathbf{w}_{t+1} - \mathbf{w}_* \rangle \\
& \overset{(4)}{\leq} \frac{1}{\eta_t}[\mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_t) - \mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_{t+1})] - (L + \beta_t)\mathsf{B}_\Phi(\mathbf{w}_t, \mathbf{w}_{t+1}) + L\mathsf{B}_\Phi(\mathbf{w}_t, \mathbf{w}_{t+1}) \\
& + \langle \zeta_t, \mathbf{w}_{t+1} - \mathbf{w}_* \rangle \\
& = \frac{1}{\eta_t}[\mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_t) - \mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_{t+1})] - \beta_t\mathsf{B}_\Phi(\mathbf{w}_t, \mathbf{w}_{t+1}) + \langle \zeta_t, \mathbf{w}_{t+1} - \mathbf{w}_t \rangle + \langle \zeta_t, \mathbf{w}_t - \mathbf{w}_* \rangle \\
& \overset{(3)}{\leq} \frac{1}{\eta_t}[\mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_t) - \mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_{t+1})] + \frac{1}{2\beta_t}\|\zeta_t\|_*^2 + \langle \zeta_t, \mathbf{w}_t - \mathbf{w}_* \rangle
\end{aligned}
$$

Summing for all t yields

$$
\begin{aligned}
& \sum_{t=1}^{T} f(\mathbf{w}_{t+1}) - f(\mathbf{w}_*) \\
& \leq \frac{1}{\eta_1}\mathsf{B}_\Phi(\mathbf{w}_*, \mathbf{w}_1) + \sum_{t=2}^{T}\mathsf{B}_\Phi(\mathbf{w}_* - \mathbf{w}_t)(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) + \sum_{t=1}^{T}\frac{\|\zeta_t\|_*^2}{2\beta_t} + \sum_{t=1}^{T}\langle \zeta_t, \mathbf{w}_t - \mathbf{w}_* \rangle \\
& \leq \frac{R^2}{\eta_1} + R^2\sum_{t=2}^{T}(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}}) + \frac{\sigma^2}{2}\sum_{t=1}^{T}\frac{1}{\beta_t} + 0 \\
& \leq \frac{R^2}{\eta_T} + \frac{\sigma R}{2}\sqrt{T} = O(LR^2 + \sigma R\sqrt{T}),
\end{aligned}
$$

where in the second in equality we used the fact $\mathbb{E}[\langle \nabla f(\mathbf{w}_t) - \mathbf{g}_t, \mathbf{w}_t - \mathbf{w}_* \rangle | \xi_1, \cdots, \xi_{t-1}] = 0$. By averaging the solutions over all iterations and applying the Jensen's inequality we obtain the convergence rate claimed in the theorem.

□

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Jacob Abernethy, Peter L Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal strategies and minimax lower bounds for online convex games. In *Proceedings of the Nineteenth Annual Conference on Computational Learning Theory*, 2008.

[2] Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *COLT*, pages 263–274, 2008.

[3] Jacob Abernethy, Elad Hazan, and Alexander Rakhlin. Interior-point methods for full-information and bandit online learning. *IEEE Transactions on Information Theory*, 58(7):4164–4175, 2012.

[4] Alekh Agarwal, Peter L. Bartlett, Pradeep D. Ravikumar, and Martin J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *IEEE Transactions on Information Theory*, 58(5):3235–3249, 2012.

[5] Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *Proceedings of The 23rd Conference on Learning Theory (COLT)*, pages 28–40, 2010.

[6] Amit Agarwal, Elad Hazan, Satyen Kale, and Robert E. Schapire. Algorithms for portfolio management based on the newton method. In *Proceedings of the 23rd international conference on Machine learning*, pages 9–16, 2006.

[7] M. Anthony and P.L. Bartlett. *Neural network learning: Theoretical foundations.* Cambridge University Press, 1999.

[8] Sanjeev Arora, László Babai, Jacques Stern, and Z Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. In *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on*, pages 724–733. IEEE, 1993.

[9] Baruch Awerbuch and Robert D. Kleinberg. Adaptive routing with end-to-end feedback: distributed learning and geometric approaches. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, STOC '04, pages 45–53, New York, NY, USA, 2004. ACM.

[10] Kazuoki Azuma et al. Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal*, 19(3):357–367, 1967.

[11] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[12] Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine Learning*, 80(2-3):111–139, 2010.

[13] Peter L Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

[14] Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

[15] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3:463–482, 2003.

[16] Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[17] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175, 2003.

[18] Robert Bell and Thomas M Cover. Game-theoretic optimal portfolios. *Management Science*, 34(6):724–733, 1988.

[19] Shai Ben-David, David Loker, Nathan Srebro, and Karthik Sridharan. Minimizing the misclassification error rate using a surrogate convex loss. In *ICML*, 2012.

[20] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT*, 2009.

[21] Dimitri P. Bertsekas. *Nonlinear Programming.* Athena Scientific, 2nd edition, 1999.

[22] Dimitri P. Bertsekas, Angelia Nedic, and Asuman E. Ozdaglar. *Convex Analysis and Optimization.* Athena Scientific, 2003.

[23] David Blackwell. An analog of the minimax theorem for vector payoffs. *Pacific Journal of Mathematics*, 1(6), 1956.

[24] David Blackwell. Minimax vs. bayes prediction. *Probability in the Engineering and Informational Sciences*, 9(01):53–58, 1995.

[25] HD Block. The perceptron: A model for brain functioning. i. *Reviews of Modern Physics*, 34(1):123, 1962.

[26] Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989.

[27] Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis.* Cambridge University Press, 1998.

[28] Jonathan M Borwein and Adrian S Lewis. *Convex analysis and nonlinear optimization: theory and examples*, volume 3. Springer, 2010.

[29] Leon Bottou and Olivier Bousquet. The tradeoffs of large scale learning. In *NIPS*, pages 161–168, 2008.

[30] Stéphane Boucheron, Gábor Lugosi, and Olivier Bousquet. Concentration inequalities. In *Advanced Lectures on Machine Learning*, pages 208–240. Springer, 2004.

[31] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence.* Oxford University Press, 2013.

[32] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

[33] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[34] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[35] Augustin Cauchy. Méthode générale pour la résolution des systemes déquations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

[36] Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

[37] Nicolo Cesa-Bianchi, Yoav Freund, David Haussler, David P Helmbold, Robert E Schapire, and Manfred K Warmuth. How to use expert advice. *Journal of the ACM (JACM)*, 44(3):427–485, 1997.

[38] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games.* Cambridge University Press, 2006.

[39] Kamalika Chaudhuri, Yoav Freund, and Daniel J Hsu. A parameter-free hedging algorithm. In *Advances in neural information processing systems*, pages 297–305, 2009.

[40] Chao-Kai Chiang, Chia-Jung Lee, and Chi-Jen Lu. Beating bandits in gradually evolving worlds. In *COLT*, pages 210–227, 2013.

[41] Chao-Kai Chiang, Tianbao Yang, Chia-Jung Lee, Mehrdad Mahdavi, Chi-Jen Lu, Rong Jin, and Shenghuo Zhu. Online optimization with gradual variations. In *Conference on Learning Theory*, 2012.

[42] Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms*, 6(4), 2010.

[43] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[44] Andrew Cotter, Ohad Shamir, Nati Srebro, and Karthik Sridharan. Better mini-batch algorithms via accelerated gradient methods. In *NIPS*, pages 1647–1655, 2011.

[45] Thomas M. Cover. Behavior of sequential predictors of binary sequences. *Transactions of the Prague Conferences on Information Theory*, 1965.

[46] Varsha Dani, Thomas P. Hayes, and Sham Kakade. The price of bandit information for online optimization. In *NIPS*, 2007.

[47] Ofer Dekel, Ran Gilad-Bachrach, Ohad Shamir, and Lin Xiao. Optimal distributed online prediction using mini-batches. *The Journal of Machine Learning Research*, 13:165–202, 2012.

[48] Ofer Dekel and Yoram Singer. Data-driven online to batch conversions. In *NIPS*, 2005.

[49] Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms.* Cambridge University Press, 2009.

[50] John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

[51] John C. Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *ICML*, pages 272–279, 2008.

[52] A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant. A general lower bound on the number of examples needed for learning. *Information and Computation*, 82(3):247–261, 1989.

[53] Tim V Erven, Wouter M Koolen, Steven D Rooij, and Peter Grünwald. Adaptive hedge. In *Advances in Neural Information Processing Systems*, pages 1656–1664, 2011.

[54] Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *SODA*, pages 385–394, 2005.

[55] Sally Floyd and Manfred Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

[56] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics*, 3, 1956.

[57] David A Freedman. On tail probabilities for martingales. *the Annals of Probability*, pages 100–118, 1975.

[58] Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.

[59] Yoav Freund and Robert E Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29(1):79–103, 1999.

[60] Michael P Friedlander and Mark Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, 2012.

[61] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[62] Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization i: a generic algorithmic framework. *SIAM Journal on Optimization*, 22(4):1469–1492, 2012.

[63] Geoffrey J Gordon. Regret bounds for prediction problems. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 29–40. ACM, 1999.

[64] John Greenstadt. On the relative efficiencies of gradient methods. *Mathematics of Computation*, pages 360–367, 1967.

[65] James Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.

[66] Steve Hanneke. *Theoretical Foundations of Active Learning.* PhD thesis, 2009.

[67] Elad Hazan. Sparse approximate solutions to semidefinite programs. In *LATIN*, pages 306–316, 2008.

[68] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[69] Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *COLT*, pages 57–68, 2008.

[70] Elad Hazan and Satyen Kale. Better algorithms for benign bandits. In *SODA*, pages 38–47, 2009.

[71] Elad Hazan and Satyen Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. *Machine learning*, 80(2-3):165–188, 2010.

[72] Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. *COLT*, 19:421–436, 2011.

[73] Elad Hazan and Satyen Kale. Projection-free online learning. In *ICML*, 2012.

[74] Elad Hazan and Tomer Koren. Optimal algorithms for ridge and lasso regression with partially observed attributes. *CoRR*, 2011.

[75] Klaus-Uwe Hoffgen, Hans-Ulrich Simon, and Kevin S Vanhorn. Robust trainability of single neurons. *Journal of Computer and System Sciences*, 50(1):114–125, 1995.

[76] Anatoli Iouditski and Yuri Nesterov. Primal-dual subgradient methods for minimizing uniformly convex functions. available at http://hal.archives-ouvertes.fr/docs/00/50/89/33/PDF/Strong-hal.pdf, 2010.

[77] Martin Jaggi. *Sparse Convex Optimization Methods for Machine Learning*. PhD thesis, ETH Zurich, October 2011.

[78] Martin Jaggi and Marek Sulovský. A simple algorithm for nuclear norm regularized problems. In *ICML*, pages 471–478, 2010.

[79] Kevin G Jamieson, Robert D Nowak, and Benjamin Recht. Query complexity of derivative-free optimization. In *NIPS*, 2012.

[80] Wenxin Jiang. Process consistency for adaboost. *The Annals of Statistics*, 32(1):13–29, 2004.

[81] Anatoli Juditsky and Yuri Nesterov. Primal-dual subgradient methods for minimizing uniformly convex functions. Technical report, 2010.

[82] Sham M. Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *NIPS*, pages 793–800, 2008.

[83] Sham M. Kakade and Ambuj Tewari. On the generalization ability of online strongly convex programming algorithms. In *Advances in Neural Information Processing Systems 21*, pages 801–808, 2008.

[84] Adam Tauman Kalai, Adam R Klivans, Yishay Mansour, and Rocco A Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

[85] Adam Tauman Kalai and Ravi Sastry. The isotron algorithm: High-dimensional isotonic regression. In *COLT*, 2009.

[86] Adam Tauman Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71:291–307, October 2005.

[87] Michael J Kearns, Robert E Schapire, and Linda M Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

[88] J. Kivinen, A. J. Smola, and R. C. Williamson. Online Learning with Kernels. *IEEE Transactions on Signal Processing*, 52:2165–2176, 2004.

[89] Jyrki Kivinen and Manfred K. Warmuth. Additive versus exponentiated gradient updates for linear prediction. In *Proceedings of the 27th annual ACM symposium on Theory of computing*, Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, pages 209–218, New York, NY, USA, 1995. ACM.

[90] Jyrki Kivinen and Manfred K Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, 1997.

[91] Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *Information Theory, IEEE Transactions on*, 47(5):1902–1914, 2001.

[92] Vladimir Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems.* Lecture Notes in mathematics. Springer, 2011.

[93] Guanghui Lan. Efficient methods for stochastic composite optimization. Technical report, 2008.

[94] Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

[95] Wee Sun Lee, Peter L. Bartlett, and Robert C. Williamson. The importance of convexity in learning with squared loss. *IEEE Transactions on Information Theory*, 44(5):1974–1980, 1998.

[96] Qihang Lin, Xi Chen, and Javier Pena. A smoothing stochastic gradient method for composite optimization. *arXiv preprint arXiv:1008.5204*, 2010.

[97] Yi Lin. A note on margin-based loss functions in classification. *Statistics & probability letters*, 68(1):73–82, 2004.

[98] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

[99] Nick Littlestone and Manfred K Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994.

[100] Jun Liu and Jieping Ye. Efficient euclidean projections in linear time. In *ICML*, pages 83–90, 2009.

[101] Gábor Lugosi and Nicolas Vayatis. On the bayes-risk consistency of regularized boosting methods. *Annals of Statistics*, pages 30–55, 2004.

[102] Mehrdad Mahdavi and Rong Jin. Passive learning with target risk. *COLT*, 2013.

[103] Mehrdad Mahdavi and Rong Jin. Excess risk bounds for exponentially concave losses. *arXiv preprint arXiv:1401.4566*, 2014.

[104] Mehrdad Mahdavi, Rong Jin, and Tianbao Yang. Trading regret for efficiency: Online convex optimization with long term constraints. *Journal of Machine Learning Research*, 13:2503–2528, 2012.

[105] Mehrdad Mahdavi, Tianbao Yang, and Rong Jin. Stochastic convex optimization with multiple objectives. In *NIPS*, 2013.

[106] Mehrdad Mahdavi, Tianbao Yang, Rong Jin, Shenghuo Zhu, and Jinfeng Yi. Stochastic gradient descent with only one projection. In *NIPS*, pages 503–511, 2012.

[107] Mehrdad Mahdavi, Lijun Zhang, and Rong Jin. Mixed optimization for smooth functions. In *NIPS*, 2013.

[108] Mehrdad Mahdavi, Lijun Zhang, and Rong Jin. Binary excess risk for smooth convex surrogates. *arXiv preprint arXiv:1402.1792*, 2014.

[109] Enno Mammen and Alexandre B Tsybakov. Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808–1829, 1999.

[110] Shie Mannor and John N. Tsitsiklis. Online learning with constraints. In *COLT*, pages 529–543, 2006.

[111] Shie Mannor, John N. Tsitsiklis, and Jia Yuan Yu. Online learning with sample path constraints. *Journal of Machine Learning Research*, 10:569–590, 2009.

[112] David A McAllester. Some pac-bayesian theorems. In *Proceedings of the eleventh annual Conference on Computational learning theory*, pages 230–234. ACM, 1998.

[113] H. Brendan McMahan and Matthew J. Streeter. Open problem: Better bounds for online logistic regression. *Journal of Machine Learning Research - Proceedings Track*, 23:44.1–44.3, 2012.

[114] Hariharan Narayanan and Alexander Rakhlin. Random walk approach to regret minimization. In *NIPS*, pages 1777–1785, 2010.

[115] Arkadai Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609, 2009.

[116] Arkadi Nemirovski. Efficient methods in convex programming. Lecture Notes, Available at http://www2.isye.gatech.edu/ nemirovs, 1994.

[117] Arkadi Nemirovski. Prox-method with rate of convergence o(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. on Optimization*, 15:229–251, 2005.

[118] Arkadi S Nemirovski and Michael J Todd. Interior-point methods for optimization. *Acta Numerica*, 17:191–234, 2008.

[119] Arkadi Nemirovsky and David Borisovich Yudin. *Problem complexity and method efficiency in optimization.* Wiley- Interscience Series in Discrete Mathematics. John Wiley, 1983.

[120] Yurii Nesterov. A method of solving a convex programming problem with convergence rate o (1/k2). In *Soviet Mathematics Doklady*, volume 27, pages 372–376, 1983.

[121] Yurii Nesterov. *Introductory lectures on convex optimization: a basic course*, volume 87 of *Applied optimization.* Kluwer Academic Publishers, 2004.

[122] Yurii Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization*, 16(1):235–249, 2005.

[123] Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[124] Yurii Nesterov. Gradient methods for minimizing composite objective function. *Mathematical Programming*, 140(1), 2013.

[125] Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 449–456, 2012.

[126] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Random averages, combinatorial parameters, and learnability. In *NIPS*, pages 1984–1992, 2010.

[127] Aaditya Ramdas and Aarti Singh. Optimal stochastic convex optimization through the lens of active learning. In *ICML*, 2013.

[128] Nicolas Le Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 2672–2680, 2012.

[129] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

[130] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[131] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability and stability in the general learning setting. *COLT*, 2009.

[132] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Stochastic convex optimization. In *COLT*, 2009.

[133] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11:2635–2670, 2010.

[134] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML*, pages 807–814, 2007.

[135] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *JMLR*, 14:567599, 2013.

[136] Ohad Shamir. On the complexity of bandit and derivative-free stochastic convex optimization. In *Conference on Learning Theory*, 2013.

[137] Ohad Shamir and Tong Zhang. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. *ICML*, 2013.

[138] Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. Smoothness, low noise and fast rates. In *NIPS*, pages 2199–2207, 2010.

[139] Karthik Sridharan. Learning from an optimization viewpoint. PhD Thesis, 2012.

[140] Karthik Sridharan, Shai Shalev-Shwartz, and Nathan Srebro. Fast rates for regularized objectives. In *NIPS*, pages 1545–1552, 2008.

[141] Ingo Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *Information Theory, IEEE Transactions on*, 51(1):128–142, 2005.

[142] Eiji Takimoto and Manfred K. Warmuth. Path kernels and multiplicative updates. *Journal Machine Learnning Research*, 4:773–818, December 2003.

[143] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization, 2009.

[144] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[145] Vladimir N Vapnik. Statistical learning theory. 1998.

[146] Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

[147] Volodimir G Vovk. Aggregating strategies. In *Proc. Third Workshop on Computational Learning Theory*, pages 371–383. Morgan Kaufmann, 1990.

[148] David H Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996.

[149] Qiang Wu and Ding-Xuan Zhou. Svm soft margin classifiers: Linear programming versus quadratic programming. *Neural Computation*, 17(5):1160–1187, 2005.

[150] Tianbao Yang, Mehrdad Mahdavi, Rong Jin, and Shenghuo Zhu. Regret bounded by gradual variation for online convex optimization. *Journal of Machine Learning Research-Proceedings Track*, 23:6–1, 2012.

[151] Yiming Ying and Peng Li. Distance metric learning with eigenvalue optimization. *JMLR.*, 13:1–26, 2012.

[152] Lijun Zhang, Mehrdad Mahdavi, and Rong Jin. Linear convergence with condition number independent access of full gradients. In *NIPS*, 2013.

[153] Lijun Zhang, Mehrdad Mahdavi, Rong Jin, Tianbao Yang, and Shenghuo Zhu. Recovering the optimal solution by dual random projection. In *COLT*, 2013.

[154] Lijun Zhang, Tianbao Yang, Rong Jin, and Xiaofei He. O(logt) projections for stochastic optimization of smooth and strongly convex functions. *ICML*, 2013.

[155] Tong Zhang. Sequential greedy approximation for certain convex optimization problems. *Information Theory, IEEE Transactions on*, 49:682–691, 2003.

[156] Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, pages 56–85, 2004.

[157] Ding-Xuan Zhou. Capacity of reproducing kernel spaces in learning theory. *Information Theory, IEEE Transactions on*, 49(7):1743–1752, 2003.

[158] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.