

ALGORITHMS AND LOWER BOUNDS FOR PARAMETER-FREE ONLINE
LEARNING

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Ashok Cutkosky

May 2018

© Copyright by Ashok Cutkosky 2018
All Rights Reserved

Ashok Cutkosky

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Kwabena Boahen) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Percy Liang)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Dan Boneh)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(John Duchi)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Aaron Sidford)

Approved for the Stanford University Committee on Graduate Studies

Acknowledgments

Many people played an important part in this work during the five years at Stanford that produced it, I would like to take this page to express my gratitude. I first thank my advisor, Kwabena Boahen, who entertained my meandering interests as I converged to the eventual topic of this dissertation, spent many hours with me distilling ideas to extract the intuition a reader could understand, provided advice on writing and presentation, and of course paid for so many group lunches. I also thank my parents Hema Srinivasan and Dale Cutkosky, who walked this path and so many others before me. Their constant encouragement and advice helped me keep going through the years. These five years were far happier for me thanks to Janet Song, who listened to me lecture about math while she drove me to dinner, and came up with a better algorithm name. I also thank all other members of Kwabena's lab, who sat around me every day and imparted a modicum of real hardware knowledge through osmosis. Finally, I thank my committee members for their attention to my efforts, and for writing some of the papers that inspired them in the first place.

To my grandfather, the inventor.

Contents

Acknowledgments	iv
1 Introduction	1
1.1 Online Learning	1
1.2 Stochastic Problems and Online-to-Batch Conversion	4
1.3 Outline of the Thesis	6
2 Basic Notions in Convex Analysis and Online Convex Optimization	8
2.1 Duals, Norms, Convexity and Notation	8
2.1.1 Subgradients and Online Linear Optimization	10
2.1.2 Strong Convexity, Smoothness, and Exp-Concavity	11
2.2 Online Subgradient Descent	13
2.3 Follow the Regularized Leader	15
2.4 Mirror Descent	16
2.5 Coin Betting	17
2.6 Online Relaxations	19
2.7 Conclusions	20
3 Prior Adaptive Online Algorithms	21
3.1 Adapting to G_{\max}	22
3.2 Adapting to $\ \hat{w}\ $	24
3.3 Strong Convexity	25
4 An Optimal Parameter-free Algorithm	27
4.1 A Lower-Bound for Adapting to $\ \hat{w}\ $ and G_{\max}	27
4.1.1 Warm-up: a Suboptimal Lower-Bound	27
4.1.2 Optimal Bound: Trade-offs in the multiplicative constant k	29
4.1.3 Optimal Bound: Trade-offs in the Logarithmic exponent γ	30
4.2 Parameter-free FTRL Analysis	31

4.2.1	Generalizing Strong Convexity	32
4.2.2	Adaptive regularizers	32
4.3	Optimal Algorithms	34
4.4	FREEEX	36
4.5	Conclusions	38
4.5.1	Other Kinds of Adaptivity	39
Appendices		40
4.A	Lower Bound Proof	40
4.B	FTRL regret	46
4.C	Facts About Strong Convexity	48
4.D	Proof of Theorem 4.9	50
4.E	Proof of Theorem 4.7	51
5	Reductions For Parameter-free Online Learning	67
5.1	Exp-Concave Optimization to Online Linear Optimization via Betting Algorithms	69
5.2	From 1D Algorithms to Dimension-Free Algorithms	71
5.2.1	Alternate Reduction Without Unit-Ball Algorithm	73
5.3	Reduction to Constrained Domains	77
5.4	Banach-space betting through ONS	79
5.5	From T to $\sum \ g_t\ _\star$	80
5.5.1	Lower bound for $\ g_t\ _\star^2$	82
5.6	Conclusions	85
Appendices		86
5.A	Banach Spaces	86
5.B	Proof of the regret bound of ONS in Banach spaces	88
5.C	Proofs of Theorems 5.1 and 5.7	94
5.D	Proof of Proposition 5.5 and Theorem 5.6	98
6	Other Applications	100
6.1	Reduction for Multi-Scale Experts	100
6.2	Sparsity	103
6.3	Conclusions	104
Appendices		105
6.A	Computing S_W for multi-scale experts	105
6.A.1	Computing a subgradient of S_W for multi-scale experts	106

7 Losses With Curvature	113
7.1 Adapting to Smoothness	114
7.1.1 Variance Reduction	117
7.2 Adapting to Strong Convexity	122
7.3 Conclusions	124
Appendices	125
7.A Proof of Theorem 7.8	125

Algorithms and Lower Bounds for Parameter-free Online Learning

Ashok Cutkosky

May 29, 2018

Abstract

Training a machine learning model today involves minimizing a loss function on datasets that are often gigantic, and so almost all practically relevant training algorithms operate in an online manner by reading in small chunks of the data at a time and making updates to the model on-the-fly. As a result, *online learning*, a popular way to analyze optimization algorithms operating on datastreams, is at the heart of modern machine learning pipelines. In order to converge to the optimal model as quickly as possible, online learning algorithms all require some user-specified parameters that reflect the shape of the loss or statistics of the input data. Examples of such parameters include the size of the gradients of the losses, the distance from some initial model to the optimal model, and the amount of variance in the data, among others. Since the true values for these parameters are often unknown, the practical implementation of online learning algorithms usually involves simply guessing (called “tuning”), which is both inefficient and inelegant. This motivates the search for parameter-free algorithms that can adapt to these unknown values. Prior algorithms have achieved adaptivity to many different unknown parameters individually - for example one may adapt to an unknown gradient sizes given a known distance to the optimal model, or adapt to the unknown distance given a known bound on gradient size. However, no algorithm could adapt to both parameters simultaneously.

This work introduces new lower bounds, algorithms, and analysis techniques for adapting to many parameters at once. We begin by proving a lower bound showing that adapting to both the size of the gradients and distance to optimal model simultaneously is fundamentally much harder than adapting to either individually, and proceed to develop the first algorithm to meet this lower bound, obtaining optimal adaptivity to both parameters at once. We then expand upon this result to design algorithms that adapt to more unknown parameters, including the variance of the data, different methods for measuring distances, and upper or lower bounds on the second derivative of the loss. We obtain these results by developing new techniques that convert non-parameter-free optimization algorithms into parameter-free algorithms. In addition to providing new and more adaptive algorithms, the relative simplicity of non-parameter-free algorithms allows these techniques to significantly reduce the complexity of many prior analyses.

Chapter 1

Introduction

1.1 Online Learning

This thesis is about Online Learning, which is an elegant and robust way to understand stochastic, streaming, and even adversarial environments [60; 53; 33; 23]. Because many modern-day machine learning applications involve huge amounts of data, most practical methods for training models process the data in an *online* manner by processing data in small chunks and making updates to the model using only this streaming view of the dataset. Due to its relative simplicity and lack of constraining assumptions, many of the popular methods used to train machine learning models today (e.g. [17; 51]) are stated and analyzed using the online learning framework. Before describing the technical contributions of this work, we will take some time to introduce the setting of online learning and provide some background on prior work.

Online learning is a game played between a learner and the environment. The game consists of a series of T rounds. In the t th round, the learner outputs a vector w_t in some specified space W (called the *domain*), and then the environment outputs a loss function $\ell_t : W \rightarrow \mathbb{R}$. The goal of the learner is to have a low *regret*, which is equal to the total loss suffered minus the total loss suffered at some comparison point \hat{w} , chosen by the environment:

$$R(\hat{w}) = \sum_{t=1}^T \ell_t(w_t) - \ell_t(\hat{w})$$

The simplest benchmark for any online learning algorithm is to obtain *sublinear regret*, which means that $\lim_{T \rightarrow \infty} \frac{R_T(\hat{w})}{T} = 0$ for each \hat{w} . Sublinear regret is a desirable characteristic because it means that in the large- T limit, the algorithm does just as well on average as the comparison point \hat{w} (the “average regret” is going to zero).

The online learning framework is appealing because it is simple enough to reason about effectively, but flexible enough that we can use it to model many practical problems of interest. As a concrete example, suppose we are tasked with predicting the weather. Each day we might have some measurement vector $x \in \mathbb{R}^3$ which records the temperature, wind speed and humidity in the morning, and we need to make a

prediction \hat{y} for the number of inches of rainfall that will occur that day. At the end of the day we observe the true rainfall y that occurred and we consider ourselves to be doing well if $|y - \hat{y}|$ is small. Suppose we want to employ a simple linear model: $\hat{y} = w \cdot x$ for some vector w we will choose. Then we can map this scenario into the online learning framework as follows:

1. Each day is a distinct round.
2. On the t th day, we choose a vector w_t , observe x_t (the measurements on the t th morning) and predict $\hat{y}_t = w_t \cdot x_t$.
3. We then observe the true value y_t and construct the loss function $\ell_t(w) = |w \cdot x_t - y_t|$, which has $\ell_t(w_t) = |\hat{y}_t - y_t|$.
4. The regret is how much error we actually accrue minus the error we *would have accrued* if we had stuck to some fixed (presumably optimal) vector \hat{w} :

$$R_T(\hat{w}) = \sum_{t=1}^T |x_t \cdot w_t - y_t| - |x_t \cdot \hat{w} - y_t|$$

This example illustrates a subtlety of online learning: the role of the comparison point \hat{w} . One might think that the right way to measure an algorithm's performance is simply the total loss $\sum_{t=1}^T \ell_t(w_t)$ rather than the regret. After all, ultimately we are interested in having good weather predictions - the people watching the morning news aren't going to be impressed by just low regret! However, this quantity confounds two factors: the performance of the algorithm, as well as any inherent difficulty introduced by some user-selected modeling assumptions. For example, in the rain prediction setting above, it is highly unlikely that there actually exists a vector \hat{w} such that $x_t \cdot \hat{w} = y_t$ every day. Therefore it would be unfair to punish our algorithm for failing to find any such \hat{w} and have low loss. We compensate for this inherent difficulty in the problem by instead measuring the performance relative to a benchmark point \hat{w} . Thus if we have low regret but high total loss, this indicates that our learning algorithm is performing well, but we need to improve our modeling assumptions.

Before moving on, we briefly sketch three more examples of scenarios in which one might apply online learning, and what it means to have sublinear regret in each scenario.

1. Suppose you are programming a robot to shoot basketballs. There are many parameters of your basketball shooting strategy, including amount of force used, angle of shot and so-on. We will encapsulate all these parameters in a vector w . Each time you take a shot is one round of an online learning game. The loss $\ell_t(w)$ is 1 if you miss the shot using strategy w , and 0 otherwise. Each round is subtly different due to wind variation, temperature, amount of air in the ball and so on. The quantity $\sum_{t=1}^T \ell_t(w_t)$ is simply the number of missed shots. The quantity $\sum_{t=1}^T \ell_t(\hat{w})$ is the number of shots that *would have been missed* if you had stuck to the fixed strategy \hat{w} . Thus an algorithm that achieves low regret has nearly the same average shooting accuracy as the best fixed strategy.

2. Now suppose you are interested in classifying email as spam or useful email. Different values of w indicate different strategies for classifying the email messages. $\ell_t(w)$ is 0 if the t th email is correctly classified by strategy w , and $\ell_t(w)$ is 1 otherwise. Then the quantity $\sum_{t=1}^T \ell_t(w_t)$ is the number of misclassified emails, and $\sum_{t=1}^T \ell_t(w_t)$ is the number of emails misclassified by the best strategy. Again, an algorithm with sublinear regret would on average misclassify the same number of emails as the best fixed strategy.
3. Finally, suppose you are training a model to produce transcripts of audio files. You may have a dataset of millions of hand-transcribed audio files. Since the dataset is so large, you can't afford to make many passes over it. So instead, you define $\ell_t(w)$ to be some error measure of the transcript produced by model w on the t th element of the dataset. Then you may use an online learning algorithm with sublinear regret to produce a model whose performance is close to the optimal error rate $\sum_{t=1}^T \ell_t(\hat{w})$. Note that in this example, the online learning algorithm technically produces a *sequence* of models with average performance close to that of \hat{w} rather than a single model, but as we will see in Section 1.2, this is a detail that can be easily remedied.

With a little thought, almost any optimization problem with many “trials” can be considered as an online learning problem. In fact, we will later see that also any optimization problem with noise (so-called stochastic optimization problems) can also be viewed as online learning problems. These types of problems abound in machine learning, and so developing better online learning algorithms improves core components in machine learning pipelines.

Although sublinear regret is a very desirable property, it turns out that it is actually impossible to achieve without making further assumptions about the environment. This is formalized in Proposition 1.1 below, which gives a simple example for which the environment causes every algorithm to obtain linear regret.

Proposition 1.1 ([11]). *Suppose $W = \{0, 1\}$. Let \mathcal{A} be an online learning algorithm with domain W . Given $w_t \in W$, let $\ell_t(w) = |1 - w_t - w|$. Then \mathcal{A} suffers regret $R_T(\hat{w}) \geq \frac{T}{2}$ for some $\hat{w} \in W$.*

Proof. Clearly $\sum_{t=1}^T \ell_t(w_t) = T$, so it suffices to show that $\sum_{t=1}^T \ell_t(\hat{w}) \leq \frac{T}{2}$ for some $\hat{w} \in \{0, 1\}$. Notice that $\ell_t(w) \geq 0$ for all $w \in \{0, 1\}$ and $\ell_t(0) + \ell_t(1) = 1$ for all t , so $\sum_{t=1}^T \ell_t(0) + \sum_{t=1}^T \ell_t(1) = T$. Therefore it cannot be that both of $\sum_{t=1}^T \ell_t(0)$ and $\sum_{t=1}^T \ell_t(1)$ are greater than $\frac{T}{2}$ and so we are done. \square

This impossibility result highlights a key aspect of the online learning setup: the environment is allowed to be *adversarial*. For example, in the proof of Proposition 1.1, the environment chooses ℓ_t after seeing the learning algorithm's output w_t in such a way as to force $\ell_t(w_t) = 1$ for all t . Thus, in order to make the problem tractable, we need to place some kind of restriction on the environment's loss functions ℓ_t .

Let's first consider the most obvious way to prevent this kind of behavior: do not allow the loss function ℓ_t to change each round and instead force the environment to choose a fixed loss ℓ . This solution clearly rules out the adversarial environment in Proposition 1.1, but it still leaves us with a more subtle problem: computational complexity. For example, if $W = \{0, 1\}^N$ and $\ell : W \rightarrow \{0, 1\}$ is the output of some Boolean

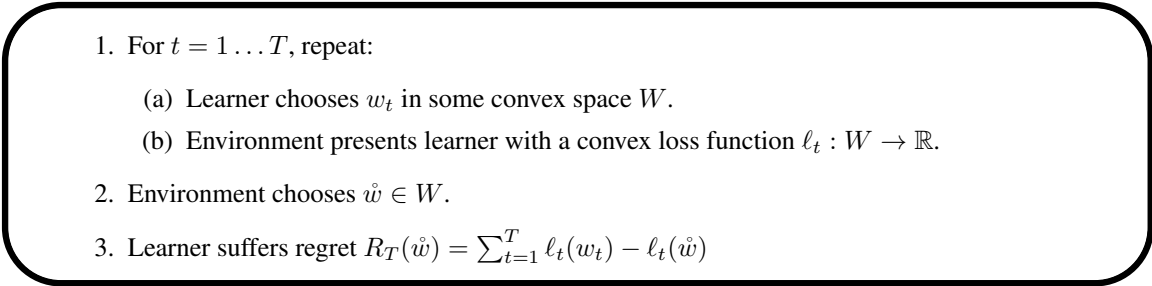
- 
1. For $t = 1 \dots T$, repeat:
 - (a) Learner chooses w_t in some convex space W .
 - (b) Environment presents learner with a convex loss function $\ell_t : W \rightarrow \mathbb{R}$.
 2. Environment chooses $\hat{w} \in W$.
 3. Learner suffers regret $R_T(\hat{w}) = \sum_{t=1}^T \ell_t(w_t) - \ell_t(\hat{w})$

Figure 1.1: Online Convex Optimization

circuit, then minimizing ℓ over W is an NP-hard problem. Thus even with a fixed loss we may be unable to obtain low regret in practice. To rule out NP-hard or worse problems, we need to impose some restriction on the *kinds* of losses ℓ_t the environment is allowed to supply.

In the general optimization literature, a very common (and successful in practice) assumption that allows for efficiently minimizing a fixed loss function ℓ is *convexity*:

Definition 1.2. *Given a convex set W of a real vector space, a function $\ell : W \rightarrow \mathbb{R}$ is convex if $\ell(tx + (1 - t)y) \leq t\ell(x) + (1 - t)\ell(y)$ for any $x, y \in W$ and $t \in [0, 1]$.*

There are vast troves of literature on both reducing seemingly non-convex optimization problems into convex optimization problems, and on efficiently solving convex optimization problems (e.g. see [6; 37]). We will therefore assume that the domain W is convex and that all our loss functions are convex. Notice that this assumption also rules out the impossibility result of Proposition 1.1 because $W = \{0, 1\}$ is not a convex set. Thus (perhaps surprisingly), it will actually not be necessary force the environment to choose a fixed loss function, or indeed make any further assumptions about the loss functions at all. Online learning with this convexity assumption is often called *online convex optimization*, and it is outlined in Figure 1.1. Online convex optimization is the main focus of this thesis. This is important, so we restate it in bold below:

All results in this thesis are for online convex optimization.

1.2 Stochastic Problems and Online-to-Batch Conversion

One of the most common problems in modern machine learning applications is *stochastic optimization*. Stochastic optimization is very similar to online learning, but has a more limited environment and a slightly different objective. In stochastic optimization, the environment presents the learner with T losses ℓ_1, \dots, ℓ_T each sampled i.i.d. from some distribution D with expectation $\mathbb{E}[\ell] = \mathcal{L}$. The learner's task is to output a single point \hat{w} such that $\mathcal{L}(\hat{w}) - \mathcal{L}(\hat{w})$ is as small as possible. This is summarized in Figure 1.2.

Our rain prediction example could also be seen as an instance of stochastic optimization. In this case the distribution D is the distribution over losses $\ell(w) = |x \cdot w - y|$ where x, y are random variables representing

1. Environment chooses distribution D over functions $\ell : W \rightarrow \mathbb{R}$ with expectation $\mathbb{E}[\ell] = \mathcal{L} : W \rightarrow \mathbb{R}$.
2. Environment chooses $\hat{w} \in W$.
3. Environment presents a sample of T i.i.d. functions $\ell_1, \dots, \ell_T \sim D$ to the learner.
4. Learner outputs $\hat{w} \in W$.
5. Learner suffers suboptimality $\mathcal{L}(\hat{w}) - \mathcal{L}(\hat{w})$

Figure 1.2: Stochastic Optimization

the measurements and rainfall on any given day. A good weather prediction algorithm would obtain a low expected loss $\mathbb{E}[\ell(\hat{w})]$. Since it may not be possible to actually obtain a truly small loss with any \hat{w} , we again measure the performance of our algorithm relative to some comparison point \hat{w} , which can be chosen by the environment to achieve a low loss.

There are two key differences between stochastic optimization and online learning. First, in stochastic optimization the learner only outputs (and is evaluated on) a single point \hat{w} . Second, the loss functions ℓ_t are all drawn i.i.d. rather than being potentially adversarially adapted to the outputs of the learner. The first point may seem to make stochastic optimization harder than online learning because the learner in some sense only has “one chance” to do well. The second point, on the other hand, makes stochastic optimization easier than online learning. It turns out that in a very general sense, the second point is the most important. It is always possible to convert an online learning algorithm into a stochastic optimization algorithm via the online-to-batch conversion method, described below:

Proposition 1.3 ([9]). *Let $w_1, \dots, w_T \in W$ be the outputs of an online learning algorithm \mathcal{A} on losses ℓ_1, \dots, ℓ_T drawn i.i.d. from some distribution D . Let \hat{w} be a randomly selected element of $\{w_1, \dots, w_T\}$. Then*

$$\mathbb{E}[\mathcal{L}(\hat{w}) - \mathcal{L}(\hat{w})] \leq \frac{1}{T} \mathbb{E}[R_T(\hat{w})]$$

Where the expectation is over all of the randomness in the losses ℓ_t , any internal randomness in \mathcal{A} , and the choice of \hat{w} . Further, if \mathcal{L} is convex, then we may set $\hat{w} = \frac{1}{T} \sum_{t=1}^T w_t$ and obtain the same result.

Proof. By definition we have

$$\begin{aligned} \mathbb{E}[\mathcal{L}(\hat{w}) - \mathcal{L}(\hat{w})] &= \mathbb{E} \left[\frac{1}{T} \sum_{t=1}^T \mathcal{L}(w_t) - \mathcal{L}(\hat{w}) \right] \\ &= \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\mathcal{L}(w_t)] - \mathcal{L}(\hat{w}) \end{aligned}$$

Now observe that ℓ_t is independent of \hat{w} and w_t since w_t only depends on $\ell_1, \dots, \ell_{t-1}$. Therefore $\mathbb{E}[\mathcal{L}(w_t)] =$

$\mathbb{E}[\ell_t(w_t)]$ and $\mathcal{L}(\hat{w}) = \mathbb{E}[\ell_t(\hat{w})]$. Thus we have

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\hat{w}) - \mathcal{L}(\hat{w})] &= \frac{1}{T} \sum_{t=1}^T \mathbb{E}[\ell_t(w_t) - \ell_t(\hat{w})] \\ &= \frac{1}{T} \mathbb{E} \left[\sum_{t=1}^T \ell_t(w_t) - \ell_t(\hat{w}) \right] = \frac{1}{T} \mathbb{E}[R_T(\hat{w})]\end{aligned}$$

Finally, when \mathcal{L} is convex, by Jensen inequality we have

$$\mathcal{L} \left(\frac{1}{T} \sum_{t=1}^T w_t \right) \leq \frac{1}{T} \sum_{t=1}^T \mathcal{L}(w_t)$$

so that we may set $\hat{w}_t = \frac{1}{T} \sum_{t=1}^T w_t$ and still conclude the first line of the proof (with an inequality rather than equality). The rest of the argument is identical. \square

This Proposition tells us that so long as $\frac{1}{T} R_T(\hat{w})$ is small - which is exactly the guarantee provided by sublinear regret - then we can turn any online learning algorithm into a corresponding stochastic optimization algorithm. Since online learning algorithms must also be able to handle adversarial environments in addition to stochastic ones, this brings some measure of robustness to the converted algorithm. One might expect that by designing an algorithm specifically for stochastic scenarios, we can sacrifice this robustness to obtain better convergence rates. Surprisingly, this is not so! Usually a converted online learning algorithm has the same performance guarantees as an algorithm that is specifically designed for stochastic optimization. As a result, online learning not only allows an algorithm to deal with streaming sources of data, it also provide simple and attractive ways to solve the stochastic optimization problems at the heart of many classification and regression tasks in machine learning.

1.3 Outline of the Thesis

The primary goal of this work is to design online learning algorithms that adapt to apriori unknown data characteristics. For example, if the loss functions ℓ_t turn out to be all the same, or if the comparison point \hat{w} happens to be equal to our initial prediction w_1 , we should expect to have lower regret than if they are changing adversarially or if \hat{w} is far away from w_1 . We will describe several techniques for accomplishing these and other goals, as well as some lower bounds showing that some goals are out of reach.

We will start in Chapter 2 by introducing several different techniques that are used to design online learning algorithms and deriving the basic theorems of the field. Next, we will review some relevant prior art for adaptive online learning in Chapter 3. Then we will move on to our original contributions. In Chapter 4 we will describe a new lower bound and matching optimal algorithm for parameter-free online learning. Our algorithm design and analysis in Chapter 4 will follow the classical FTRL technique introduced in Chapter 2. In Chapter 5 we will present an alternative method for designing parameter-free algorithms that significantly

simplifies the analysis, and also sometimes provides superior results. Then in Chapters 6 and 7 we will apply these reductions to obtain algorithms that adapt to other parameters relating to the curvature or sparsity of the losses.

Chapter 2

Basic Notions in Convex Analysis and Online Convex Optimization

In this chapter we will outline the most common techniques used to design online learning algorithms. We will start by reviewing some basic facts about convex functions, and then use this background to directly derive the online subgradient descent algorithm. From there we will go over the Follow-the-Regularized-Leader and Mirror Descent frameworks for designing online learning algorithms, and show how the gradient descent algorithm can be realized and analyzed as special cases of these frameworks. Then we will move on to describe slightly less classical approaches based on coin betting and online relaxations.

2.1 Duals, Norms, Convexity and Notation

In this section and throughout this thesis we will assume a basic familiarity with abstract linear algebra. In particular, we make use of the concepts of inner products, Hilbert spaces, Banach spaces, dual spaces, norms and dual norms. For completeness, we provide here a short overview of some the background definitions and theorems we will use. A further description of some concepts relating to Banach spaces can be found in Section 5.A.

First, we introduce the notion of a norms:

Definition 2.1. *Given a real vector space V , a norm is a function $\|\cdot\| : V \rightarrow \mathbb{R}$ that satisfies:*

1. $\|cv\| = |c|\|v\|$ for any $c \in \mathbb{R}$ and $v \in V$.
2. $\|v + w\| \leq \|v\| + \|w\|$ for any $v, w \in V$.
3. $\|v\| = 0$ if and only if v is the zero element of V .

All of the normed vector spaces we consider in this thesis are *Banach spaces*, defined below:

Definition 2.2. A real Banach space is a real vector space V equipped with a norm $\|\cdot\|$ such that V is topologically complete with respect to the metric $d(x, y) = \|x - y\|$.

Now we can define the concept of dual space. Notice that our definition makes use of the notion of a continuous map on a vector space. The definition of dual space with this extra qualification is sometimes called the *analytic* dual space, in contrast to the *algebraic* dual space, which can be defined on any vector space regardless of whether it possesses any topology.

Definition 2.3. Given a real Banach space V , the dual space V^* is the set of all continuous linear maps $v^* : V \rightarrow \mathbb{R}$. Given $v^* \in V^*$ and $v \in V$, we will denote $v^*(v)$ by $\langle v^*, v \rangle$.

In the classic example of $V = \mathbb{R}^n$, the dual space V^* is isomorphic to \mathbb{R}^n , and we identify $\langle v^*, v \rangle$ with the dot product $v^* \cdot v$ when v^* is viewed as an element of \mathbb{R}^n .

The dual of the dual space, $(V^*)^*$ naturally contains the original space V via the identification:

$$v \in V \mapsto (v^* \mapsto \langle v^*, v \rangle) \in (V^*)^*$$

When this identification is also a surjection (i.e. when $(V^*)^*$ is isomorphic to V), then we say the space V is *reflexive*. All finite dimensional spaces are reflexive.

Given a norm $\|\cdot\|$ on a vector space V , we can define the dual norm $\|\cdot\|_*$, which can be shown to be a norm on V^* :

Definition 2.4. Let $\|\cdot\| : V \rightarrow \mathbb{R}$ be a norm on a vector space V . The dual norm, $\|\cdot\|_* : V^* \rightarrow \mathbb{R}$ is defined by

$$\|v^*\|_* = \sup_{\|v\| \leq 1} \langle v^*, v \rangle$$

If V is a Banach space with a norm $\|\cdot\|$, then its dual V^* is also a Banach space with the norm $\|\cdot\|_*$.

A critical consequence of the dual norm is the generalized Cauchy-Schwarz inequality:

Proposition 2.5. For any $v \in V$ and $v^* \in V^*$, $\langle v^*, v \rangle \leq \|v\| \|v^*\|_*$.

Of particular interest are the p -norms, defined on \mathbb{R}^n as:

$$\|(x_1, \dots, x_n)\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$$

The dual of the p -norm is the q -norm, where $\frac{1}{p} + \frac{1}{q} = 1$. In particular, the 2-norm is dual to itself.

A special case we will often be interested in is when V is a real *Hilbert space*:

Definition 2.6. A real Hilbert space is a real Banach space V equipped (by re-use of notation) with map $\langle \cdot, \cdot \rangle : V \times V \rightarrow \mathbb{R}$ called an inner product such that for all $s \in \mathbb{R}$ and $a, b, C \in V$:

1. $\langle a, b \rangle = \langle b, a \rangle$.

2. $\langle a, \cdot \rangle : V \rightarrow \mathbb{R}$ is a linear map.
3. $\langle a, a \rangle = \|a\|^2$

Any Hilbert space V has the property that V is isomorphic to V^* by $v \mapsto \langle v, \cdot \rangle$ (which justifies our re-use of $\langle \cdot, \cdot \rangle$ notation). Further, using this isomorphism we have $\|\cdot\| = \|\cdot\|_*$. When working in a Hilbert space we will often make use of this identification to perform operations in V rather than V^* .

Finally, throughout this thesis we make use a compressed sum notation where subscripts with colons indicate summations: $\sum_{t=1}^T g_t = g_{1:T}$, $\sum_{t=1}^T \|g_t\|^2 = \|g\|_{1:T}^2$, $\sum_{t=1}^T \langle g_t, w_t \rangle = \langle g, w \rangle_{1:T}$ and similarly for other indexed sums.

2.1.1 Subgradients and Online Linear Optimization

Recall from Definition 1.2 that a convex function ℓ satisfies $\ell(tx + (1-t)y) \leq t\ell(x) + (1-t)\ell(y)$. For our purposes, the most important property of a convex function is the existence of *subgradients*:

Definition 2.7. Let W be a convex subset of some vector space and let $\ell : W \rightarrow \mathbb{R}$ be a convex function. For any $x \in W$, a vector $g \in W^*$ is called a *subgradient* of ℓ at x , if $\ell(y) \geq \ell(x) + \langle g, y - x \rangle$ for all $y \in W$. The set of subgradients of ℓ at x is written $\partial\ell(x)$ and is called the *subdifferential*, and $\partial\ell(x) \neq \emptyset$ for all x .

Subgradients should be viewed as a generalization of the usual gradient to non-differentiable convex functions. In particular, we have the following:

Proposition 2.8. Suppose $f : W \rightarrow \mathbb{R}$ is a convex function. If f is differentiable at some point w in the interior of W , then $\{\nabla f(w)\} = \partial f(w)$.

The critical consequence of subgradients in online convex optimization is that the hardest type of convex losses are actually simple *linear losses*. The argument is simple: set $g_t \in \partial\ell_t(w_t)$. Then by definition of subgradient, we have

$$\ell_t(w_t) - \ell_t(\hat{w}) \leq \langle g_t, w_t - \hat{w} \rangle$$

from which we conclude:

$$R_T(\hat{w}) = \sum_{t=1}^T \ell_t(w_t) - \ell_t(\hat{w}) \leq \sum_{t=1}^T \langle g_t, w_t \rangle - \langle g_t, \hat{w} \rangle$$

This simple fact is extraordinarily powerful. It implies that we can *replace* the potentially complicated convex loss function $w \mapsto \ell_t(w)$ with the significantly simpler linear loss function $w \mapsto \langle g_t, w \rangle$, and the regret will only increase. Therefore an algorithm that guarantees low regret when the losses are always linear (which is called online linear optimization, or OLO) can automatically guarantee low regret with arbitrary convex losses. Because of this fact, for most of the rest of this manuscript we will consider only linear losses of the form $w \mapsto \langle g_t, w \rangle$, and take $R_T(\hat{w}) = \sum_{t=1}^T \langle g_t, w_t - \hat{w} \rangle$.

Another important property of a function is *Lipschitzness*:

Definition 2.9. A function $f : W \rightarrow \mathbb{R}$ is G -Lipschitz with respect to a norm $\|\cdot\|$ if

$$|f(x) - f(y)| \leq G\|x - y\|$$

for all x, y in W .

There is a natural connection between Lipschitzness and subgradients, which we will often use implicitly:

Proposition 2.10. If $f : W \rightarrow \mathbb{R}$ is differentiable, then f is G -Lipschitz if and only if $\|\nabla f(w)\|_* \leq G$ for all $w \in W$. If f is convex but non-differentiable, then f is G -Lipschitz if and only if all subgradients $g \in \partial f(w)$ satisfy $\|g\|_* \leq G$ for all $w \in W$.

2.1.2 Strong Convexity, Smoothness, and Exp-Concavity

We will occasionally make use of the notion of strong convexity, which provides a way of quantifying “how convex” a function is, with 0-strong convexity being equivalent to the ordinary definition of convex:

Definition 2.11. Given a convex set W , a function $f : W \rightarrow \mathbb{R}$ is a μ -strongly convex with respect to a norm $\|\cdot\|$ if $h(y) = f(x) - \frac{\mu}{2}\|x - y\|^2$ is a convex function for all $x \in W$.

A critical fact about strongly-convex functions is the following identity:

Proposition 2.12. Let $f : W \rightarrow \mathbb{R}$ be a μ -strongly convex function. Then for all $x, y \in W$ and all subgradients $g \in \partial f(x)$,

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

An inverse concept to strong convexity that we will also use is smoothness, which is defined below:

Definition 2.13. A differentiable function f is L -smooth with respect to some norm $\|\cdot\|$ if ∇f is L -Lipschitz with respect to the dual norm $\|\cdot\|_*$. When f is convex, this is equivalent to

$$f(y) \leq f(x) + \langle g, y - x \rangle + \frac{L}{2}\|y - x\|^2$$

for all x, y and $g \in \partial f(x)$.

We will also make use of the concept of an exp-concave function:

Definition 2.14. A function $f : W \rightarrow \mathbb{R}$ is exp-concave if $\exp(-f)$ is a concave function. All exp-concave functions are convex, and all Lipschitz strongly-convex functions are exp-concave.

Finally, we will occasionally make use of the notions of *fenchel conjugate* and *Bregman Divergences*, which we define below:

Definition 2.15. Let $W \subset V$ where V is a real vector space. Suppose $f : W \rightarrow \mathbb{R}$ is a function (not necessarily convex). The fenchel conjugate of f is denoted $f^* : V^* \rightarrow \{\mathbb{R}, \infty\}$ and is defined by

$$f^*(v^*) = \sup_{w \in W} \langle v^*, w \rangle - f(w)$$

Definition 2.16. Let $\psi : V \rightarrow \mathbb{R}$ be a differentiable convex function. The Bregman divergence associated with ψ is denoted by $B_\psi : V \times V \rightarrow \mathbb{R}$ and is given by

$$B_\psi(v, w) = \psi(v) - \psi(w) - \langle \nabla \psi(w), v - w \rangle$$

The Bregman divergence should be thought of as a kind of “squared distance” between v and w that depends on the function ψ . For the special case $\psi(x) = \|x\|^2$ and $\|\cdot\|$ is the 2-norm, $B_\psi(v, w) = \|v - w\|^2$. Further, notice that the ψ is μ -strongly convex with respect to a norm if and only if $B_\psi(v, w) \geq \frac{\mu}{2} \|v - w\|^2$.

The use of the notation f^* , commonly reserved for “dual objects” is justified by the following duality theorem:

Proposition 2.17. Let $W \subset V$ be a convex set and V be a reflexive Banach space. For any function $f : W \rightarrow \mathbb{R}$, f^* is a convex function and $(f^*)^*(w) \leq f(w)$ for all $w \in W$. When f is itself a convex function, then $(f^*)^*(w) = f(w)$ for all $w \in W$.

Proof. First, observe that $\langle v^*, w \rangle - f(w)$ is convex in v^* for all w . Therefore f^* is a supremum of convex functions and so must be convex.

For the second statement, since V is a reflexive Banach space, $(V^*)^*$ is naturally isomorphic to V , and so we may consider $(f^*)^*$ as a map $V \rightarrow \mathbb{R}$. Then tracing through the definitions we have:

$$\begin{aligned} (f^*)^*(w) &= \sup_{v^* \in V^*} \langle v^*, w \rangle - f^*(v^*) \\ &= \sup_{v^* \in V^*} \langle v^*, w \rangle - \left(\sup_{v \in W} \langle v^*, v \rangle - f(v) \right) \\ &= \sup_{v^* \in V^*} \inf_{v \in W} \langle v^*, w - v \rangle + f(v) \end{aligned}$$

Now observe that for any v^* , $\inf_{v \in W} \langle v^*, w - v \rangle + f(v) \leq f(w)$ simply by setting $v = w$. Therefore $(f^*)^*(w) \leq f(w)$.

For the last statement, let $g \in V^*$ be a subgradient of f at w (which exists because f is convex). Then we have

$$\begin{aligned}
 (f^*)^*(w) &= \sup_{v^* \in V^*} \inf_{v \in W} \langle v^*, w - v \rangle + f(v) \\
 &\geq \sup_{v^* \in V^*} \inf_{v \in W} \langle v^*, w - v \rangle + f(w) + \langle g, v - w \rangle \\
 &= \sup_{v^* \in V^*} \inf_{v \in W} \langle v^* - g, w - v \rangle + f(w) \\
 &= f(w) + \sup_{v^* \in V^*} \inf_{v \in W} \langle v^* - g, w - v \rangle
 \end{aligned}$$

Now observe that for $v^* \neq g$, $\inf_{v \in W} \langle v^* - g, w - v \rangle = -\infty$, and for $v^* = g$, $\sup_{v^* \in V^*} \inf_{v \in W} \langle v^* - g, w - v \rangle = 0$. Therefore $(f^*)^*(w) \geq f(w)$, which completes the proof. \square

2.2 Online Subgradient Descent

The simplest online linear optimization algorithm is *online subgradient descent*. Suppose our domain W is a bounded subset of a Hilbert space H , and make the standard identification of H with its dual space, recalling that the dual of the Hilbert space norm is itself. Let $\Pi_W(x) = \operatorname{argmin}_{y \in W} \|x - y\|$. Then the online subgradient descent algorithm chooses some initial point w_1 and from then on sets:

$$w_{t+1} = \Pi_W(w_t - \eta g_t)$$

for some parameter η , called the *learning rate*. This algorithm comes with a simple (yet surprisingly powerful) regret guarantee:

Proposition 2.18 ([60]). *Suppose W is a subset of a Hilbert space H so that $H \simeq H^*$ and $\|\cdot\| = \|\cdot\|_*$. Then online subgradient descent guarantees regret*

$$R_T(\dot{w}) = \sum_{t=1}^T \langle g_t, w_t - \dot{w} \rangle \leq \frac{\|w_1 - \dot{w}\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^T \|g_t\|^2$$

Further, suppose $w_1 = 0 \in W$ and each g_t satisfies $\|g_t\| \leq G_{\max}$. Then if we set $\eta = \frac{\|\dot{w}\|}{G\sqrt{T}}$, we have

$$R_T(\dot{w}) \leq \|\dot{w}\| G_{\max} \sqrt{T}$$

Proof. Let $x_{t+1} = w_t - \eta g_t$. Then we have $w_{t+1} = \Pi_W(x_{t+1})$ so that

$$\begin{aligned}
 \|w_{t+1} - \dot{w}\|^2 &\leq \|x_{t+1} - \dot{w}\|^2 \\
 &= \|w_t - \eta g_t - \dot{w}\|^2 \\
 &= \|w_t - \dot{w}\|^2 - 2\eta \langle g_t, w_t - \dot{w} \rangle + \eta^2 \|g_t\|^2 \\
 \langle g_t, w_t - \dot{w} \rangle &\leq \frac{1}{2\eta} (\|w_t - \dot{w}\|^2 - \|w_{t+1} - \dot{w}\|^2) + \frac{\eta}{2} \|g_t\|^2 \\
 R_T(\dot{w}) &\leq \frac{1}{2\eta} \sum_{t=1}^T (\|w_t - \dot{w}\|^2 - \|w_{t+1} - \dot{w}\|^2) + \frac{\eta}{2} \sum_{t=1}^T \|g_t\|^2 \\
 &= \frac{1}{2\eta} (\|w_1 - \dot{w}\|^2 - \|w_{T+1} - \dot{w}\|^2) + \frac{\eta}{2} \sum_{t=1}^T \|g_t\|^2 \\
 &\leq \frac{\|w_1 - \dot{w}\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^T \|g_t\|^2
 \end{aligned}$$

The last statement follows from observing that if each $\|g_t\| \leq G_{\max}$, then $\sum_{t=1}^T \|g_t\|^2 \leq G_{\max}^2 T$. \square

This simple Proposition shows that (with the right learning rate η) online subgradient descent already obtains sublinear regret! Surprisingly, this simple algorithm is actually optimal in a certain sense: there exist domains W and environments such that for any specified G_{\max} , no algorithm can guarantee better regret than $\frac{G_{\max} D \sqrt{T}}{2}$.

Proposition 2.19 ([1]). *Let $n \geq 3$ and set W to be the ball of radius $\frac{D}{2}$ in \mathbb{R}^n . Then for any G_{\max} and any online learning algorithm playing points $w_t \in W$ in response to $g_t \in \mathbb{R}^n$ with $\|g_t\| \leq G_{\max}$, there exists a strategy for choosing g_t such that*

$$R_T(\dot{w}) = \|\dot{w}\| G_{\max} \sqrt{T}$$

for some $\dot{w} \in W$ with $\|\dot{w}\| = D/2$.

Proof. The argument is surprisingly simple: since $n \geq 3$, given any two vectors $x, y \in \mathbb{R}^n$, we can always find a vector z with $\langle z, x \rangle = \langle z, y \rangle = 0$ and $\|z\| = G_{\max}$. Thus for each t we pick g_t such that $\langle g_t, w_t \rangle = \langle g_t, \sum_{i=1}^{t-1} g_i \rangle = 0$ and $\|g_t\| = G_{\max}$. We set $\dot{w} = -\frac{D}{2} \frac{\sum_{t=1}^T g_t}{\|\sum_{t=1}^T g_t\|} \in W$, and then compute:

$$\begin{aligned}
 R_T(\dot{w}) &= \sum_{t=1}^T \langle g_t, w_t - \dot{w} \rangle \\
 &= - \sum_{t=1}^T \langle g_t, \dot{w} \rangle \\
 &= \frac{D}{2} \left\| \sum_{t=1}^T g_t \right\|
 \end{aligned}$$

So now it remains to show that $\left\| \sum_{t=1}^T g_t \right\| = G_{\max} \sqrt{T}$. We proceed by induction. Suppose $\left\| \sum_{t=1}^k g_t \right\| = G_{\max} \sqrt{k}$ for some k . Then

$$\begin{aligned} \left\| \sum_{t=1}^{k+1} g_t \right\|^2 &= \left\| \sum_{t=1}^k g_t \right\|^2 + 2 \left\langle \sum_{t=1}^{k+1} g_t, g_{k+1} \right\rangle + \|g_{k+1}\|^2 \\ &= G_{\max}^2 k + 0 + G_{\max}^2 \\ &= G_{\max}^2 (k+1) \end{aligned}$$

which completes the induction as well as the proof. \square

This algorithm and lower bound set the basic benchmarks for the field of online convex optimization: a good algorithm should guarantee regret $R_T(\hat{w}) = O(\sqrt{T})$, and no algorithm can guarantee a better dependence on T in the worst case. Of course if we make additional assumptions about the environment (e.g. strongly-convex losses) it is possible to do better than $O(\sqrt{T})$ regret, and we will explore these possibilities later.

2.3 Follow the Regularized Leader

In this section we introduce the follow-the-regularized-leader (FTRL) approach to designing online learning algorithms. This is a classic and very powerful technique that has been used to great effect in the literature (see [33; 52] for more comprehensive surveys). The intuition behind this technique stems from the closely related follow-the-leader (FTL) approach, which we will describe first.

The FTL algorithm first chooses some w_1 via some arbitrary rule, and then sets w_{t+1} according to:

$$w_{t+1} = \operatorname{argmin}_{w \in W} \sum_{i=1}^t \ell_i(w)$$

This has the intuitively pleasing property of being directly analogous to the empirical risk minimization (ERM) procedure in stochastic optimization: each w_{t+1} is essentially minimizing the “training error” at time t .

Unfortunately, however, FTL is not guaranteed to perform well. To see this, consider the following classical counterexample: in an online linear optimization game, set $W = [-1, 1]$, $g_1 = \frac{1}{2}$, and $g_t = (-1)^t$ for $t > 1$. Then we have $\sum_{i=1}^t \ell_i(w) = \frac{1}{2}(-1)^{t+1}w$ for all t , and so the FTL algorithm will set $w_t = (-1)^t$ for all $t > 1$. Therefore $\sum_{t=1}^T g_t w_t \geq T - \frac{1}{2}$. This yields $R_T(0) \geq T - \frac{1}{2}$, which is not sublinear regret.

The failure of FTL in this scenario is due to some inherent instability of the algorithm: it spends all its time bouncing back and forth between -1 and 1 , while stabilizing on any constant value would actually suffice. This is directly analogous to the phenomenon of over-fitting in stochastic optimization, and the way to fix it is also directly analogous: regularization.

The FTRL approach is simple modification to FTL: at each t we choose a *regularizer* function $\psi_t : W \rightarrow \mathbb{R}$, and set

$$w_{t+1} = \operatorname{argmin}_{w \in W} \psi_t(w) + \sum_{i=1}^t \ell_i(w)$$

Intuitively, we should choose regularizers ψ_t that stabilize the w_t predictions, for example by “pulling” them in towards the origin.

A classic example is the constant sequence of regularizers $\psi_t(w) = \frac{1}{2\eta} \|w\|^2$ for all t , for some η . A little bit of algebra shows that on linear losses with an unbounded W , this yields $w_{t+1} = w_t - \eta g_t$, which is exactly the online subgradient descent update! More formally, the update is

$$w_{t+1} = \Pi_W \left(-\eta \sum_{i=1}^t g_i \right)$$

which is often called online subgradient descent with *lazy* projections rather than the *greedy* projections of our algorithm in the previous section.

The classic analysis of the FTRL algorithm stems from the so-called FTRL-lemma:

Lemma 2.20 (FTRL-Lemma, see [33; 52] for proofs). *Suppose that $\psi_t(w) \geq \psi_{t-1}(w)$ for all w and t . Then the regret of the FTRL algorithm is bounded by*

$$R_T(\hat{w}) \leq \psi_T(\hat{w}) + \sum_{t=1}^T \ell_t(w_t) - \ell_t(w_{t+1})$$

From this Lemma, we have the following important corollary:

Corollary 2.21. *Suppose ψ_t is μ_t -strongly convex and $\psi_t(w) \geq \psi_{t-1}(w)$ for all w and t . Further, suppose ℓ_t is G_t Lipschitz. Then the regret of the FTRL algorithm is bounded by*

$$R_T(\hat{w}) \leq \psi_T(\hat{w}) + \sum_{t=1}^T \frac{G_t^2}{\mu_t}$$

If we apply this corollary to the regularizers $\psi_t(w) = \frac{1}{2\eta} \|w\|^2$, which are $\frac{1}{\eta}$ -strongly convex, then we get a regret of

$$R_T(\hat{w}) \leq \frac{\|\hat{w}\|^2}{2\eta} + TG_{\max}^2 \eta$$

so that by setting $\eta = \frac{\|\hat{w}\|}{G_{\max} \sqrt{T}}$, we again recover $O(G_{\max} \|\hat{w}\| \sqrt{T})$ regret.

2.4 Mirror Descent

The Mirror Descent (MD) framework is an alternative method for designing online linear optimization algorithms [55; 5]. Instead of enforcing stability by adding a regularizer to the the empirical risk minimizer, as in

FTRL, MD enforces stability by explicitly encouraging w_{t+1} to be close to w_t .

Unlike FTRL, whose update formula can be stated for general online learning problems, our Mirror Descent update is usually only given for online linear optimization problems. To compute w_{t+1} given w_t and g_t , choose a regularizer function $\psi_t : W \rightarrow \mathbb{R}$ and set w_{t+1} by the formula:

$$w_{t+1} = \operatorname{argmin}_{w \in W} \langle g_t, w \rangle + B_{\psi_t}(w, w_t)$$

where here B_{ψ_t} denotes the Bregman divergence (see Definition 2.16). Again, when we set $\psi_t(w) = \frac{1}{\eta} \|w\|^2$, we observe

$$w_{t+1} = \Pi_W(w_t - \eta g_t)$$

which is exactly the same as our (greedy) online subgradient descent update.

Mirror Descent enjoys the regret bound of the following theorem:

Lemma 2.22 (see e.g. [18]). *Mirror descent achieves regret:*

$$\begin{aligned} R_T(\hat{w}) &\leq B_{\psi_1}(\hat{w}, w_1) - B_{\psi_T}(\hat{w}, w_{T+1}) + \sum_{t=2}^T B_{\psi_t}(\hat{w}, w_t) - B_{\psi_{t-1}}(\hat{w}, w_t) \\ &\quad + \sum_{t=1}^T -B_{\psi_t}(w_{t+1}, w_t) + \langle g_t, w_t - w_{t+1} \rangle \end{aligned}$$

Just as Lemma 2.20 implies Corollary 2.21 for FTRL with strongly-convex regularizers, the above Lemma 2.22 implies the following Corollary for strongly-convex regularizers:

Corollary 2.23. *Suppose each $\psi_t = \frac{1}{\eta_t} \psi$ for some fixed function ψ non-increasing sequence η_t . Suppose ψ is 1-strongly convex with respect to a norm $\|\cdot\|$. Then the regret of Mirror Descent is bounded by*

$$R_T(\hat{w}) \leq \frac{1}{\eta_T} \max_{t \leq T} B_{\psi_t}(\hat{w}, w_t) + \sum_{t=1}^T \frac{\eta_t \|g_t\|_*^2}{2}$$

If we set $\psi_t(w) = \frac{G_{\max}}{2D\sqrt{T}} \|w\|^2$, which gives us the subgradient descent update, we can apply the above corollary to recover the subgradient descent regret guarantee:

$$R_T(\hat{w}) \leq O(DG_{\max}\sqrt{T})$$

where D is the diameter of W .

2.5 Coin Betting

The coin betting framework is a recent addition to the general online convex optimization toolkit [40], being originally introduced in the context of portfolio optimization [46]. Similar to Mirror Descent, the coin betting

framework is only formulated for the online linear optimization framework. To simplify exposition here, we will confine ourselves to a 1-dimensional online linear optimization setting in which g_t must be in $[-1, 1]$ for all t (i.e. $G_{\max} = 1$). This will allow us to very briefly sketch the main idea - we will provide a more in-depth description later in Chapter 5.

In our 1-dimensional online linear optimization setup, the regret can be written as:

$$R_T(\dot{w}) = \sum_{t=1}^T g_t w_t - g_t \dot{w}$$

The conceptual idea of the coin betting framework is to pretend that the quantity $-\sum_{t=1}^T g_t w_t$ represents some amount of “money” won by a gambler. To aid this concept, we specify a number $\epsilon > 0$ called the “initial endowment”, and define the “wealth” of the algorithm at time T as

$$\text{Wealth}_T = \epsilon - \sum_{t=1}^T g_t w_t$$

A coin betting algorithm “bets” a signed fraction $v_t \in [-1, 1]$ of its current wealth on the outcome of the “coin” $g_t \in [-1, 1]$ by playing $w_t = v_t \text{Wealth}_{t-1}$, so that $\text{Wealth}_t = \text{Wealth}_{t-1} - g_t v_t \text{Wealth}_{t-1}$. The advantage of this approach is that high wealth is equivalent to low regret [35], but lower-bounding the wealth of an algorithm may be conceptually simpler than upper-bounding the regret because \dot{w} does not appear in the definition of wealth.

To see the connection between high wealth and low regret, we can re-write the regret as

$$R_T(\dot{w}) = \epsilon - \text{Wealth}_T - \sum_{t=1}^T g_t \dot{w}$$

Now suppose we can lower bound the wealth with a statement of the form

$$\text{Wealth}_T \geq f_T \left(- \sum_{t=1}^T g_t \right) \tag{2.1}$$

for some function f_T . Then we can conclude

$$\begin{aligned} R_T(\dot{w}) &= \epsilon - \text{Wealth}_T - \sum_{t=1}^T g_t \dot{w} \\ &\leq \epsilon + \left(- \sum_{t=1}^T g_t \right) \dot{w} - f_T \left(- \sum_{t=1}^T g_t \right) \\ &\leq \epsilon + \sup_X X \dot{w} - f_T(X) \\ &= \epsilon + f_T^*(\dot{w}) \end{aligned}$$

where the last equality is simply the definition of f_T^* (see Definition 2.15). Thus by obtaining a lower-bound on the wealth, we obtain an upper-bound on the regret.

Prior analyses of coin betting algorithms [40; 43] use particular *potential functions*, similar to the regularizers of FTRL, to choose the betting fractions v_t . Our methods for choosing v_t , outlined in Chapter 5, are rather divorced from this approach and we will postpone all discussion of how to choose v_t to that Chapter.

2.6 Online Relaxations

The final algorithmic design framework we will discuss is online relaxations [19; 50]. This framework, like coin betting, is somewhat less classical than the FTRL and Mirror Descent methods, and so we will only briefly summarize the big idea in this section. The starting point for the online relaxation framework is the minimax strategy: suppose we are given that all loss functions ℓ_t will lie in some set \mathcal{L} . Then we might play a w_t according to the “worst-case” rule:

$$w_t = \operatorname{argmin}_{w_t \in W} \sup_{\ell_t \in \mathcal{L}} \dots \inf_{w_t \in W} \sup_{\ell_T \in \mathcal{L}} \sup_{\dot{w} \in W} \left[\sum_{t=1}^T \ell_t(w_t) - \ell_t(\dot{w}) \right]$$

Essentially, the minimax strategy chooses each point w_t so as to minimize the worst-case effect of the environment. When we use this strategy, we obtain the regret guarantee:

$$R_T(\dot{w}) \leq \inf_{w_1 \in W} \sup_{\ell_1 \in \mathcal{L}} \dots \inf_{w_t \in W} \sup_{\ell_T \in \mathcal{L}} \sup_{\dot{w} \in W} \left[\sum_{t=1}^T \ell_t(w_t) - \ell_t(\dot{w}) \right]$$

which is known as the *value* of the online learning game. The minimax strategy is choosing the point w_t that minimizes the *conditional value* of the game:

$$V_i = V(\ell_1, \dots, \ell_t, w_1, \dots, w_t) = \inf_{w_t \in W} \sup_{\ell_t \in \mathcal{L}} \dots \inf_{w_t \in W} \sup_{\ell_T \in \mathcal{L}} \sup_{\dot{w} \in W} \left[\sum_{i=1}^T \ell_i(w_i) - \sum_{i=1}^T \ell_i(\dot{w}) \right]$$

Unfortunately, the minimax strategy may not be easy to compute. This is where the idea of relaxations come in. The basic idea is to design a more computationally tractable function Rel that upper bounds the conditional value of the game:

$$\operatorname{Rel}_i \geq V_i$$

Then (under suitable conditions on the function Rel_T), we play

$$w_t = \operatorname{argmin}_w \sup_{\ell_t} [\ell_t(w_t) + \operatorname{Rel}_{t-1}] \tag{2.2}$$

and we obtain regret

$$R_T(\hat{w}) \leq \text{Rel}_T$$

A good relaxation should have two important properties. First, it should be possible to efficiently compute the value w_t via (2.2). Second, the gap $\text{Rel}_i - V_i$ should be small so that the regret $R_T(\hat{w})$ is close to the optimal value.

2.7 Conclusions

In this chapter we have presented an overview of key facts from convex analysis as well as a very rapid look at the major prior techniques for designing online learning algorithms. Of these, we will make use of the FTRL and coin betting frameworks in this thesis; the Mirror Descent and online relaxations frameworks are presented here only for completeness. In the next chapter we will introduce the problem of hyperparameter tuning and give an overview of the prior work in this area. **For the next three chapters, we will consider exclusively the online linear optimization problem**, in which each loss ℓ_t has the form $\ell_t(w) = \langle g_t, w \rangle$. We will return to nonlinear losses in Chapter 7

Chapter 3

Prior Adaptive Online Algorithms

Now that we have covered the basic mathematical concepts, it is time to introduce the notion of hyper-parameters in online learning. To do this we recall the online subgradient descent algorithm online linear optimization when W is a subset of a Hilbert space, presented in Section 2.2:

$$w_{t+1} = \Pi_W(w_t - \eta g_t)$$

If we assume $w_1 = 0$ (WLOG) and that $\|g_t\| \leq G_{\max}$ for all t , then by Proposition 2.18 we have

$$R_T(\hat{w}) \leq \frac{\|\hat{w}\|^2}{2\eta} + \frac{\eta}{2} T G_{\max}^2$$

So that with the optimal $\eta = \frac{\|\hat{w}\|}{G_{\max}\sqrt{T}}$ we obtain

$$R_T(\hat{w}) \leq \|\hat{w}\| G_{\max} \sqrt{T} \tag{3.1}$$

Further, recall from Proposition 2.19 that, at least up to constant factors, we actually cannot improve on this regret bound.

Although the regret bound (3.1) is optimal, we were only able to achieve it by a particular setting for the parameter η , which is usually called the *learning rate*. If we set η incorrectly, then the regret may be quite a bit worse.

To illustrate this, consider a simple example: set $W = \mathbb{R}$ and let $\ell_t(w) = |1 - w| + z_t w$ where each z_t is a uniformly random element of $\{\pm 2\}$. Then clearly we have $\mathbb{E}[\ell_t(w)] = |1 - w|$, so the natural choice for \hat{w} is 1. We plot the value of $R_T(\hat{w})$ with $T = 1000$ for various choices of η in Figure 3.1. From the plot, one can observe that there is a clear optimal choice for η .

In many practical settings we do not know the true value of the parameters G_{\max} or $\|\hat{w}\|$, and so our only recourse is to simply guess a value. This necessarily results in slower optimization, and so suggests the

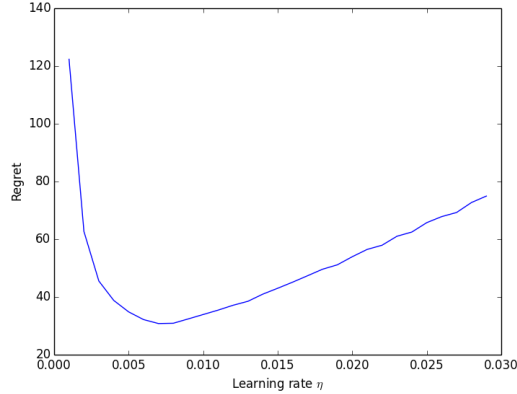


Figure 3.1: Regret of Online Subgradient Descent after 1000 iterations as a function of η

following question, which is central to this thesis:

How can we adapt to unknown parameters?

In this chapter we will give a brief overview of some prior approaches to this problem.

3.1 Adapting to G_{\max}

First we will cover methods for adapting to the Lipschitz constant G_{\max} . The result presented here is deliberately given using elementary techniques, but significantly improved and more general results can be found in prior works [25; 17; 36].

In order to adapt to unknown values for G_{\max} , we will need to slightly generalize our analysis of online subgradient descent to allow for varying learning rates. Let us consider the case where our domain W is an entire Hilbert space, and we are given the value of $\|\dot{w}\|$ ahead of time. Let B be the ball of radius $\|\dot{w}\|$ in W . Then set $w_1 = 0$ and consider the update equation

$$w_{t+1} = \Pi_B(w_t - \eta_t g_t)$$

where now η_t is allowed to vary from round to round, and even may depend on g_t . We can mimic the proof of Proposition 2.18 to obtain:

Proposition 3.1. *Suppose $w_1 = 0$ and $w_{t+1} = \Pi_B w_t - \eta_t g_t$ for all $t \geq 2$. Further, suppose $\eta_t \leq \eta_{t-1}$ for all t . Then we have*

$$R_T(\dot{w}) \leq \frac{5\|\dot{w}\|^2}{2\eta_T} + \sum_{t=1}^T \frac{\eta_t}{2} \|g_t\|^2$$

Proof. Similar to the proof of Proposition 2.18, let $x_{t+1} = w_t - \eta_t g_t$. Then we have $w_{t+1} = \Pi_B(x_{t+1})$ so that

$$\begin{aligned} \|w_{t+1} - \dot{w}\|^2 &\leq \|x_{t+1} - \dot{w}\|^2 \\ &= \|w_t - \eta_t g_t - \dot{w}\|^2 \\ &= \|w_t - \dot{w}\|^2 - 2\eta_t \langle g_t, w_t - \dot{w} \rangle + \eta_t^2 \|g_t\|^2 \\ \langle g_t, w_t - \dot{w} \rangle &\leq \frac{1}{2\eta_t} (\|w_t - \dot{w}\|^2 - \|w_{t+1} - \dot{w}\|^2) + \frac{\eta_t}{2} \|g_t\|^2 \end{aligned}$$

Now observe that $R_T(\dot{w}) = \sum_{t=1}^T \langle g_t, w_t - \dot{w} \rangle$ to obtain

$$\begin{aligned} R_T(\dot{w}) &\leq \sum_{t=1}^T \frac{1}{2\eta_t} (\|w_t - \dot{w}\|^2 - \|w_{t+1} - \dot{w}\|^2) + \frac{\eta_t}{2} \sum_{t=1}^T \|g_t\|^2 \\ &= \left(\frac{1}{2\eta_1} \|w_1 - \dot{w}\|^2 - \frac{1}{2\eta_T} \|w_{T+1} - \dot{w}\|^2 \right) + \sum_{t=2}^T \left(\frac{1}{2\eta_t} - \frac{1}{2\eta_{t-1}} \right) \|w_t - \dot{w}\|^2 + \sum_{t=1}^T \frac{\eta_t}{2} \|g_t\|^2 \\ &\leq \left(\frac{1}{2\eta_1} \|w_1 - \dot{w}\|^2 - \frac{1}{2\eta_T} \|w_{T+1} - \dot{w}\|^2 \right) + \sum_{t=2}^T \left(\frac{1}{2\eta_t} - \frac{1}{2\eta_{t-1}} \right) 4\|\dot{w}\|^2 + \sum_{t=1}^T \frac{\eta_t}{2} \|g_t\|^2 \\ &= \left(\frac{1}{2\eta_1} \|w_1 - \dot{w}\|^2 - \frac{1}{2\eta_T} \|w_{T+1} - \dot{w}\|^2 \right) + \frac{2\|\dot{w}\|^2}{\eta_T} - \frac{2\|\dot{w}\|^2}{\eta_1} + \sum_{t=1}^T \frac{\eta_t}{2} \|g_t\|^2 \\ &\leq \frac{5\|\dot{w}\|^2}{2\eta_T} + \sum_{t=1}^T \frac{\eta_t}{2} \|g_t\|^2 \end{aligned}$$

where in the last line we used $w_1 = 0$ and dropped negative values. \square

The final ingredient is a generalization of the observation that $\sum_{t=1}^T \frac{1}{\sqrt{t}} \leq 2\sqrt{T}$:

Lemma 3.2. *If x_1, \dots, x_T are non-negative numbers, such that $x_1 \geq 0$ then*

$$\sum_{t=1}^T \frac{x_t}{\sqrt{\sum_{i=1}^t x_i}} \leq 2\sqrt{\sum_{t=1}^T x_t}$$

Proof. By convexity of square root, we have

$$\frac{x_t}{2\sqrt{\sum_{i=1}^t x_i}} \leq \sqrt{\sum_{i=1}^t x_i} - \sqrt{\sum_{i=1}^{t-1} x_i}$$

Summing over t and telescoping the right hand side completes the proof. \square

With this Lemma and Proposition in hand, we can propose an adaptive learning rate schedule: set

$$\eta_t = \frac{\|\dot{w}\|}{\sqrt{\sum_{i=1}^t \|g_i\|^2}}$$

to obtain

$$\begin{aligned} R_T(\dot{w}) &\leq \frac{5\|\dot{w}\|^2}{2\eta_T} + \sum_{t=1}^T \frac{\eta_t}{2} \|g_t\|^2 \\ &\leq \frac{7}{2} \|\dot{w}\| \sqrt{\sum_{t=1}^T \|g_t\|^2} \\ &\leq \frac{7}{2} \|\dot{w}\| G_{\max} \sqrt{T} \end{aligned}$$

This result says that, so long as we know the value of $\|\dot{w}\|$, we can adapt to an unknown G_{\max} value simply by adaptively tuning the learning rate in online subgradient descent. This is extremely encouraging: we are able to pay only a constant factor for this extra adaptivity! In fact, the bound $\|\dot{w}\| \sqrt{\sum_{t=1}^T \|g_t\|^2}$ can be quite a bit better than $\|\dot{w}\| G_{\max} \sqrt{T}$ in some settings, as we will discuss later in Section 7.1.

3.2 Adapting to $\|\dot{w}\|$

In this section we summarize prior work on adapting to an unknown value for $\|\dot{w}\|$, given the value of G_{\max} . The algorithms in this section are quite a bit more complicated than in the previous section, so we will only state results without any proofs.

The first algorithm to adapt to $\|\dot{w}\|$ we are aware of is that of [32]. In a 1-dimensional problem with $W = \mathbb{R}$, and given any user-specified ϵ , their algorithm obtains a regret bound of

$$R_T(\dot{w}) \leq \frac{\sqrt{2}}{\sqrt{2}-1} |\dot{w}| \sqrt{T G_{\max}^2 + 1} \left(\log \left(\frac{|\dot{w}|}{\epsilon} (2G_{\max}^2 T + 2)^{5/2} \right) - 1 \right) + \epsilon$$

There are two peculiarities of this bound. The first is the presence of the extra parameter ϵ . The second is the additional logarithmic term. Let us unpack each of these in turn.

The parameter ϵ is the *regret at the origin*: $R_T(0) \leq \epsilon$. It is fairly easy to see that no algorithm can actually guarantee $R_T(0) \leq 0$ without knowing $\|\dot{w}\| = 0$ ahead of time, and so the value ϵ parametrizes how much origin-regret we are willing to tolerate. Notice that while this is certainly an extra parameter, it has a relatively mild effect on the bound: the additive ϵ term does not grow with T , and ϵ is only present inside the logarithm of the term that does grow with T .

The logarithmic term makes this bound asymptotically worse than the ideal $\|\dot{w}\| G_{\max} \sqrt{T}$ we are looking for. Unfortunately, it turns out to be unavoidable: in the same work, [32] prove that any algorithm that

guarantees $R_T(0) \leq \epsilon$ can be made to suffer regret at least:

$$R_T(\hat{w}) \geq \Omega \left(\|\hat{w}\| G_{\max} \sqrt{T \log \left(\frac{\|\hat{w}\| G_{\max} \sqrt{T}}{\epsilon} \right)} \right)$$

for some $\|\hat{w}\|$. Notice that if we set $\epsilon = \Omega(\|\hat{w}\| G_{\max} \sqrt{T})$, then the logarithmic term disappears. Unfortunately, since we do not know $\|\hat{w}\|$, this is not possible. This lower bound implies that the adapting to unknown $\|\hat{w}\|$ is in fact fundamentally harder than adapting to an unknown G_{\max} , by an extra logarithmic factor.

After the initial work of [32], [38] developed an improved algorithm that operates when W is a Hilbert space rather than simple 1-dimensional domain, again achieving a regret guarantee of

$$R_T(\hat{w}) = O(\|\hat{w}\| G_{\max} \sqrt{T} \log(\|\hat{w}\| G_{\max} \sqrt{T} / \epsilon + 1) + \epsilon)$$

Later a variety of subsequent works [39; 35; 19; 40] were able to improve the logarithmic factor in the algorithms to achieve regret matching the lower bound in any Hilbert space:

$$R_T(\hat{w}) = O(\|\hat{w}\| G_{\max} \sqrt{T \log(\|\hat{w}\| G_{\max} \sqrt{T} / \epsilon + 1) + \epsilon})$$

Interestingly, although we saw that algorithms that adapt to unknown G_{\max} can achieve a dependence on $\sqrt{\sum_{t=1}^T \|g_t\|_*^2}$ rather than $G_{\max} \sqrt{T}$, the best that any of these algorithms that adapt to $\|\hat{w}\|$ obtain is $\sqrt{G_{\max} \sum_{t=1}^T \|g_t\|_*}$. We will be able to improve upon this later in Chapter 5.

3.3 Strong Convexity

In this section we will consider *strongly convex* losses ℓ_t . Recall that a function ℓ_t is μ -strongly convex if

$$\ell_t(x + \delta) \geq \ell_t(x) + \langle g, \delta \rangle + \frac{\mu}{2} \|\delta\|^2$$

for all x and δ and $g \in \partial \ell_t(x)$. This implies that

$$\begin{aligned} \ell_t(w_t) - \ell_t(\hat{w}) &\leq \langle g_t, w_t - \hat{w} \rangle - \frac{\mu}{2} \|w_t - \hat{w}\|^2 \\ R_T(\hat{w}) &\leq \sum_{t=1}^T \langle g_t, w_t - \hat{w} \rangle - \frac{\mu}{2} \|w_t - \hat{w}\|^2 \end{aligned}$$

The above observation gives us a hint that strong convexity may make an online learning problem fundamentally easier than online linear optimization because the $-\frac{\mu}{2} \|w_t - \hat{w}\|^2$ strictly decreases the regret. We can indeed take advantage of this inequality using online subgradient descent with a different learning rate (as observed in [24]): instead of $\eta_t = O(1/\sqrt{t})$, we use $\eta_t = O(1/\mu t)$.

Proposition 3.3. *Suppose $w_1 = 0$ and $w_{t+1} = \Pi_B w_t - \eta_t g_t$ for all $t \geq 2$ with $\eta_t = \frac{1}{\mu t}$. Then we have*

$$R_T(\hat{w}) \leq \frac{G_{\max}^2}{\mu} \log(T)$$

Proof. Similar to the proof of Proposition 3.1, we have

$$\langle g_t, w_t - \hat{w} \rangle \leq \frac{1}{2\eta_t} (\|w_t - \hat{w}\|^2 - \|w_{t+1} - \hat{w}\|^2) + \frac{\eta_t}{2} \|g_t\|^2$$

Now observe that $R_T(\hat{w}) = \sum_{t=1}^T \langle g_t, w_t - \hat{w} \rangle - \frac{\mu}{2} \|w_t - \hat{w}\|^2$ to obtain

$$\begin{aligned} R_T(\hat{w}) &\leq \sum_{t=1}^T \frac{1}{2\eta_t} (\|w_t - \hat{w}\|^2 - \|w_{t+1} - \hat{w}\|^2) - \frac{\mu}{2} \|w_t - \hat{w}\|^2 + \frac{\eta_t}{2} \sum_{t=1}^T \|g_t\|^2 \\ &= \left(\frac{1}{2\eta_1} - \frac{\mu}{2} \right) \|w_1 - \hat{w}\|^2 - \frac{1}{2\eta_T} \|w_{T+1} - \hat{w}\|^2 + \sum_{t=2}^T \left(\frac{1}{2\eta_t} - \frac{\mu}{2} \frac{1}{2\eta_{t-1}} \right) \|w_t - \hat{w}\|^2 + \sum_{t=1}^T \frac{\eta_t}{2} \|g_t\|^2 \\ &= \sum_{t=1}^T \frac{\eta_t}{2} \|g_t\|^2 \\ &\leq \frac{G_{\max}^2}{2\mu} \sum_{t=1}^T \frac{1}{t} \\ &\leq \frac{G_{\max}^2}{\mu} \log(T) \end{aligned}$$

□

Thus we see that strong convexity allows us to go from \sqrt{T} regret to $\log(T)$ regret. Notice that the algorithm seems to adapt to G_{\max} without any information about it, but that it does require knowledge of μ .

This concludes our tour of the prior work in adaptive online learning. The next chapters will describe our contributions to the field, beginning with lower bounds for adapting to $\|w\|$ and G_{\max} simultaneously.

Chapter 4

An Optimal Parameter-free Algorithm

In this chapter we first prove a lower bound on the regret of any algorithm that must adapt to both G_{\max} and $\|\dot{w}\|$, and then provide an algorithm that meets this lower bound in any real Hilbert space. I originally published a sub-optimal lower bound and algorithm in [13], and then improved both to the optimal results presented here in [12], which forms the core of the prose in this chapter.

4.1 A Lower-Bound for Adapting to $\|\dot{w}\|$ and G_{\max}

The main result of this section is that, while prior work can adapt to either $\|\dot{w}\|$ or G_{\max} with relatively small penalty to the regret, it is actually impossible to adapt to both simultaneously without suffering a potentially exponential penalty¹. If we define $G_t = \max_{i \leq t} \|g_i\|_*$, there is in fact a *frontier* of lower bounds that trade-off between a $\|\dot{w}\| G_{\max} \sqrt{T} \log(\|\dot{w}\| T)$ term and a $\exp(\max_t G_t / G_{t-1})$ term along two dimensions, which we parametrize by k and γ . Along the first dimension, the exponential penalty is reduced to $\exp((G_t / G_{t-1}) / k^2)$ for any $k > 0$ at the expense of rescaling the regret's \sqrt{T} term to $k \|\dot{w}\| G_{\max} \sqrt{T} \log(\|\dot{w}\| T)$. Along the second dimension, the logarithm's power in the \sqrt{T} term is reduced to $\|\dot{w}\| G_{\max} \sqrt{T} \log^\gamma(\|\dot{w}\| T)$ for any $\gamma \in (1/2, 1]$ at the expense of increasing the exponential penalty to $\exp((G_t / G_{t-1})^{1/(2\gamma-1)})$. We prove the lower bounds by constructing a specific adversarial loss sequence, and then we provide a family of algorithms whose regret matches the lower bound frontier for any k and γ . First we describe our adversarial loss sequence and lower bound frontier along the k dimension, and then we extend the argument to obtain the full two dimensional frontier parametrized by both k and γ .

4.1.1 Warm-up: a Suboptimal Lower-Bound

Before describing our full frontier, we prove a somewhat worse bound (Theorem 4.1) whose proof communicates the main intuition behind our optimal method. The proof of the full lower bound (Theorem 4.3) is

¹This lower bound resolved a COLT 2016 open problem (Parameter-Free and Scale-Free Online Algorithms) [41] in the negative.

significantly more complicated, and postponed to Section 4.A.

Theorem 4.1. *For any constants $c, d, \epsilon > 0$, there exists a T and an adversarial strategy picking $g_t \in \mathbb{R}$ in response to the outputs of an online learning algorithm choosing $w_t \in \mathbb{R}$ such that regret is:*

$$\begin{aligned} R_T(\dot{w}) &= \sum_{t=1}^T g_t w_t - g_t \dot{w} \\ &\geq (d + c\|\dot{w}\| \log \|\dot{w}\|) G_{\max} \sqrt{T} \log(G_{\max} + 1) + dG_{\max} \exp((2T)^{1/2-\epsilon}) \\ &\geq (d + c\|\dot{w}\| \log \|\dot{w}\|) G_{\max} \sqrt{T} \log(G_{\max} + 1) + dG_{\max} \exp \left[\left(\max_t \frac{G_t}{G_{t-1}} \right)^{1/2-\epsilon} \right] \end{aligned}$$

for some $\dot{w} \in \mathbb{R}$ where $G_{\max} = \max_{t \leq T} \|g_t\|$ and $G_t = \max_{t' \leq t} \|g_{t'}\|$.

Proof. We prove the theorem by showing that for sufficiently large T , the adversary can “checkmate” the learner by presenting it only with the subgradient $g_t = -1$. If the learner fails to have w_t increase quickly, then there is a $\dot{w} \gg 1$ against which the learner has high regret. On the other hand, if the learner ever does make w_t higher than a particular threshold, the adversary immediately punishes the learner with a subgradient $g_t = 2T$, again resulting in high regret.

Let T be large enough such that both of the following hold:

$$\frac{T}{4} \exp\left(\frac{T^{1/2}}{4 \log(2)c}\right) > k \log(2) \sqrt{T} + k \exp((2T)^{1/2-\epsilon}) \quad (4.1)$$

$$\frac{T}{2} \exp\left(\frac{T^{1/2}}{4 \log(2)c}\right) > 2kT \exp((2T)^{1/2-\epsilon}) + 2kT \sqrt{T} \log(2T + 1) \quad (4.2)$$

The adversary plays the following strategy: for all $t \leq T$, so long as $w_t < \frac{1}{2} \exp(T^{1/2}/4 \log(2)c)$, give $g_t = -1$. As soon as $w_t \geq \frac{1}{2} \exp(T^{1/2}/4 \log(2)c)$, give $g_t = 2T$ and $g_t = 0$ for all subsequent t . Let’s analyze the regret at time T in these two cases.

Case 1: $w_t < \frac{1}{2} \exp(T^{1/2}/4 \log(2)c)$ for all t :

In this case, let $u = \exp(T^{1/2}/4 \log(2)c)$. Then $G_{\max} = 1$, $\max_t \frac{G_t}{G_{t-1}} = 1$, and using (4.1) the learner’s regret is at least

$$\begin{aligned} R_T(u) &\geq Tu - T \frac{1}{2} \exp\left(\frac{T^{1/2}}{4 \log(2)c}\right) \\ &= \frac{1}{2} Tu \\ &= cu \log(u) \sqrt{T} \log(2) + \frac{T}{4} \exp\left(\frac{T^{1/2}}{4 \log(2)c}\right) \\ &> cu \log(u) G_{\max} \sqrt{T} \log(G_{\max} + 1) + k G_{\max} \sqrt{T} \log(L_{\max} + 1) + k L_{\max} \exp((2T)^{1/2-\epsilon}) \\ &= (k + cu \log u) G_{\max} \sqrt{T} \log(G_{\max} + 1) + k G_{\max} \exp \left[(2T)^{1/2-\epsilon} \right] \end{aligned}$$

Case 2: $w_t \geq \frac{1}{2} \exp(T^{1/2}/4 \log(2)c)$ for some t :

In this case, $G_{\max} = 2T$ and $\max_t \frac{G_t}{G_{t-1}} = 2T$. For $u = 0$, using (4.2), the regret is at least

$$\begin{aligned} R_T(u) &\geq \frac{T}{2} \exp\left(\frac{T^{1/2}}{4\log(2)c}\right) \\ &\geq 2kT \exp((2T)^{1/2-\epsilon}) + 2kT\sqrt{T} \log(2T + 1) \\ &= kG_{\max} \exp((2T)^{1/2-\epsilon}) + kG_{\max}\sqrt{T} \log(G_{\max} + 1) \\ &= (k + cu \log u)L_{\max}\sqrt{T} \log(G_{\max} + 1) + kG_{\max} \exp\left[(2T)^{1/2-\epsilon}\right] \end{aligned}$$

□

4.1.2 Optimal Bound: Trade-offs in the multiplicative constant k

In the next two sections we will provide the tight version of our lower bound. Given an algorithm, we establish a lower bound on its performance by constructing an adversarial sequence of subgradients $g_t \in \mathbb{R}$. This sequence sets $g_t = -1$ for $T - 1$ iterations, where T is chosen adversarially but can be made arbitrarily large, then sets $g_T = O(k\sqrt{T})$ in a very similar manner to the warm-up in the previous section. We tighten the analysis to prove that this simple strategy forces the algorithm to experience regret that is exponential in \sqrt{T}/k . We then express \sqrt{T}/k as a constant multiple of $\frac{1}{k^2} G_t/G_{t-1}$, where $G_t = \max_{t' \leq t} |g_{t'}|$, capturing the algorithm's sensitivity to the big jump in the gradients between $T - 1$ and T in the adversarial sequence. Note that although our lower bound is stated for the case where $W = \mathbb{R}$, it equally well applies to any real vector space by choosing any 1-dimensional subspace and considering the projections of w_t to that subspace.

The cost that an algorithm pays when faced with the adversarial sequence is stated formally in the following Theorem.

Theorem 4.2. *For any $k > 0$, $T_0 > 0$, and any online learning algorithm picking $w_t \in \mathbb{R}$, there exists a $T > T_0$, a $\dot{w} \in \mathbb{R}$, and a fixed sequence $g_t \in \mathbb{R}$ on which the regret is:*

$$\begin{aligned} R_T(\dot{w}) &= \sum_{t=1}^T g_t w_t - g_t \dot{w} \\ &\geq k \|\dot{w}\| G_{\max} \log(T \|\dot{w}\| + 1) \sqrt{T} + \frac{G_{\max}}{T-1} \exp\left(\frac{\sqrt{T-1}}{8k}\right) \\ &\geq k \|\dot{w}\| G_{\max} \log(T \|\dot{w}\| + 1) \sqrt{T} + \max_{t \leq T} G_{\max} \frac{G_{t-1}^2}{\|g\|_{1:t-1}^2} \exp\left[\frac{1}{2} \left(\frac{G_t/G_{t-1}}{288k^2}\right)\right] \end{aligned}$$

where $G_t = \max_{t' \leq t} \|g_{t'}\|$, and $G_{\max} = G_T = \max_{t \leq T} \|g_t\|$.

The first inequality in this bound demonstrates that it is impossible to guarantee sublinear regret without prior information about $\|\dot{w}\|$ or G_{\max} while maintaining $O(G_{\max} \|\dot{w}\| \log(\|\dot{w}\|))$ dependence on G_{\max} and $\|\dot{w}\|^2$,² but the second inequality provides hope that if the loss sequence is limited to small jumps in G_t , then

²it is possible to guarantee sublinear regret in exchange for $O(G_{\max} \|\dot{w}\|^2)$ dependence, see Orabona and Pál [42]

we might be able to obtain sublinear regret. Specifically, from the first inequality, observe that in order to bring the exponential term to lower than $O(T)$, the value of k needs to be at least $\Omega(\sqrt{T}/\log(T))$, which causes the non-exponential term to become $O(T)$. However, the second inequality emphasizes that our high regret is the result of a large jump in the value of G_t , so that we might expect to do better if there are no such large jumps. Our upper bounds are given in the form of algorithms that guarantee regret matching the second inequality of this lower bound for any k , showing that we can indeed do well so long as G_t does not increase too quickly.

4.1.3 Optimal Bound: Trade-offs in the Logarithmic exponent γ

To extend the frontier to the γ dimension, we modify our adversarial sequence by setting $g_T = O(\gamma k^{1/\gamma} T^{1-1/2\gamma})$ instead of $O(k\sqrt{T})$. This results in a penalty that is exponential in $(\sqrt{T}/k)^{1/\gamma}$, which we express as a multiple of $(G_t/\gamma k^2 G_{t-1})^{1/(2\gamma-1)}$. Since $\gamma \in (1/2, 1]$, we are getting a larger exponential penalty even though the adversarial subgradients have decreased in size, illustrating that decreasing the logarithmic factor is very expensive.

The full frontier is stated formally in the following Theorem.

Theorem 4.3. *For any $\gamma \in (1/2, 1]$, $k > 0$, $T_0 > 0$, and any online learning algorithm picking $w_t \in \mathbb{R}$, there exists a $T > T_0$, a $u \in \mathbb{R}$, and a sequence $g_1, \dots, g_T \in \mathbb{R}$ with $\|g_t\| \leq \max(1, 18\gamma(4k)^{1/\gamma}(t-1)^{1-1/2\gamma})$ on which the regret is:³*

$$\begin{aligned} R_T(\hat{w}) &= \sum_{t=1}^T g_t w_t - g_t \hat{w} \\ &\geq k \|\hat{w}\| G \log^\gamma(T \|\hat{w}\| + 1) \sqrt{T} + \frac{G_{\max}}{T-1} \exp\left(\frac{(T-1)^{1/2\gamma}}{2(4k)^{1/\gamma}}\right) \\ &\geq k \|\hat{w}\| G_{\max} \log^\gamma(T \|\hat{w}\| + 1) \sqrt{T} + \max_{t \leq T} G_{\max} \frac{G_{t-1}^2}{\|g\|_{1:t-1}^2} \exp\left[\frac{1}{2} \left(\frac{G_t/G_{t-1}}{288\gamma k^2}\right)^{1/(2\gamma-1)}\right] \end{aligned}$$

where $G_t = \max_{t' \leq t} \|g_{t'}\|$ and $G_{\max} = G_T = \max_{t \leq T} \|g_t\|$.

Again, the first inequality tells us that adversarial sequences can always deny the algorithm sublinear regret and the second inequality says that so long as G_t grows slowly, we can still hope for sublinear regret. This time, however, the second inequality appears to blow up when $\gamma \rightarrow 1/2$. In this case, $G_{\max} = O(k^2)$ regardless of T and so the value of G_t/G_{t-1} is never very large, keeping the exponent in the second inequality less than 1 so that the singularity in the exponent does not send the bound to infinity. This singularity at $\gamma = 1/2$ tells us that the adversary does not need to be “very adversarial” in order to force us to experience exponential regret.

To gain some more intuition for what happens at $\gamma = 1/2$, consider a model in which the adversary must commit ahead of time to some G_{\max} (which corresponds to picking k), unknown to the optimization

³The same result holds with in expectation for randomized algorithms with a deterministic sequence g_t .

algorithm, such that $\|g_t\| \leq G_{\max}$ for all t . When a bound $G_{\text{bound}} \geq G_{\max}$ is known to the algorithm ahead of time, then it is possible to achieve $O(\|\dot{w}\|G_{\text{bound}}\sqrt{T\log(\|\dot{w}\|T)})$ regret (e.g. see Orabona and Pál [40]). However, note that when $\gamma = 1/2$, committing to an appropriate G_{\max} would not prevent an adversary from using the sequence of Theorem 4.3. Therefore, Theorem 4.3 tells us that algorithms which achieve $O(\|\dot{w}\|G_{\text{bound}}\sqrt{T\log(\|\dot{w}\|T)})$ regret are inherently very fragile because if the bound is incorrect (which happens for large enough k), then the adversary can force the algorithm to suffer $G \exp(O(T/G))$ regret for arbitrarily large T .

Continuing with the model in which the adversary must commit to some unknown G ahead of time, suppose we are satisfied with $O(\|\dot{w}\|G_{\max}\sqrt{T}\log^\gamma(\|\dot{w}\|T))$ regret for some $\gamma > 1/2$. In this case, after some (admittedly possibly very large) number of iterations, the exponential term in the second inequality no longer grows with T , and the adversarial strategy of Theorem 4.3 is not available because this strategy requires a choice of G that depends on T . Therefore an algorithm that guarantees regret matching the second inequality for some k and γ will obtain an asymptotic dependence on T that is only $\log^\gamma(T)\sqrt{T}$.

These lower bounds show that there is a fundamental frontier of tradeoffs parameterized γ and k . Now we proceed to derive parameter-free algorithms that match any point on the frontier.

4.2 Parameter-free FTRL Analysis

In this section we provide the tools used to derive algorithms whose regret matches the lower bounds in the previous section. Our algorithms make use of the Follow-the-Regularized-Leader (FTRL) framework, as previously introduced in Section 2.3. We briefly recall their main idea: after seeing the t^{th} loss of the online learning game, an FTRL algorithm chooses a function ψ_t (called a *regularizer*), and picks w_{t+1} according to:

$$w_{t+1} = \operatorname{argmin}_{w \in W} \psi_t(w) + \sum_{t'=1}^t \ell_{t'}(w)$$

Careful choice of regularizers is obviously crucial to the success of such an algorithm, and in the following we provide simple conditions on ψ sufficient for FTRL to achieve optimal regret without knowing $\|\dot{w}\|$ or G_{\max} . Our analysis generalizes many previous works for online learning with unconstrained W (e.g. Orabona [38, 39]; Cutkosky and Boahen [13]) in which regret bounds were proved via arduous ad-hoc constructions. Further, our techniques improve the regret bound in the algorithm that does not require prior information of Cutkosky and Boahen [13]. We note that an alternative set of conditions on regularizers was given in Orabona and Pál [40] via an elegant reduction to coin-betting algorithms, but this prior analysis requires a known bound on G_{\max} .

Our regularizers ψ_t take the form $\psi_t(w) = \frac{k}{a_t \eta_t} \psi(a_t w)$ for some fixed function ψ and numbers a_t and η_t . The value k specifies the corresponding tradeoff parameter in the lower-bound frontier, while the function ψ specifies the value of γ . The values for a_t and η_t do not depend on k or ψ , but are carefully chosen functions of the observed gradients g_1, \dots, g_t that guarantee the desired asymptotics in the regret bound.

4.2.1 Generalizing Strong Convexity

Prior analyses of FTRL (such as the basic theorems presented in Section 2.3) often make use of strongly-convex regularizers to simplify regret analysis, but it turns out that strongly-convex regularizers cannot match our lower bounds. Fortunately, there is a simple generalization of strong-convexity that will suffice for our purposes. This generalized notion is very similar to a dual version of the “local smoothness” condition used in Orabona [38]. We define this generalization of strong-convexity below.

Definition 4.4. *Let W be a convex space and let $\sigma : W^2 \rightarrow \mathbb{R}$ by an arbitrary function. We say a convex function $f : W \rightarrow \mathbb{R}$ is $\sigma(\cdot, \cdot)$ -strongly convex with respect to a norm $\|\cdot\|$ if for all $x, y \in W$ and $g \in \partial f(x)$ we have*

$$f(y) \geq f(x) + g \cdot (y - x) + \frac{\sigma(x, y)}{2} \|x - y\|^2$$

As a special case (and by abuse of notation), for any function $\sigma : W \rightarrow \mathbb{R}$ we define $\sigma(w, z) = \min(\sigma(w), \sigma(z))$ and define $\sigma(\cdot)$ -strong convexity accordingly.

We’ll usually just write σ -strongly convex instead of $\sigma(\cdot, \cdot)$ -strongly convex since our definition is purely a generalization of the standard one. We will also primarily make use of the special case $\sigma(w, z) = \min(\sigma(w), \sigma(z))$.

4.2.2 Adaptive regularizers

Now we present a few definitions that will allow us to easily construct sequences of regularizers that achieve parameter-free regret bounds. Intuitively, we require that our regularizers ψ_t grow super-linearly in order to ensure that $\psi_t(w) + g_{1:t}w$ always has a minimal value. However, we do not want ψ_t to grow quadratically because this will result in $O(\|\dot{w}\|^2)$ regret. The formal requirements on the shape of ψ_t are presented in the following definition:

Definition 4.5. *Let W be a closed convex subset of a vector space such that $0 \in W$. Any differentiable function $\psi : W \rightarrow \mathbb{R}$ that satisfies the following conditions:*

1. $\psi(0) = 0$.
2. $\psi(x)$ is σ -strongly-convex with respect to some norm $\|\cdot\|$ for some $\sigma : W \rightarrow \mathbb{R}$ such that $\|x\| \geq \|y\|$ implies $\sigma(x) \leq \sigma(y)$.
3. For any C , there exists a B such that $\psi(x)\sigma(x) \geq C$ for all $\|x\| \geq B$.

is called a $(\sigma, \|\cdot\|)$ -adaptive regularizer. We also define the useful auxiliary function $h(w) = \psi(w)\sigma(w)$ and by mild abuse of notation, we define $h^{-1}(x) = \max_{h(w) \leq x} \|w\|$.

We will use adaptive regularizers as building blocks for our FTRL regularizers ψ_t , so it is important to have examples of such functions. We will provide some tools for finding adaptive regularizers in Section

4.3, but to keep an example in mind for now, we remark that $\psi(w) = (\|w\| + 1) \log(\|w\| + 1) - \|w\|$ is a $\left(\frac{1}{\|\cdot\|+1}, \|\cdot\|\right)$ -adaptive regularizer where $\|\cdot\|$ is the G_2 norm.

The following definition specifies the sequences η_t and a_t which we use to turn an adaptive regularizer into the regularizers used for our FTRL algorithms:

Definition 4.6. Let $\|\cdot\|$ be a norm and $\|\cdot\|_*$ be the dual norm ($\|x\|_* = \sup_{\|y\|=1} x \cdot y$). Let g_1, \dots, g_T be a sequence of subgradients and set $G_t = \max_{t' \leq t} \|g_{t'}\|_*$. Define the sequences $\frac{1}{\eta_t}$ and a_t recursively by:

$$\begin{aligned} \frac{1}{\eta_0^2} &= 0 \\ \frac{1}{\eta_t^2} &= \max \left(\frac{1}{\eta_{t-1}^2} + 2\|g_t\|_*^2, G_t \|g_{1:t}\|_* \right) \\ a_1 &= \frac{1}{(G_1 \eta_1)^2} \\ a_t &= \max \left(a_{t-1}, \frac{1}{(G_t \eta_t)^2} \right) \end{aligned}$$

Suppose ψ is a $(\sigma, \|\cdot\|)$ -adaptive regularizer and $k > 0$. Define

$$\begin{aligned} \psi_t(w) &= \frac{k}{\eta_t a_t} \psi(a_t w) \\ w_{t+1} &= \operatorname{argmin}_{w \in W} \psi_t(w) + g_{1:t} \cdot w \end{aligned}$$

Now without further ado, we give our regret bound for FTRL using these regularizers.

Theorem 4.7. Suppose ψ is a $(\sigma, \|\cdot\|)$ -adaptive regularizer and g_1, \dots, g_T is some arbitrary sequence of subgradients. Let $k \geq 1$, and let ψ_t be defined as in Definition 4.6.

Set

$$\begin{aligned} \sigma_{\min} &= \inf_{\|w\| \leq h^{-1}(10/k^2)} k\sigma(w) \\ D &= \max_t \frac{G_{t-1}^2}{(\|g\|_*^2)_{1:t-1}} h^{-1} \left(\frac{5G_t}{k^2 G_{t-1}} \right) \\ Q_T &= 2 \frac{\|g\|_{1:T}}{G_{\max}} \end{aligned}$$

Then FTRL with regularizers ψ_t achieves regret

$$\begin{aligned} R_T(\hat{w}) &\leq \frac{k}{Q_T \eta_T} \psi(Q_T u) + \frac{45G_{\max}}{\sigma_{\min}} + 2G_{\max} D \\ &\leq kG_{\max} \frac{\psi(2uT)}{\sqrt{2T}} + \frac{45G_{\max}}{\sigma_{\min}} + 2G_{\max} D \end{aligned}$$

This bound consists of three terms, the first of which will correspond to the \sqrt{T} term in our lower bounds and the last of which will correspond to the exponential penalty. The middle term is a constant independent of u and T . To unpack a specific instantiation of this bound, consider the example adaptive regularizer $\psi(w) = (\|w\| + 1) \log(\|w\| + 1) - \|w\|$. For this choice of ψ , we have $\psi(2uT)/\sqrt{2T} = O(\|\dot{w}\| \sqrt{T} \log(T\|\dot{w}\| + 1))$ so that the first term in the regret bound matches the \sqrt{T} term in our lower bound with $\gamma = 1$. Roughly speaking, $h(w) \approx \log(w)$, so that $h^{-1}(x) \approx \exp(x)$ and the quantity $D = \max_t \frac{G_{t-1}^2}{(\|g\|_{1:t-1}^2)} h^{-1}\left(\frac{5G_t}{k^2 G_{t-1}}\right)$ matches the exponential penalty in our lower bound. In the following section we formalize this argument and exhibit a family of adaptive regularizers that enable us to design algorithms whose regret matches any desired point on the lower bound frontier.

4.3 Optimal Algorithms

In this section we construct specific adaptive regularizers in order to obtain optimal algorithms using our regret upper bound of Theorem 4.7. The results in the previous section hold for arbitrary norms, but from this point on we will focus on the G_2 norm. Our regret upper bound expresses regret in terms of the function h^{-1} . Inspection of the bound shows that if $h^{-1}(x)$ is exponential in $x^{1/(2\gamma-1)}$, and $\psi(w) = O(\|w\| \log^\gamma(\|w\| + 1))$, then our upper bound will match (the second inequality in) our lower bound frontier. The following Corollary formalizes this observation.

Corollary 4.8. *If ψ is an $(\sigma, \|\cdot\|)$ -adaptive regularizer such that*

$$\begin{aligned}\psi(x)\sigma(x) &\geq \Omega(\gamma \log^{2\gamma-1}(\|x\|)) \\ \psi(x) &\leq O(\|x\| \log^\gamma(\|x\| + 1))\end{aligned}$$

then for any $k \geq 1$, FTRL with regularizers $\psi_t(w) = \frac{k}{a_t \eta_t} \psi(a_t w)$ yields regret

$$R_T(\dot{w}) \leq O \left[k G_{\max} \sqrt{T} \|\dot{w}\| \log^\gamma(T \|\dot{w}\| + 1) + \max_t \frac{G_{\max} G_{t-1}^2}{\|g\|_{1:t-1}^2} \exp \left[O \left(\left(\frac{G_t}{k^2 \gamma G_{t-1}} \right)^{1/(2\gamma-1)} \right) \right] \right]$$

We call regularizers that satisfy these conditions γ -optimal.

With this Corollary in hand, to match our lower bound frontier we need only construct a γ -optimal adaptive regularizer for all $\gamma \in (1/2, 1]$. Constructing adaptive regularizers is made much simpler with Proposition 4.9 below. This proposition allows us to design adaptive regularizers in high dimensional spaces by finding simple one-dimensional functions. It can be viewed as taking the place of arguments in prior work [35; 40; 13] that reduce high dimensional problems to one-dimensional problems by identifying a “worst-case” direction for each subgradient g_t .

Proposition 4.9. *Let $\|\cdot\|$ be the G_2 norm ($\|w\| = \|w\|_2 = \sqrt{w \cdot w}$). Let ϕ be a three-times differentiable function from the non-negative reals to the reals that satisfies*

1. $\phi(0) = 0$.
2. $\phi'(x) \geq 0$.
3. $\phi''(x) \geq 0$.
4. $\phi'''(x) \leq 0$.
5. $\lim_{x \rightarrow \infty} \phi(x)\phi''(x) = \infty$.

Then $\psi(w) = \phi(\|w\|)$ is a $(\phi''(\|\cdot\|), \|\cdot\|)$ -adaptive regularizer.

Now we are finally ready to derive our first optimal regularizer:

Proposition 4.10. *Let $\|\cdot\|$ be the G_2 norm. Let $\phi(x) = (x+1)\log(x+1) - x$. Then $\psi(w) = \phi(\|w\|)$ is a 1-optimal, $(\phi''(\|\cdot\|), \|\cdot\|)$ -adaptive regularizer.*

Proof. We can use Proposition 4.9 to prove this with a few simple calculations:

$$\begin{aligned}\phi(0) &= 0 \\ \phi'(x) &= \log(x+1) \\ \phi''(x) &= \frac{1}{x+1} \\ \phi'''(x) &= -\frac{1}{(x+1)^2} \\ \phi(x)\phi''(x) &= \left(\log(x+1) - \frac{x}{x+1}\right)\end{aligned}$$

Now the conclusion of the Proposition is immediate from Proposition 4.9 and inspection of the above equations. \square

A simple application of Corollary 4.8 shows that FTRL with regularizers $\psi_t(w) = \frac{k}{\eta t} ((\|w\|+1)\log(\|w\|+1) - \|w\|)$ matches our lower bound with $\gamma = 1$ for any desired k .

In fact, the result of Proposition 4.10 is a more general phenomenon:

Proposition 4.11. *Let $\|\cdot\|$ be the G_2 norm. Given $\gamma \in (1/2, 1]$, set $\phi(x) = \int_0^x \log^\gamma(z+1) dz$. Then $\psi(w) = \phi(\|w\|)$ is a γ -optimal, $(\phi''(\|\cdot\|), \|\cdot\|)$ -adaptive regularizer.*

Proof.

$$\begin{aligned}\phi(0) &= 0 \\ \phi'(x) &= \log^\gamma(x+1) \\ \phi''(x) &= \gamma \frac{\log^{\gamma-1}(x+1)}{x+1} \\ \phi'''(x) &= \gamma(\gamma-1) \frac{\log^{\gamma-2}(x+1)}{(x+1)^2} - \gamma \frac{\log^{\gamma-1}(x+1)}{(x+1)^2}\end{aligned}$$

Since $\gamma \leq 1$, $\phi'''(x) \leq 0$ and so ϕ satisfies the first four conditions of Proposition 4.9. It remains to characterize $\phi(x)$ and $\phi(x)\phi''(x)$, which we do by finding lower and upper bounds on $\phi(x)$:

For a lower bound, we have

$$\begin{aligned} \frac{1}{2} \frac{d}{dx} x \log^\gamma(x+1) &= \frac{1}{2} \left(\log^\gamma(x+1) + \gamma \frac{x}{x+1} \log^{\gamma-1}(x+1) \right) \\ &\leq \log^\gamma(x+1) \end{aligned}$$

where the inequality follows since $\frac{x}{x+1} \leq \log(x+1)$, which can be verified by differentiating both sides. Therefore $\phi(x) \geq \frac{1}{2} x \log^\gamma(x+1)$. This lower-bound implies

$$\phi(x)\phi''(x) \geq \frac{1}{2} \gamma \frac{x}{x+1} \log^{2\gamma-1}(x+1)$$

which gives us the last condition in Proposition 4.9, as well as the first condition for γ -optimality.

Similarly, we have

$$\begin{aligned} \frac{d}{dx} x \log^\gamma(x+1) &= \left(\log^\gamma(x+1) + \gamma \frac{x}{x+1} \log^{\gamma-1}(x+1) \right) \\ &\geq \log^\gamma(x+1) \end{aligned}$$

This implies $\phi(x) \leq x \log(x+1)$ which gives us the second condition for γ -optimality. \square

Thus, by applying Theorem 4.7 to the regularizers of Proposition 4.11, we have a family of algorithms that matches our family of lower-bounds up to constants. The updates for these regularizers are extremely simple:

$$w_{t+1} = -\frac{g_{1:t}}{a_t \|g_{1:t}\|} \left[\exp \left((\eta_t \|g_{1:t}\|/k)^{1/\gamma} \right) - 1 \right]$$

The guarantees of Theorem 4.7 do not make any assumptions on how k is chosen, so that we could choose k using prior knowledge if it is available. For example, if a bound on G_t/G_{t-1} is known, we can set $k \geq \sqrt{\max_t G_t/G_{t-1}}$. This reduces the exponentiated quantity $\max_t G_t/k^2 G_{t-1}$ to a constant, leaving a regret of $O(\|\hat{w}\| \log(T\|\hat{w}\| + 1) G_{\max} \sqrt{T \max_t G_t/G_{t-1}})$. This bound holds without requiring a bound on G_{\max} . Thus our algorithms open up an intermediary realm in which we have no bounds on $\|\hat{w}\|$ or G_{\max} , and yet we can leverage some other information to avoid the exponential penalty.

4.4 FREEREX

Now we explicitly describe an algorithm, along with a fully worked-out regret bound. The norm $\|\cdot\|$ used in the following is the G_2 norm ($\|w\| = \sqrt{w \cdot w}$), and our algorithm uses the adaptive regularizer $\psi(w) = (\|w\| + 1) \log(\|w\| + 1) - \|w\|$. Similar calculations could be performed for arbitrary γ using the regularizers

of Proposition 4.11, but we focus on the $\gamma = 1$ because it allows for simpler and tighter analysis through our closed-form expression for ψ . Since we do not require any information about the losses, we call our algorithm FREEREX for Information-free Regret via exponential updates.

Algorithm 1 FREEREX

Input: k .
Initialize: $\frac{1}{\eta_0^2} \leftarrow 0$, $a_0 \leftarrow 0$, $w_1 \leftarrow 0$, $G_0 \leftarrow 0$, $\psi(w) = (\|w\| + 1) \log(\|w\| + 1) - \|w\|$.
for $t = 1$ **to** T **do**
 Play w_t , receive subgradient $g_t \in \partial \ell_t(w_t)$.
 $G_t \leftarrow \max(G_{t-1}, \|g_t\|)$.
 $\frac{1}{\eta_t^2} \leftarrow \max\left(\frac{1}{\eta_{t-1}^2} + 2\|g_t\|^2, G_t\|g_{1:t}\|\right)$.
 $a_t \leftarrow \max(a_{t-1}, 1/(G_t\eta_t)^2)$.
 //Set w_{t+1} using FTRL update
 $w_{t+1} \leftarrow -\frac{g_{1:t}}{a_t\|g_{1:t}\|} \left[\exp\left(\frac{\eta_t\|g_{1:t}\|}{k}\right) - 1 \right] // = \operatorname{argmin}_w \left[\frac{k\psi(a_t w)}{a_t \eta_t} + g_{1:t} w \right]$
end for

Theorem 4.12. *The regret of FREEREX (Algorithm 1) is bounded by*

$$R_T(\hat{w}) \leq k\|\hat{w}\| \sqrt{2\|g\|_{1:T}^2 + G_{\max} \max_{t \leq T} \|g_{1:t}\|} \log\left(\frac{2\|g\|_{1:T} \|\hat{w}\| + 1}{G_{\max}}\right) + \frac{45G_{\max}}{k} \exp(10/k^2 + 1) \\ + 2G_{\max} \max_t \frac{G_{t-1}^2}{\|g\|_{1:t-1}^2} \left[\exp\left(\frac{5G_t}{k^2 G_{t-1}} + 1\right) - 1 \right]$$

Proof. Define $\phi(x) = (x+1) \log(x+1) - x$. Then $\psi(w) = (\|w\|+1) \log(\|w\|+1) - \|w\|$ is a $(\phi''(\|\cdot\|), \|\cdot\|)$ -adaptive regularizer by Proposition 4.10. Therefore we can immediately apply Theorem 4.7 to obtain

$$R_T(\hat{w}) \leq \frac{k}{Q_T \eta_T} \psi(Q_T u) + \frac{45G_{\max}}{\phi''_{\min}} + 2G_{\max} D$$

where we've defined $\phi''_{\min} = \inf_{\|w\| \leq h^{-1}(10/k^2)} k\phi''(\|w\|)$.

We can compute (for non-negative x):

$$\begin{aligned} \phi(x) &\leq (x+1) \log(x+1) \\ \phi''(x) &= \frac{1}{x+1} \\ h(w) &= \phi(\|w\|) \phi''(\|w\|) = \left(\log(\|w\| + 1) - \frac{\|w\|}{\|w\| + 1} \right) \\ &\geq \log(\|w\| + 1) - 1 \end{aligned}$$

From Proposition 4.20 (part 2) we have $\frac{1}{\eta_T} \leq \sqrt{2\|g\|_{1:T}^2 + G_{\max} \max_{t \leq T} \|g_{1:t}\|}$. We also have $(\|w\| +$

1) $\log(\|w\| + 1) - \|w\| = \|w\| \log(\|w\| + 1) + \log(\|w\| + 1) - \|w\| \leq \|w\| \log(\|w\| + 1)$, so we are left with

$$\begin{aligned} R_T(\hat{w}) &\leq \frac{k}{\eta_T} \|\hat{w}\| \log(Q_T \|\hat{w}\| + 1) + \sup_{\|w\| \leq h^{-1}(\frac{10}{k^2})} \frac{45(\|w\| + 1)}{k} + 2G_{\max} D \\ &= k \sqrt{2\|g\|_{1:T}^2 + G_{\max} \max_{t \leq T} \|g_{1:t}\|} \|\hat{w}\| \log(a_T \|\hat{w}\| + 1) + \frac{45G_{\max}}{k} \left[h^{-1} \left(\frac{10}{k^2} \right) + 1 \right] \\ &\quad + 2G_{\max} D \end{aligned}$$

Now it remains to bound $h^{-1}(10/k^2)$ and D . From our expression for h , we have

$$h^{-1}(x/k^2) \leq \exp \left[\frac{x}{k^2} + 1 \right] - 1$$

Therefore we have

$$\begin{aligned} h^{-1}(10/k^2) &\leq \exp(10/k^2 + 1) - 1 \\ D &= 2 \max_t \frac{G_{t-1}^2}{(\|g\|_{1:t-1}^2)_*} h^{-1} \left(\frac{5G_t}{k^2 G_{t-1}} \right) \\ &\leq 2 \max_t \frac{G_{t-1}^2}{(\|g\|_{1:t-1}^2)_*} \left[\exp \left(\frac{5G_t}{k^2 G_{t-1}} + 1 \right) - 1 \right] \end{aligned}$$

Substituting the value $Q_T = 2 \frac{\|g\|_{1:T}}{G_{\max}}$, we conclude

$$\begin{aligned} R_T(\hat{w}) &\leq k \sqrt{2\|g\|_{1:T}^2 + G_{\max} \max_{t \leq T} \|g_{1:t}\|} \|\hat{w}\| \log \left(\frac{2\|g\|_{1:T}}{G_{\max}} \|\hat{w}\| + 1 \right) \\ &\quad + \frac{45G_{\max}}{k} \exp(10/k^2 + 1) + 2G_{\max} D \end{aligned}$$

From which the result follows by substituting in our expression for D . □

As a specific example, for $k = \sqrt{5}$ we numerically evaluate the bound to get

$$\begin{aligned} R_T(\hat{w}) &\leq \|\hat{w}\| \sqrt{10\|g\|_{1:T}^2 + 5G_{\max} \max_{t \leq T} \|g_{1:t}\|} \log \left(\frac{2\|g\|_{1:T}}{G_{\max}} \|\hat{w}\| + 1 \right) + 405G_{\max} \\ &\quad + 2G_{\max} \max_t \frac{G_{t-1}^2}{\|g\|_{1:t-1}^2} \left[\exp \left(\frac{G_t}{G_{t-1}} + 1 \right) - 1 \right] \end{aligned}$$

4.5 Conclusions

In this chapter, we presented a frontier of lower bounds on the worst-case regret of any parameter-free on-line convex optimization algorithm. This frontier demonstrates a fundamental trade-off at work between

$k\|\dot{w}\|G_{\max} \log^\gamma(T\|\dot{w}\| + 1)$ and $\exp\left[\left(\max_t \frac{G_t}{\gamma k^2 G_{t-1}}\right)^{\frac{1}{2\gamma-1}}\right]$ terms. We also present some easy-to-use theorems that allow us to construct algorithms that match our lower bound for any chosen k and γ . Note that our algorithms require the essentially unavoidable trade-off parameters k and γ . However, k and γ are a fundamentally different kind of parameter than the values of G_{\max} or $\|\dot{w}\|$ required by other algorithms. Since our analysis does not make assumptions about the loss functions or comparison point \dot{w} , the parameters k and γ can be freely chosen by the user. There are no unknown constraints on these parameters, and they effect only constants in the regret analysis.

Our results also open a new perspective on optimization algorithms by casting using different parameter values (other than just $\|\dot{w}\|$ or G_{\max}) as a tool to avoid the exponential penalty. Previous algorithms that require bounds on the diameter of W or G_{\max} can be viewed as addressing this issue. We show that it also possible to avoid the exponential penalty by using a known bound on $\max_t G_t/G_{t-1}$, leading to a regret of $\tilde{O}(\|\dot{w}\|G_{\max}\sqrt{T \max_t G_t/G_{t-1}})$.

4.5.1 Other Kinds of Adaptivity

The lower-bound in this chapter throws some cold water on the goal of adapting to $\|\dot{w}\|$ and G_{\max} simultaneously, but there are many other forms of adaptivity we can address. In particular, all prior algorithms that adapt to $\|\dot{w}\|$ (including FREEREX), obtain a term like $\sqrt{G_{\max} \sum_{t=1}^T \|g_t\|}$, while algorithms that adapt to G_{\max} can obtain $\sqrt{\sum_{t=1}^T \|g_t\|^2}$, which is smaller.

In the next chapters, we will focus on obtaining adaptivity to various additional properties, usually assuming a known bound on G_{\max} . We find this assumption more appealing than a bound on $\|\dot{w}\|$ because in practice the value G_{\max} is often at least partially under the user's control through the choice of loss function. For example, many popular kernels used in SVM classification (including the Gaussian kernel) are bounded, leading to known values for G_{\max} . Further, in practice one can always employ the heuristic of setting G_{\max} to be $2G_t$, which has no clear analogue that can be used to set $\|\dot{w}\|$. Using this heuristic we should expect to only update our guess for G_{\max} a small number of times and so we might at least achieve an asymptotic regret of $\|\dot{w}\|G_{\max}\sqrt{T}$.

Appendix

4.A Lower Bound Proof

Before getting started, we need one technical observation:

Proposition 4.13. *Let $k > 0$, $\gamma \in (1/2, 1]$. Set*

$$Z_t = \frac{t^{1-1/2\gamma}}{2t} \left[\exp\left(\frac{t^{1/2\gamma}}{(4k)^{1/\gamma}}\right) - 1 \right]$$

and set $r_t = Z_t - Z_{t-1}$. Then for all sufficiently large T ,

$$r_T \geq \frac{Z_{T-1}}{3\gamma(4k)^{1/\gamma}(T-1)^{1-1/2\gamma}}$$

Proof. We have

$$\frac{d}{dt} \Big|_{t=T} Z_t = \frac{1}{4\gamma(4k)^{1/\gamma}T} \exp\left(\frac{T^{1/2\gamma}}{(4k)^{1/\gamma}}\right) + \frac{1}{4\gamma}T^{-1-1/2\gamma} - \frac{1}{4\gamma}T^{-1-1/2\gamma} \exp\left(\frac{T^{1/2\gamma}}{(4k)^{1/\gamma}}\right)$$

For sufficiently large T , this quantity is positive and increasing in T . Therefore for sufficiently large T ,

$$\begin{aligned} r_T &\geq \frac{d}{dt} \Big|_{t=T-1} Z_t \\ &= \frac{1}{4\gamma(4k)^{1/\gamma}(T-1)} \exp\left(\frac{(T-1)^{1/2\gamma}}{(4k)^{1/\gamma}}\right) + \frac{1}{4\gamma}(T-1)^{-1-1/2\gamma} - \frac{1}{4\gamma}(T-1)^{-1-1/2\gamma} \exp\left(\frac{(T-1)^{1/2\gamma}}{(4k)^{1/\gamma}}\right) \\ &\geq \frac{1}{5\gamma(4k)^{1/\gamma}(T-1)} \exp\left(\frac{(T-1)^{1/2\gamma}}{(4k)^{1/\gamma}}\right) \\ &= \frac{2}{5\gamma(4k)^{1/\gamma}(T-1)^{1-1/2\gamma}} \left(Z_{T-1} + \frac{(T-1)^{1-1/2\gamma}}{2(T-1)} \right) \\ &\geq \frac{1}{3\gamma(4k)^{1/\gamma}(T-1)^{1-1/2\gamma}} Z_{T-1} \end{aligned}$$

where the third inequality holds only for sufficiently large T . □

Now we prove Theorem 4.3, restated below. Theorem 4.2 is an immediate consequence of Theorem 4.3, so we do not prove it separately.

Theorem 4.3. *For any $\gamma \in (1/2, 1]$, $k > 0$, $T_0 > 0$, and any online learning algorithm picking $w_t \in \mathbb{R}$, there exists a $T > T_0$, a $u \in \mathbb{R}$, and a sequence $g_1, \dots, g_T \in \mathbb{R}$ with $\|g_t\| \leq \max(1, 18\gamma(4k)^{1/\gamma}(t-1)^{1-1/2\gamma})$ on which the regret is:⁴*

$$\begin{aligned} R_T(\hat{w}) &= \sum_{t=1}^T g_t w_t - g_t \hat{w} \\ &\geq k \|\hat{w}\| G \log^\gamma(T \|\hat{w}\| + 1) \sqrt{T} + \frac{G_{\max}}{T-1} \exp\left(\frac{(T-1)^{1/2\gamma}}{2(4k)^{1/\gamma}}\right) \\ &\geq k \|\hat{w}\| G_{\max} \log^\gamma(T \|\hat{w}\| + 1) \sqrt{T} + \max_{t \leq T} G_{\max} \frac{G_{t-1}^2}{\|g\|_{1:t-1}^2} \exp\left[\frac{1}{2} \left(\frac{G_t/G_{t-1}}{288\gamma k^2}\right)^{1/(2\gamma-1)}\right] \end{aligned}$$

where $G_t = \max_{t' \leq t} \|g_{t'}\|$ and $G_{\max} = G_T = \max_{t \leq T} \|g_t\|$.

Proof. We prove the Theorem for randomized algorithms and expected regret, as this does not overly complicate the argument. Our proof technique is very similar to that of the warm-up Theorem 4.1 in Section 4.1.1, but we use more careful analysis to improve the bound. Again, the adversarial sequence foils the learner by repeatedly presenting it with the subgradient $g_t = -1$ until the learner's expected prediction $\mathbb{E}[w_t]$ crosses some threshold. If $\mathbb{E}[w_t]$ does not increase fast enough to pass the threshold, then we show that there is some large $\hat{w} \gg 1$ for which $R_T(\hat{w})$ exceeds our bound. However, if $\mathbb{E}[w_t]$ crosses this threshold, then the adversary presents a large positive gradient which forces the learner to have a large $R_T(0)$.

Define $\hat{w}_t = \mathbb{E}[w_t | g_{t'} = -1 \text{ for all } t' < t]$. Without loss of generality, assume $\hat{w}_1 = 0$. Note that \hat{w}_t can be computed by an adversary without access to the algorithm's internal randomness.

Let $S_n = \sum_{t=1}^n \hat{w}_t$. Let $Z_t = \frac{t^{1-1/2\gamma}}{2t} \left[\exp\left(\frac{t^{1/2\gamma}}{(4k)^{1/\gamma}}\right) - 1 \right]$, and set $r_t = Z_t - Z_{t-1}$. Suppose $T_1 > T_0$ is such that

1. For all $t_1 > t_2 > T_1$, $Z_{t_1} > Z_{t_2}$.
2. For all $t > T_1$, $r_t \geq \frac{Z_{t-1}}{3\gamma(4k)^{1/\gamma}(t-1)^{1-1/2\gamma}}$ (by Proposition 4.13).
3. For all $t > T_1$,

$$\frac{1}{4} \left[\exp\left(\frac{t^{1/2\gamma}}{(4k)^{1/\gamma}}\right) - 1 \right] \geq \frac{1}{t-1} \exp\left(\frac{(t-1)^{1/2\gamma}}{2(4k)^{1/\gamma}}\right)$$

4. for all $t > T_1$,

$$\frac{1}{36\gamma(4k)^{1/\gamma}(t-1)} \left[\exp\left(\frac{(t-1)^{1/2\gamma}}{(4k)^{1/\gamma}}\right) - 1 \right] \geq \frac{1}{(t-1)} \exp\left(\frac{(t-1)^{1/2\gamma}}{2(4k)^{1/\gamma}}\right)$$

⁴The same result holds with in expectation for randomized algorithms with a deterministic sequence g_t .

5. For all $t > T_1$,

$$\frac{1}{t-1} \exp\left(\frac{(t-1)^{1/2\gamma}}{2(4k)^{1/\gamma}}\right) \geq \exp\left[\frac{1}{4} \left(\frac{1}{288\gamma k^2}\right)^{1/(2\gamma-1)}\right]$$

6. For all $t > T_1$,

$$18\gamma(4k)^{1/\gamma}(T-1)^{1-1/2\gamma} \geq 1$$

We consider the quantity $\liminf_{n \rightarrow \infty} \frac{S_n}{Z_n}$. There are two cases, either the lim inf is less than 1, or it is not.

Case 1: $\liminf_{n \rightarrow \infty} \frac{S_n}{Z_n} < 1$

In this case, there must be some $T > T_1$ such that $S_T < Z_T$. We use the adversarial strategy of simply giving $g_t = -1$ for all $t \leq T$. Because of this, $\mathbb{E}[w_t | g_1, \dots, g_{t-1}] = \hat{w}_t$ so that

$$\begin{aligned} \mathbb{E}[R_T(\hat{w})] &= \sum_{t=1}^T g_t \mathbb{E}[w_t | g_1, \dots, g_{t-1}] - g_t \hat{w} \\ &= \sum_{t=1}^T g_t \hat{w}_t - g_t \hat{w} \\ &= T\hat{w} - S_T \\ &\geq T\hat{w} - \frac{T^{1-\frac{1}{2\gamma}}}{2T} \left[\exp\left(\frac{T^{1/2\gamma}}{(4k)^{1/\gamma}}\right) - 1 \right] \\ &\geq T\hat{w} - \frac{1}{2} \left[\exp\left(\frac{T^{1/2\gamma}}{(4k)^{1/\gamma}}\right) - 1 \right] \end{aligned}$$

Set $\hat{w} = \frac{1}{T} \left[\exp\left(\frac{T^{1/2\gamma}}{(4k)^{1/\gamma}}\right) - 1 \right]$. Then clearly

$$\begin{aligned} \mathbb{E}[R_T(\hat{w})] &\geq T\hat{w} - \frac{1}{2} \left[\exp\left(\frac{T^{1/2\gamma}}{(4k)^{1/\gamma}}\right) - 1 \right] \\ &\geq \frac{1}{2} T\hat{w} \\ &= \frac{1}{4} T\hat{w} + \frac{1}{4} \left[\exp\left(\frac{T^{1/2\gamma}}{(4k)^{1/\gamma}}\right) - 1 \right] \end{aligned}$$

Now observe that we have chosen \hat{w} carefully so that

$$\sqrt{T} = 4k \log^\gamma(T\hat{w} + 1)$$

Therefore we can write

$$\begin{aligned}\mathbb{E}[R_T(\hat{w})] &\geq \frac{1}{4}T\hat{w} + \frac{1}{4} \left[\exp\left(\frac{T^{1/2\gamma}}{(4k)^{1/\gamma}}\right) - 1 \right] \\ &= k\|\hat{w}\| \log^\gamma(T\|\hat{w}\| + 1)\sqrt{T} + \frac{1}{4} \left[\exp\left(\frac{T^{1/2\gamma}}{(4k)^{1/\gamma}}\right) - 1 \right] \\ &= k\|\hat{w}\|G_{\max} \log^\gamma(T\|\hat{w}\| + 1)\sqrt{T} + \frac{G_{\max}}{4} \left[\exp\left(\frac{T^{1/2\gamma}}{(4k)^{1/\gamma}}\right) - 1 \right]\end{aligned}$$

where we have used $G_{\max} = 1$ to insert factors of G_{\max} where appropriate.

Observing that $G_t/G_{t-1} = 1$ for all t , we can also easily conclude (using properties 3 and 5 of T_1):

$$\begin{aligned}\mathbb{E}[R_T(\hat{w})] &\geq k\|\hat{w}\|G_{\max} \log^\gamma(T\|\hat{w}\| + 1)\sqrt{T} + \frac{G_{\max}}{T-1} \exp\left(\frac{(T-1)^{1/2\gamma}}{2(4k)^{1/\gamma}}\right) \\ &\geq k\|\hat{w}\|G_{\max} \log^\gamma(T\|\hat{w}\| + 1)\sqrt{T} + \max_{t \leq T} G_{\max} \frac{G_{t-1}^2}{\sum_{t'=1}^{t-1} \|g_{t'}\|^2} \exp\left[\frac{1}{2} \left(\frac{G_t/G_{t-1}}{288\gamma k^2}\right)^{1/(2\gamma-1)}\right]\end{aligned}$$

Case 2: $\liminf_{n \rightarrow \infty} \frac{S_n}{Z_n} \geq 1$

By definition of \liminf , there exists some $T_2 > T_1$ and $Q \geq 1$ such that $S_{T_2} \leq \frac{3}{2}QZ_{T_2}$ and for all $t > T_2$, $S_t > \frac{3Q}{4}Z_t$.

Suppose for contradiction that $\hat{w}_t \leq \frac{Q}{2}r_t$ for all $t > T_2$. Then for all $T > T_2$,

$$\begin{aligned}S_T &= S_{T_2} + \sum_{t=T_2+1}^T \hat{w}_t \\ &\leq \frac{3}{2}QZ_{T_2} + \frac{Q}{2}Z_T - \frac{Q}{2}Z_{T_2} \\ &= \frac{Q}{2}Z_T + QZ_{T_2}\end{aligned}$$

Since the second term does not depend on T , this implies that for sufficiently large T , $\frac{S_T}{Z_T} \leq \frac{3}{4}QZ_T$, which contradicts our choice of T_2 . Therefore $\hat{w}_t > \frac{Q}{2}r_t$ for some $t > T_2$.

Let T be the smallest index $T > T_2$ such that $\hat{w}_T > \frac{Q}{2}r_T$. Since $\hat{w}_t \leq \frac{Q}{2}r_t$ for $t < T$, we have

$$S_{T-1} \leq \frac{Q}{2}Z_{T-1} + QZ_{T_2} \leq 2QZ_{T-1}$$

where we have used property 1 of T_1 to conclude $Z_{T_2} \leq Z_{T-1}$.

Our adversarial strategy is to give $g_t = -1$ for $t < T$, then $g_T = 18\gamma(4k)^{1/\gamma}(T-1)^{1-1/2\gamma}$. We evaluate

the regret at $\hat{w} = 0$ and iteration T . Since $g_t = -1$ for $t < T$, $\mathbb{E}[w_t | g_1, \dots, g_{t-1}] = \hat{w}_t$ for $t \leq T$ and so

$$\begin{aligned} \mathbb{E}[R_T(\hat{w})] &= -S_{T-1} + g_T w_T \\ &\geq g_T \frac{Q}{2} r_T - 2QZ_{T-1} \\ &\geq \frac{Q}{2} \frac{18\gamma(4k)^{1/\gamma}(T-1)^{1-1/2\gamma}}{3\gamma(4k)^{1/\gamma}(T-1)^{1-1/2\gamma}} Z_{T-1} - 2QZ_{T-1} \\ &= QZ_{T-1} \\ &\geq Z_{T-1} \end{aligned}$$

where we have used $Q \geq 1$ in the last line. Now we use the fact that $G_{\max} = 18\gamma(4k)^{1/\gamma}(T-1)^{1-1/2\gamma}$ (by property 6 of T_1) to write

$$\begin{aligned} \mathbb{E}[R_T(\hat{w})] &\geq Z_{T-1} \\ &= \frac{1}{18\gamma(4k)^{1/\gamma}} \frac{G_{\max}}{T-1} \left[\exp\left(\frac{(T-1)^{1/2\gamma}}{(4k)^{1/\gamma}}\right) - 1 \right] \\ &\geq \frac{G_{\max}}{T-1} \exp\left(\frac{(T-1)^{1/2\gamma}}{2(4k)^{1/\gamma}}\right) \end{aligned}$$

where we have used the fourth assumption on T_1 in the last line.

Since we are considering $\hat{w} = 0$, we can always insert arbitrary multiples of \hat{w} :

$$\begin{aligned} \mathbb{E}[R_T(\hat{w})] &\geq \frac{G_{\max}}{T-1} \exp\left(\frac{(T-1)^{1/2\gamma}}{2(4k)^{1/\gamma}}\right) \\ &= k\|\hat{w}\| G_{\max} \log^\gamma(T\|\hat{w}\| + 1) \sqrt{T} + \frac{G_{\max}}{T-1} \exp\left(\frac{(T-1)^{1/2\gamma}}{2(4k)^{1/\gamma}}\right) \end{aligned}$$

Now we relate the quantity in the exponent to G_t/G_{t-1} . We have $G_T = g_T$ and $G_{T-1} = 1$ so that

$$G_T/G_{T-1} = 18\gamma(4k)^{1/\gamma}(T-1)^{1-1/2\gamma}$$

Therefore

$$\begin{aligned}
(T-1)^{1/2\gamma} &= \left(\frac{G_T/G_{T-1}}{18\gamma(4k)^{1/\gamma}} \right)^{\frac{1}{2\gamma(1-1/2\gamma)}} \\
&= \left(\frac{G_T/G_{T-1}}{18\gamma(4k)^{1/\gamma}} \right)^{1/(2\gamma-1)} \\
\frac{(T-1)^{1/2\gamma}}{(4k)^{1/\gamma}} &= \left(\frac{G_T/G_{T-1}}{18\gamma(4k)^2} \right)^{1/(2\gamma-1)} \\
&= \left(\frac{G_T/G_{T-1}}{288\gamma k^2} \right)^{1/(2\gamma-1)}
\end{aligned}$$

Now observe that $\frac{1}{T-1} = \frac{G_{T-1}^2}{\sum_{t=1}^{T-1} \|g_t\|^2}$ so that we have

$$\frac{G_{\max}}{T-1} \exp\left(\frac{(T-1)^{1/2\gamma}}{2(4k)^{1/\gamma}}\right) = G_{\max} \frac{G_{T-1}^2}{\sum_{t=1}^{T-1} \|g_t\|^2} \exp\left[\frac{1}{2} \left(\frac{G_T/G_{T-1}}{288\gamma k^2}\right)^{1/(2\gamma-1)}\right]$$

Further, since $\frac{1}{t-1} = \frac{G_{t-1}^2}{\sum_{t'=1}^{t-1} \|g_{t'}\|^2}$ for all $t \leq T$, condition 5 on T_1 tells us that

$$\begin{aligned}
\frac{G_{\max}}{T-1} \exp\left(\frac{(T-1)^{1/2\gamma}}{2(4k)^{1/\gamma}}\right) &\geq G_{\max} \exp\left[\frac{1}{2} \left(\frac{1}{288\gamma k^2}\right)^{1/(2\gamma-1)}\right] \\
&= \max_{t \leq T-1} G_{\max} \frac{G_{t-1}^2}{\sum_{t'=1}^{t-1} \|g_{t'}\|^2} \exp\left[\frac{1}{2} \left(\frac{G_t/G_{t-1}}{288\gamma k^2}\right)^{1/(2\gamma-1)}\right]
\end{aligned}$$

so that

$$\frac{G_{\max}}{T-1} \exp\left(\frac{(T-1)^{1/2\gamma}}{2(4k)^{1/\gamma}}\right) = \max_{t \leq T} G_{\max} \frac{G_{t-1}^2}{\sum_{t'=1}^{t-1} \|g_{t'}\|^2} \exp\left[\frac{1}{2} \left(\frac{G_t/G_{t-1}}{288\gamma k^2}\right)^{1/(2\gamma-1)}\right]$$

Therefore we can put everything together to get

$$\begin{aligned}
\mathbb{E}[R_T(\hat{w})] &\geq k\|\hat{w}\| G_{\max} \log^\gamma(T\|\hat{w}\| + 1) \sqrt{T} + \frac{G_{\max}}{T-1} \exp\left(\frac{(T-1)^{1/2\gamma}}{2(4k)^{1/\gamma}}\right) \\
&\geq k\|\hat{w}\| G_{\max} \log^\gamma(T\|\hat{w}\| + 1) \sqrt{T} + \max_{t \leq T} G_{\max} \frac{G_{t-1}^2}{\sum_{t'=1}^{t-1} \|g_{t'}\|^2} \exp\left[\frac{1}{2} \left(\frac{G_t/G_{t-1}}{288\gamma k^2}\right)^{1/(2\gamma-1)}\right]
\end{aligned}$$

□

4.B FTRL regret

We prove a general bound on the regret of FTRL. Our bound is not fundamentally tighter than the many previous analyses of FTRL, but we decompose the regret in a new way that makes our analysis much easier. We make use of “shadow regularizers”, ψ_t^+ that can be used to characterize regret more easily. Our bound bears some similarity in form to the adaptive online mirror descent bound of [45] and the analysis of FTRL with varying regularizers of [13].

Theorem 4.14. *Let ℓ_t, \dots, ℓ_T be an arbitrary sequence of loss functions. Define $\ell_0(w) = 0$ for notational convenience. Let $\psi_0, \psi_1, \dots, \psi_{T-1}$ be a sequence of regularizer functions, such that ψ_t is chosen without knowledge of $\ell_{t+1}, \dots, \ell_T$. Let $\psi_1^+, \dots, \psi_T^+$ be an arbitrary sequences of regularizer functions (possibly chosen with knowledge of the full loss sequence). Define w_1, \dots, w_T to be the outputs of FTRL with regularizers ψ_t : $w_{t+1} = \operatorname{argmin} \psi_t + \ell_{1:t}$, and define w_t^+ for $t = 2, \dots, T+1$ by $w_{t+1}^+ = \operatorname{argmin} \psi_t^+ + \ell_{1:t}$. Then FTRL with regularizers ψ_t obtains regret*

$$\begin{aligned} R_T(\hat{w}) &= \sum_{t=1}^T \ell_t(w_t) - \ell_t(\hat{w}) \\ &\leq \psi_T^+(\hat{w}) - \psi_0(w_2^+) + \sum_{t=1}^T \psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) + \ell_t(w_t) - \ell_t(w_{t+1}^+) \\ &\quad + \sum_{t=1}^{T-1} \psi_t^+(w_{t+2}^+) - \psi_t(w_{t+2}^+) \end{aligned}$$

Proof. We define $X_t = w_{t+2}^+$ for $t < T$ and $X_T = u$. We’ll use the symbols X_t as intermediate variables in our proof in an attempt to keep the algebra cleaner. By definition of w_{t+1}^+ , for all $t \leq T$ we have

$$\begin{aligned} \psi_t^+(w_{t+1}^+) + \ell_{1:t}(w_{t+1}^+) &\leq \psi_t^+(X_t) + \ell_{1:t}(X_t) \\ \ell_t(w_t) &\leq \ell_t(w_t) + \ell_{1:t}(X_t) - \ell_{1:t}(w_{t+1}^+) + \psi_t^+(X_t) - \psi_t^+(w_{t+1}^+) \\ &= \ell_t(w_t) - \ell_t(w_{t+1}^+) + \ell_{1:t}(X_t) - \ell_{1:t-1}(w_{t+1}^+) \\ &\quad + \psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) \\ &\quad + \psi_t^+(X_t) - \psi_{t-1}(w_{t+1}^+) \end{aligned}$$

Summing this inequality across all t we have

$$\begin{aligned} \sum_{t=1}^T \ell_t(w_t) &\leq \sum_{t=1}^T \ell_t(w_t) - \ell_t(w_{t+1}^+) \\ &\quad + \sum_{t=1}^T \ell_{1:t}(X_t) - \sum_{t=1}^T \ell_{1:t-1}(w_{t+1}^+) \\ &\quad + \sum_{t=1}^T \psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) \\ &\quad + \sum_{t=1}^T \psi_t^+(X_t) - \psi_{t-1}(w_{t+1}^+) \end{aligned}$$

Notice that $\sum_{t=1}^T \ell_{1:t-1}(w_{t+1}^+) = \sum_{t=2}^T \ell_{1:t-1}(w_{t+1}^+)$ since the first term is zero. Thus after some re-indexing we have

$$\begin{aligned} \sum_{t=1}^T \ell_t(w_t) &\leq \sum_{t=1}^T \ell_t(w_t) - \ell_t(w_{t+1}^+) \\ &\quad + \ell_{1:T}(X_T) + \sum_{t=2}^T \ell_{1:t-1}(X_{t-1}) - \sum_{t=2}^T \ell_{1:t-1}(w_{t+1}^+) \\ &\quad + \sum_{t=1}^T \psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) \\ &\quad + \psi_T^+(X_T) - \psi_0(w_2^+) + \sum_{t=1}^{T-1} \psi_t^+(X_t) - \sum_{t=1}^{T-1} \psi_t(w_{t+2}^+) \end{aligned}$$

Now we substitute our values of $X_t = w_{t+2}^+$ for $t < T$ and $X_T = \hat{w}$ to obtain

$$\begin{aligned} \sum_{t=1}^T \ell_t(w_t) &\leq \sum_{t=1}^T \ell_t(w_t) - \ell_t(w_{t+1}^+) \\ &\quad + \ell_{1:T}(\hat{w}) + \psi_T^+(\hat{w}) - \psi_0(w_2^+) \\ &\quad + \sum_{t=1}^T \psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) \\ &\quad + \sum_{t=1}^{T-1} \psi_t^+(w_{t+2}^+) - \sum_{t=1}^{T-1} \psi_t(w_{t+2}^+) \end{aligned}$$

so that subtracting $\ell_{1:T}(\hat{w})$ from both sides we get a regret bound:

$$\begin{aligned}
R_T(\hat{w}) &= \sum_{t=1}^T \ell_t(w_t) - \ell_t(\hat{w}) \\
&\leq \sum_{t=1}^T \ell_t(w_t) - \ell_t(w_{t+1}^+) \\
&\quad + \psi_T^+(\hat{w}) - \psi_0(w_2^+) \\
&\quad + \sum_{t=1}^T \psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) \\
&\quad + \sum_{t=1}^{T-1} \psi_t^+(w_{t+2}^+) - \sum_{t=1}^{T-1} \psi_t(w_{t+2}^+)
\end{aligned}$$

□

4.C Facts About Strong Convexity

In this section we prove some basic facts about our generalized strong convexity.

Proposition 4.15. *Suppose $\psi : W \rightarrow \mathbb{R}$ is σ -strongly convex. Then:*

1. $\psi + f$ is σ -strongly convex for any convex function f .
2. $c\psi$ is $c\sigma$ -strongly convex for any $c \geq 0$.
3. Suppose $c \geq 0$ and $\phi(w) = \psi(cw)$. Let $\sigma'(x, y) = \sigma(cx, cy)$. Then ϕ is $c^2\sigma'$ -strongly convex.

Proof. 1. Let $x, y \in W$ and let $g \in \partial\psi(x)$ and $b \in \partial f(x)$. Then $g + b \in \partial(\psi + f)(x)$. By convexity and strongly convexity respectively we have:

$$\begin{aligned}
\psi(y) &\geq \psi(x) + g \cdot (y - x) + \frac{\sigma(x, y)}{2} \|x - y\|^2 \\
f(y) &\geq f(x) + b \cdot (y - x)
\end{aligned}$$

so that adding these equations shows that $\psi + f$ is σ -strongly convex.

2. This follows immediately by multiplying the defining equation for strong convexity of ψ by c .
3. Let $x, y \in W$ and let $g \in \partial\psi(cx)$. Then $cg \in \partial\phi(x)$.

$$\begin{aligned}
\psi(cy) &\geq \psi(cx) + g \cdot (cy - cx) + \frac{\sigma(cx, cy)}{2} \|cx - cy\|^2 \\
\phi(y) &\geq \phi(x) + cg \cdot (y - x) + \frac{\sigma(cx, cy)}{2} c^2 \|x - y\|^2
\end{aligned}$$

□

Note that for any linear function $f(w) = g \cdot w$, if ψ is σ -strongly convex, then $\psi + f$ is also σ -strongly convex.

We show that the following lemma from [33] about strongly-convex functions continues to hold under our more general definition. The proof of this lemma (and the next) are identical to the standard ones, but we include them here for completeness.

Lemma 4.16. *Suppose A and B are arbitrary convex functions such that $A + B$ is σ -strongly convex. Let $w_1 = \operatorname{argmin} A$ and $w_2 = \operatorname{argmin} A + B$ and let $g \in \partial B(w_1)$. Then*

$$\|w_1 - w_2\| \leq \frac{\|g\|_\star}{\sigma(w_1, w_2)}$$

Proof. Since $w_2 \in \operatorname{argmin} A + B$, we have $0 \in \partial(A + B)(w_2)$ and so by definition of strong convexity we have

$$\frac{\sigma(w_1, w_2)}{2} \|w_1 - w_2\|^2 \leq A(w_1) + B(w_1) - A(w_2) - B(w_2)$$

Now let $g \in \partial B(w_1)$. Consider the function $\hat{A}(w) = A(w) + B(w) - \langle g, w \rangle$. Then we must have $0 \in \partial \hat{A}(w_1)$ and so by strong-convexity again we have

$$\frac{\sigma(w_1, w_2)}{2} \|w_1 - w_2\|^2 \leq A(w_2) + B(w_2) - \langle g, w_2 \rangle - A(w_1) - B(w_1) + \langle g, w_1 \rangle$$

Adding these two equations yields:

$$\sigma(w_1, w_2) \|w_1 - w_2\|^2 \leq \langle g, w_1 - w_2 \rangle \leq \|g\|_\star \|w_1 - w_2\|$$

and so we obtain the desired statement. □

Finally, we have an analog of a standard way to check for strong-convexity:

Proposition 4.17. *Suppose $\psi : W \rightarrow \mathbb{R}$ is twice-differentiable and $v^T \nabla^2 \psi(x) v \geq \sigma(x) \|v\|^2$ for all x and v for some norm $\|\cdot\|$ and $\sigma : W \rightarrow \mathbb{R}$ where $\sigma(x + t(y - x)) \geq \min(\sigma(x), \sigma(y))$ for all $x, y \in W$ and $t \in [0, 1]$. Then ψ is σ -strongly convex with respect to the norm $\|\cdot\|$.*

Proof. We integrate the derivative:

$$\begin{aligned}
\psi(x) - \psi(y) &= \int_0^1 \frac{d}{dt} \psi(x + t(y-x)) dt \\
&= \int_0^1 \nabla \psi(x + t(y-x)) \cdot (y-x) dt \\
&= \nabla \psi(x) \cdot (y-x) \\
&\quad + \int_0^1 \int_0^t (y-x)^T \nabla^2 \psi(x + k(y-x)) (y-x) dk dt \\
&\geq \nabla \psi(x) \cdot (y-x) + \|y-x\|^2 \int_0^1 \int_0^t \sigma(x + k(y-x)) dk dt \\
&\geq \nabla \psi(x) \cdot (y-x) + \|y-x\|^2 \int_0^1 t \min(\sigma(x), \sigma(y)) dt \\
&= \nabla \psi(x) \cdot (y-x) + \frac{\min(\sigma(x), \sigma(y))}{2} \|y-x\|^2
\end{aligned}$$

□

4.D Proof of Theorem 4.9

First we prove a proposition that allows us to generate a strongly convex function easily:

Proposition 4.18. *Suppose $\phi : \mathbb{R} \rightarrow \mathbb{R}$ is such that $\frac{\phi'(x)}{x} \geq \phi''(x) \geq 0$ and $\phi'''(x) \leq 0$ for all $x \geq 0$. Let W be a Hilbert Space and $\psi : W \rightarrow \mathbb{R}$ be given by $\psi(w) = \phi(\|w\|)$. Then ψ is $\phi''(\|w\|)$ -strongly convex with respect to $\|\cdot\|$.*

Proof. Let $x, y \in W$. We have

$$\begin{aligned}
\nabla \psi(x) &= \phi'(\|x\|) \frac{x}{\|x\|} \\
\nabla^2 \psi(x) &= \left(\phi''(\|x\|) - \frac{\phi'(\|x\|)}{\|x\|} \right) \frac{xx^T}{\|x\|^2} + \frac{\phi'(\|x\|)}{\|x\|} I \\
&\succeq \phi''(\|x\|) I
\end{aligned}$$

Where the last line follows since $\frac{\phi'(x)}{x} \geq \phi''(x)$ for all $x \geq 0$. Since $\phi'''(x) \leq 0$, $\phi''(x)$ is always decreasing for positive x and so we have

$$\phi''(\|x + t(y-x)\|) \geq \min(\phi''(\|x\|), \phi''(\|y\|))$$

for all $t \in [0, 1]$. Therefore we can apply Proposition 4.17 to conclude that ψ is $\phi''(\|w\|)$ -strongly convex. □

Now we prove Proposition 4.9, restated below:

Proposition 4.9. *Let $\|\cdot\|$ be the G_2 norm ($\|w\| = \|w\|_2 = \sqrt{w \cdot w}$). Let ϕ be a three-times differentiable function from the non-negative reals to the reals that satisfies*

1. $\phi(0) = 0$.
2. $\phi'(x) \geq 0$.
3. $\phi''(x) \geq 0$.
4. $\phi'''(x) \leq 0$.
5. $\lim_{x \rightarrow \infty} \phi(x)\phi''(x) = \infty$.

Then $\psi(w) = \phi(\|w\|)$ is a $(\phi''(\|\cdot\|), \|\cdot\|)$ -adaptive regularizer.

Proof. It's clear that $\psi(0) = 0$ so the first condition for being an adaptive regularizer is satisfied.

Next we will show that $\frac{\phi'(x)}{x} \geq \phi''(x)$ so that we can apply Proposition 4.18. It suffices to show

$$\phi'(x) - x\phi''(x) \geq 0$$

Clearly this identity holds for $x = 0$. Differentiating the right-hand-side of the equation, we have

$$\phi''(x) - x\phi'''(x) - \phi''(x) = -x\phi'''(x) \geq 0$$

since $\phi'''(x) \leq 0$ and $x \geq 0$. Thus $\phi'(x) - x\phi''(x)$ is non-decreasing and so must always be non-negative.

Therefore, by Proposition 4.18, ψ is $(\phi''(\|\cdot\|), \|\cdot\|)$ -strongly convex. Also, since $\phi'''(x) \leq 0$, $\phi''(\|x\|) \leq \phi''(\|y\|)$ when $\|x\| \geq \|y\|$ so that ψ satisfies the second condition for being an adaptive regularizer.

Finally, observe that $\lim_{x \rightarrow \infty} \phi(x)\phi''(x)$ implies by definition that for any C there exists a B such that $\phi(x)\phi''(x) \geq C$ whenever $x \geq B$. Therefore we immediately see that $\psi(x)\phi''(\|x\|) \geq C$ for all $\|x\| \geq B$ so that the third condition is satisfied. \square

4.E Proof of Theorem 4.7

First we define new regularizers ψ_t^+ analogously to ψ_t that we will use in conjunction with Theorem 4.14:

Definition 4.19. *Given a norm $\|\cdot\|$ and a sequence of subgradients g_1, \dots, g_T , define G_t and $\frac{1}{\eta_t}$ as in Definition 4.6, and define $G_0 = G_1$. We define $\frac{1}{\eta_t^+}$ recursively by:*

$$\begin{aligned} \frac{1}{\eta_0^+} &= \frac{1}{\eta_0} \\ \frac{1}{(\eta_t^+)^2} &= \max \left(\frac{1}{\eta_{t-1}^2} + 2\|g_t\|_* \min(\|g_t\|_*, G_{t-1}), G_{t-1}\|g_{1:t}\|_* \right) \end{aligned}$$

Further, given a $k \geq 1$ and a non-decreasing sequence of positive numbers a_t , define ψ_t^+ by:

$$\begin{aligned}\psi_t^+(w) &= \frac{k}{\eta_t^+ a_{t-1}} \psi(a_{t-1} w) \\ w_{t+1}^+ &= \underset{w \in W}{\operatorname{argmin}} \psi_t^+(w) + g_{1:t} \cdot w\end{aligned}$$

Throughout the following arguments we will assume η_t and η_t^+ are the sequences defined in Definitions 4.6 and 4.19.

The next proposition establishes several identities that we will need in proving our bounds.

Proposition 4.20. *Suppose ψ is a $(\sigma, \|\cdot\|)$ -adaptive regularizer, and g_1, \dots, g_T be some sequence of subgradients. Then the following identities hold:*

1.

$$2\|g_t\|_* G_{t-1} \eta_t^+ \geq \left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}} \right) \geq \|g_t\|_* \min(\|g_t\|_*, G_{t-1}) \eta_t^+$$

2.

$$\begin{aligned}\frac{1}{\eta_t} &\leq \sqrt{2G_t(\|g\|_*)_{1:t}} \\ \frac{1}{\eta_t} &\leq \sqrt{2(\|g\|_*^2)_{1:t} + G_{\max} \max_{t' \leq t} \|g_{1:t'}\|_*}\end{aligned}$$

3.

$$\|w_t - w_{t+1}^+\| \leq \frac{\|g_t\|_* \eta_t^+ + \left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}} \right) \frac{1}{G_{t-1}}}{a_{t-1} k \sigma(a_{t-1} w_t, a_{t-1} w_{t+1}^+)}$$

4. *Let $\hat{\psi}$ be such that $\hat{\psi}(a_{t-1} w) = \psi(a_{t-1} w)$ for $w \in W$ and $\hat{\psi}(a_{t-1} w) = \infty$ for $w \notin W$. There exists some subgradient of $\hat{\psi}$ at $a_{t-1} w_t$, which with mild abuse of notation we call $\nabla \hat{\psi}(a_{t-1} w_t)$, such that:*

$$|\nabla \hat{\psi}(a_{t-1} w_t) \cdot (w_t - w_{t+1}^+)| \leq 3 \frac{\frac{\|g_t\|_*}{G_{t-1}}}{a_{t-1} k^2 \sigma(a_{t-1} w_t, a_{t-1} w_{t+1}^+)}$$

5.

$$g_t \cdot (w_t - w_{t+1}^+) \leq \frac{\|g_t\|_*^2 \eta_t^+ + \left(\frac{1}{\eta_{t-1}} - \frac{1}{\eta_t^+} \right) \frac{\|g_t\|_*}{G_{t-1}}}{a_{t-1} k \sigma(a_{t-1} w_t, a_{t-1} w_{t+1}^+)}$$

6.

$$\frac{1}{\eta_t^+} \leq \sqrt{2G_{\max}(\|g\|_{\star})_{1:T-1} + 2G_{\max}G_{t-1}}$$

Proof. Let $\hat{\psi}$ be such that $\hat{\psi}(a_{t-1}w) = \psi(a_{t-1}w)$ for $w \in W$ and $\hat{\psi}(a_{t-1}w) = \infty$ for $w \notin W$. Then we can write $w_t = \operatorname{argmin}_{w \in W} \frac{k}{a_{t-1}\eta_{t-1}} \psi(a_{t-1}w) + g_{1:t-1} \cdot w = \operatorname{argmin}_{w \in W} \frac{k}{a_{t-1}\eta_{t-1}} \hat{\psi}(a_{t-1}w) + g_{1:t-1}$. From this it follows that there is some subgradient of $\hat{\psi}$ at $a_{t-1}w_t$, which we refer to (by mild abuse of notation) as $\nabla \hat{\psi}(a_{t-1}w_t)$ such that

$$\nabla \hat{\psi}(a_{t-1}w_t) = -\frac{\eta_{t-1}g_{1:t-1}}{k}$$

Note that we must appeal to a subgradient rather than the actual gradient in order to encompass the case that $a_{t-1}w_t$ is on the boundary of W .

Next, observe that

$$\eta_t^+ \eta_{t-1} \|g_{1:t-1}\|_{\star} \leq (\eta_{t-1})^2 \|g_{1:t-1}\|_{\star} \leq \frac{1}{G_{t-1}}$$

Now we are ready to prove the various parts of the Proposition.

1. By definition of η_{t-1} and η_t^+ we have

$$\begin{aligned} \frac{1}{(\eta_t^+)^2} - \frac{1}{(\eta_{t-1})^2} &\geq 2\|g_t\|_{\star} \min(\|g_t\|_{\star}, G_{t-1}) \\ \left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}}\right) \left(\frac{1}{\eta_t^+} + \frac{1}{\eta_{t-1}}\right) &\geq 2\|g_t\|_{\star} \min(\|g_t\|_{\star}, G_{t-1}) \\ \left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}}\right) \left(1 + \frac{\eta_t^+}{\eta_{t-1}}\right) &\geq 2\|g_t\|_{\star} \min(\|g_t\|_{\star}, G_{t-1}) \eta_t^+ \\ \frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}} &\geq \|g_t\|_{\star} \min(\|g_t\|_{\star}, G_{t-1}) \eta_t^+ \end{aligned}$$

where in the last line we used the fact that $\eta_t^+ \leq \eta_{t-1}$ to conclude that $1 + \frac{\eta_t^+}{\eta_{t-1}} \leq 2$.

For the other direction, we have two cases:

1. $\frac{1}{(\eta_t^+)^2} = \frac{1}{(\eta_{t-1})^2} + 2\|g_t\|_{\star} \min(\|g_t\|_{\star}, G_{t-1})$.
2. $\frac{1}{(\eta_t^+)^2} = G_{t-1} \|g_{1:t}\|_{\star}$.

Case 1 $\frac{1}{(\eta_t^+)^2} = \frac{1}{(\eta_{t-1})^2} + 2\|g_t\|_{\star} \min(\|g_t\|_{\star}, G_{t-1})$:

In this case we have

$$\begin{aligned}
\frac{1}{(\eta_t^+)^2} - \frac{1}{(\eta_{t-1})^2} &= 2\|g_t\|_\star \min(\|g_t\|_\star, G_{t-1}) \\
\left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}}\right) \left(\frac{1}{\eta_t^+} + \frac{1}{\eta_{t-1}}\right) &= 2\|g_t\|_\star \min(\|g_t\|_\star, G_{t-1}) \\
\left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}}\right) \left(1 + \frac{\eta_t^+}{\eta_{t-1}}\right) &= 2\|g_t\|_\star \min(\|g_t\|_\star, G_{t-1})\eta_t^+ \\
\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}} &\leq 2\|g_t\|_\star \min(\|g_t\|_\star, G_{t-1})\eta_t^+
\end{aligned}$$

where in the last line we used the fact that $1 + \frac{\eta_t^+}{\eta_{t-1}} \geq 1$.

Case 2 $\frac{1}{(\eta_t^+)^2} = G_{t-1}\|g_{1:t}\|_\star$:

$$\begin{aligned}
\frac{1}{(\eta_t^+)^2} - \frac{1}{(\eta_{t-1})^2} &\leq G_{t-1}\|g_{1:t}\|_\star - G_{t-1}\|g_{1:t-1}\|_\star \\
&\leq G_{t-1}\|g_t\|_\star \leq G_{t-1}\|g_t\|_\star
\end{aligned}$$

Now we follow the exact same argument as in Case 1 to show $\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}} \leq G_{t-1}\|g_t\|_\star\eta_t^+$, which proves the desired result.

2. We proceed by induction for both claims. The statements are clear for $\frac{1}{\eta_1} = \sqrt{2}\|g_1\|_\star$. Suppose

$$\begin{aligned}
\frac{1}{\eta_t} &\leq \sqrt{2G_t(\|g\|_\star)_{1:t}} \\
\frac{1}{\eta_t} &\leq \sqrt{2(\|g\|_\star^2)_{1:t} + G_{\max} \max_{t' \leq t} \|g_{1:t'}\|_\star}
\end{aligned}$$

Then observe that $\frac{1}{\eta_t^2} + 2\|g_{t+1}\|_\star^2 \leq 2G_{t+1}(\|g\|_\star)_{1:t+1}$ by the induction hypothesis, and $G_{t+1}\|g_{1:t+1}\|_\star \leq 2G_{t+1}(\|g\|_\star)_{1:t+1}$. Therefore $\frac{1}{\eta_{t+1}} \leq \sqrt{2G_{t+1}(\|g\|_\star)_{1:t+1}}$, proving the first claim.

The induction step for the second claim follows from the observations:

$$\begin{aligned}
2(\|g\|_\star^2)_{1:t+1} + G_{\max} \max_{t' \leq t+1} \|g_{1:t'}\|_\star &\geq 2(\|g\|_\star^2)_{1:t} + G_{\max} \max_{t' \leq t} \|g_{1:t'}\|_\star + 2\|g_{t+1}\|_\star^2 \\
2(\|g\|_\star^2)_{1:t+1} + G_{\max} \max_{t' \leq t+1} \|g_{1:t'}\|_\star &\geq G_t\|g_{1:t+1}\|_\star
\end{aligned}$$

so that $\frac{1}{\eta_{t+1}} \leq \sqrt{2(\|g\|_\star^2)_{1:t+1} + G_{\max} \max_{t' \leq t+1} \|g_{1:t'}\|_\star}$ as desired.

3. Let $I_{a_{t-1}W}(w)$ be the indicator of the set $a_{t-1}W - I_{a_{t-1}W}(a_{t-1}w) = 0$ if $w \in W$ and ∞ otherwise. Observe that $\hat{\psi}(w) = \psi(w) + I_{a_{t-1}W}(w)$. Observe that $\hat{\psi}(w) = I_{a_{t-1}W}(w) + \psi(w)$.

Now the third equation follows from Lemma 4.16, setting $A(w) = I_{a_{t-1}}W(w) + \frac{k}{a_{t-1}\eta_{t-1}}\psi(w) + \frac{g_{1:t-1}}{a_{t-1}} \cdot w$ and $B(w) = I_{a_{t-1}}W(w) + \frac{g_t}{a_{t-1}} \cdot w + \left(\frac{1}{a_{t-1}\eta_t^+} - \frac{k}{a_{t-1}\eta_{t-1}}\right)\psi(w)$. Then by inspection of the definitions of w_t and w_{t+1}^+ , we have $a_{t-1}w_t = \operatorname{argmin} A$ and $a_{t-1}w_{t+1}^+ = \operatorname{argmin} A + B$. Further, by Corollary 4.15, $A + B$ is $\frac{k\sigma}{a_{t-1}\eta_t^+}$ -strongly convex. We can re-write A and B in terms of $\hat{\psi}$ by simply replacing the ψ s with $\hat{\psi}$ s and removing the $I_{a_{t-1}}W$ s. Now we use the facts noted at the beginning of the proof:

$$\begin{aligned}\nabla\hat{\psi}(a_{t-1}w_t) &= -\frac{\eta_{t-1}g_{1:t-1}}{k} \\ \eta_t^+\eta_{t-1}\|g_{1:t-1}\| &\leq \frac{1}{G_{t-1}}\end{aligned}$$

Applying these identities with Lemma 4.16 we have:

$$\begin{aligned}\|a_{t-1}w_t - a_{t-1}w_{t+1}^+\| &\leq a_{t-1}\eta_t^+ \frac{\left\|\frac{g_t}{a_{t-1}} + \left(\frac{k}{a_{t-1}\eta_t^+} - \frac{k}{a_{t-1}\eta_{t-1}}\right)\nabla\hat{\psi}(a_{t-1}w_t)\right\|_*}{k\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \\ &\leq \frac{\eta_t^+\|g_t\|_*}{\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} + \frac{\eta_t^+ \left(\frac{k}{\eta_t^+} - \frac{k}{\eta_{t-1}}\right) \frac{\eta_{t-1}\|g_{1:t-1}\|_*}{k}}{k\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \\ &\leq \frac{\eta_t^+\|g_t\|_*}{k\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} + \frac{\left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}}\right) \frac{1}{G_{t-1}}}{k\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)}\end{aligned}$$

And we divide by a_{t-1} to conclude the desired identity.

4. Using the already-proved parts 1 and 3 of this Proposition and definition of dual norm, we have

$$\begin{aligned}|\nabla\hat{\psi}(a_{t-1}w_t) \cdot (w_t - w_{t+1}^+)| &\leq \|\nabla\hat{\psi}(a_{t-1}w_t)\|_* \|w_t - w_{t+1}^+\| \\ &\leq \frac{\eta_{t-1}\|g_{1:t-1}\|_*}{k} \frac{\eta_t^+\|g_t\|_*}{a_{t-1}k\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \\ &\quad + \frac{\eta_{t-1}\|g_{1:t-1}\|_*}{k} \frac{\left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}}\right) \frac{1}{G_{t-1}}}{a_{t-1}k\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \\ &\leq \frac{\frac{\|g_t\|_*}{G_{t-1}}}{a_{t-1}k^2\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} + \frac{\eta_t^+\eta_{t-1}\|g_{1:t-1}\|_* 2G_{t-1}\|g_t\| \frac{1}{G_{t-1}}}{a_{t-1}k^2\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \\ &\leq \frac{\frac{\|g_t\|_*}{G_{t-1}}}{a_{t-1}k^2\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} + \frac{\frac{1}{G_{t-1}^2} 2G_{t-1}\|g_t\|_*}{a_{t-1}k^2\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \\ &\leq 3 \frac{\frac{\|g_t\|_*}{G_{t-1}}}{a_{t-1}k^2\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)}\end{aligned}$$

5. The fifth part of the Proposition follows directly from part 3 by the definition of dual norm.

6. By part 2, we have

$$\frac{1}{\eta_{t-1}} \leq \sqrt{2G_{\max}(\|g\|_{\star})_{1:t-1}}$$

We consider the two cases:

Case 1 $\frac{1}{(\eta_t^+)^2} = \frac{1}{(\eta_{t-1})^2} + 2\|g_t\|_{\star} \min(\|g_t\|_{\star}, G_{t-1})$: In this case we have

$$\begin{aligned} \frac{1}{(\eta_t^+)^2} &\leq 2G_{\max}(\|g\|_{\star})_{1:t-1} + 2\|g_t\|_{\star} \min(\|g_t\|_{\star}, G_{t-1}) \\ &\leq 2G_{\max}(\|g\|_{\star})_{1:t-1} + 2G_{\max}G_{t-1} \end{aligned}$$

Case 2 $\frac{1}{(\eta_t^+)^2} = G_{t-1}\|g_{1:t}\|_{\star}$:

$$\begin{aligned} \frac{1}{(\eta_t^+)^2} &\leq G_{t-1}\|g_{1:t}\|_{\star} \\ &\leq G_{t-1}\|g_{1:t-1}\| + G_{t-1}\|g_t\| \\ &\leq G_{\max}(\|g\|_{\star})_{1:t-1} + G_{\max}G_{t-1} \end{aligned}$$

□

Lemma 4.21. *Suppose ψ a $(\sigma, \|\cdot\|)$ -adaptive regularizer and g_1, \dots, g_T is some sequence of subgradients. We use the terminology of Definition 4.6. Recall that we define $h(w) = \psi(w)\sigma(w)$ and $h^{-1}(x) = \max_{h(w) \leq x} \|w\|$. Suppose either of the follow holds:*

1. $\|w_{t+1}^+\| \geq \frac{h^{-1}\left(2\frac{G_t}{k^2G_{t-1}}\right)}{a_{t-1}}$ and $\|w_{t+1}^+\| \geq \|w_t\|$.
2. $\|w_t\| \geq \frac{h^{-1}\left(5\frac{G_t}{k^2G_{t-1}}\right)}{a_{t-1}}$ and $\|w_t\| \geq \|w_{t+1}^+\|$.

Then

$$\psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) + g_t(w_t - w_{t+1}^+) \leq 0$$

Proof. As in Proposition 4.20, we use $\nabla\psi(x)$ to simply mean some particular subgradient of ψ at x .

Case 1: $\|w_{t+1}^+\| \geq \frac{h^{-1}\left(2\frac{G_t}{k^2G_{t-1}}\right)}{a_{t-1}}$ and $\|w_{t+1}^+\| \geq \|w_t\|$:

By definition of adaptive regularizer (part 2), we must have $\sigma(a_{t-1}w_{t+1}^+) \leq \sigma(a_{t-1}w_t)$ since $\|w_{t+1}^+\| \geq \|w_t\|$. Therefore $\sigma(a_{t-1}w_{t+1}^+, a_{t-1}w_t) = \sigma(a_{t-1}w_{t+1}^+)$.

By definition of h , when $\|w_{t+1}^+\| \geq \frac{h^{-1}\left(2\frac{G_t}{k^2G_{t-1}}\right)}{a_{t-1}}$ we can apply Proposition 4.20 (parts 1 and 5) to obtain

$$\begin{aligned}
\psi(a_{t-1}w_{t+1}^+)\sigma(a_{t-1}w_{t+1}^+) &\geq 2\frac{G_t}{k^2G_{t-1}} \\
\left(\frac{1}{a_{t-1}\eta_t^+} - \frac{1}{a_{t-1}\eta_{t-1}}\right)\psi(a_{t-1}w_{t+1}^+) &\geq \frac{\left(\frac{1}{a_{t-1}\eta_t^+} - \frac{1}{a_{t-1}\eta_{t-1}}\right)2\frac{G_t}{G_{t-1}}}{k^2\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \\
\left(\frac{k}{a_{t-1}\eta_t^+} - \frac{k}{a_{t-1}\eta_{t-1}}\right)\psi(a_{t-1}w_{t+1}^+) &\geq \frac{\left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}}\right)}{a_{t-1}k\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)}2\frac{G_t}{G_{t-1}} \\
\psi_t^+(w_{t+1}^+) - \psi_{t-1}(w_{t+1}^+) &\geq \frac{\|g_t\|_* \min(\|g_t\|_*, G_{t-1})\eta_t^+ \frac{G_t}{G_{t-1}} + \left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}}\right) \frac{G_t}{G_{t-1}}}{a_{t-1}k\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \\
&\geq \frac{\|g_t\|_*^2\eta_t^+ + \left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}}\right) \frac{\|g_t\|_*}{G_{t-1}}}{a_{t-1}k\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \\
&\geq g_t \cdot (w_t - w_{t+1}^+)
\end{aligned}$$

We remark that in the calculations above, we showed

$$\frac{\left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}}\right)2\frac{G_t}{G_{t-1}}}{a_{t-1}\sigma(a_{t-1}kw_t, a_{t-1}w_{t+1}^+)} \geq g_t(w_t - w_{t+1}^+)$$

which we will re-use in Case 2.

Case 2 $\|w_t\| \geq \frac{h^{-1}\left(5\frac{\|g_t\|_*}{k^2G_{t-1}}\right)}{a_{t-1}}$, and $\|w_t\| \geq \|w_{t+1}^+\|$:

Again, by definition of adaptive regularizer (part 2), we must have $\sigma(a_{t-1}w_{t+1}^+) \geq \sigma(a_{t-1}w_t)$ since $\|w_{t+1}^+\| \leq \|w_t\|$. Therefore $\sigma(a_{t-1}w_{t+1}^+, a_{t-1}w_t) = \sigma(a_{t-1}w_t)$. Let $\hat{\psi}$ be as in Proposition 4.20 part 4. Observe that w_{t+1}^+ and w_t are both in W , so that we have $\psi(a_{t-1}w_{t+1}^+) = \hat{\psi}(a_{t-1}w_{t+1}^+)$ and $\psi(a_{t-1}w_t) =$

$\hat{\psi}(a_{t-1}w_t)$. Then we have:

$$\begin{aligned}
\psi_t^+(w_{t+1}^+) - \psi_{t-1}(w_{t+1}^+) &= \left(\frac{k}{a_{t-1}\eta_t^+} - \frac{k}{a_{t-1}\eta_{t-1}} \right) \psi(a_{t-1}w_{t+1}^+) \\
&= \left(\frac{k}{a_{t-1}\eta_t^+} - \frac{k}{a_{t-1}\eta_{t-1}} \right) \hat{\psi}(a_{t-1}w_{t+1}^+) \\
&\geq \left(\frac{k}{a_{t-1}\eta_t^+} - \frac{k}{a_{t-1}\eta_{t-1}} \right) \left(\hat{\psi}(a_{t-1}w_t) - \left| a_{t-1} \nabla \hat{\psi}(a_{t-1}w_t) \cdot (w_{t+1}^+ - w_t) \right| \right) \\
&\geq \left(\frac{k}{\eta_t^+} - \frac{k}{\eta_{t-1}} \right) \left(\frac{\psi(a_{t-1}w_t)}{a_{t-1}} - 3 \frac{\frac{\|g_t\|_*}{G_{t-1}}}{a_{t-1}k^2\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \right) \\
&\geq \left(\frac{k}{\eta_t^+} - \frac{k}{\eta_{t-1}} \right) \left(\frac{\psi(a_{t-1}w_t)}{a_{t-1}} - 3 \frac{\frac{G_t}{G_{t-1}}}{a_{t-1}k^2\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \right)
\end{aligned}$$

Now by definition of h , when $\|w_t\| \geq \frac{h^{-1}(5\frac{G_t}{k^2G_{t-1}})}{a_{t-1}}$ we have

$$\begin{aligned}
\psi(a_{t-1}w_t)\sigma(a_{t-1}w_t) &\geq 5 \frac{G_t}{k^2G_{t-1}} \\
\left(\frac{k}{a_{t-1}\eta_t^+} - \frac{k}{a_{t-1}\eta_{t-1}} \right) \psi(a_{t-1}w_t) &\geq \frac{\left(\frac{k}{\eta_t^+} - \frac{k}{\eta_{t-1}} \right) 5 \frac{G_t}{G_{t-1}}}{a_{t-1}k^2\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \\
\left(\frac{k}{\eta_t^+} - \frac{k}{\eta_{t-1}} \right) \left(\frac{\psi(a_{t-1}w_t)}{a_{t-1}} - 3 \frac{\frac{G_t}{G_{t-1}}}{a_{t-1}k^2\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \right) &\geq \frac{\left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}} \right) 2 \frac{G_t}{G_{t-1}}}{a_{t-1}k\sigma(a_{t-1}w_t, a_{t-1}w_{t+1}^+)} \\
\psi_t^+(w_{t+1}^+) - \psi_{t-1}(w_{t+1}^+) &\geq g_t \cdot (w_t - w_{t+1}^+)
\end{aligned}$$

□

The next theorem is a general fact about adaptive regularizers that is useful for controlling $\psi_t^+ - \psi_t$:

Proposition 4.22. *Suppose $\psi : W \rightarrow \mathbb{R}$ is a $(\sigma, \|\cdot\|)$ -adaptive regularizer. Then $\frac{\psi(aw)}{a}$ is an increasing function of a for all $a > 0$ for all $w \in W$.*

Proof. Let's differentiate: $\frac{d}{da} \frac{\psi(aw)}{a} = \frac{\nabla \psi(aw) \cdot w}{a} - \frac{\psi(aw)}{a^2}$. Thus it suffices to show

$$\nabla \psi(aw) \cdot aw \geq \psi(aw)$$

But this follows immediately from the definition of subgradient, since $\psi(0) = 0$. □

Lemma 4.23. *Suppose ψ is a $(\sigma, \|\cdot\|)$ -adaptive regularizer and g_1, \dots, g_T is an arbitrary sequence of subgradients (possibly chosen adaptively). Using the terminology of Definition 4.6,*

$$\psi_t^+(w_{t+2}^+) - \psi_t(w_{t+1}^+) \leq 0$$

for all t

Proof. This follows from the fact that $a_{t-1} \leq a_t$, and property 4 of an adaptive regularizer ($\psi(ax)/a$ is a non-decreasing function of a). By Proposition 4.20 (part 1), we have $\frac{1}{\eta_t^+} \leq \frac{1}{\eta_t}$. Therefore:

$$\begin{aligned} \psi_t^+(w_{t+2}^+) &= \frac{k}{\eta_t^+ a_{t-1}} \psi(a_{t-1} w_{t+2}^+) \\ &\leq \frac{k}{\eta_t a_{t-1}} \psi(a_{t-1} w_{t+2}^+) \\ &\leq \frac{k}{\eta_t a_t} \psi(a_t w_{t+2}^+) \\ &= \psi_t(w_{t+2}^+) \end{aligned}$$

□

Lemma 4.24. *Suppose ψ is a $(\sigma, \|\cdot\|)$ -adaptive regularizer and g_1, \dots, g_T is an arbitrary sequence of subgradients (possibly chosen adaptively). We use the regularizers of Definition 4.6. Recall that we define $h(w) = \psi(w)\sigma(w)$ and $h^{-1}(x) = \operatorname{argmax}_{h(w) \leq x} \|w\|$. Define*

$$\sigma_{\min} = \inf_{\|w\| \leq h^{-1}(10/k^2)} k\sigma(w)$$

and

$$D = 2 \max_t \frac{h^{-1}\left(5 \frac{G_t}{k G_{t-1}}\right)}{a_{t-1}}$$

Then

$$\begin{aligned} &\psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) + g_t(w_t - w_{t+1}^+) \\ &\leq \begin{cases} \|g_t\|_* \min(D, \max_t(\|w_t - w_{t+1}^+\|)) & \text{when } \|g_t\| > 2G_{t-1} \\ \frac{3\|g_t\|_*^2 \eta_t^+}{a_{t-1} \sigma_{\min}} & \text{otherwise} \end{cases} \end{aligned}$$

Proof. By Lemma 4.21, whenever either $\|w_{t+1}^+\| \geq \frac{h^{-1}\left(5 \frac{G_t}{k^2 G_{t-1}}\right)}{a_{t-1}} \geq \frac{h^{-1}\left(2 \frac{G_t}{k^2 G_{t-1}}\right)}{a_{t-1}}$ or $\|w_t\| \geq \frac{h^{-1}\left(5 \frac{G_t}{k^2 G_{t-1}}\right)}{a_{t-1}}$ we must have

$$\psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) + g_t(w_t - w_{t+1}^+) \leq 0$$

Therefore, we have:

$$\psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) + g_t(w_t - w_{t+1}^+) \leq \begin{cases} g_t \cdot (w_t - w_{t+1}^+) & \text{when } \max(\|w_t\|, \|w_{t+1}^+\|) \leq \frac{h^{-1}\left(5 \frac{G_t}{k^2 G_{t-1}}\right)}{a_{t-1}} \\ 0 & \text{otherwise} \end{cases}$$

When $\|g_t\|_\star \leq 2G_{t-1}$, then we have $h^{-1}\left(5\frac{G_t}{k^2G_{t-1}}\right) \leq h^{-1}(10/k^2)$. Thus when $\max(\|w_t\|, \|w_{t+1}^+\|) \leq \frac{h^{-1}\left(5\frac{G_t}{k^2G_{t-1}}\right)}{a_{t-1}}$ and $\|g_t\|_\star \leq 2G_{t-1}$, by Proposition 4.20 (part 5), we have

$$g_t(w_t - w_{t+1}^+) \leq \frac{\|g_t\|_\star^2 \eta_t^+ + \left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}}\right) \frac{\|g_t\|_\star}{G_{t-1}}}{a_{t-1} \sigma_{\min}}$$

Therefore when $\|g_t\|_\star \leq 2G_{t-1}$ we have (using Proposition 4.20 part 1):

$$\begin{aligned} \psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) + g_t(w_t - w_{t+1}^+) &\leq \frac{\|g_t\|_\star^2 \eta_t^+ + \left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}}\right) \frac{\|g_t\|_\star}{G_{t-1}}}{a_{t-1} \sigma_{\min}} \\ &\leq \frac{\|g_t\|_\star^2 \eta_t^+ + 2\|g_t\|_\star G_{t-1} \eta_t^+ \frac{\|g_t\|_\star}{G_{t-1}}}{a_{t-1} \sigma_{\min}} \\ &\leq \frac{3\|g_t\|_\star^2 \eta_t^+}{a_{t-1} \sigma_{\min}} \end{aligned}$$

so that we can improve our conditional bound to:

$$\begin{aligned} &\psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) + g_t(w_t - w_{t+1}^+) \\ &\leq \begin{cases} g_t \cdot (w_t - w_{t+1}^+) & \text{when } \max(\|w_t\|, \|w_{t+1}^+\|) \leq \frac{h^{-1}\left(5\frac{G_t}{k^2G_{t-1}}\right)}{a_{t-1}} \text{ and } \|g_t\|_\star > 2G_{t-1} \\ \frac{3\|g_t\|_\star^2 \eta_t^+}{a_{t-1} \sigma_{\min}} & \text{when } \max(\|w_t\|, \|w_{t+1}^+\|) \leq \frac{h^{-1}\left(5\frac{G_t}{k^2G_{t-1}}\right)}{a_{t-1}} \text{ and } \|g_t\|_\star \leq 2G_{t-1} \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

When both $\|w_{t+1}^+\|$ and $\|w_t\|$ are less than than $\frac{h^{-1}\left(5\frac{G_t}{k^2G_{t-1}}\right)}{a_{t-1}}$ then we also have

$$\|w_t - w_{t+1}^+\| \leq \min\left(D, \max_t \|w_t - w_{t+1}^+\|\right)$$

where we define

$$D = 2 \max_t \frac{h^{-1}\left(5\frac{G_t}{k^2G_{t-1}}\right)}{a_{t-1}}$$

Therefore we have

$$\begin{aligned}
& \psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) + g_t(w_t - w_{t+1}^+) \\
& \leq \begin{cases} g_t \cdot (w_t - w_{t+1}^+) & \text{when } \max(\|w_t\|, \|w_{t+1}^+\|) \leq \frac{h^{-1}\left(5\frac{G_t}{G_{t-1}}\right)}{a_{t-1}} \text{ and } \|g_t\|_* > 2G_{t-1} \\ \frac{3\|g_t\|_*^2 \eta_t^+}{a_{t-1} \sigma_{\min}} & \text{when } \max(\|w_t\|, \|w_{t+1}^+\|) \leq \frac{h^{-1}\left(5\frac{G_t}{k^2 G_{t-1}}\right)}{a_{t-1}} \text{ and } \|g_t\|_* \leq 2G_{t-1} \\ 0 & \text{otherwise} \end{cases} \\
& \leq \begin{cases} \|g_t\|_* \min(D, \max_t \|w_t - w_{t+1}^+\|), & \text{when } \|g_t\|_* > 2G_{t-1} \\ \frac{3\|g_t\|_*^2 \eta_t^+}{a_{t-1} \sigma_{\min}} & \text{otherwise} \end{cases}
\end{aligned}$$

□

Now we have three more technical lemmas:

Lemma 4.25. *Let a_1, \dots, a_M be a sequence of non-negative numbers such that $a_{i+1} \geq 2a_i$. Then*

$$\sum_{i=1}^M a_i \leq 2a_M$$

Proof. We proceed by induction on M . For the base case, we observe that $a_1 \leq 2a_1$. Suppose $\sum_{i=1}^{M-1} a_i \leq 2a_{M-1}$. Then we have

$$\begin{aligned}
\sum_{i=1}^M a_i &= a_M + \sum_{i=1}^{M-1} a_i \\
&\leq a_M + 2a_{M-1} \\
&\leq a_M + a_M = 2a_M
\end{aligned}$$

□

The next lemma establishes some identities analogous to the bounds $\sum_{t=1}^T \frac{1}{\sqrt{t}} = O(\sqrt{T})$, and $\sum_{t=1}^T \frac{1}{T^2} = O(1)$. These are useful for dealing with increasing a_t in our regret bounds.

Lemma 4.26. *1.*

$$\sum_{t \mid \|g_t\|_* \leq 2G_{t-1}} \|g_t\|_*^2 \eta_t^+ \leq \frac{2}{\eta_T^+}$$

2. Suppose α_t is defined by

$$\alpha_0 = \frac{1}{(G_1\eta_1)^2}$$

$$\alpha_t = \max\left(\alpha_{t-1}, \frac{1}{(G_t\eta_t)^2}\right)$$

then

$$\sum_{t \mid \|g_t\|_* \leq 2G_{t-1}} \|g_t\|_*^2 \frac{\eta_t^+}{\alpha_{t-1}} \leq 15G_{\max}$$

Proof. 1. Using part 1 from Proposition 4.20, and observing that $\eta_t^+ \geq \eta_t$, we have

$$\begin{aligned} \sum_{t \mid \|g_t\|_* \leq 2G_{t-1}} \|g_t\|_*^2 \eta_t^+ &\leq \sum_{t \mid \|g_t\|_* \leq 2G_{t-1}} 2\|g_t\|_* \min(\|g_t\|_*, G_{t-1}) \eta_t^+ \\ &\leq \sum_{t \mid \|g_t\|_* \leq 2G_{t-1}} 2\left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}}\right) \\ &\leq \sum_{t \mid \|g_t\|_* \leq 2G_{t-1}} 2\left(\frac{1}{\eta_t^+} - \frac{1}{\eta_{t-1}^+}\right) \\ &\leq 2\eta_T^+ \end{aligned}$$

2. For the second part of the lemma, we observe that for $\|g_t\|_* \leq 2G_{t-1}$,

$$\begin{aligned} \frac{1}{(\eta_t^+)^2} &\geq \frac{1}{(\eta_{t-1})^2} + 2\|g_t\|_* \min(G_{t-1}, \|g_t\|_*) \\ &\geq \frac{1}{(\eta_{t-1})^2} + \|g_t\|_*^2 \\ &\geq (\|g\|_*^2)_{1:t} \end{aligned}$$

Similarly, we also have $(\|g\|_\star^2)_{1:t} \leq (1 + \frac{G_t^2}{G_{t-1}^2})(\|g\|_\star^2)_{1:t-1}$ so that

$$\begin{aligned}
\frac{1}{\alpha_{t-1}} &\leq G_{t-1}^2 \eta_{t-1}^2 \\
&\leq \frac{G_{t-1}^2}{2(\|g\|_\star^2)_{1:t-1}} \\
&\leq \frac{G_{t-1}}{G_t} \frac{G_t^2}{2(\|g\|_\star^2)_{1:t-1}} \\
&\leq \frac{G_{t-1}}{G_t} \left(1 + \frac{G_t^2}{G_{t-1}^2}\right) \frac{G_t^2}{2(\|g\|_\star^2)_{1:t}} \\
&= \left(\frac{G_{t-1}}{G_t} + \frac{G_t}{G_{t-1}}\right) \frac{G_t^2}{2(\|g\|_\star^2)_{1:t}} \\
&\leq \frac{5}{4} \frac{G_t^2}{(\|g\|_\star^2)_{1:t}}
\end{aligned}$$

where in the last line we have used $G_t/G_{t-1} \leq 2$.

Combining these two calculations, we have

$$\sum_{t \mid \|g_t\|_\star \leq 2G_{t-1}} \|g_t\|_\star^2 \frac{\eta_t^+}{\alpha_{t-1}} \leq \frac{5}{4} \sum_{t \mid \|g_t\|_\star \leq 2G_{t-1}} \frac{\|g_t\|_\star^2 G_t^2}{(\|g\|_\star^2)_{1:t}^{3/2}}$$

Let T_1, T_2, \dots, T_n be the indices such that $\|g_{T_i}\|_\star > 2G_{T_i-1}$, and define $T_n = T + 1$. We will show that for any i with $T_{i+1} > T_i + 1$,

$$\sum_{t=T_i+1}^{T_{i+1}-1} \frac{\|g_t\|_\star^2 G_t^2}{(\|g\|_\star^2)_{1:t}^{3/2}} \leq 6G_{T_{i+1}-1} \quad (4.3)$$

Observe that for $N = T_i + 1$, we have

$$\sum_{t=T_i+1}^N \frac{\|g_t\|_\star^2 G_t^2}{(\|g\|_\star^2)_{1:t}^{3/2}} \leq 6G_N - \frac{2G_N^2}{\sqrt{(\|g\|_\star^2)_{1:N}}} \quad (4.4)$$

We'll prove by induction that equation (4.4) holds for all $N \leq T_{i+1} - 1$. Suppose it holds for some $N < T_{i+1} - 1$. Then by concavity of $-\frac{1}{\sqrt{x}}$, we have

$$\left(6G_{N+1} - \frac{2G_{N+1}^2}{\sqrt{(\|g\|_\star^2)_{1:N+1}}}\right) - \left(6G_N - \frac{2G_N^2}{\sqrt{(\|g\|_\star^2)_{1:N}}}\right) \geq \frac{\|g_{N+1}\|_\star^2 G_{N+1}^2}{(\|g\|_\star^2)_{1:N+1}^{3/2}}$$

So using the inductive hypothesis:

$$\begin{aligned}
\sum_{t=1}^{N+1} \frac{\|g_t\|_*^2 G_t^2}{(\|g\|_*^2)_{1:t}^{3/2}} &\leq \left(6G_N - \frac{2G_N^2}{\sqrt{(\|g\|_*^2)_{1:N}}}\right) + \frac{\|g_{N+1}\|_*^2 G_{N+1}^2}{(\|g\|_*^2)_{1:N+1}^{3/2}} \\
&= \left(6G_{N+1} - \frac{2G_{N+1}^2}{\sqrt{(\|g\|_*^2)_{1:N}}}\right) + \frac{\|g_{N+1}\|_*^2 G_{N+1}^2}{(\|g\|_*^2)_{1:N+1}^{3/2}} + 6(G_N - G_{N+1}) - \frac{2(G_N^2 - G_{N+1}^2)}{\sqrt{(\|g\|_*^2)_{1:N}}} \\
&\leq 6G_{N+1} - \frac{2G_{N+1}^2}{\sqrt{(\|g\|_*^2)_{1:N+1}}} + 6(G_N - G_{N+1}) - \frac{2(G_N^2 - G_{N+1}^2)}{\sqrt{(\|g\|_*^2)_{1:N}}}
\end{aligned}$$

To finish the induction, we show that $6(G_N - G_{N+1}) - \frac{2(G_N^2 - G_{N+1}^2)}{\sqrt{(\|g\|_*^2)_{1:N}}} \leq 0$. We factor out the non-negative quantity $G_{N+1} - G_N$, and then observe that $G_{N+1} \leq 2G_N$ since $T_i + 1 \leq N < N + 1 \leq T_{i+1} - 1$ (and in particular, $G_{N+1} \neq T_i$ for any i).

$$\begin{aligned}
-6 + \frac{2(G_N + G_{N+1})}{\sqrt{(\|g\|_*^2)_{1:N}}} &\leq -6 + \frac{6G_N}{\sqrt{(\|g\|_*^2)_{1:N}}} \\
&\leq 0
\end{aligned}$$

Therefore equation (4.4) holds for all $N \leq T_{i+1} - 1$, so that we have

$$\sum_{t=T_i+1}^{T_{i+1}-1} \frac{\|g_t\|_*^2 G_t^2}{(\|g\|_*^2)_{1:t}^{3/2}} \leq 6G_{T_{i+1}-1} - \frac{2G_{T_{i+1}-1}^2}{\sqrt{(\|g\|_*^2)_{1:T}}} \leq 6G_{T_{i+1}-1} \quad (4.5)$$

so that equation (4.3) holds. Now we write (using the convention that $\sum_{t=x}^z y_t = 0$ if $z < x$):

$$\begin{aligned}
\sum_{t \mid \|g_t\|_* \leq 2G_{t-1}} \frac{\|g_t\|_*^2 G_t^2}{(\|g\|_*^2)_{1:t}^{3/2}} &= \sum_{i=1}^{n+1} \sum_{t=T_i+1}^{T_{i+1}-1} \frac{\|g_t\|_*^2 G_t^2}{(\|g\|_*^2)_{1:t}^{3/2}} \\
&\leq \sum_{i=1}^{n+1} 6G_{T_{i+1}-1} \\
&\leq 12G_{\max}
\end{aligned}$$

where in the last step we have observed that by definition of T_i , $G_{T_{i+1}-1} \geq 2G_{T_i-1}$ for all i and used Lemma 4.25.

Finally, we conclude

$$\begin{aligned}
\sum_{t \mid \|g_t\|_* \leq 2G_{t-1}} \|g_t\|_*^2 \frac{\eta_t^+}{a_t} &\leq \frac{5}{4} \sum_{t=1}^T \frac{\|g_t\|_*^2 G_t^2}{(\|g\|_*^2)_{1:t}^{3/2}} \\
&\leq 15G_{\max}
\end{aligned}$$

□

Lemma 4.27. *Let α_t be defined by*

$$\alpha_0 = \frac{1}{(G_1\eta_1)^2}$$

$$\alpha_t = \max\left(\alpha_{t-1}, \frac{1}{(G_t\eta_t)^2}\right)$$

Then

$$\frac{2(\|g\|_{\star})_{1:t}}{G_t} \geq a_t \geq \frac{2(\|g\|_{\star}^2)_{1:t}}{G_t^2}$$

Proof. Since $\frac{1}{\eta_t^2} \geq 2(\|g\|_{\star}^2)_{1:t}$, we immediately recover the lower bound on a_t . The upper bound follows from Proposition 4.20 (part 2), which states $\frac{1}{\eta_t^2} \leq 2G_t(\|g\|_{\star})_{1:t}$ □

Now we're ready to prove Theorem 4.7, which we restate for reference:

Theorem 4.7. *Suppose ψ is a $(\sigma, \|\cdot\|)$ -adaptive regularizer and g_1, \dots, g_T is some arbitrary sequence of subgradients. Let $k \geq 1$, and let ψ_t be defined as in Definition 4.6.*

Set

$$\sigma_{\min} = \inf_{\|w\| \leq h^{-1}(10/k^2)} k\sigma(w)$$

$$D = \max_t \frac{G_{t-1}^2}{(\|g\|_{\star}^2)_{1:t-1}} h^{-1} \left(\frac{5G_t}{k^2 G_{t-1}} \right)$$

$$Q_T = 2 \frac{\|g\|_{1:T}}{G_{\max}}$$

Then FTRL with regularizers ψ_t achieves regret

$$R_T(\hat{w}) \leq \frac{k}{Q_T\eta_T} \psi(Q_T u) + \frac{45G_{\max}}{\sigma_{\min}} + 2G_{\max}D$$

$$\leq kG_{\max} \frac{\psi(2uT)}{\sqrt{2T}} + \frac{45G_{\max}}{\sigma_{\min}} + 2G_{\max}D$$

Proof. Using Theorem 4.14 and Lemmas 4.23 and 4.24, our regret is bounded by

$$\begin{aligned}
R_T(\hat{w}) &\leq \psi_T^+(\hat{w}) + \sum_{t=1}^T \psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) + g_t(w_t - w_{t+1}^+) \\
&\quad + \sum_{t=1}^T \psi_t^+(w_{t+2}^+) - \psi_t(w_{t+2}^+) \\
&\leq \psi_T^+(\hat{w}) + \sum_{t=1}^T \psi_{t-1}(w_{t+1}^+) - \psi_t^+(w_{t+1}^+) + g_t(w_t - w_{t+1}^+) \\
&\leq \psi_T^+(\hat{w}) + \sum_{\|g_t\|_* \leq 2G_{t-1}} \frac{3\|g_t\|^2 \eta_t^+}{a_{t-1} \sigma_{\min}} + \sum_{\|g_t\|_* > 2G_{t-1}} \|g_t\|_* D'
\end{aligned}$$

where D' is defined by

$$D' = 2 \max_t \frac{h^{-1} \left(5 \frac{G_t}{kG_{t-1}} \right)}{a_{t-1}}$$

Now we use Lemma 4.27 to conclude that

$$D' \leq D = \max_t \frac{G_{t-1}^2}{(\|g\|_*^2)_{1:t-1}} h^{-1} \left(5 \frac{G_t}{kG_{t-1}} \right)$$

so that we have

$$R_T(\hat{w}) \leq \psi_T^+(\hat{w}) + \sum_{\|g_t\|_* \leq 2G_{t-1}} \frac{3\|g_t\|^2 \eta_t^+}{a_{t-1} \sigma_{\min}} + \sum_{\|g_t\|_* > 2G_{t-1}} \|g_t\|_* D$$

Now using Lemma 4.26 we can simplify this to

$$R_T(\hat{w}) \leq \frac{k}{a_T \eta_T^+} \psi(a_T \hat{w}) + \frac{45G_{\max}}{\sigma_{\min}} + \sum_{\|g_t\|_* > 2G_{t-1}} \|g_t\|_* D$$

Finally, observe that each value of $\|g_t\|_*$ in the sum $\sum_{\|g_t\|_* > 2G_{t-1}} \|g_t\|_* D$ is at least twice the previous value, so that by Lemma 4.25 we conclude

$$R_T(\hat{w}) \leq \frac{k}{a_T \eta_T^+} \psi(a_T \hat{w}) + \frac{45G_{\max}}{\sigma_{\min}} + 2G_{\max} D$$

Finally, we observe that (by Lemma 4.27), $a_T \leq 2 \frac{\|g\|_{1:T}}{G_T} = Q_T$, which gives the first inequality in the Theorem statement.

Using the fact that $\frac{1}{\eta_t} \leq \sqrt{2G_{\max} (\|g\|_*)_{1:t}}$ (from Proposition 4.20 part 2), we have $\eta_T^+ \geq \frac{1}{G_{\max} \sqrt{2T}}$ and it is clear that $a_T \leq 2T$, so that we recover the second inequality as well. \square

Chapter 5

Reductions For Parameter-free Online Learning

In this chapter we introduce an alternative approach to the design of parameter-free algorithms that dramatically simplifies their design and analysis. Recall that in the previous chapter, we analyzed algorithms using the classic FTRL approach, which resulted in extremely complicated proofs because we were unable to use strongly-convex regularizers (at least in the usual sense of the term). The techniques in this chapter are instead based on the coin betting framework for designing algorithm (recall Section 2.5), which turns out to be much better suited for parameter-free algorithms. To respect the exponential lower-bound in the previous chapter, throughout this chapter we will assume a known bound on G_{\max} and instead focus on providing algorithms that adapt to finer-grained statistics of the g_t . In particular, we will assume WLOG that $G_{\max} = 1$, potentially by rescaling all losses by a constant factor. We will eventually show that adapting to these more fine-grained statistics, while on the surface a relatively subtle distinction, actually leads to significant asymptotic savings when the losses ℓ_t are strongly-convex or smooth (recall Definitions 2.11 and 2.13 (without knowing the strong convexity or smoothness parameters)). Most of the material in this chapter is taken from my paper with Francesco Orabona [16], with a few new additions in the later sections.

Our primary techniques are a series of three reductions that streamline the design of parameter-free algorithms by constructing them from simpler algorithms. First, we show that algorithms for online exp-concave optimization imply parameter-free algorithms for OLO (Section 5.1). Second, we show a general reduction from online learning in arbitrary dimensions with any norm to one-dimensional online learning (Section 5.2). Third, given any two convex sets $W \subset V$, we construct an online learning algorithm over W from an online learning algorithm over V (Section 5.3).

All of our reductions are very general. We make no assumptions about the inner workings of the base algorithms and are able to consider any norm, so that W may be a subset of a Banach space rather than a Hilbert space or \mathbb{R}^d . Each reduction is of independent interest, even for non-parameter-free algorithms, but

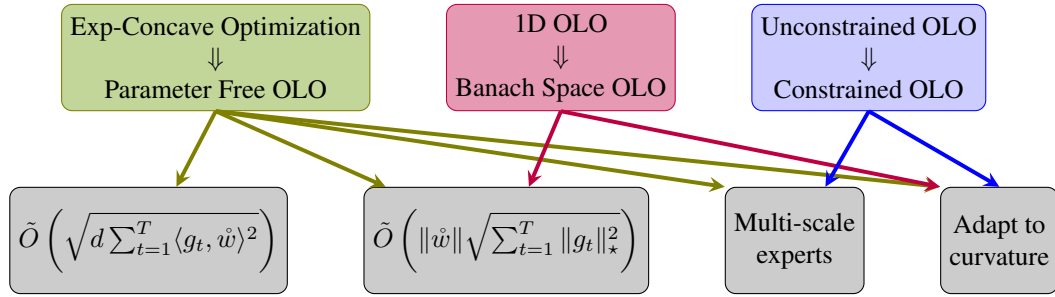


Figure 5.1: We prove three reductions (top row), and use these reductions to obtain specific algorithms and regret bounds (bottom row). Arrows indicate which reductions are used in each algorithm.

by combining them we can produce powerful new algorithms.

First, we use our reductions to design a new parameter-free algorithm that improves upon the prior regret bounds, achieving

$$R_T(\hat{w}) \leq \|\hat{w}\| \sqrt{\sum_{t=1}^T \|g_t\|_*^2 \ln \left(\|\hat{w}\| \sum_{t=1}^T \|g_t\|_*^2 + 1 \right)},$$

where $\|\cdot\|$ is any norm and $\|\cdot\|_*$ is the dual norm ($\|g_t\|_* = \|g_t\|$ when $\|\cdot\|$ is the 2-norm). Previous parameter-free algorithms [32; 35; 39; 40; 19; 12; 44] (including FREEREX) obtain at best an exponent of 1 in their dependence on $\|g_t\|_*$ (which is worse because $\|g_t\|_* \leq 1$ by our 1-Lipschitz assumption). Achieving $\|g_t\|_*^2$ rather than $\|g_t\|_*$ can imply asymptotically lower regret when the losses ℓ_t are smooth [56], so this is not merely a cosmetic difference. In addition to the worse regret bound, all prior analyses we are aware of are quite complicated, and are usually limited to the Hilbert spaces and the 2-norm. In contrast, the techniques presented in this chapter are both simpler and more general

We will later demonstrate the power of our reductions through several more applications. In Section 5.4 we prove a regret bound of the form $R_T(\hat{w}) = \tilde{O} \left(\sqrt{d \sum_{t=1}^T \langle g_t, \hat{w} \rangle^2} \right)$ for d -dimensional Banach spaces, extending the results of [27] to unconstrained domains. In Section 6.1, we consider the multi-scale experts problem studied in [20; 7] and improve prior regret guarantees and runtimes. Finally, in Chapter 7, we will investigate the consequences of these reductions for smooth or strongly convex losses, including creating an algorithm obtaining $\tilde{O}(\sqrt{T})$ regret for general convex losses, but logarithmic regret for strongly-convex losses using only first-order information, similar to [59; 14], but with runtime improved to match gradient descent. We summarize these results in Figure 5.1.

5.1 Exp-Concave Optimization to Online Linear Optimization via Betting Algorithms

In this section we show *how to convert an online exp-concave algorithm to construct a 1D parameter-free algorithm*. Our approach relies on the coin-betting abstraction for the design of parameter-free algorithms (see Section 2.5). Recall that coin betting strategies record the *wealth* of the algorithm, which is defined by some initial (i.e. user-specified) ϵ plus the total “reward” $\sum_{t=1}^T -g_t w_t$ it has gained:

$$\text{Wealth}_T = \epsilon - \sum_{t=1}^T g_t w_t. \quad (5.1)$$

Also recall that given this wealth measurement, coin betting algorithms bet a signed fraction $v_t \in (-1, 1)$ of their current wealth on the outcome of the coin $g_t \in [-1, 1]$ by playing $w_t = v_t \text{Wealth}_{t-1}$. Since high wealth is equivalent to a low regret the question is how to pick betting fractions v_t that guarantee high wealth. This is usually accomplished through careful design of bespoke potential functions and meticulous algebraic manipulation, but we take a different and simpler path.

At a high level, our approach is to re-cast the problem of choosing betting fractions v_t as itself an online learning problem. We show that this online learning problem has *exp-concave* losses rather than linear losses. Exp-concave losses are known to be much easier to optimize than linear losses and it is possible to obtain $\ln(T)$ regret using the Online Newton Step (ONS) algorithm rather than the \sqrt{T} limit for linear optimization [24]. So by using an exp-concave optimization algorithm such as ONS, we find the optimal betting fraction \hat{v} very quickly, and obtain high wealth. The pseudocode for the resulting strategy is in Algorithm 2.

Algorithm 2 Coin-Betting through ONS

Require: Initial wealth $\epsilon > 0$

- 1: **Initialize:** $\text{Wealth}_0 = \epsilon$, initial betting fraction $v_1 = 0$
 - 2: **for** $t = 1$ **to** T **do**
 - 3: Bet $w_t = v_t \text{Wealth}_{t-1}$
 - 4: Receive $g_t \in [-1, 1]$
 - 5: Update $\text{Wealth}_t = \text{Wealth}_{t-1} - g_t w_t$
 - 6: //compute new betting fraction $v_{t+1} \in [-1/2, 1/2]$ via ONS update on losses $-\ln(1 - g_t v)$
 - 7: Set $z_t = \frac{d}{dv_t} (-\ln(1 - g_t v_t)) = \frac{g_t}{1 - g_t v_t}$
 - 8: Set $A_t = 1 + \sum_{i=1}^t z_i^2$
 - 9: $v_{t+1} = \max\left(\min\left(v_t - \frac{2}{2 - \ln(3)} \frac{z_t}{A_t}, 1/2\right), -1/2\right)$
 - 10: **end for**
-

Later (in Section 5.4), we will see that this same 1D argument holds seamlessly in Banach spaces, where now the betting fraction v_t is a vector in the Banach space and the outcome of the coin g_t is a vector in the dual space with norm bounded by 1. We therefore postpone computing exact constants for the Big-O notation in Theorem 5.1 to the more general Theorem 5.7.

It is important to note that ONS in 1D is extremely simple to implement. Even the projection onto a bounded set becomes just a truncation between two real numbers, so that Algorithm 2 can run quickly. We can show the following regret guarantee:

Theorem 5.1. *For $|g_t| \leq 1$, Algorithm 2, guarantees the regret bound:*

$$R_T(\hat{w}) = O \left[\epsilon + \max \left(|\hat{w}| \ln \left[\frac{|\hat{w}| \sum_{t=1}^T g_t^2}{\epsilon} \right], |\hat{w}| \sqrt{\sum_{t=1}^T g_t^2 \ln \left[\frac{|\hat{w}|^2 \sum_{t=1}^T g_t^2}{\epsilon^2} + 1 \right]} \right) \right].$$

Proof. Define $\text{Wealth}_T(\hat{v})$ to be wealth of the betting algorithm that bets the constant (signed) fraction \hat{v} on every round, starting from initial wealth $\epsilon > 0$.

We begin with the regret-reward duality that is the start of all coin-betting analyses [40]. Suppose that we obtain a bound $\text{Wealth}_T \geq f_T \left(-\sum_{t=1}^T g_t \right)$ for some f_T . Then,

$$R_T(\hat{w}) - \epsilon = -\text{Wealth}_T - \sum_{t=1}^T g_t \hat{w} \leq -\sum_{t=1}^T g_t \hat{w} - f_T \left(-\sum_{t=1}^T g_t \right) \leq \sup_{G \in \mathbb{R}} G \hat{w} - f_T(G) = f_T^*(\hat{w}),$$

where f_T^* indicates the Fenchel conjugate, defined by $f_T^*(x) = \sup_{\theta} \theta x - f_T(\theta)$.

So, now it suffices to prove a wealth lower bound. First, observing that $\text{Wealth}_T = \text{Wealth}_{T-1} - \text{Wealth}_{T-1} g_t v_t$, we derive a simple expression for $\ln \text{Wealth}_T$ by recursion:

$$\ln \text{Wealth}_T = \ln(\text{Wealth}_{T-1}(1 - g_t v_t)) = \ln(\epsilon) + \sum_{t=1}^T \ln(1 - v_t g_t).$$

Similarly, we have $\ln \text{Wealth}_T(\hat{v}) = \ln(\epsilon) + \sum_{t=1}^T \ln(1 - \hat{v} g_t)$. We subtract the identities to obtain

$$\ln \text{Wealth}_T(\hat{v}) - \ln \text{Wealth}_T = \sum_{t=1}^T -\ln(1 - v_t g_t) - (-\ln(1 - \hat{v} g_t)). \quad (5.2)$$

Now, the key insight of this analysis: we interpret equation (5.2) as the regret of an algorithm playing v_t on losses $\ell_t(v) = -\ln(1 - v g_t)$, so that we can write

$$\ln \text{Wealth}_T = \ln \text{Wealth}_T(\hat{v}) - R_T^v(\hat{v}), \quad (5.3)$$

where $R_T^v(\hat{v})$ is the regret of our method for choosing v_t .

For the next step, observe that $-\ln(1 - g_t v)$ is exp-concave (a function f is exp-concave if $\exp(-f)$ is concave), so that choosing v_t is an online exp-concave optimization problem. Prior work on exp-concave optimization allows us to obtain $R_T^v(\hat{v}) = O \left(\ln \left(\sum_{t=1}^T g_t^2 \right) \right)$ for any $|\hat{v}| \leq \frac{1}{2}$ using the ONS algorithm. Therefore (dropping all constants for simplicity), we use (5.3) to obtain $\text{Wealth}_T \geq \text{Wealth}_T(\hat{v}) / \sum_{t=1}^T g_t^2$ for all $|\hat{v}| \leq \frac{1}{2}$.

Finally, we need to show that there exists \hat{v} such that $\text{Wealth}_T(\hat{v}) / \sum_{t=1}^T g_t^2$ is high enough to guarantee

low regret on our original problem. Consider $\hat{v} = \frac{-\sum_{t=1}^T g_t}{2\sum_{t=1}^T g_t^2 + 2|\sum_{t=1}^T g_t|} \in [-1/2, 1/2]$. Then, we invoke the tangent bound $\ln(1+x) \geq x - x^2$ for $x \in [-1/2, 1/2]$ (e.g. see [8]) to see:

$$\ln \text{Wealth}_T(\hat{v}) - \ln(\epsilon) = \sum_{t=1}^T \ln(1 - g_t \hat{v}) \geq -\sum_{t=1}^T g_t \hat{v} - \sum_{t=1}^T (g_t \hat{v})^2 \geq \frac{(\sum_{t=1}^T g_t)^2}{4\sum_{t=1}^T g_t^2 + 4|\sum_{t=1}^T g_t|}.$$

Overall we have obtained

$$\text{Wealth}_T \geq \epsilon \exp \left[\frac{(\sum_{t=1}^T g_t)^2}{4\sum_{t=1}^T g_t^2 + 4|\sum_{t=1}^T g_t|} \right] / \sum_{t=1}^T g_t^2 = f_T \left(\sum_{t=1}^T g_t \right),$$

where $f_T(x) = \epsilon \exp[x^2 / (4\sum_{t=1}^T g_t^2 + 4|x|)] / \sum_{t=1}^T g_t^2$. To obtain the desired result, we recall that $\text{Wealth}_T \geq f_T(\sum_{t=1}^T g_t)$ implies $R_T(\hat{w}) \leq \epsilon + f_T^*(\hat{w})$, and calculate f_T^* (see Lemma 5.27).

In order to implement the algorithm, we observe that our chosen reference betting fraction \hat{v} lies in $[-1/2, 1/2]$, so we can safely run ONS restricted to the domain $[-1/2, 1/2]$. Exact constants can be computed by substituting the constants coming from the ONS regret guarantee, as we do in the proof of Theorem 5.7. \square

5.2 From 1D Algorithms to Dimension-Free Algorithms

A common strategy for designing parameter-free algorithms is to first create an algorithm for 1D problems (as we did in the previous section), and then invoke some particular algorithm-specific analysis to extend the algorithm to high dimensional spaces [40; 13; 35]. This strategy is unappealing for a couple of reasons. First, these arguments are often somewhat tailored to the algorithm at hand, and so a new argument must be made for a new 1D algorithm (indeed, it is not clear that any prior dimensionality extension arguments apply to our Algorithm 2). Secondly, all such arguments we know of apply only to Hilbert spaces and so do not allow us to design algorithms that consider norms other than the standard Euclidean 2-norm. In this section we address both concerns by providing a *black-box reduction from optimization in any Banach space to 1D optimization*. In further contrast to previous work, our reduction can be proven in just a few lines.

Our reduction takes two inputs: an algorithm \mathcal{A}_{1D} that operates with domain \mathbb{R} and achieves regret $R_T^1(\hat{w})$ for any $\hat{w} \in \mathbb{R}$, and an algorithm \mathcal{A}_S that operates with domain equal to the unit ball S in some Banach space B , $S = \{x \in B : \|x\| \leq 1\}$ and obtains regret $R_T^{\mathcal{A}_S}(\hat{w})$ for any $\hat{w} \in S$. In the case when B is \mathbb{R}^d or a Hilbert space, then online gradient descent with adaptive step sizes can obtain $R_T^{\mathcal{A}_S}(\hat{w}) = \sqrt{2\sum_{t=1}^T \|g_t\|_2^2}$ (which is independent of \hat{w}) [25].

Given these inputs, the reduction uses the 1D algorithm \mathcal{A}_{1D} to learn a “magnitude” z and the unit-ball algorithm \mathcal{A}_S to learn a “direction” y . This direction and magnitude are multiplied together to form the final output $w = zy$. Given a gradient g , the “magnitude error” is given by $\langle g, y \rangle$, which is intuitively the component of the gradient parallel to w . The “direction error” is just g . Our reduction is described formally

in Algorithm 3 below.

Algorithm 3 One Dimensional Reduction

Require: 1D Online learning algorithm \mathcal{A}_{1D} , Banach space B and Online learning algorithm \mathcal{A}_S with domain equal to unit ball $S \subset B$

- 1: **for** $t = 1$ **to** T **do**
 - 2: Get point $z_t \in \mathbb{R}$ from \mathcal{A}_{1D}
 - 3: Get point $y_t \in S$ from \mathcal{A}_S
 - 4: Play $w_t = z_t y_t \in B$
 - 5: Receive subgradient g_t
 - 6: Set $s_t = \langle g_t, y_t \rangle$
 - 7: Send s_t as the t th subgradient to \mathcal{A}_{1D}
 - 8: Send g_t as the t th subgradient to \mathcal{A}_S
 - 9: **end for**
-

Theorem 5.2. Suppose \mathcal{A}_S obtains regret $R_T^{\mathcal{A}_S}(\hat{w})$ for any competitor \hat{w} in the unit ball and \mathcal{A}_{1D} obtains regret $R_T^1(\hat{w})$ for any competitor $\hat{w} \in \mathbb{R}$. Then Algorithm 3 guarantees regret:

$$R_T(\hat{w}) \leq R_T^1(\|\hat{w}\|) + \|\hat{w}\| R_T^{\mathcal{A}_S}(\hat{w}/\|\hat{w}\|).$$

Where by slight abuse of notation we set $\hat{w}/\|\hat{w}\| = 0$ when $\hat{w} = 0$. Further, the subgradients s_t sent to \mathcal{A}_{1D} satisfy $|s_t| \leq \|g_t\|_*$.

Proof. First, observe that $|s_t| \leq \|g_t\|_* \|y_t\| \leq \|g_t\|_*$ since $\|y_t\| \leq 1$ for all t . Now, compute:

$$\begin{aligned} R_T(\hat{w}) &= \sum_{t=1}^T \langle g_t, w_t - \hat{w} \rangle = \sum_{t=1}^T \langle g_t, z_t y_t \rangle - \langle g_t, \hat{w} \rangle \\ &= \sum_{t=1}^T \underbrace{\langle g_t, y_t \rangle z_t - \langle g_t, y_t \rangle \|\hat{w}\|}_{\text{regret of } \mathcal{A}_{1D} \text{ at } \|\hat{w}\| \in \mathbb{R}} + \langle g_t, y_t \rangle \|\hat{w}\| - \langle g_t, \hat{w} \rangle \\ &= R_T^1(\|\hat{w}\|) + \sum_{t=1}^T \langle g_t, y_t \rangle \|\hat{w}\| - \langle g_t, \hat{w} \rangle \\ &\leq R_T^1(\|\hat{w}\|) + \|\hat{w}\| \sum_{t=1}^T \underbrace{\langle g_t, y_t \rangle - \langle g_t, \hat{w}/\|\hat{w}\| \rangle}_{\text{regret of } \mathcal{A}_S \text{ at } \hat{w}/\|\hat{w}\| \in S} \\ &\leq R_T^1(\|\hat{w}\|) + \|\hat{w}\| R_T^{\mathcal{A}_S}(\hat{w}/\|\hat{w}\|), \end{aligned}$$

□

With this reduction in hand, designing dimension-free and parameter-free algorithms is now exactly as easy as designing 1D algorithms, so long as we have access to a unit-ball algorithm \mathcal{A}_S . As mentioned, for any Hilbert space we indeed have such an algorithm. In general, algorithms \mathcal{A}_S exist for most other Banach

spaces of interest [57], and in particular one can achieve $R_T^{As}(\dot{w}) \leq O\left(\sqrt{\frac{1}{\lambda} \sum_{t=1}^T \|g_t\|_*^2}\right)$ whenever B is $(2, \lambda)$ -uniformly convex [47] using the Follow-the-Regularized-Leader algorithm with regularizers scaled by $\frac{\sqrt{\lambda}}{\sqrt{\sum_{i=1}^t \|g_i\|_*^2}}$ [34]. When B is instead (p, D) -uniformly convex, for $p < 2$, then it is actually impossible to obtain $O(\sqrt{T})$ regret. However, the Mirror Descent algorithm, with appropriate regularizers, can obtain $O(T^{1/p})$ regret in the unit ball, which gives us a corresponding regret guarantee for unconstrained domains.

Applying Algorithm 3 to our 1D Algorithm 2, for any $(2, \lambda)$ -uniformly convex B , we obtain:

$$R_T(\dot{w}) = O \left[\|\dot{w}\| \max \left(\ln \frac{\|\dot{w}\| \sum_{t=1}^T \|g_t\|_*^2}{\epsilon}, \sqrt{\sum_{t=1}^T \|g_t\|_*^2} \ln \left(\frac{\|\dot{w}\|^2 \sum_{t=1}^T \|g_t\|_*^2}{\epsilon^2} + 1 \right) \right) + \frac{\|\dot{w}\|}{\sqrt{\lambda}} \sqrt{\sum_{t=1}^T \|g_t\|_*^2 + \epsilon} \right].$$

Spaces that satisfy this property include Hilbert spaces such as \mathbb{R}^d with the 2-norm (in which case $\lambda = 1$), as well the \mathbb{R}^d with the p -norm for $p \in (1, 2]$ (in which case $\lambda = p - 1$). Finally, observe that the runtime of this reduction is equal to the runtime of \mathcal{A}_{1D} plus the runtime of \mathcal{A}_S , which in many cases (including \mathbb{R}^d with 2-norm or Hilbert spaces) is the same as online gradient descent.

Not only does this provide the fastest known parameter-free algorithm for an arbitrary norm, it is also the first parameter-free algorithm to obtain a dependence on the gradients of $\|g_t\|_*^2$ rather than $\|g_t\|_*$. This improved bound immediately implies much lower regret in easier settings, such as smooth losses with small loss values at \dot{w} [56].

5.2.1 Alternate Reduction Without Unit-Ball Algorithm

The reduction of Algorithm 3 relies on the unit-ball algorithm \mathcal{A}_S . Although these algorithms do exist for all (p, λ) -uniformly convex Banach spaces, it might be more desirable to have a more explicit reduction. We address this concern with Algorithm 4 below, which provides an alternative reduction with an explicit formula for the “direction” vector. Interestingly, the strategy for choosing the direction does not appear to be a valid unit-ball online learning algorithm and so this reduction is not a special case of the previous one.

Theorem 5.3. *Suppose W is a (q, D) -uniformly convex reflexive Banach space for some $q \geq 1$ such that $\|\cdot\|_*^p$ is Frechet differentiable. Let \mathcal{A} be a 1-dimensional online learning algorithm that guarantees regret $R_T^1(\dot{w})$. Then:*

1. For $p = \frac{q}{q-1}$, there exists C such that $\|\cdot\|^p$ satisfies

$$\|x + y\|_*^p \leq \|x\|_*^p + \langle \nabla \|x\|_*^p, y \rangle + C \|y\|_*^p$$

2. The reduction Algorithm 4 guarantees:

$$\|g_{1:T}\| \leq |s_{1:T}| + (2^p + C\|g\|_{1:T}^p)^{1/p}$$

and each s_t satisfies $|s_t| \leq \|g_t\|$.

3.

$$R_T(\hat{w}) \leq \max_{c \in \{\pm 1\}} R_T^1(c\|\hat{w}\|) + \|\hat{w}\| (2^p + C\|g\|_{1:T}^p)^{1/p}$$

Algorithm 4 One dimensional to p -smooth banach space reduction

Input: Online learning algorithm \mathcal{A}

for $t = 1$ **to** T **do**

 Get point $z_t \in \mathbb{R}$ from \mathcal{A} .

 Play $w_t = z_t \text{sign}(s_{1:t-1}) \frac{\nabla \|g_{1:t-1}\|_*^p}{p \|g_{1:t-1}\|_*^{p-1}}$, (or if $g_{1:t-1} = 0$ play $w_t = z_t v$ for some arbitrary unit vector v)

 Receive subgradient g_t .

 Set $s_t = \text{sign}(s_{1:t-1}) \frac{\langle \nabla \|g_{1:t-1}\|_*^p, g_t \rangle}{p \|g_{1:t-1}\|_*^{p-1}}$ (or $s_t = g_t \cdot v$ if $g_{1:t-1} = 0$ for the same arbitrary unit vector v).

 Send s_t as the t th gradient to \mathcal{A} .

end for

By giving an explicit formula for generating direction vectors in Banach spaces, we can actually make slightly more detailed bounds by inspecting the properties of the direction vectors. For example, in a Hilbert space we have $p = 2$ and a little calculation shows that $|s_t| \leq \|g_t^\parallel\|_*$, where g_t^\parallel is the projection of g_t onto the 1-D subspace spanned by $g_{1:t-1}$. Now suppose we have $R_T^1(X) = \tilde{O}(|X| \sqrt{s_{1:T}^2})$. Then by tracing through the proof of Theorem 5.3, we actually get an improved regret bound of

$$\mathbb{R}_T(\hat{w}) \leq \tilde{O} \left(\|\hat{w}\| \sqrt{\|g^\parallel\|_{*1:T}^2} \right) + O \left(\|\hat{w}\| \sqrt{\|g\|_{*1:T}^2} \right)$$

So that we lose the logarithmic term in the \tilde{O} if $g_t^\parallel = 0$ for all t (in other words, if g_t is always perpendicular to $g_{1:t-1}$). This property is actually shared by many previous algorithms (including FREEREX), but is not easily extracted from the analysis. Here we are able to clearly articulate this behavior, and even observe the graceful degradation of the bound to obtain the extra logarithmic factor.

proof of Theorem 5.3. First, the existence of the constant C follows because the dual of a (q, D) -uniformly convex Banach space is necessarily $(p, C/2)$ -smooth for some C (see [48]). Then since $p \geq 1$, $\|\cdot\|_*^p$ is convex and so by Lemma 5.14 we have the desired statement.

Now we prove the second part of the theorem by induction. Suppose $\|g_{1:K}\|_* \leq |s_{1:K}| + (1+C\|g\|_{*1:K}^p)^{1/p}$ for some K . Then we will show $\|g_{1:K+1}\|_* \leq |s_{1:K+1}| + (1+C\|g\|_{*1:K+1}^p)^{1/p}$. We consider two cases, either $\|g_{1:K}\|_* + \frac{\langle \nabla \|g_{1:K}\|_*^p, g_{K+1} \rangle}{p \|g_{1:K}\|_*^{p-1}} \geq 0$, or not.

Case 1: $\|g_{1:K}\|_{\star} + \frac{\langle \nabla \|g_{1:K}\|_{\star}^p, g_{K+1} \rangle}{p \|g_{1:K}\|_{\star}^{p-1}} \geq 0$:

By smoothness, we have

$$\begin{aligned} \|g_{1:K} + g_{K+1}\|_{\star} &= (\|g_{1:K} + g_{K+1}\|_{\star}^p)^{1/p} \\ &\leq (\|g_{1:K}\|_{\star}^p + \langle \nabla \|g_{1:K}\|_{\star}^p, g_{K+1} \rangle + C \|g_{K+1}\|_{\star}^p)^{1/p} \\ &\leq \left[\left(\|g_{1:K}\|_{\star} + \frac{\langle \nabla \|g_{1:K}\|_{\star}^p, g_{K+1} \rangle}{p \|g_{1:K}\|_{\star}^{p-1}} \right)^p + C \|g_{K+1}\|_{\star}^p \right]^{1/p} \end{aligned}$$

Where the third line follows because

$$x^p + pyx^{p-1} \leq (x + y)^p$$

whenever $x \geq 0$ and $x + y \geq 0$ by convexity of the function $x \mapsto x^p$ for positive x .

Now use the induction hypothesis:

$$\begin{aligned} \|g_{1:K} + g_{K+1}\|_{\star} &\leq \left(\left(|s_{1:K}| + (2^p + C \|g\|_{\star_{1:K}}^p)^{1/p} + \frac{\langle \nabla \|g_{1:K}\|_{\star}^p, g_{K+1} \rangle}{p \|g_{1:K}\|_{\star}^{p-1}} \right)^p + C \|g_{K+1}\|_{\star}^p \right)^{1/p} \\ &\leq \left(\left(|s_{1:K}| + \frac{\langle \nabla \|g_{1:K}\|_{\star}^p, g_{K+1} \rangle}{p \|g_{1:K}\|_{\star}^{p-1}} \right)^p + (2^p + C \|g\|_{\star_{1:K}}^p)^{1/p} \right)^p + C \|g_{K+1}\|_{\star}^p \right)^{1/p} \end{aligned}$$

Next we need a technical observation: for any positive A, B and x , we have

$$(A + B)^p - B^p \leq (A + B + x)^p - (B + x)^p$$

which can be verified by differentiating with respect to x .

Therefore we have, with $A = \left| |s_{1:K}| + \frac{\langle \nabla \|g_{1:K}\|_{\star}^p, g_{K+1} \rangle}{p \|g_{1:K}\|_{\star}^{p-1}} \right|$, $B = (2^p + C \|g\|_{\star_{1:K}}^p)^{1/p}$ and $B + x = (2^p + C \|g\|_{\star_{1:K+1}}^p)^{1/p}$:

$$\begin{aligned} \|g_{1:K} + g_{K+1}\|_{\star} &\leq ((A + B)^p - B^p + B^p + C \|g_{K+1}\|_{\star}^p)^{1/p} \\ &\leq ((A + B + x)^p - (B + x)^p + B^p + C \|g\|_{\star}^p)^{1/p} \\ &= ((A + B + x)^p - 2^p - C \|g\|_{\star_{1:K+1}}^p + 2^p + C \|g\|_{\star_{1:K}}^p + C \|g_{K+1}\|_{\star}^p)^{1/p} \\ &= A + B + x \\ &= \left| |s_{1:K}| + \frac{\langle \nabla \|g_{1:K}\|_{\star}^p, g \rangle}{p \|g_{1:K}\|_{\star}^{p-1}} \right| + (2^p + C \|g\|_{\star_{1:K+1}}^p)^{1/p} \end{aligned}$$

Next observe:

$$\begin{aligned} |s_{1:K+1}| &= \left| s_{1:K} + \text{sign}(s_{1:K}) \frac{\langle \nabla \|g_{1:K}\|_*^p, g_{K+1} \rangle}{p \|g_{1:K}\|_*^{p-1}} \right| \\ &= \left| s_{1:K} + \frac{\langle \nabla \|g_{1:K}\|_*^p, g_{K+1} \rangle}{p \|g_{1:K}\|_*^{p-1}} \right| \end{aligned}$$

Combining these two calculations:

$$\begin{aligned} \|g_{1:K} + g_{K+1}\|_* &\leq \left| s_{1:K} + \frac{\langle \nabla \|g_{1:K}\|_*^p, g_{K+1} \rangle}{p \|g_{1:K}\|_*^{p-1}} \right| + (2^p + C \|g_{\star 1:K+1}\|^p)^{1/p} \\ &= |s_{1:K+1}| + (2^p + C \|g_{\star 1:K+1}\|^p)^{1/p} \end{aligned}$$

Case 2 $\|g_{1:K}\|_* + \frac{\langle \nabla \|g_{1:K}\|_*^p, g_{K+1} \rangle}{p \|g_{1:K}\|_*^{p-1}} < 0$:

In this case, observe that $\nabla \|g_{1:K}\|_*^p = p \|g_{1:K}\|_*^{p-1} \nabla \|g_{1:K}\|_*$ where $\nabla \|g_{1:K}\|_*$ denotes the gradient (Frechet derivative) of $\psi(x) = \|x\|_*$ at $x = g_{1:K}$. Therefore,

$$\begin{aligned} 0 > \|g_{1:K}\|_* + \frac{\langle \nabla \|g_{1:K}\|_*^p, g_{K+1} \rangle}{p \|g_{1:K}\|_*^{p-1}} \\ &= \|g_{1:K}\|_* + \langle \nabla \|g_{1:K}\|_*, g_{K+1} \rangle \end{aligned}$$

Next, for any differentiable G -Lipschitz function ψ ,

$$\begin{aligned} \psi(x+y) &\leq \psi(x) + G \|y\|_* \\ &= \psi(x) + \langle \nabla \psi(x), y \rangle + G \|y\|_* - \langle \nabla \psi(x), y \rangle \\ &\leq \psi(x) + \langle \nabla \psi(x), y \rangle + 2G \|y\|_* \end{aligned}$$

Now since all norms are 1-Lipschitz with respect to themselves, we have

$$\begin{aligned} \|g_{1:K} + g_{K+1}\|_* &\leq \|g_{1:K}\|_* + \langle \nabla \|g_{1:K}\|_*, g_{K+1} \rangle + 2 \|g_{K+1}\|_* \\ &\leq 2 \|g_{K+1}\|_* \\ &\leq 2 \\ &\leq |s_{1:K+1}| + (2^p + C \|g_{\star 1:K+1}\|^p)^{1/p} \end{aligned}$$

This concludes the proof of the first statement.

To see that $|s_t| \leq \|g_t\|_*$, recall that $\nabla \|g_{1:K}\|_*^p = p \|g_{1:K}\|_*^{p-1} \nabla \|g_{1:K}\|_*$. Now since $\|\cdot\|_*$ is 1-Lipschitz with respect to itself, this implies as a corollary that $|s_t| \leq |\langle \nabla \|g_{1:K}\|_*, g_t \rangle| \leq \|g_t\|_*$.

Finally, it remains to show the regret bound. To do this we will make use of the first part of the theorem:

$$\begin{aligned}
R_T(\hat{w}) &= \sum_{t=1}^T \langle g_t, w_t \rangle - \langle g_t, \hat{w} \rangle \\
&= \sum_{t=1}^T \left\langle z_t \text{sign}(s_{1:t-1}) \frac{\nabla \|g_{1:t-1}\|_*^p}{p \|g_{1:t-1}\|^{p-1}}, g_t \right\rangle - \langle g_t, \hat{w} \rangle \\
&= \sum_{t=1}^T \left\langle \text{sign}(s_{1:t-1}) \frac{\nabla \|g_{1:t-1}\|_*^p}{p \|g_{1:t-1}\|^{p-1}}, g_t \right\rangle z_t - \langle g_t, \hat{w} \rangle \\
&= \sum_{t=1}^T s_t z_t - \langle g_t, \hat{w} \rangle \\
&= \underbrace{\sum_{t=1}^T s_t z_t + s_{1:T} \text{sign}(s_{1:T}) \|\hat{w}\| - s_{1:T} \text{sign}(s_{1:T}) \|\hat{w}\|}_{\text{Regret of } \mathcal{A} \text{ at } \text{sign}(s_{1:T}) \|\hat{w}\|} - \sum_{t=1}^T \langle g_t, \hat{w} \rangle \\
&\leq \sup_{c \in \{\pm 1\}} R_T^1(c \|\hat{w}\|) - s_{1:T} \text{sign}(s_{1:T}) \|\hat{w}\| - \sum_{t=1}^T \langle g_t, \hat{w} \rangle \\
&\leq \sup_{c \in \{\pm 1\}} R_T^1(c \|\hat{w}\|) - |s_{1:T}| \|\hat{w}\| + \|g_{1:T}\|_* \|\hat{w}\| \\
&\leq \sup_{c \in \{\pm 1\}} R_T^1(c \|\hat{w}\|) + \|\hat{w}\| (\|g_{1:T}\|_* - |s_{1:T}|) \\
&\leq \sup_{c \in \{\pm 1\}} R_T^1(c \|\hat{w}\|) + \|\hat{w}\| (2^p + C \|g_{1:T}\|_*^p)^{1/p}
\end{aligned}$$

where the last inequality uses the second part of the Theorem. \square

5.3 Reduction to Constrained Domains

The previous algorithms have dealt with optimization over an entire vector space. Although this is a common and important case in practice, sometimes we must perform optimization with constraints in which each w_t and the comparison point \hat{w} must lie in some convex domain W that is not an entire vector space. This constrained problem is often solved with the classical Mirror Descent [60] or Follow-the-Regularized-Leader [52] analysis. However, these approaches have drawbacks: for unbounded sets, they typically maintain regret bounds that have suboptimal dependence on \hat{w} , or, for bounded sets, they depend explicitly on the diameter of W . We will address these issues with a simple reduction. Given any convex domain $V \supset W$ and an algorithm \mathcal{A} that maintains regret $R_T^{\mathcal{A}}(\hat{w})$ for any $\hat{w} \in V$, we obtain an algorithm that maintains $2R_T^{\mathcal{A}}(\hat{w})$ for any \hat{w} in W .

Before giving the reduction, we define the distance to a convex set W as $S_W(x) = \inf_{d \in W} \|x - d\|$ as

well as the projection to W as $\Pi_W(x) = \{d \in W : \|d - x\| \leq \|c - x\|, \forall c \in W\}$. Note that if B is reflexive,¹ $\Pi_W(x) \neq \emptyset$ and that it is a singleton if B is a Hilbert space [30, Exercise 4.1.4].

The intuition for our reduction is as follows: given a vector $z_t \in V$ from \mathcal{A} , we predict with any $w_t \in \Pi_W(z_t)$. Then give \mathcal{A} a subgradient at z_t of the surrogate loss function $\langle g_t, \cdot \rangle + \|g_t\|_* S_W$, which is just the original linearized loss plus a multiple of S_W . The additional term S_W serves as a kind of Lipschitz barrier that penalizes \mathcal{A} for predicting with any $z_t \notin W$. Pseudocode for the reduction is given in Algorithm 5.

Algorithm 5 Constraint Set Reduction

Require: Convex closed domain W in a reflexive Banach space B , Online learning algorithm \mathcal{A} with domain $V \supset W$

- 1: **for** $t = 1$ **to** T **do**
 - 2: Get point $z_t \in V$ from \mathcal{A}
 - 3: Play $w_t \in \Pi_W(z_t)$
 - 4: Receive $g_t \in \partial \ell_t(w_t)$
 - 5: Set $\tilde{\ell}_t(x) = \frac{1}{2} (\langle g_t, x \rangle + \|g_t\|_* S_W(x))$
 - 6: Compute $\tilde{g}_t \in \partial \tilde{\ell}_t(z_t)$
 - 7: Send \tilde{g}_t as t th subgradient to \mathcal{A}
 - 8: **end for**
-

Theorem 5.4. *Assume that the algorithm \mathcal{A} obtains regret $R_T^{\mathcal{A}}(\hat{w})$ for any $\hat{w} \in V$. Then Algorithm 5 guarantees regret:*

$$R_T(\hat{w}) = \sum_{t=1}^T \langle g_t, w_t - \hat{w} \rangle \leq 2R_T^{\mathcal{A}}(\hat{w}), \quad \forall \hat{w} \in W.$$

Further, the subgradients \tilde{g}_t sent to \mathcal{A} satisfy $\|\tilde{g}_t\|_* \leq \|g_t\|_*$.

Before proving this Theorem, we need a small technical Proposition, proved in Appendix 5.D.

Proposition 5.5. *S_W is convex and 1-Lipschitz for any closed convex set W in a reflexive Banach space B . of Theorem 5.4. From Proposition 5.5, we observe that since S_W is convex and $\|g_t\|_* \geq 0$, $\tilde{\ell}_t$ is convex for all t . Therefore, by \mathcal{A} 's regret guarantee, we have*

$$\sum_{t=1}^T \tilde{\ell}_t(z_t) - \tilde{\ell}_t(\hat{w}) \leq R_T^{\mathcal{A}}(\hat{w}).$$

Next, since $\hat{w} \in W$, $\langle g_t, \hat{w} \rangle = 2\tilde{\ell}_t(\hat{w})$ for all t . Further, since $w_t \in \Pi_W(z_t)$, we have $\langle g_t, z_t \rangle + \|g_t\|_* \|w_t - z_t\| = 2\tilde{\ell}_t(z_t)$. Finally, by the definition of dual norm we have

$$\langle g_t, w_t - \hat{w} \rangle \leq \langle g_t, z_t - \hat{w} \rangle + \|g_t\|_* \|w_t - z_t\| = 2\tilde{\ell}_t(z_t) - 2\tilde{\ell}_t(\hat{w}).$$

Combining these two lines proves the regret bound of the theorem. The bound on $\|\tilde{g}_t\|_*$ follows because S_W is 1-Lipschitz, from Proposition 5.5. □

¹All Hilbert spaces and finite-dimensional Banach spaces are reflexive - see Section 2.1.

Algorithm 6 Banach-space betting through ONS

Require: Real Banach space B , initial linear operator $L : B \rightarrow B^*$, initial wealth $\epsilon > 0$

- 1: **Initialize:** Wealth₀ = ϵ , initial betting fraction $v_1 = 0 \in S = \{x \in B : \|x\| \leq \frac{1}{2}\}$
- 2: **for** $t = 1$ **to** T **do**
- 3: Bet $w_t = v_t$ Wealth _{$t-1$}
- 4: Receive g_t , with $\|g_t\|_* \leq 1$
- 5: Update Wealth _{t} = Wealth _{$t-1$} - $\langle g_t, w_t \rangle$
- 6: //compute new betting fraction $v_{t+1} \in S$ via ONS update on losses $-\ln(1 - \langle g_t, v \rangle)$:
- 7: Set $z_t = \frac{d}{dv_t} (-\ln(1 - \langle g_t, v_t \rangle)) = \frac{g_t}{1 - \langle g_t, v_t \rangle}$
- 8: Set $A_t(x) = L(x) + \sum_{i=1}^t z_i \langle z_i, x \rangle$
- 9: $v_{t+1} = \Pi_S^{A_t}(v_t - \frac{2}{2 - \ln(3)} A_t^{-1}(z_t))$, where $\Pi_S^{A_t}(x) = \operatorname{argmin}_{y \in S} \langle A_t(y - x), y - x \rangle$
- 10: **end for**

We conclude this section by observing that in many cases it is very easy to compute an element of Π_W and a subgradient of S_W . For example, when W is a unit ball, it is easy to see that $\Pi_W(x) = \frac{x}{\|x\|}$ and $\partial S_W(x) = \partial \|x\|$ for any x not in the ball. In general, we provide the following result that often simplifies computing the subgradient of S_W (proved in Appendix 5.D):

Theorem 5.6. *Let B be a reflexive Banach space such that for every $0 \neq b \in B$, there is a unique dual vector b^* such that $\|b^*\|_* = 1$ and $\langle b^*, b \rangle = \|b\|$. Let $W \subset B$ a closed convex set. Given $x \in B$ and $x \notin W$, let $p \in \Pi_W(x)$. Then $\{(x - p)^*\} = \partial S_W(x)$.*

5.4 Banach-space betting through ONS

In this section, we present the Banach space version of the one-dimensional Algorithm 2. The pseudocode is in Algorithm 6. We state the algorithm in its most general Banach space formulation, which obscures some of its simplicity in more common scenarios. For example, when B is \mathbb{R}^d equipped with the p -norm, then the linear operator L can be taken to be simply the identity map $I : \mathbb{R}^d \rightarrow \mathbb{R}^d \cong (\mathbb{R}^d)^*$, and the ONS portion of the algorithm is the standard d -dimensional ONS algorithm.

We give the regret guarantee of Algorithm 6 in Theorem 5.7. The proof, modulo technical details of ONS in Banach spaces, is identical to Theorem 5.1, and can be found in Appendix 5.C.

Theorem 5.7. *Let B be a d -dimensional real Banach space and $u \in B$ be an arbitrary unit vector. Then, there exists a linear operator L such that using the Algorithm 6, we have for any $\hat{w} \in B$,*

$$R_T(\hat{w}) \leq \epsilon + \|\hat{w}\| \max \left[\frac{d}{2} - 8 + 8 \ln \left[\frac{8 \|\hat{w}\| \left(1 + 4 \sum_{t=1}^T \|g_t\|_*^2 \right)^{4.5d}}{\epsilon} \right], \right. \\ \left. 2 \sqrt{\sum_{t=1}^T \langle g_t, \hat{w} \rangle^2 \ln \left(\frac{5 \|\hat{w}\|^2}{\epsilon^2} \left(8 \sum_{t=1}^T \|g_t\|^2 + 2 \right)^{9d+1} + 1 \right)} \right].$$

The main particularity of this bound is the presence of the terms $\sqrt{d \sum_{t=1}^T \langle g_t, \hat{w} \rangle^2}$ rather than the usual $\|\hat{w}\| \sqrt{\sum_{t=1}^T \|g_t\|_\star^2}$. We can interpret this bound as being adaptive to any sequence of norms $\|\cdot\|_1, \dots, \|\cdot\|_t$ because $\sqrt{d \sum_{t=1}^T \langle g_t, \hat{w} \rangle^2} \leq \sqrt{d \sum_{t=1}^T \|\hat{w}\|_t^2 (\|g_t\|_t)_\star^2}$. A similar kind of “many norm adaptivity” was recently achieved in [20], which competes with the best *fixed* L_p norm (or the best fixed norm in any finite set) using a multi-scale experts algorithm. Our bound in Theorem 5.7 is a factor of \sqrt{d} worse,² but we can compete with any possible *sequence* of norms rather than with any fixed one.

Similar regret bounds to our Theorem 5.7 have already appeared in the literature. The first one we are aware of is the Second Order Perceptron [10] whose *mistake bound* is exactly of the same form. Recently, a similar bound was also proven in [27], but this result holds only for domains of the form $W = \{\hat{v} : \langle g_t, \hat{v} \rangle \leq C\}$, for a known C . Also, Kotłowski [28] proved the same bound under the assumptions that the losses are of the form $\ell_t(w_t) = \ell(y_t, w_t \cdot x_t)$ and the algorithm receives x_t before its prediction. In contrast, we can deal with unbounded W and arbitrary convex losses through the use of subgradients. Interestingly, all these algorithms have a $O(d^2)$ complexity per update.

5.5 From T to $\sum \|g_t\|_\star$

The algorithms in the previous sections obtain regret bounds improve the generic $G_{\max} \sqrt{T}$ bound to $\sqrt{\sum_{t=1}^T \|g_t\|_\star^2}$ or $\sqrt{G_{\max} \sum_{t=1}^T \|g_t\|_\star}$, so that they adapt to specific statistics of the loss sequence. However, there exist algorithms with less adaptive guarantees that nevertheless improve upon the logarithmic dependence in T . For example, an algorithm in [35] obtains

$$R_T(\hat{w}) \leq O \left(\|\hat{w}\| \sqrt{T \log \left(\frac{\|\hat{w}\| \sqrt{T} \log^2(T+1)}{\epsilon} + 1 \right)} + \epsilon \right) \quad (5.4)$$

In this section we show how to automatically add some more adaptivity to these algorithms. Our argument is very simple, but seems actually powerful enough to replicate (or even improve upon) some prior literature with significantly less effort (e.g. [43]). We will continue to assume $G_{\max} = 1$ for simplicity, and consider algorithms that obtain a regret guarantee

$$R_T(\hat{w}) = \sum_{t=1}^T g_t \cdot (x_t - \hat{w}) \leq \psi_T(\hat{w})$$

where $\psi_T(\hat{w})$ depends only on \hat{w} and T and in particular not on the individual members of the sequence g_t . We will replace this with the more adaptive regret bound

$$R_T(\hat{w}) \leq \psi_G(\hat{w})$$

²The dependence on d is unfortunately unimprovable, as shown by [31].

where $G \leq 1 + \sum_{t=1}^T \|g_t\|$.

Algorithm 7 non-adaptive to adaptive reduction

Input: Online learning algorithm \mathcal{A} .

Get point z_1 from \mathcal{A} .

$k \leftarrow 1$.

$G_k \leftarrow 0$.

for $t = 1$ **to** T **do**

 Plat $x_t = z_k$.

 Receive subgradient g_t .

$G_k \leftarrow G_k + g_t$.

if $\|G_k\|_* \geq 1$ **then**

 Send $G_k/2$ to \mathcal{A} as k th gradient.

 Get point z_{k+1} from \mathcal{A} .

$G_{k+1} \leftarrow 0$

$k \leftarrow k + 1$.

$t_k \leftarrow t$

end if

end for

Theorem 5.8. *Suppose \mathcal{A} is an online linear optimization algorithm that satisfies the regret bound $R_T(\dot{w}) \leq \psi_T(\dot{w})$ such that $\psi_T(\dot{w})$ is an increasing function of T for all \dot{w} . Then there is an online linear optimization algorithm that satisfies $R_T(\dot{w}) \leq 2\psi_{1+\|g\|_{1:T}}(\dot{w})$.*

The bound is actually likely quite a bit better than $\|g\|_{1:T}$, as can be observed from the proof, but in worst-case it is $\|g\|_{1:T}$ (in particular it is not $\|g\|_{1:T}^2$). If we apply it to the bound in (5.4), we obtain an algorithm that guarantees

$$R_T(\dot{w}) \leq O \left(\|\dot{w}\| \sqrt{\sum_{t=1}^T \|g_t\|_*} \log \left(\frac{\|\dot{w}\| \sqrt{\sum_{t=1}^T \|g_t\|_*} \log^2(\sum_{t=1}^T \|g_t\|_* + 1)}{\epsilon} + 1 \right) + \epsilon \right)$$

Proof. Let K be the maximum value of k , and let $0 = t_1, \dots, t_K = T$ be such that $G_k = \sum_{t=t_k+1}^{t_{k+1}} g_t$. Notice that since $\|G_k - g_{t_{k+1}}\|_* \leq 1$ by definition of G_k , we must have $\|G_k\|_* \leq 2$ so that $\|G_k/2\|_* \leq 1$.

Therefore by \mathcal{A} 's non-adaptive regret guarantee we have

$$\begin{aligned}
\psi_K(\hat{w}) &\geq \sum_{k=1}^K \frac{G_k}{2} \cdot (z_k - \hat{w}) \\
&= \frac{1}{2} \sum_{k=1}^K \sum_{t=t_k}^{t_{k+1}} g_t(z_k - \hat{w}) \\
&= \frac{1}{2} \sum_{k=1}^K \sum_{t=t_k}^{t_{k+1}} g_t(x_t - \hat{w}) \\
&= \frac{1}{2} R_T(\hat{w})
\end{aligned}$$

so now all we need do is show that $K \leq 1 + \sum_{t=1}^T \|g_t\|_\star$:

$$\begin{aligned}
\sum_{t=1}^T \|g_t\|_\star &= \sum_{k=1}^K \sum_{t=t_k}^{t_{k+1}} \|g_t\|_\star \\
&\geq \sum_{k=1}^K \left\| \sum_{t=t_k}^{t_{k+1}} g_t \right\|_\star \\
&\geq \|G_K\| + \sum_{k=1}^{K-1} \|G_k\| \\
&\geq \|G_K\| + \sum_{k=1}^{K-1} 1 \\
&= K - 1
\end{aligned}$$

Therefore, since $\psi_K(\hat{w})$ is an increasing function of K , $R_T(\hat{w}) \leq 2\psi_K(\hat{w}) \leq 2\psi_{1+\|g\|_{1:T}}(\hat{w})$. \square

5.5.1 Lower bound for $\|g_t\|_\star^2$

In this section we will show that we cannot replace the $\sum_{t=1}^T \|g_t\|_\star$ term in Theorem 5.8 with the more desirable $\sum_{t=1}^T \|g_t\|_\star^2$ - at least not without some nontrivial alteration to the result. We accomplish this by showing that the extra $\log(\sum_{t=1}^T \|g_t\|_\star^2)$ term of our regret bound in Theorem 5.1 is actually necessary. In particular, it is impossible to obtain a regret bound like

$$R_T(\hat{w}) \leq \epsilon + A\|\hat{w}\| \sqrt{1 + \sum_{t=1}^T \|g_t\|_\star^2 \log(\|\hat{w}\|T/\epsilon)}$$

Since it is possible to guarantee regret $R_T(\hat{w}) \leq \epsilon + A\|\hat{w}\| \sqrt{T \log(\|\hat{w}\|T/\epsilon)}$, this shows that we cannot replace all instances of T with $\sum_{t=1}^T \|g_t\|_\star^2$ in a black-box manner.

Theorem 5.9. *Suppose \mathcal{A} is one-dimensional online linear optimization algorithm that guarantees origin-regret $\epsilon(t)$ on 1-Lipschitz losses at time t for where ϵ is non-decreasing and $\epsilon(t)$ is $O(t^d)$ for some d . Then for any $k > 0$ and any $1/2 < \gamma \leq 1$ and any $\tau > 0$, there exists a T_1 such that for all $T > T_1$ there is an adversarial strategy picking $g_t \in \mathbb{R}$ in response to the learner's output $w_t \in \mathbb{R}$ such that there exists $\hat{w} \in \mathbb{R}$ with*

$$R_T(\hat{w}) = \sum_{t=1}^T g_t w_t - g_t u > k|u| \log^\gamma(T|\hat{w}| + 1) \sqrt{1 + |g_{1:T}^{2(\gamma+\tau)}|} + \epsilon(T)$$

This Theorem rules out bounds with $\tilde{O}\left(\sqrt{\sum_{t=1}^T \|g_t\|_*^2}\right)$ regret unless there is a term that is at least $O(\log(T))$ rather than the $O(\sqrt{\log(T)})$ bound we can get using Theorem 5.8. Observe that there are relatively mild conditions on ϵ so that one cannot dodge these lower bounds by moving some small T -dependence into ϵ .

Proof. Without loss of generality, assume $w_1 = 0$. Define the ‘‘reward’’ at time t by $r_t = -\sum_{t'=1}^t g_{t'} w_{t'}$. Observe that the regret can be written as

$$R_T(\hat{w}) = -r_T - g_{1:T} \hat{w}$$

Therefore the condition that \mathcal{A} ensures $R_t(0) \leq \epsilon(t)$ for all t requires that $r_t \geq -\epsilon(t)$ for all t .

Suppose for contradiction that there is some $t \leq T$ and a sequence g_1, \dots, g_{t-1} such that $w_t > r_{t-1} + \epsilon(T)$. Consider the adversary that gives g_1, \dots, g_{t-1} for the first $t-1$ rounds, then gives $g_t = 1$ and $g_{t+1} = g_{t+2} = \dots = g_T = 0$. Then we have $r_T = r_t = r_{t-1} - w_t < -\epsilon(T) < -\epsilon(t)$, which contradicts our assumption about \mathcal{A} . Therefore $w_t \leq r_{t-1} + \epsilon(T)$ for all t on all loss sequences of length at most T .

With this in hand, we now make an assumption about T_1 that will come in handy later. Choose T_1 large enough that for all $T > T_1$,

$$\log(2\epsilon(T)\sqrt{T}) + (T-1) \log\left(1 + \frac{1}{\sqrt{T}}\right) + 1 < \left(\frac{\sqrt{T}}{k\sqrt{1+T^{1-\gamma-\tau}}}\right)^{1/\gamma} \quad (5.5)$$

$$2\epsilon(T) \left(1 + \frac{1}{\sqrt{T}}\right)^{T-1} \sqrt{T} \geq 1 \quad (5.6)$$

This is possible because $\epsilon(T)$ is nondecreasing and grows at most polynomially in T , and $\left[(T-1) \log\left(1 + \frac{1}{\sqrt{T}}\right)\right]^\gamma \leq T^{\gamma/2} = \frac{\sqrt{T}}{\sqrt{T^{1-\gamma}}} \leq \frac{1}{2} \left(\frac{\sqrt{T}}{2k\sqrt{1+T^{1-\gamma-\tau}}}\right)$ for sufficiently large T for any $\tau > 0$.

Consider the loss sequence $g_t = -\frac{1}{\sqrt{T}}$ for all $t \leq T$. Then we have $r_t = r_{t-1} + \frac{x_t}{\sqrt{T}}$. Since $w_t \leq$

$r_{t-1} + \epsilon(T)$, we have

$$\begin{aligned} r_t &\leq r_{t-1} \left(1 + \frac{1}{\sqrt{T}}\right) + \frac{\epsilon(T)}{\sqrt{T}} \\ (r_t + \epsilon(T)) &\leq (r_{t-1} + \epsilon(T)) \left(1 + \frac{1}{\sqrt{T}}\right) \\ r_T &\leq \epsilon(T) \left(1 + \frac{1}{\sqrt{T}}\right)^{T-1} - \epsilon(T) \end{aligned}$$

where in the last line we have unrolled the recursion and observed that $r_1 = 0$.

Now consider $\dot{w} = 2\epsilon(T) \frac{\left(1 + \frac{1}{\sqrt{T}}\right)^{T-1}}{\sqrt{T}}$. Then

$$\begin{aligned} R_T(\dot{w}) &= -r_T - g_{1:T}\dot{w} \\ &\geq \epsilon(T) - \epsilon(T) \left(1 + \frac{1}{\sqrt{T}}\right)^{T-1} + \sqrt{T}\dot{w} \\ &= \epsilon(T) + \epsilon(T) \left(1 + \frac{1}{\sqrt{T}}\right)^{T-1} \\ &= \epsilon(T) + \frac{\sqrt{T}}{2}\dot{w} \end{aligned}$$

From our choice of \dot{w} and our assumptions on T_1 , we compute:

$$\begin{aligned} \log(Tu + 1) &\leq \log(Tu) + 1 \quad \text{because } T\dot{w} \geq 1 \text{ by (5.6)} \\ &= \log(2\epsilon\sqrt{T}) + (T-1) \log\left(1 + \frac{1}{\sqrt{T}}\right) + 1 \\ &< \left(\frac{\sqrt{T}}{2k\sqrt{1 + T^{1-\gamma-\tau}}}\right)^{1/\gamma} \quad \text{by (5.5)} \\ \frac{\sqrt{T}}{2} &> k\sqrt{1 + T^{1-\gamma-\tau}} \log^\gamma(T\dot{w} + 1) \end{aligned}$$

Combining this calculation with our regret bound we have:

$$\begin{aligned} R_T(\dot{w}) &\geq \epsilon(T) + \frac{\sqrt{T}}{2}\dot{w} \\ &> \epsilon(T) + k\sqrt{1 + T^{1-\gamma-\tau}}\dot{w} \log^\gamma(T\dot{w} + 1) \end{aligned}$$

Finally, observe that $1 + |g|_{1:T}^{2(\gamma+\tau)} = 1 + T^{1-\gamma-\tau}$ so that we can conclude

$$\begin{aligned} R_T(u) &> \epsilon(T) + k\sqrt{1 + T^{1-\gamma-\tau}}\dot{w} \log^\gamma(T\dot{w} + 1) \\ &\geq \epsilon(T) + k\dot{w} \log^\gamma(Tu + 1) \sqrt{1 + |g|_{1:T}^{2(\gamma+\tau)}} \end{aligned}$$

as desired. □

5.6 Conclusions

In this chapter we introduced a series of reductions showing that parameter-free online learning algorithms can be obtained from online exp-concave optimization algorithms, that optimization in a vector space with any norm can be obtained from 1D optimization, and that online learning with constraints is no harder than optimization without constraints. Our reductions not only result in simpler arguments in many applications, they also provide often better algorithms in terms of regret bounds or runtime.

We also remark that the dependence of our regret bounds on the term $\sqrt{1 + \sum_{t=1}^T \|g_t\|_*^2}$ when $G_{\max} \leq 1$ suggests that the assumption of a known bound on G_{\max} is perhaps a little less bad it might appear. If we re-scale the losses by the bound G_{\max} , then the regret bound depends on the term $\sqrt{G_{\max}^2 + \sum_{t=1}^T \|g_t\|_*^2} \leq G_{\max} + \sqrt{\|g_t\|_*^2}$. Thus if our chosen bound for G_{\max} is too conservative, it will actually have a subasymptotic effect on the regret bound. On the other hand, a conservative bound on $\|\hat{w}\|$ would have the effect of linearly scaling the regret bound.

In the next chapters we will apply these reductions to design new algorithms and regret bounds.

Appendix

This appendix is organized as follows:

1. In Section 5.A we collect some background information about Banach spaces, their duals, and other properties.
2. In Section 5.B we provide an analysis of the ONS algorithm in Banach spaces that is useful for proving Theorem 5.7.
3. In Section 5.C we apply this analysis of ONS in Banach spaces to prove Theorem 5.7, and provide the missing Fenchel conjugate calculation required to prove Theorem 5.1, which are our reductions from parameter-free online learning to Exp-concave optimization.
4. In Section 5.D we prove Proposition 5.5, used in our reduction from constrained optimization to unconstrained optimization in Section 5.3. In this section we also prove Theorem 5.6, which simplifies computing subgradients of S_W in many cases.

5.A Banach Spaces

Given any vector space V , there is a natural injection $V \rightarrow V^{**}$ given by $x \mapsto \langle \cdot, x \rangle$. When this injection is an isomorphism of Banach spaces, then the space V is called *reflexive*. All finite-dimensional Banach spaces are reflexive.

Given any linear map of Banach spaces $T : X \rightarrow Y$, we define the *adjoint* map $T^* : Y^* \rightarrow X^*$ by $T^*(y^*)(x) = \langle y^*, T(x) \rangle$. T^* has the property (by definition) that $\langle y^*, T(x) \rangle = \langle T^*(y^*), x \rangle$. As a special case, if B is a reflexive Banach space and $T : B \rightarrow B^*$, then we can use the natural identification between B^{**} and B to view T^* as $T^* : B \rightarrow B^*$. Thus, in this case it is possible to have $T = T^*$, in which case we call T self-adjoint.

Definition 5.10. We define a Banach space B as (p, D) uniformly convex if [47]:

$$\|x + y\|^p + \|x - y\|^p \geq 2\|x\|^p + 2D\|y\|^p, \quad \forall x, y \in B. \quad (5.7)$$

From this definition, we can see that if B is $(2, D)$ uniformly convex, then $\|\cdot\|^2$ is a D -strongly convex function with respect to $\|\cdot\|$:

Lemma 5.11. *Let $f(x)$ a convex function that satisfies*

$$f\left(\frac{x+y}{2}\right) \leq \frac{1}{2}f(x) + \frac{1}{2}f(y) - \frac{D}{2p}\|x-y\|^p.$$

Then, f satisfies $f(x+\delta) \geq f(x) + \langle g, \delta \rangle + D\frac{\|\delta\|^p}{p}$ for any subgradient $g \in \partial f(x)$. In particular for $p = 2$, f is D strongly convex with respect to $\|\cdot\|$.

Proof. Set $y = x + 2\delta$ for some arbitrary δ . Let $g \in \mathbb{X}^*$ be an arbitrary subgradient of f at x . Let $R_x(\tau) = f(x+\tau) - (f(x) + g(\tau))$. Then

$$f(x) + g(\delta) \leq f\left(\frac{x+y}{2}\right) \leq \frac{f(x) + f(x+2\delta)}{2} - \frac{D\|2\delta\|^p}{2p} = f(x) + g(\delta) + \frac{R_x(2\delta)}{2} - \frac{D\|2\delta\|^p}{2p},$$

that implies $\frac{D}{p}\|2\delta\|^p \leq R_x(2\delta)$. So that $f(x+\tau) = f(x) + g(\tau) + R_x(\tau) \geq f(x) + g(\tau) + \frac{D}{p}\|\tau\|^p$ as desired. \square

Lemma 5.12. *Let B be a $(2, D)$ uniformly convex Banach space, then $f(x) = \frac{1}{2}\|x\|^2$ is D -strongly convex.*

Proof. Let $x = u + v$ and $y = u - v$. Then, from the definition of $(2, D)$ uniformly convex Banach space, we have

$$2\|u+v\|^2 + 2D\|u-v\|^2 \leq 4\|u\|^2 + 4\|v\|^2,$$

that is

$$\frac{1}{2}\left\|\frac{u+v}{2}\right\|^2 \leq \frac{1}{2}\|u\|^2 + \frac{1}{2}\|v\|^2 - \frac{D}{4}\|u-v\|^2.$$

Using Lemma 5.11, we have the stated bound. \square

Any Hilbert space is $(2, 1)$ -strongly convex. As a slightly more exotic example, \mathbb{R}^d equipped with the p -norm is $(2, p-1)$ strongly-convex for $p \in (1, 2]$.

The notion of uniform convexity is dual to the notion of uniform smoothness:

Definition 5.13. *A Banach space V is (p, D) uniformly smooth if*

$$\|x+y\|^p + \|x-y\|^p \leq 2\|x\|^p + 2D\|y\|^p, \quad \forall x, y \in B.$$

It turns out that if B is (q, D) -uniformly convex, then there exists some C such that B^* is (p, C) -uniformly smooth with $\frac{1}{p} + \frac{1}{q} = 1$ (see [48] chapter 6 and [49] chapter 4).

Similar to with uniform convexity, with this definition we can observe that if B is a $(2, D/4)$ -uniformly smooth Banach space then $\|\cdot\|^2$ is D -smooth with respect to $\|\cdot\|$.

Lemma 5.14. *Let $f(x)$ a convex function that satisfies*

$$\frac{f(x+y)}{2} + \frac{f(x-y)}{2} \leq f(x) + D\|y\|^p.$$

Then, f satisfies $f(x+\delta) \leq f(x) + \langle g, \delta \rangle + 2D\frac{\|\delta\|^p}{p}$ for any subgradient $g \in \partial f(x)$. In particular for $p = 2$, f is $4D$ smooth with respect to $\|\cdot\|$.

Proof. Set $y = \delta$ for some arbitrary δ . Let $g \in \partial f(x)$ be an arbitrary subgradient of f at x .

$$\begin{aligned} f(x+\delta) + f(x-\delta) &\leq 2f(x) + 2D\|\delta\|^p \\ f(x+\delta) &\leq 2f(x) - f(x-\delta) + 2D\|\delta\|^p \\ &\leq 2f(x) - f(x) - f(x) + \langle g, \delta \rangle + 2D\|\delta\|^p \\ &= f(x) + \langle g, \delta \rangle + 2D\|\delta\|^p \end{aligned}$$

that implies $\frac{D}{p}\|2\delta\|^p \leq R_x(2\delta)$. So that $f(x+\tau) = f(x) + g(\tau) + R_x(\tau) \geq f(x) + g(\tau) + \frac{D}{p}\|\tau\|^p$ as desired. \square

Finally, we discuss the notion of Frechet differentiability, which is a simple generalization of the standard notion familiar from basic calculus.

Definition 5.15. *Given a real Banach space B , a function $f : B \rightarrow \mathbb{R}$ is Frechet differentiable at $x \in B$ if there exists a $g \in B^*$ such that*

$$\lim_{\delta \rightarrow 0} \frac{|f(x+\delta) - f(x) - \langle g, \delta \rangle|}{\|\delta\|} = 0$$

the value g is called the Frechet derivative, denoted $\nabla f(x)$.

This definition maintains all the important familiar properties, including the chain rule and the fact that $\nabla f(x) \in \partial f(x)$ for any convex f .

5.B Proof of the regret bound of ONS in Banach spaces

First, we need some additional facts about self-adjoint operators. These are straight-forward properties in Hilbert spaces, but may be less familiar in Banach spaces so we present them below for completeness.

Proposition 5.16. *Suppose X and Y are Banach spaces and $T : X \rightarrow Y$ is invertible. Then, T^* is invertible and $(T^{-1})^* = (T^*)^{-1}$.*

Proof. Let $y^* \in Y^*$. Let $x \in X$. Recall that by definition $\langle T^*(y^*), x \rangle = \langle y^*, T(x) \rangle$. Then we have

$$\langle (T^{-1})^*(T^*(y^*)), x \rangle = \langle T^*(y^*), T^{-1}(x) \rangle = \langle y^*, x \rangle,$$

Algorithm 8 ONS in Banach Spaces

Require: Real Banach space B , convex subset $S \subset B$, initial linear operator $L : B \rightarrow B^*$, $\tau, \beta > 0$

- 1: **Initialize:** $v_1 = 0 \in S$
- 2: **for** $t = 1$ **to** T **do**
- 3: Play v_t
- 4: Receive $z_t \in B^*$
- 5: Set $A_t(x) = \tau L(x) + \sum_{i=1}^t z_i \langle z_i, x \rangle$
- 6: $v_{t+1} = \Pi_S^{A_t}(v_t - \frac{1}{\beta} A_t^{-1}(z_t))$, where $\Pi_S^{A_t}(x) = \operatorname{argmin}_{y \in S} \langle A_t(y - x), y - x \rangle$
- 7: **end for**

where we used the definition of adjoint twice. Therefore, $(T^{-1})^*(T^*(y^*)) = y^*$ and so $(T^{-1})^* = (T^*)^{-1}$. \square

Proposition 5.17. Suppose B is a reflexive Banach space and $T : B \rightarrow B^*$ is such that

$$T(x) = \sum_{i=1}^N \langle b^i, x \rangle b^i$$

for some vectors $b^i \in B^*$. Then $T^* = T$.

Proof. Let $g, f \in B$. Since B is reflexive, g corresponds to the function $\langle \cdot, g \rangle \in B^{**}$. Now, we compute:

$$T^*(g)(f) = \langle T(f), g \rangle = \sum_{i=1}^N \langle b^i, f \rangle \langle b^i, g \rangle = \langle T(g), f \rangle = T(g)(f).$$

\square

Proposition 5.18. Suppose $\tau > 0$, B is a d -dimensional real Banach space, b^1, \dots, b^d are a basis for B^* and g_1, \dots, g_T are elements of B^* . Then, $A : B \rightarrow B^*$ defined by $A(x) = \tau \sum_{i=1}^d \langle b^i, x \rangle b^i + \sum_{t=1}^T \langle g_t, x \rangle g_t$ is invertible and self-adjoint, and $\langle Ax, x \rangle > 0$ for all $x \neq 0$.

Proof. First, A is self-adjoint by Proposition 5.17.

Next, we show A is invertible. Suppose otherwise. Then, since B and B^* are both d -dimensional, A must have a non-trivial kernel element x . Therefore,

$$0 = \langle Ax, x \rangle = \tau \sum_{i=1}^d \langle b^i, x \rangle^2 + \sum_{t=1}^T \langle g_t, x \rangle^2, \quad (5.8)$$

so that $\langle b^i, x \rangle = 0$ for all i . Since the b^i form a basis for B^* , this implies $\langle y, x \rangle = 0$ for all $y \in B^*$, which implies $x = 0$. Therefore, A has no kernel and so must be invertible.

Finally, observe that since (5.8) holds for any x , we must have $\langle Ax, x \rangle > 0$ if $x \neq 0$. \square

Now we state the ONS algorithm in Banach spaces and prove its regret guarantee:

Theorem 5.19. *Using the notation of Algorithm 8, suppose $L(x) = \sum_{i=1}^d \langle b^i, x \rangle$ for some basis $b^i \in B^*$ and that B is d -dimensional. Then for any $\hat{v} \in S$,*

$$\sum_{t=1}^T \left(\langle z_t, v_t - \hat{v} \rangle - \frac{\beta}{2} \langle z_t, v_t - \hat{v} \rangle^2 \right) \leq \frac{\beta\tau}{2} \langle L(\hat{v}), \hat{v} \rangle + \frac{2}{\beta} \sum_{t=1}^T \langle z_t, A_t^{-1}(z_t) \rangle.$$

Proof. First, observe by Proposition 5.18 that A_t is invertible and self-adjoint for all t .

Now, define $x_{t+1} = v_t - \frac{1}{\beta} A_t^{-1}(z_t)$ so that $v_{t+1} = \Pi_S^{A_t}(x_{t+1})$. Then, we have

$$x_{t+1} - \hat{v} = v_t - \hat{v} - \frac{1}{\beta} A_t^{-1}(z_t),$$

that implies

$$A_t(x_{t+1} - \hat{v}) = A_t(v_t - \hat{v} - \frac{1}{\beta} A_t^{-1}(z_t)) = A_t(v_t - \hat{v}) - \frac{1}{\beta} z_t,$$

and

$$\begin{aligned} & \langle A_t(x_{t+1} - \hat{v}), x_{t+1} - \hat{v} \rangle \\ &= \langle A_t(v_t - \hat{v}) - \frac{1}{\beta} z_t, x_{t+1} - \hat{v} \rangle \\ &= \langle A_t(v_t - \hat{v}), x_{t+1} - \hat{v} \rangle - \frac{1}{\beta} \langle z_t, x_{t+1} - \hat{v} \rangle \\ &= \langle A_t(v_t - \hat{v}), x_{t+1} - \hat{v} \rangle - \frac{1}{\beta} \langle z_t, v_t - \hat{v} - \frac{1}{\beta} A_t^{-1}(z_t) \rangle \\ &= \langle A_t(v_t - \hat{v}), x_{t+1} - \hat{v} \rangle - \frac{1}{\beta} \langle z_t, v_t - \hat{v} \rangle + \frac{1}{\beta^2} \langle z_t, A_t^{-1}(z_t) \rangle \\ &= \langle A_t(v_t - \hat{v}), v_t - \hat{v} - \frac{1}{\beta} A_t^{-1}(z_t) \rangle - \frac{1}{\beta} \langle z_t, v_t - \hat{v} \rangle + \frac{1}{\beta^2} \langle z_t, A_t^{-1}(z_t) \rangle \\ &= \langle A_t(v_t - \hat{v}), v_t - \hat{v} \rangle - \frac{1}{\beta} \langle A_t(v_t - \hat{v}), A_t^{-1}(z_t) \rangle - \frac{1}{\beta} \langle z_t, v_t - \hat{v} \rangle + \frac{1}{\beta^2} \langle z_t, A_t^{-1}(z_t) \rangle \\ &= \langle A_t(v_t - \hat{v}), v_t - \hat{v} \rangle - \frac{2}{\beta} \langle z_t, v_t - \hat{v} \rangle + \frac{1}{\beta^2} \langle z_t, A_t^{-1}(z_t) \rangle, \end{aligned}$$

where in the last line we used $\langle A_t(v_t - \hat{v}), A_t^{-1}(z_t) \rangle = \langle (v_t - \hat{v}), A_t^* A_t^{-1}(z_t) \rangle$ and $A_t^* = A_t$. We now use the Lemma 8 from [24], extended to Banach spaces thanks to the last statement of Proposition 5.18, to have

$$\langle A_t(x_{t+1} - \hat{v}), x_{t+1} - \hat{v} \rangle \geq \langle A_t(v_{t+1} - \hat{v}), v_{t+1} - \hat{v} \rangle$$

to have

$$\langle z_t, v_t - \hat{v} \rangle \leq \frac{\beta}{2} \langle A_t(v_t - \hat{v}), v_t - \hat{v} \rangle - \frac{\beta}{2} \langle A_t(v_{t+1} - \hat{v}), v_{t+1} - \hat{v} \rangle + \frac{2}{\beta} \langle z_t, A_t^{-1}(z_t) \rangle.$$

Summing over $t = 1, \dots, T$, we have

$$\begin{aligned}
\sum_{t=1}^T \langle z_t, v_t - \hat{v} \rangle &\leq \frac{\beta}{2} \langle A_1(v_1 - \hat{v}), v_1 - \hat{v} \rangle + \frac{\beta}{2} \sum_{t=2}^T \langle A_t(v_t - \hat{v}) - A_{t-1}(v_t - \hat{v}), v_t - \hat{v} \rangle \\
&\quad - \frac{\beta}{2} \langle A_T(v_{T+1} - \hat{v}), v_{T+1} - \hat{v} \rangle + \frac{2}{\beta} \sum_{t=1}^T \langle z_t, A_t^{-1}(z_t) \rangle \\
&\leq \frac{\beta}{2} \langle A_1(v_1 - \hat{v}), v_1 - \hat{v} \rangle + \frac{\beta}{2} \sum_{t=2}^T \langle z_t \langle z_t, v_t - \hat{v} \rangle, v_t - \hat{v} \rangle + \frac{2}{\beta} \sum_{t=1}^T \langle z_t, A_t^{-1}(z_t) \rangle \\
&= \frac{\beta}{2} \langle \tau L(\hat{v}), \hat{v} \rangle + \frac{\beta}{2} \sum_{t=1}^T \langle z_t, v_t - \hat{v} \rangle^2 + \frac{2}{\beta} \sum_{t=1}^T \langle z_t, A_t^{-1}(z_t) \rangle.
\end{aligned}$$

□

It remains to choose L properly and analyze the sum $\sum_{t=1}^T \langle z_t, A_t^{-1}(z_t) \rangle$. In order to do this, we introduce the concept of an Auerbach basis (e.g. see [22] Theorem 1.16):

Theorem 5.20. *Let B be a d -dimensional Banach space. Then there exists a basis of b_1, \dots, b_d of B and a basis b^1, \dots, b^d of B^* such that $\|b_i\| = \|b^i\|_* = 1$ for all i and $\langle b_i, b^j \rangle = \delta_{ij}$. Any bases (b_i) and (b^i) satisfying these conditions is called an Auerbach basis.*

We will use an Auerbach basis to define L , and also to provide a coordinate system that makes it easier to analyze the sum $\sum_{t=1}^T \langle z_t, A_t^{-1}(z_t) \rangle$.

Theorem 5.21. *Suppose B is d -dimensional. Let (b_i) and (b^i) be an Auerbach basis for B . Set $L(x) = \sum_{i=1}^d \langle b^i, x \rangle b^i$. Define A_t as in Algorithm 6. Then, for any $\hat{v} \in S$, the following holds*

$$\frac{\beta\tau}{2} \langle L(\hat{v}), \hat{v} \rangle + \frac{2}{\beta} \sum_{t=1}^T \langle z_t, A_t^{-1}(z_t) \rangle \leq \frac{\beta\tau}{2} d \|\hat{v}\|^2 + \frac{2}{\beta} d \ln \left(\frac{\sum_{t=1}^T \|z_t\|_*^2}{\tau} + 1 \right).$$

Proof. First, we show that $\frac{\beta}{2} \langle L(\hat{v}), \hat{v} \rangle \leq \frac{\beta d}{2} \|\hat{v}\|^2$. To see this, observe that for any $x \in B$,

$$\langle L(x), x \rangle = \sum_{i=1}^d \langle b^i, x \rangle^2 \leq \sum_{i=1}^d \|b^i\|_*^2 \|x\|^2 \leq d \|x\|^2.$$

Now, we characterize the sum part of the bound. The basic idea is to use the Auerbach basis to identify B with \mathbb{R}^d (equivalently, we view $\langle L(x), x \rangle$ as an inner product on B). We use this identification to translate all quantities in B and B^* to vectors in \mathbb{R}^d , and observe that the 2-norm of any g_t in \mathbb{R}^d is at most d . Then we use analysis of the same sum terms in the classical analysis of ONS in \mathbb{R}^d [24] to prove the bound.

We spell these identifications explicitly for clarity. Define a map $F : B \rightarrow \mathbb{R}^d$ by

$$F(x) = (\langle b^1, x \rangle, \dots, \langle b^d, x \rangle).$$

We have an associated map $F^* : B^* \rightarrow \mathbb{R}^d$ given by

$$F^*(x^*) = (\langle x^*, b_1 \rangle, \dots, \langle x^*, b_d \rangle).$$

Since $\langle b^i, b_j \rangle = \delta_{ij}$, these maps respect the action of dual vectors in B^* . That is,

$$\langle x, y \rangle = F^*(x) \cdot F(y).$$

Further, since each $\|b_i\| = \|b_i\|_* = 1$, we have

$$\|F(x)\|^2 = \sum_{i=1}^d \langle b^i, x \rangle^2 \leq d\|x\|^2.$$

and

$$\|F^*(x)\|^2 = \sum_{i=1}^d \langle x, b_i \rangle^2 \leq d\|x\|_*^2.$$

where the norm in \mathbb{R}^d is the 2-norm. To make the correspondence notation cleaner, we write $\bar{x} = F(x)$ for $x \in B$ and $\bar{y} = F^*(y)$ for $y \in B^*$. \bar{x}_i indicates the i th coordinate of \bar{x} .

Given any linear map $M : B \rightarrow B^*$ (which we denote by $M \in \mathcal{L}(B, B^*)$), there is an associated map $\bar{M} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ given by

$$\bar{M} = F^* M F^{-1}.$$

Further, when written as a matrix, the ij th element of \bar{M} is

$$\bar{M}_{ij} = (F^* M F^{-1} e_j) \cdot e_i,$$

where e_j represents the j th standard basis element in \mathbb{R}^d . A symmetric statement holds for any linear map $B^* \rightarrow B$, in which $\bar{M} = F M (F^*)^{-1}$.

These maps all commute properly: $\overline{Mx} = \bar{M}\bar{x}$ for any $M \in \mathcal{L}(B, B^*)$ and $x \in B$, and similarly $\overline{Mx} = \bar{M}\bar{x}$ for any $M \in \mathcal{L}(B^*, B)$ and $x \in B^*$. It follows that $\overline{M^{-1}} = \bar{M}^{-1}$ for any M as well.

Now, let's calculate \bar{L}_{ij} :

$$\bar{L}_{ij} = (F^* L F^{-1} e_j) \cdot e_i = \langle L b_j, b_i \rangle = \delta_{ij},$$

so that the matrix \bar{L} is the identity.

Finally, if $M_g : B \rightarrow B^*$ is the map $M_g(x) = \langle g, x \rangle g$, then a simple calculation shows

$$\overline{M_g} = \overline{g} \overline{g}^T.$$

With these details described, recall that we are trying to bound the sum

$$\sum_{t=1}^T \langle z_t, A_t^{-1}(z_t) \rangle.$$

We transfer to \mathbb{R}^d coordinates:

$$\sum_{t=1}^T \langle z_t, A_t^{-1}(z_t) \rangle = \sum_{t=1}^T \overline{z}_t \cdot \overline{A}_t^{-1} \overline{z}_t.$$

We have $\|\overline{z}_t\| \leq \sqrt{d} \|z_t\|_*$ and

$$\overline{A}_t = \tau \overline{L} + \sum_{s=1}^t \overline{z}_s \overline{z}_s^T,$$

so that by [24] Lemma 11,

$$\sum_{t=1}^T \overline{z}_t \cdot \overline{A}_t^{-1} \overline{z}_t \leq \ln \frac{|\overline{A}_T|}{|\overline{A}_0|} \leq d \ln \left(\frac{\sum_{t=1}^T \|\overline{z}_t\|^2}{d\tau} + 1 \right) \leq d \ln \left(\frac{\sum_{t=1}^T \|z_t\|_*^2}{\tau} + 1 \right),$$

where in the second inequality we used the fact that the determinant is maximized when all the eigenvalues are equal to $\frac{\sum_{t=1}^T \|\overline{z}_t\|^2}{d}$. \square

For completeness, we also state the regret bound and the setting of the parameters β and τ to obtain a regret bound for exp-concave functions. Note that we use a different settings in Algorithms 2 and 6, tailored to our specific setting.

Theorem 5.22. *Suppose we run Algorithm 6 on α exp-concave losses. Let D be the diameter of the domain S and $\|\nabla f(x)\|_* \leq Z$ for all the x in S . Then set $\beta = \frac{1}{2} \min\left(\frac{1}{4ZD}, \alpha\right)$ and $\tau = \frac{1}{\beta^2 D^2}$. Then*

$$R_T(\hat{v}) \leq 4d \left(ZD + \frac{1}{\alpha} \right) (1 + \ln(T+1)).$$

Proof. First, observe that classic analysis of α exp-concave functions [24, Lemma 3] shows that for any $x, y \in S$,

$$f(x) \geq f(y) + \langle \nabla f(y), x - y \rangle + \frac{\beta}{2} \langle \nabla f(y), x - y \rangle^2.$$

(Note that although the original proof is stated in \mathbb{R}^d , the exact same argument applies in a Banach space)

Therefore, by Theorems 5.19 and 5.21, we have

$$R_T(u) \leq \frac{\beta\tau}{2} d\|u\|^2 + \frac{2}{\beta} d \ln(Z^2 T / \tau + 1).$$

Substitute our values for β and τ to conclude

$$R_T(u) \leq \frac{d}{2\beta} (1 + \ln(Z^2 T \beta^2 D^2 + 1)) \leq 4d \left(ZD + \frac{1}{\alpha} \right) (1 + \ln(T + 1)),$$

where in the last line we used $\frac{1}{\beta} \leq 8(ZD + 1/\alpha)$. \square

5.C Proofs of Theorems 5.1 and 5.7

In order to prove Theorem 5.1 and 5.7, we first need some technical lemmas. In particular, first we show in Lemma 5.25 that ONS gives us a logarithmic regret against the functions $\ell_t(\beta) = \ln(1 + \langle g_t, \beta \rangle)$. Then, we will link the wealth to the regret with respect to an arbitrary unitary vector thanks to Theorem 5.29.

Lemma 5.23. *For $-1 < x \leq 2$, we have*

$$\ln(1 + x) \leq x - \frac{2 - \ln(3)}{4} x^2.$$

Lemma 5.24. *Define $\ell_t(v) = -\ln(1 - \langle g_t, v \rangle)$. Let $\|\hat{v}\|, \|v\| \leq \frac{1}{2}$ and $\|g_t\|_* \leq 1$. Then*

$$\ell_t(v) - \ell_t(\hat{v}) \leq \langle \nabla \ell_t(v), v - \hat{v} \rangle - \frac{2 - \ln(3)}{2} \frac{1}{2} \langle \nabla \ell_t(v), v - \hat{v} \rangle^2.$$

Proof. We have

$$\ln(1 - \langle g_t, \hat{v} \rangle) = \ln(1 - \langle g_t, v \rangle + \langle g_t, v - \hat{v} \rangle) = \ln(1 - \langle g_t, v \rangle) + \ln \left(1 + \frac{\langle g_t, v - \hat{v} \rangle}{1 - \langle g_t, v \rangle} \right).$$

Now, observe that since $1 - \langle g_t, \hat{v} \rangle \geq 0$ and $1 - \langle g_t, v \rangle \geq 0$, $1 + \frac{\langle g_t, v - \hat{v} \rangle}{1 - \langle g_t, v \rangle} \geq 0$ as well so that $\frac{\langle g_t, v - \hat{v} \rangle}{1 - \langle g_t, v \rangle} \geq -1$. Further, since $\|\hat{v} - v\| \leq 1$ and $1 - \langle g_t, v \rangle \geq 1/2$, $\frac{\langle g_t, v - \hat{v} \rangle}{1 - \langle g_t, v \rangle} \leq 2$. Therefore, by Lemma 5.23 we have

$$\ln(1 - \langle g_t, \hat{v} \rangle) \leq \ln(1 - \langle g_t, v \rangle) + \frac{\langle g_t, v - \hat{v} \rangle}{1 - \langle g_t, v \rangle} - \frac{2 - \ln(3)}{4} \frac{\langle g_t, v - \hat{v} \rangle^2}{(1 - \langle g_t, v \rangle)^2}.$$

Using the fact that $\nabla \ell_t(v) = \frac{g_t}{1 - \langle g_t, v \rangle}$ finishes the proof. \square

Lemma 5.25. *Define $S = \{v \in B : \|v\| \leq \frac{1}{2}\}$ and $\ell_t(v) : S \rightarrow \mathbb{R}$ as $\ell_t(v) = -\ln(1 - \langle g_t, v \rangle)$, where*

$\|g_t\|_* \leq 1$. If we run ONS in Algorithm 6 with $\beta = \frac{2-\ln(3)}{2}$, $\tau = 1$, and $S = \{v : \|v\| \leq \frac{1}{2}\}$, then

$$\sum_{t=1}^T \ell_t(v_t) - \ell_t(\hat{v}) \leq d \left(\frac{1}{17} + 4.5 \ln \left(1 + 4 \sum_{t=1}^T \|g_t\|_*^2 \right) \right).$$

Proof. From Lemma 5.24, we have

$$\sum_{t=1}^T \ell_t(v_t) - \ell_t(\hat{v}) \leq \sum_{t=1}^T \left(\langle \nabla \ell_t(v_t), v_t - \hat{v} \rangle - \frac{\beta}{2} \langle \nabla \ell_t(v_t), v_t - \hat{v} \rangle^2 \right).$$

So, using Lemma 5.19 we have

$$\sum_{t=1}^T \left(\langle \nabla \ell_t(v_t), v_t - \hat{v} \rangle - \frac{\beta}{2} \langle \nabla \ell_t(v_t), v_t - \hat{v} \rangle^2 \right) \leq \frac{\beta}{2} \langle L(\hat{v}), \hat{v} \rangle + \frac{2}{\beta} \sum_{t=1}^T \langle z_t, A_t^{-1}(z_t) \rangle,$$

where $z_t = \nabla \ell_t(v_t)$. Now, use Theorem 5.21 so that

$$\frac{\beta}{2} \langle L(\hat{v}), \hat{v} \rangle + \frac{2}{\beta} \sum_{t=1}^T \langle z_t, A_t^{-1}(z_t) \rangle \leq \frac{d\beta}{8} + \frac{2d}{\beta} \ln \left(1 + \sum_{t=1}^T \|z_t\|_*^2 \right),$$

where we have used $\|\hat{v}\| \leq 1/2$. Then observe that $\|z_t\|_*^2 = \frac{\|g_t\|_*^2}{(1+\langle g_t, \beta_t \rangle)^2} \leq 4\|g_t\|_*^2$ so that $\ln(1 + \sum_{t=1}^T \|z_t\|_*^2) \leq \ln(1 + 4 \sum_{t=1}^T \|g_t\|_*^2)$. Finally, substitute the specified value of β and numerically evaluate to conclude the bound. \square

Now, we collect some Fenchel conjugate calculations that allow us to convert our wealth lower-bounds into regret upper-bounds:

Lemma 5.26. *Let $f(x) = a \exp(b|x|)$, where $a, b > 0$. Then*

$$f^*(\theta) = \begin{cases} \frac{|\theta|}{b} \left(\ln \frac{|\theta|}{ab} - 1 \right), & \frac{|\theta|}{ab} > 1 \\ -a, & \text{otherwise.} \end{cases} \leq \frac{|\theta|}{b} \left(\ln \frac{|\theta|}{ab} - 1 \right).$$

Lemma 5.27. *Let $f(x) = a \exp(b \frac{x^2}{|x|+c})$, where $a, b > 0$ and $c \geq 0$. Then*

$$f^*(\theta) \leq |\theta| \max \left(\frac{2}{b} \left(\ln \frac{2|\theta|}{ab} - 1 \right), \sqrt{\frac{c}{b} \ln \left(\frac{c\theta^2}{a^2b} + 1 \right)} - a \right).$$

Proof. By definition we have

$$f^*(\theta) = \sup_x \theta x - f(x).$$

It is easy to see that the sup cannot be attained at infinity, hence we can safely assume that it is attained at $x^* \in \mathbb{R}$. We now do a case analysis, based on x^* .

Case $|x^*| \leq c$. In this case, we have that $f(x^*) \geq a \exp(b \frac{x^2}{2c})$, so

$$\begin{aligned} f^*(\theta) &= \theta x^* - f(x^*) \leq \theta x^* - a \exp\left(b \frac{(x^*)^2}{2c}\right) \\ &\leq \sup_x \theta x - a \exp\left(b \frac{x^2}{2c}\right) \leq |\theta| \sqrt{\frac{c}{b} \ln\left(\frac{c\theta^2}{a^2 b} + 1\right)} - a, \end{aligned}$$

where the last inequality is from Lemma 18 in [40].

Case $|x^*| > c$. In this case, we have that $f(x^*) \geq a \exp\left(b \frac{(x^*)^2}{2|x^*|}\right) = a \exp\left(\frac{b}{2}|x^*|\right)$, so

$$\begin{aligned} f^*(\theta) &= \theta x^* - f(x^*) \leq \theta x^* - a \exp\left(\frac{b}{2}|x^*|\right) \\ &\leq \sup_x \theta x - a \exp\left(\frac{b}{2}|x|\right) \leq \frac{2|\theta|}{b} \left(\ln \frac{2|\theta|}{ab} - 1\right), \end{aligned}$$

where the last inequality is from Lemma 5.26.

Considering the max over the two cases gives the stated bound. □

Theorem 5.28. *Let u be an arbitrary unit vector and $\|g_t\|_* \leq 1$ for $t = 1, \dots, T$. Then*

$$\sup_{\|v\| \leq \frac{1}{2}} \sum_{t=1}^T \ln(1 - \langle g_t, v \rangle) \geq \frac{1}{4} \frac{\langle \sum_{t=1}^T g_t, u \rangle^2}{\sum_{t=1}^T \langle g_t, u \rangle^2 + \left| \langle \sum_{t=1}^T g_t, u \rangle \right|}.$$

Proof. Recall that $\ln(1+x) \geq x - x^2$ for $|x| \leq 1/2$. Then, we compute

$$\begin{aligned} \sup_{\|v\| \leq 1/2} \sum_{t=1}^T \ln(1 - \langle g_t, v \rangle) &\geq \sup_{\|v\| \leq 1/2} \sum_{t=1}^T (-\langle g_t, v \rangle - \langle g_t, v \rangle^2) \\ &= \sup_{\|v\| \leq 1/2} -\left\langle \sum_{t=1}^T g_t, v \right\rangle - \sum_{t=1}^T \langle g_t, v \rangle^2. \end{aligned}$$

Choose $v = \frac{u}{2} \frac{\langle \sum_{t=1}^T g_t, u \rangle}{\sum_{t=1}^T \langle g_t, u \rangle^2 + \left| \langle \sum_{t=1}^T g_t, u \rangle \right|}$. Then, clearly $\|v\| \leq \frac{1}{2}$. Thus, we have

$$\begin{aligned} \sup_{\|v\| \leq 1/2} \sum_{t=1}^T \ln(1 - \langle g_t, v \rangle) &\geq \sup_{\|v\| \leq 1/2} -\left\langle \sum_{t=1}^T g_t, v \right\rangle - \sum_{t=1}^T \langle g_t, v \rangle^2 \\ &\geq \frac{1}{2} \frac{\langle \sum_{t=1}^T g_t, u \rangle^2}{\sum_{t=1}^T \langle g_t, u \rangle^2 + \left| \langle \sum_{t=1}^T g_t, u \rangle \right|} - \frac{\langle \sum_{t=1}^T g_t, u \rangle^2}{4 \left(\sum_{t=1}^T \langle g_t, u \rangle^2 + \left| \langle \sum_{t=1}^T g_t, u \rangle \right| \right)^2} \sum_{t=1}^T \langle g_t, u \rangle^2 \\ &\geq \frac{1}{4} \frac{\langle \sum_{t=1}^T g_t, u \rangle^2}{\sum_{t=1}^T \langle g_t, u \rangle^2 + \left| \langle \sum_{t=1}^T g_t, u \rangle \right|}. \end{aligned}$$

□

Lemma 5.29. *Let u be an arbitrary unit vector in B and $t > 0$. Then, using the Algorithm 6, we have*

$$R_T(tu) \leq \epsilon + t \max \left[\frac{d}{2} - 8 + 8 \ln \frac{8t \left(4 \sum_{t=1}^T \|g_t\|_*^2 + 1 \right)^{4.5d}}{\epsilon}, \right. \\ \left. 2 \sqrt{\sum_{t=1}^T \langle g_t, u \rangle^2 \ln \left(\frac{5t^2}{\epsilon^2} \exp \left(\frac{d}{17} \right) \left(4 \sum_{t=1}^T \|g_t\|^2 + 1 \right)^{9d+1} + 1 \right)} \right].$$

Proof. Let's compute a bound on our wealth, Wealth_T . We have that

$$\text{Wealth}_t = \text{Wealth}_{t-1} - \langle g_t, w_t \rangle = \text{Wealth}_{t-1} (1 - \langle g_t, v_t \rangle) = \epsilon \prod_{t=1}^T (1 - \langle g_t, v_t \rangle),$$

and taking the logarithm we have

$$\ln \text{Wealth}_t = \ln \epsilon + \sum_{t=1}^T \ln(1 - \langle g_t, v_t \rangle).$$

Hence, using Lemma 5.25, we have

$$\ln \text{Wealth}_t \geq \ln \epsilon + \max_{\|v\| \leq \frac{1}{2}} \sum_{t=1}^T \ln(1 + \langle g_t, v \rangle) - d \left(\frac{1}{17} + 4.5 \ln \left(1 + \sum_{t=1}^T 4 \|g_t\|_*^2 \right) \right).$$

Using Theorem 5.28, we have

$$\text{Wealth}_T \geq \frac{\epsilon}{\exp \left[d \left(\frac{1}{17} + 4.5 \ln \left(1 + 4 \sum_{t=1}^T \|g_t\|_*^2 \right) \right) \right]} \exp \left[\frac{1}{4} \frac{\langle \sum_{t=1}^T g_t, u \rangle^2}{\sum_{t=1}^T \langle g_t, u \rangle^2 + \left| \langle \sum_{t=1}^T g_t, u \rangle \right|} \right].$$

Defining

$$f(x) = \frac{\epsilon}{\exp \left[d \left(\frac{1}{17} + 4.5 \ln \left(1 + 4 \sum_{t=1}^T \|g_t\|_*^2 \right) \right) \right]} \exp \left[\frac{1}{4} \frac{x^2}{\sum_{t=1}^T \langle g_t, u \rangle^2 + |x|} \right],$$

we have

$$\begin{aligned}
R_T(tu) &= \epsilon - \text{Wealth}_T - t \left\langle \sum_{t=1}^T g_t, u \right\rangle \\
&\leq \epsilon - t \left\langle \sum_{t=1}^T g_t, u \right\rangle - f \left(\left\langle \sum_{t=1}^T g_t, u \right\rangle \right) \\
&\leq \epsilon + f^*(-t) \\
&\leq \epsilon + t \max \left[8 \left(\ln \frac{8t}{\epsilon} + \frac{d}{17} + 4.5d \ln \left(4 \sum_{t=1}^T \|g_t\|_*^2 + 1 \right) - 1 \right), \right. \\
&\quad \left. \sqrt{4 \sum_{t=1}^T \langle g_t, u \rangle^2 \ln \left(\frac{5t^2}{\epsilon^2} \exp \left(\frac{d}{17} \right) \left(4 \sum_{t=1}^T \|g_t\|^2 + 1 \right)^{9d} \sum_{t=1}^T \langle g_t, u \rangle^2 + 1 \right)} \right] \\
&\leq \epsilon + t \max \left[\frac{d}{2} - 8 + 8 \ln \frac{8t \left(4 \sum_{t=1}^T \|g_t\|_*^2 + 1 \right)^{4.5d}}{\epsilon}, \right. \\
&\quad \left. 2 \sqrt{\sum_{t=1}^T \langle g_t, u \rangle^2 \ln \left(\frac{5t^2}{\epsilon^2} \exp \left(\frac{d}{17} \right) \left(4 \sum_{t=1}^T \|g_t\|^2 + 1 \right)^{9d+1} + 1 \right)} \right],
\end{aligned}$$

where we have used the calculation of Fenchel conjugate of f from Lemma 5.27. Then observe that $\exp(d/17) \leq \exp((9d+1)/153) \leq 2^{9d+1}$ to conclude:

$$\begin{aligned}
R_T(tu) &\leq \epsilon + t \max \left[\frac{d}{2} - 8 + 8 \ln \frac{8t \left(4 \sum_{t=1}^T \|g_t\|_*^2 + 1 \right)^{4.5d}}{\epsilon}, \right. \\
&\quad \left. 2 \sqrt{\sum_{t=1}^T \langle g_t, u \rangle^2 \ln \left(\frac{5t^2}{\epsilon^2} \left(8 \sum_{t=1}^T \|g_t\|^2 + 2 \right)^{9d+1} + 1 \right)} \right].
\end{aligned}$$

□

Proof of Theorem 5.7. Given some \hat{w} , set $u = \frac{\hat{w}}{\|\hat{w}\|}$ and $t = \|\hat{w}\|$. Then observe that $t^2 \sum_{t=1}^T \langle g_t, u \rangle^2 = \sum_{t=1}^T \langle g_t, \hat{w} \rangle^2$ and apply the previous Lemma 5.29 to conclude the desired result. □

5.D Proof of Proposition 5.5 and Theorem 5.6

We restate Proposition 5.5 below:

Proposition 5.5. S_W is convex and 1-Lipschitz for any closed convex set W in a reflexive Banach space B .

Proof. Let $x, y \in B$, $t \in [0, 1]$, $x' \in \Pi_W(x)$, and $y' \in \Pi_W(y)$. Then

$$\begin{aligned} S_W(tx + (1-t)y) &= \min_{d \in W} \|tx + (1-t)y - d\| \leq \|tx + (1-t)y - tx' - (1-t)y'\| \\ &= \|t(x - x') + (1-t)(y - y')\| \leq t\|x - x'\| + (1-t)\|y - y'\| \\ &= tS_W(x) + (1-t)S_W(y). \end{aligned}$$

For the Lipschitzness, let $x \in B$ and $x' \in \Pi_W(x)$, and observe that

$$S_W(x + \delta) = \inf_{d \in W} \|x + \delta - d\| \leq \|x + \delta - x'\| \leq S_W(x) + \|\delta\|.$$

Similarly, let $x \in B$, δ such that $x + \delta \in B$ and $x' \in \Pi_W(x + \delta)$, then

$$S_W(x) = \min_{d \in W} \|x - d\| \leq \|x + \delta - \delta - x'\| \leq S_W(x + \delta) + \|\delta\|.$$

So that $|S_W(x) - S_W(x + \delta)| \leq \|\delta\|$. □

Now we restate and prove Theorem 5.6:

Theorem 5.6. *Let B be a reflexive Banach space such that for every $0 \neq b \in B$, there is a unique dual vector b^* such that $\|b^*\|_* = 1$ and $\langle b^*, b \rangle = \|b\|$. Let $W \subset B$ a closed convex set. Given $x \in B$ and $x \notin W$, let $p \in \Pi_W(x)$. Then $\{(x - p)^*\} = \partial S_W(x)$.*

Proof. Let $x' = \frac{x+p}{2}$. Then clearly $S_W(x') \leq \|x' - p\| = \frac{\|x-p\|}{2} = S_W(x) - \|x - x'\|$. Since S_W is 1-Lipschitz, $S_W(x') \geq S_W(x) - \|x - x'\|$ and so $S_W(x') = S_W(x) - \|x - x'\|$.

Suppose $g \in \partial S_W(x)$. Then $\langle g, x' - x \rangle + S_W(x) \leq S_W(x') = S_W(x) - \|x - x'\|$. Therefore, $\langle g, x' - x \rangle \leq -\|x - x'\|$. Since $\|g\|_* \leq 1$, we must have $\|g\|_* = 1$ and $\langle g, x - p \rangle = \|x - p\|$. By assumption, this uniquely specifies the vector $(x - p)^*$. Since ∂S_W is not the empty set, $\{(x - p)^*\} = \partial S_W(x)$. □

Chapter 6

Other Applications

In this chapter we introduce two applications of our reductions in Chapter 5. First, we consider the *multi-scale experts* problem, which is an optimization problem over the probability simplex and so makes use of our unconstrained-to-constrained reduction. Then we return to the unconstrained setting and demonstrate that a simple coordinate-wise update scheme can produce an algorithm that adapts to sparsity in either \hat{w} of g_t while maintaining nearly dimension-free regret when neither quantity is sparse.

6.1 Reduction for Multi-Scale Experts

In this section, we apply our reductions to the multi-scale experts problem considered in [20; 7]. Our algorithm improves upon both prior algorithms: the approach of [7] has a mildly sub-optimal dependence on the prior distribution, while the approach of [20] takes time $O(T)$ per update, resulting in a quadratic total runtime. Our algorithm matches the regret bound of [20] while running in the same time complexity as online gradient descent.

The multi-scale experts problem is an online linear optimization problem over the probability simplex $\{x \in \mathbb{R}_{\geq 0}^N : \sum_{i=1}^N x_i = 1\}$ with linear losses $\ell_t(w) = g_t \cdot w$ such that each $g_t = (g_{t,1}, \dots, g_{t,N})$ satisfies $|g_{t,i}| \leq c_i$ for some known quantities c_i . Given a prior discrete distribution (π_1, \dots, π_N) , the objective is to guarantee that the regret with respect to the i th basis vector e_i (the i th “expert”) scales with c_i . Formally, we want $R_T(\hat{w}) = O(\sum_{i=1}^N c_i |\hat{w}_i| \sqrt{T \log(c_i |\hat{w}_i| T / \pi_i)})$. As discussed in depth by [20], such a guarantee allows us to combine many optimization algorithms into one meta-algorithm that converges at the rate of the best algorithm *in hindsight*.

We accomplish this through two reductions. First, given any distribution (π_1, \dots, π_N) and any family of 1-dimensional OLO algorithms $\mathcal{A}(\epsilon)$ that guarantees $R(u) \leq O(\epsilon + |u| \sqrt{\log(|u|T/\epsilon)T})$ on 1-Lipschitz losses for any given ϵ (such as our Algorithm 2 or many other parameter-free algorithms), we apply the classic “coordinate-wise updates” trick [58] to generate an N -dimensional OLO algorithm with regret $R_T(u) = O(\epsilon + \sum_{i=1}^N |u_i| \sqrt{\log(|u_i|T/(\epsilon\pi_i))T})$ on losses that are 1-Lipschitz with respect to the 1-norm.

Algorithm 9 Coordinate-Wise Updates

Require: parametrized family of 1-D online learning algorithm $\mathcal{A}(\epsilon)$, prior $\pi \in \mathbb{R}^N$, $\epsilon > 0$

- 1: **Initialize:** N copies of \mathcal{A} : $\mathcal{A}_1(\epsilon\pi_1), \dots, \mathcal{A}_N(\epsilon\pi_N)$
- 2: **for** $t = 1$ **to** T **do**
- 3: Get points $z_{t,i}$ from \mathcal{A}_i for all i to form vector $z_t = (z_{t,1}, \dots, z_{t,N})$
- 4: Play z_t , get loss $g_t \in \mathbb{R}^N$ with $\|g_t\|_\infty \leq 1$
- 5: Send $g_{t,i}$ to \mathcal{A}_i for all i
- 6: **end for**

Theorem 6.1. *Suppose for any $\epsilon > 0$, $\mathcal{A}(\epsilon)$ guarantees regret*

$$R_T(\hat{w}) \leq R_T(\epsilon, \hat{w}, g_1, \dots, g_T)$$

when run on 1-dimensional subgradients $g_1, \dots, g_T \in \mathbb{R}$ with $|g_t| \leq 1$. Then Algorithm 9 guarantees regret

$$R_T(\hat{w}) \leq \sum_{i=1}^N R_T(\epsilon/\pi_i, \hat{w}_i, g_{1,i}, \dots, g_{T,i})$$

Proof. By \mathcal{A} 's regret guarantee we have

$$\sum_{t=1}^T w_{t,i} g_{t,i} - \hat{w}_i g_{t,i} \leq R_T(\epsilon\pi_i, \hat{w}_i, g_{1,i}, \dots, g_{T,i})$$

Summing over all i we obtain:

$$\begin{aligned} R_T(\hat{w}) &= \sum_{t=1}^T \langle g_t, w_t - \hat{w} \rangle \\ &= \sum_{i=1}^N \sum_{t=1}^T g_{t,i} (w_{t,i} - \hat{w}_i) \\ &\leq \sum_{i=1}^N \pi_i R_T(\epsilon\pi_i, \hat{w}_i, g_{1,i}, \dots, g_{T,i}). \end{aligned}$$

□

When e apply this result to a 1-D algorithm that guarantees

$$R_T(u) \leq O\left(\epsilon + |u| \sqrt{\log(|u|T/\epsilon)T}\right)$$

such as described in the previous section, or in prior works [35], we obtain

$$R_T(\hat{w}) = O \left(\epsilon + \sum_{i=1}^N |\hat{w}_i| \sqrt{\sum_{t=1}^T g_{t,i}^2 \log(|\hat{w}_i|^2 \sum_{t=1}^T g_{t,i}^2 / \epsilon^2 \pi_i^2)} \right)$$

Algorithm 10 Multi-Scale Experts

Require: parametrized 1-D Online learning algorithm $\mathcal{A}(\epsilon)$, prior π , scales c_1, \dots, c_N

- 1: **Initialize:** coordinate-wise algorithm \mathcal{A}_π with prior π using $\mathcal{A}(\epsilon)$
 - 2: Define $W = \{x : x_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^N x_i/c_i = 1\}$
 - 3: Let \mathcal{A}_π^W be the result of applying the unconstrained-to-constrained reduction to \mathcal{A}_π with constraint set W using $\|\cdot\|_1$
 - 4: **for** $t = 1$ **to** T **do**
 - 5: Get point $z_t \in W$ from \mathcal{A}_π^W
 - 6: Set $x_t \in \mathbb{R}^N$ by $x_{t,i} = z_{t,i}/c_i$. Observe that x_t is in the probability simplex
 - 7: Play x_t , get loss vector g_t
 - 8: Set $\tilde{g}_t \in \mathbb{R}^N$ by $\tilde{g}_{t,i} = \frac{g_{t,i}}{c_i}$
 - 9: Send \tilde{g}_t to \mathcal{A}_π^W
 - 10: **end for**
-

With this in hand, notice that applying our reduction Algorithm 5 with the 1-norm easily yields an algorithm over the probability simplex W with the same regret (up to a factor of 2), as long as $\|g_t\|_\infty \leq 1$. Then, we apply an affine change of coordinates to make our multi-scale experts losses have $\|g_t\|_\infty \leq 1$, so that applying this algorithm yields the desired result (see Algorithm 10).

Theorem 6.2. *If g_t satisfies $|g_{t,i}| \leq c_i$ for all t and i and $\mathcal{A}(\epsilon)$ guarantees the regret bound*

$$R_T(u) \leq O \left(\epsilon + |u| \sqrt{\log(|u|T/\epsilon)T} \right)$$

then, for any \hat{w} in the probability simplex, Algorithm 10 satisfies the regret bound

$$R_T(\hat{w}) \leq O \left(\epsilon + \sum_{i=1}^N C_i |\hat{w}_i| \sqrt{\log \left(\frac{C_i |\hat{w}_i| T}{\epsilon \pi_i} \right) T} \right).$$

Proof. Given any \hat{w} in the probability simplex, define $\tilde{w} \in \mathbb{R}^N$ by $\tilde{w}_i = \hat{w}_i c_i$. Observe that $\tilde{w} \in W$. Further, observe that since $|g_{t,i}| \leq c_i$, $\|\tilde{g}_t\|_\infty \leq 1$. Now, by Theorem 6.1 and Theorem 5.4 we have

$$\sum_{t=1}^T \tilde{g}_t \cdot z_t - \tilde{g}_t \cdot \tilde{w} \leq O \left(\epsilon + \sum_{i=1}^N |\tilde{w}_i| \sqrt{\log \left(\frac{|\tilde{w}_i| T}{\epsilon \pi_i} \right) T} \right) = O \left(\epsilon + \sum_{i=1}^N c_i |\hat{w}_i| \sqrt{\log \left(\frac{c_i |\hat{w}_i| T}{\epsilon \pi_i} \right) T} \right),$$

where in the equality we simply substitute the definition of \tilde{w} . Finally, observe that $\tilde{g}_t \cdot z_t = \sum_{i=1}^N \tilde{g}_{t,i} z_{t,i} = \sum_{i=1}^N \frac{g_{t,i}}{c_i} c_i x_{t,i} = g_t \cdot x_t$ and similarly $\tilde{g}_t \cdot \tilde{w} = g_t \cdot \hat{w}$. Thus $\sum_{t=1}^T \tilde{g}_t \cdot z_t - \tilde{g}_t \cdot \tilde{w} = \sum_{t=1}^T g_t \cdot (x_t - \hat{w})$, which completes the proof. \square

In Appendix 6.A we show how to compute the projection Π_S and a subgradient of S_W in $O(N)$ time via a simple greedy algorithm. As a result, our entire reduction runs in $O(N)$ time per update.

6.2 Sparsity

In this section we analyze the coordinate-wise reduction (Algorithm 9) from the previous section and show that we can adapt to sparsity in the subgradients g_t as well as the comparison point \hat{w} , while still maintaining good performance when neither the subgradients nor the comparison point is sparse. More concretely, let us assume that our domain W is \mathbb{R}^N for some N , and that each subgradient g_t satisfies $\|g_t\|_\infty \leq 1$. Then if we apply Theorem 6.1 to our 1-D coin-betting strategy (Algorithm 2) with $\pi_i = \frac{1}{N}$ for all i and $\epsilon = 1$, we obtain

$$R_T(\hat{w}) \leq O \left(\sum_{i=1}^N |\hat{w}_i| \sqrt{\sum_{t=1}^T g_{t,i}^2 \log \left(N^2 |\hat{w}_i|^2 \sum_{t=1}^T g_{t,i}^2 \right)} \right) \quad (6.1)$$

where \hat{w}_i and $g_{t,i}$ indicate the i th coordinates of the corresponding vectors.

It is interesting to compare this regret bound to the one we obtain in Chapter 5 via our reduction from Banach space online learning to 1 dimensional online learning using the 2-norm (Algorithm 3). Recall that applying this reduction using the 2-norm will give regret:

$$R_T(\hat{w}) = O \left[\|\hat{w}\| \max \left(\ln \frac{\|\hat{w}\| \sum_{t=1}^T \|g_t\|_*^2}{\epsilon}, \sqrt{\sum_{t=1}^T \|g_t\|_*^2 \ln \left(\frac{\|\hat{w}\|^2 \sum_{t=1}^T \|g_t\|_*^2}{\epsilon^2} + 1 \right)} \right) \right]$$

In contrast, applying Cauchy-Schwarz inequality to (6.1) yields:

$$R_T(\hat{w}) \leq O \left(\|\hat{w}\|_2 \sqrt{\sum_{t=1}^T \|g_t\|_2^2 \log(N^2 \|\hat{w}\|_\infty^2 T)} \right) \quad (6.2)$$

which differs by only a $\log(N)$ factor. Thus using coordinate-wise updates seems to lose only a very small dimension-dependence over using the reduction of Algorithm 3 for the 2-norm.

Next, consider a case in which we expect \hat{w} to be very sparse. For example, perhaps we are performing linear regression and we expect most of our regressors to actually carry no signal. Without loss of generality,

let $\dot{w}_i \neq 0$ for $i \leq K$ and $\dot{w}_i = 0$ for $i \geq K$. Then (6.1) immediately implies:

$$R_T(\dot{w}) \leq O \left(\sum_{i=1}^K |\dot{w}_i| \sqrt{\sum_{t=1}^T g_{t,i}^2 \log(N^2 |\dot{w}_i|^2 T)} \right) \quad (6.3)$$

$$\leq O \left(\|\dot{w}\|_1 \sqrt{\sum_{t=1}^T \|g_t\|_\infty^2 \log(N^2 \|\dot{w}\|_\infty^2 T)} \right) \quad (6.4)$$

$$(6.5)$$

This means that, at the cost of a $\log(N)$ factor, we can add as many irrelevant coordinates as we like without affecting the regret bound!

Finally, we report an example by [17] which suggests that (6.1) implies lower regret when the g_t are sparse as well: suppose each $g_{t,i} \neq 0$ with probability proportional to $1/i^2$, and is 1 otherwise. Then we have

$$\begin{aligned} R_T(\dot{w}) &\leq O \left(\|\dot{w}\|_\infty \sum_{i=1}^N \sqrt{\sum_{t=1}^T g_{t,i}^2 \log(N^2 |\dot{w}_i|^2 T)} \right) \\ &\leq O \left(\|\dot{w}\|_\infty \sum_{i=1}^N \sqrt{\frac{T \log(N^2 \|\dot{w}\|_\infty^2 T)}{i^2}} \right) \\ &\leq O \left(\|\dot{w}\|_\infty \log(N) \sqrt{T \log(N^2 \|\dot{w}\|_\infty^2 T)} \right) \end{aligned}$$

In contrast, if \dot{w} is a dense vector, we would expect $\|\dot{w}\|_2 \approx \|\dot{w}\|_\infty \sqrt{N}$, and so we have saved a factor of \sqrt{N} over the bound (6.2).

6.3 Conclusions

The results in this chapter are surprisingly straightforward to derive using our reductions from chapter 5 (with the exception of computing the gradient of S_W in Algorithm 10). Nevertheless, their guarantees are also surprisingly strong. This exemplifies the power of using reductions to derive algorithms: by peeling away complicated internal details of individual algorithms, we are able to zero in on only the critical parts of the analysis and remove unnecessary work. It is my hope that more applications and improved reductions will continue this process in the future.

Appendix

6.A Computing S_W for multi-scale experts

In this section we show how to compute $\Pi_W(x)$ and a subgradient of $S_W(x)$ used in Algorithm 10. First we tackle $\Pi_W(x)$. Without loss of generality, assume the c_i are ordered so that $c_1 \geq c_2 \geq \dots \geq c_N$. We also consider $W_k = \{x : x_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^N x_i/c_i = k\}$ instead of $W = W_1$. Obviously we are particularly interested in the case $k = 1$, but working in this mild generality allows us to more easily state an algorithm for computing $\Pi_W(x)$ in a recursive manner.

Proposition 6.3. *Let $N > 1$ and $W_k = \{x : x_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^N x_i/c_i = k\}$, and let $S_{W_k}(x) = \inf_{y \in W_k} \|x - y\|_1$. Suppose the c_i are ordered so that $c_1 \geq c_2 \geq \dots \geq c_N$. Then for any $x = (x_1, \dots, x_n)$, there exists a $y = (y_1, \dots, y_n) \in \Pi_{W_k}(x)$ such that*

$$y_1 = \begin{cases} 0, & x_1 < 0 \\ x_1, & x_1 \in [0, kc_1] \\ kc_1, & x_1 > kc_1 \end{cases}$$

Proof. First, suppose $N = 1$. Then clearly there is only one element of W_k and so the choice of $\Pi_{W_k}(x)$ is forced. So now assume $N > 1$.

Let $(y_1, \dots, y_N) \in \Pi_{W_k}(x_1, \dots, x_N)$ be such that $|y_1 - x_1|$ is as small as possible (such a point exists because W_k is compact).

We consider three cases: either $x_1 > kc_1$, $x_1 < 0$ or $x_1 \in [0, kc_1]$.

Case 1: $x > kc_1$. Suppose $y_1 < kc_1$. Let i be the largest index such that $y_i \neq 0$. $i \neq 1$ since $y_1/c_1 < k$. Choose $0 < \epsilon < \min(y_i \frac{c_1}{c_i}, kc_1 - y_1)$. Then let y' be such that $y'_1 = y_1 + \epsilon$, $y'_i = y_i - \epsilon \frac{c_i}{c_1}$ and $y'_j = y_j$ otherwise. Then by definition of ϵ , $y'_i \geq 0$ and $y'_1 \leq kc_1$. Further, $\sum_{j=1}^N y'_j/c_j = \epsilon/c_1 - \frac{c_i}{c_1} \epsilon/c_i + \sum_{j=1}^N y_j/c_j = k$ so that $y' \in W_k$. However, since $x_1 > kc_1$, $\|y' - x\|_1 \leq \|y - x\|_1 - \epsilon + \epsilon \frac{c_i}{c_1} \leq \|y - x\|_1$. Therefore, $y' \in \Pi_{W_k}(x)$, but $|y'_1 - x_1| < |y_1 - x_1|$, contradicting our choice of y_1 . Therefore, $y_1 = kc_1$.

Case 2: $x < 0$. This case is very similar to the previous case. Suppose $y_1 > 0$. Let i be the largest index such that $y_i \neq kc_i$. $i \neq 1$ since otherwise $\sum_{j=1}^N y_j/c_j > \sum_{j=2}^N k = k(N-1) \geq k$, which is not possible. Choose $0 < \epsilon < \min(y_1, c_1(kc_i - y_i)/c_i)$. Set y' such that $y'_1 = y_1 - \epsilon$, $y'_i = y_i + \epsilon \frac{c_i}{c_1}$. Then, again we have

$y' \in W_k$ and $\|y' - x\|_1 \leq \|y - x\|_1 - \epsilon + \epsilon \frac{c_i}{c_1} \leq \|y - x\|_1$ so that $y' \in \Pi_{W_k}(x)$, but $|y'_1 - x_1| < |y_1 - x_1|$. Therefore, we cannot have $y_1 > 0$ and so $y_1 = 0$.

Case 3: $x \in [0, kc_1]$. Suppose $y_1 < x_1 \leq kc_1$. Then by the same the argument as for Case 1, there is some $i > 1$ such that for any $0 < \epsilon < \min(y_i \frac{c_1}{c_i}, x_1 - y_1)$, we can construct y' with $y' \in \Pi_{W_k}(x)$ and $|y'_1 - x_1| < |y_1 - x_1|$. Therefore, $y_1 \geq x_1$.

Similarly, if $y_1 > x_1$, then by the same argument as for Case 2, there is some $i > 1$ such that for any $0 < \epsilon < \min(y_1 - x_1, c_1(kc_i - y_i)/c_i)$, we again construct y' with $y' \in \Pi_{W_k}(x)$ and $|y'_1 - x_1| < |y_1 - x_1|$. Therefore, $y_1 = x_1$. \square

This result suggests an explicit algorithm for choosing $y \in \Pi_W(x) = \Pi_{W_1}(x)$. Using the Proposition we can pick y_1 such that there is a $y \in \Pi_{W_1}(x)$ with first coordinate y_1 . If $y \in \Pi_{W_k}(x)$ has first coordinate y_1 , then if $W_k^2 = \{(y_2, \dots, y_n) : y_i \geq 0 \text{ for all } i \text{ and } \sum_{i=2}^N y_i/c_i = k\}$, then $(y_2, \dots, y_N) \in \Pi_{W_{k-y_1/c_1}}^2(x_2, \dots, x_N)$. Therefore, we can use a greedy algorithm to choose each y_i in increasing order of i and obtain a point $y \in \Pi_{W_k}(x)$ in $O(N)$ time. This procedure is formalized in Algorithm 11.

Algorithm 11 Computing $\Pi_W(x)$

Require: $(x_1, \dots, x_N) \in \mathbb{R}^N$

- 1: **Initialize:** $k_1 = 1, i = 1$
 - 2: **for** $i = 1$ **to** N **do**
 - 3: **if** $i = N$ **then**
 - 4: Set $y_i = k_i c_i$
 - 5: **else**
 - 6: **if** $x_i \leq 0$ **then**
 - 7: Set $y_i = 0$
 - 8: **end if**
 - 9: **if** $x_i > k_i c_i$ **then**
 - 10: Set $y_i = k_i c_i$
 - 11: **end if**
 - 12: **if** $x_i \in (0, k_i c_i]$ **then**
 - 13: Set $y_i = x_i$
 - 14: **end if**
 - 15: Set $k_{i+1} = k_i - y_i/c_i$
 - 16: **end if**
 - 17: **end for**
 - 18: **return** (y_1, \dots, y_N)
-

6.A.1 Computing a subgradient of S_W for multi-scale experts

Unfortunately, $\|\cdot\|_1$ does not satisfy the hypotheses of Theorem 5.6 and so we need to do a little more work to compute a subgradient.

Proposition 6.4. *Let (y_1, \dots, y_n) be the output of Algorithm 11 on input $x = (x_1, \dots, x_N)$. Then if $i = N$, $\frac{\partial S_W(x)}{\partial x_i} = \text{sign}(x_N - y_N)$. Let M be the smallest index such that $y_M = k_M c_M$, where k_i is defined in*

Algorithm 11. There exists a subgradient $g \in \partial S_W(x)$ such that

$$g_i = \begin{cases} -1, & x_i \leq 0 \\ 1, & x_i > k_i c_i \\ \text{sign}(x_M - y_M) \frac{c_M}{c_i}, & x_i \in (0, k_i c_i], x_M \neq k_M c_M \\ \frac{c_M}{c_i}, & x_i \in (0, k_i c_i], x_M = k_M c_M \end{cases}$$

Proof. We start with a few reductions. First, we show that by a small perturbation argument we can assume $x_M \neq k_M c_M$. Next, we show that it suffices to prove that S_W is linear on a small L_∞ ball near x . Then we go about proving the Proposition for that L_∞ ball, which is the meat of the argument.

Before we start the perturbation argument, we need a couple observations about M . First, observe that $k_i = y_i = 0$ for all $i > M$.

Next, we show that either have $M = N$, or $x_M \geq k_M c_M$. If $M \neq N$, then by inspection of the Algorithm 11, we must have $x_M \leq 0$ and $k_M = 0$ or $x_M \geq k_M c_M$. If $k_M = 0$, then we have $0 = k_M = k_{M-1} - \frac{y_{M-1}}{c_{M-1}}$. This implies $k_{M-1} c_{M-1} = y_{M-1}$, which contradicts our choice of M as the smallest index with $y_M = k_M c_M$. Therefore, we must have $x_M \geq k_M c_M$. Therefore, we must have $M = N$, or $x_M \geq k_M c_M$.

Now, we show that we may assume $x_M \neq k_M c_M$. Let $\delta > 0$. If $x_M \neq k_M c_M$, set $x_\delta = x$. Otherwise, set $x_\delta = x + \delta e_M$. By inspecting Algorithm 11, we observe that the output on x_δ is unchanged from the output on x , and M is still the smallest index such that $y_i = k_i c_i$.

We claim that it suffices to prove $g \in \partial S_W(x_\delta)$ for all δ rather than $g \in \partial S_W(x)$. To see this, observe that by 1-Lipschitzness, $|S_W(x_\delta) - S_W(x)| \leq \delta$, so that if $g \in \partial S_W(x_\delta)$, then for any w ,

$$S_W(w) \geq S_W(x_\delta) + \langle g, w - x_\delta \rangle \geq S_W(x) + \langle g, w - x \rangle - 2\delta.$$

By taking $\delta \rightarrow 0$, we see that g must be a subgradient of S_W at x if $g \in \partial S_W(x_\delta)$ for all δ . This implies that if we prove the Proposition for any x_δ , which has $x_M \neq k_M c_M$, we have proved the proposition for x .

Following this perturbation argument, for the rest of the proof we consider only the case $x_M \neq k_M c_M$.

Now, we claim that to show the Proposition, it suffices to exhibit a closed L_∞ ball B such that x is on the boundary of B and for $z \in B$, $S_W(z) = \langle g, z \rangle + F$ for some constant F . To see this, first suppose that we have such a B . Then observe that g is the derivative, and therefore a subgradient, of S_W for any point in the interior of B . Let z be in the interior of B and let w be an arbitrary point in \mathbb{R}^N . Then since g is a subgradient at z , we have $S_W(w) \geq S_W(z) + \langle g, w - z \rangle$. Further, since x is on the boundary of B (and therefore in B), $S_W(x) = S_W(z) + \langle g, x - z \rangle$. Putting these identities together:

$$\begin{aligned} S_W(w) &\geq S_W(z) + \langle g, w - z \rangle \\ &= S_W(z) + \langle g, x - z \rangle + \langle g, w - x \rangle \\ &= S_W(x) + \langle g, w - x \rangle. \end{aligned}$$

Therefore, g is a subgradient of S_W at x .

Next, we turn to identifying the particular L_∞ ball we will work with. Let

$$\begin{aligned} q &= \frac{1}{2} \min_{x_i > 0} x_i, \\ d &= \frac{1}{2} \min_{j | x_j \neq k_j c_j} \min(1/c_1, 1) |x_j - c_j k_j|, \\ h &= \min(q, d) \min(c_N, 1)/N. \end{aligned}$$

Consider the L_∞ ball given by

$$B = \{x + (\epsilon_1, \dots, \epsilon_N) \mid \epsilon_j \in [-h, 0]\}.$$

Clearly, x is on the boundary of B . Now, we proceed to show that S_W is linear on the interior of B , which will prove the Proposition by the above discussion.

Let $x' = x + \epsilon$ be an element of B . We will compute $S_W(x')$ by computing the output y' of running Algorithm 11 on x' . We will also refer to the internally generated variables k_i as k'_i to distinguish between the k s generated when computing y versus when computing y' . The overall strategy is to show that all of the conditional branches in Algorithm 11 will evaluate to the same branch on x as on x' .

Specifically we show the following claim by induction:

Claim 6.5. *for any $i < M$:*

$$\begin{aligned} y'_i &= \begin{cases} 0 & x_i \leq 0 \\ x'_i & x_i \in (0, k_i c_i] \end{cases}, \\ k'_{i+1} &= k_{i+1} + \sum_{j \leq i, x_j \in (0, k_j c_j]} -\epsilon_j / c_j, \\ k_{i+1} &\leq k'_{i+1} \leq k_{i+1} + d \frac{i}{2N}, \\ |y'_i - x'_i| &= \begin{cases} |y_i - x_i| - \epsilon_i & x_i \leq 0 \\ |y_i - x_i| & x_i \in (0, k_i c_i] \end{cases}. \end{aligned}$$

For $i = M$,

$$\begin{aligned} y'_i &= k'_i c_i, \\ k'_{i+1} &= 0, \\ |y'_i - x'_i| &= |y_i - x_i| + \text{sign}(x_i - y_i) \epsilon_M + \sum_{j < M \mid x_j \in (0, k_j c_j]} c_M \epsilon_j / c_j. \end{aligned}$$

And for $i > M$:

$$\begin{aligned} y'_i &= 0, \\ k'_{i+1} &= 0, \\ |y'_i - x'_i| &= \begin{cases} |y_i - x_i| - \epsilon_i & x_i \leq 0 \\ |y_i - x_i| + \epsilon_i & x_i > 0 \end{cases}. \end{aligned}$$

First we do the base case. Observe that $k'_1 = k_1$. Then we consider three cases, either $x_1 \leq 0$, $x_1 \in (0, k_1 c_1]$, or $x_1 > k_1 c_1$. These cases correspond to $y_1 = 0$, $y_1 = x_1$, or $y_1 = k_1 c_1$.

Case 1 ($x_1 \leq 0$): Since $\epsilon_1 \leq 0$, we have $x'_1 = x_1 + \epsilon_1 \leq 0$. Therefore, by inspecting the condition blocks in Algorithm 11, $y'_1 = y_1 = 0$ and $k'_2 = k_2$.

Case 2 ($x_1 \in (0, k_1 c_1]$): Since $x_1 > 0$, we have $|\epsilon_1| \leq q \leq x_1/2$. Therefore, $x'_1 > 0$. Since $\epsilon_1 \leq 0$, $x'_1 \leq x_1 \leq k_1 c_1 = k'_1 c_1$ so that $x'_1 \in (0, k'_1 c_1]$. This implies $y'_1 = x'_1$ and

$$\begin{aligned} k'_2 &= k'_1 - \frac{x'_1}{c_1} \\ &= k_1 - \frac{x_1 + \epsilon_1}{c_1} \\ &= k_2 - \frac{\epsilon_1}{c_1}. \end{aligned}$$

Case 3 ($x_1 > k_1 c_1$): In this last case, observe that $|\epsilon_1| < d \leq (x_1 - k_1 c_1)/2$ so that $x_1 \geq x'_1 > k_1 c_1 = k'_1 c_1$. This implies $y'_1 = k'_1 c_1 = k_1 c_1$ and $k'_2 = 0$.

The values for $|y'_1 - x'_1|$ can also be checked via the casework. First, suppose $1 = M$. Then we must have $x_1 > k_1 c_1$ (because we assume $x_M \neq k_M c_M$ by our perturbation argument). Therefore, $y_1 = y'_1 = k_1 c_1$ and the base case is true.

When $1 < M$, then we consider the cases $x_1 \leq 0$ and $x_1 \in (0, k_1 c_1]$. The case $x_1 > k_1 c_1$ does not occur because $1 < M$. When $x_1 \leq 0$, then by the above casework we must have $x'_1 \leq 0$ and $y'_1 = y_1 = 0$. Therefore,

$$|y'_1 - x'_1| = |x'_1| = |x_1| + |\epsilon_1| = |y_1 - x_1| - \epsilon_1,$$

where we have used $\epsilon_1 \leq 0$ to conclude $|x'_1| = |x_1| + |\epsilon_1|$.

When $x_1 \in (0, k_1 c_1]$, we have $y_1 = x_1$, and by the above casework we have and $y'_1 = x'_1$. Thus $|y'_1 - x'_1| = 0 = |y_1 - x_1|$. This concludes the base case of the induction.

Now, we move on to the inductive step. Suppose the claim holds for all $j < i$. To show the claim also holds for i , we consider the three cases $i < M$, $i = M$ and $i > M$ separately:

Case 1 ($i < M$): We must consider two sub-cases, either $x_i \leq 0$, or $x_i \in (0, k_i c_i]$. The case $x_i > k_i c_i$

does not occur because $i < M$.

Case 1a ($x_i \leq 0$): In this case, we have $y_i = 0$ and $k_{i+1} = k_i$. By definition, $\epsilon_i \leq 0$ so that $x'_i \leq 0$. Then by inspection of Algorithm 11, $y'_i = 0 = y_i$ so that $k'_{i+1} = k'_i$. By the induction assumption, this implies

$$k'_{i+1} = k'_i = k_i + \sum_{j < i, x_j \in (0, k_j c_j]} -\epsilon_j / c_j = k_{i+1} + \sum_{j \leq i, x_j \in (0, k_j c_j]} -\epsilon_j / c_j.$$

Also, $k'_{i+1} = k'_i \geq k_i = k_{i+1}$ and also

$$|k'_{i+1} - k_{i+1}| = |k'_i - k_i| \leq d \frac{i-1}{N} \leq d \frac{i}{N}.$$

Finally, since $y'_i = 0 = y_i$ and $x_i, x'_i \leq 0$, we have

$$|y'_i - x'_i| = |x'_i| = -x'_i = -x_i - \epsilon_i = |x_i| - \epsilon_i = |y_i - x_i| - \epsilon_i.$$

Thus all parts of the claim continue to hold.

Case 1b ($x_i \in (0, k_i c_i]$): In this case we show that $x'_i \in (0, k'_i c_i]$. Observe that $y_i = x_i$ and $k_{i+1} = k_i - x_i / c_i$. By definition again, $\epsilon_i \leq 0$, and also $|\epsilon_i| \leq q \leq x_i / 2$, so that $x'_i > 0$. Finally, since $k'_i \geq k_i$,

$$x'_i \leq x_i \leq c_i k_i \leq c_i k'_i.$$

Therefore, $x'_i \in (0, k'_i c_i]$ so that $y'_i = x'_i$ and

$$\begin{aligned} k'_{i+1} &= k'_i - x'_i / c_i \\ &= k_i + (k'_i - k_i) - x_i / c_i - \epsilon_i / c_i \\ &= k_{i+1} + (k'_i - k_i) - \epsilon_i / c_i \\ &= k_{i+1} + \sum_{j \leq i, x_j \in (0, k_j c_j]} -\epsilon_j / c_j, \end{aligned}$$

where the last equality uses the induction assumption. Now, since $\epsilon_j \leq 0$ for all j , this implies $k'_{i+1} \geq k_{i+1}$. Further, $|\epsilon_i / c_i| \leq d c_N / (N c_i) \leq d / N$ and by the inductive assumption, $|k'_i - k_i| \leq d \frac{i-1}{N}$ so that $|k'_{i+1} - k_{i+1}| \leq d \frac{i}{N}$ as desired. Finally, since $y'_i = x'_i$ and $y_i = x_i$, $|y'_i - x'_i| = 0 = |y_i - x_i|$.

Case 2 ($i = M$): First we show that $y'_i = k'_i c_i$, which implies $k'_{i+1} = 0$, and then we prove the expression for $|y'_i - x'_i|$. Since $x_M \neq k_M c_M$, we must have either $x_i > k_i c_i$ or $M = N$.

If $M = N$, then the claim $y'_i = k'_i c_i$ is immediate by inspection of Algorithm 11. So suppose $x_i > k_i c_i$. By the inductive assumption, $k'_i \leq k_i + d \frac{i}{N} \leq k_i + d$. Now, we observe that $d \leq \frac{1}{2c_1} (x_i - c_i k_i) \leq$

$\frac{1}{2c_i}(x_i - c_i k_i)$, which implies

$$\begin{aligned} c_i k'_i &\leq c_i k_i + c_i d \\ &\leq c_i k_i + (x_i - c_i k_i)/2 \\ &\leq x_i - (x_i - c_i k_i)/2. \end{aligned}$$

Next, observe that $d \leq \frac{1}{2}(x_i - c_i k_i)$ to conclude

$$\begin{aligned} c_i k'_i &\leq x_i - (x_i - c_i k_i)/2 \\ &\leq x_i - d \\ &\leq x_i - h \\ &\leq x'_i. \end{aligned}$$

Therefore, $x'_i \geq k'_i c_i$, so that $y'_i = c_i k'_i$.

It remains to compute $|y'_i - x'_i|$. By the induction assumption, we have

$$k'_i = k_i + \sum_{j < i, x_j \in (0, k_j c_j]} -\epsilon_j / c_j.$$

Therefore,

$$x'_i - y'_i = x_i + \epsilon_M - y_i + c_M \sum_{j < i, x_j \in (0, k_j c_j]} \epsilon_j / c_j. \quad (6.6)$$

Observe that $\epsilon_M + c_M \sum_{j < i, x_j \in (0, k_j c_j]} \epsilon_j / c_j \leq 0$ since $\epsilon_i \leq 0$ for all $i \leq M$. Now, since $c_M \leq c_j$ for $j \leq M$, we have

$$\left| \epsilon_M + c_M \sum_{j < i, x_j \in (0, k_j c_j]} \epsilon_j / c_j \right| \leq Nh \leq d.$$

Now, since $x_M \neq x_M k_M$, and $i = M$, we have $d \leq \frac{|x_i - c_i k_i|}{2}$ by definition so that

$$\left| \epsilon_M + c_M \sum_{j < i, x_j \in (0, k_j c_j]} \epsilon_j / c_j \right| \leq |x_i - c_i k_i|/2 = \frac{|x_i - y_i|}{2}.$$

Now, recalling equation (6.6) we have

$$\begin{aligned} \text{sign}(x'_i - y'_i) &= \text{sign} \left(x_i - y_i + \left[\epsilon_M + c_M \sum_{j < i, x_j \in (0, k_j c_j]} \epsilon_j / c_j \right] \right) \\ &= \text{sign}(x_i - y_i), \end{aligned}$$

where in the last line we have used $\left| \epsilon_M + c_M \sum_{j < i, x_j \in (0, k_j c_j]} \epsilon_j / c_j \right| \leq \frac{|x_i - y_i|}{2}$. Therefore, we have

$$\begin{aligned} |x'_i - y'_i| &= \text{sign}(x'_i - y'_i)(x'_i - y'_i) \\ &= \text{sign}(x_i - y_i) \left(x_i - y_i + \epsilon_M + c_M \sum_{j < i, x_j \in (0, k_j c_j]} \epsilon_j / c_j \right) \\ &= |x_i - y_i| + \text{sign}(x_i - y_i) \left(\epsilon_M + c_M \sum_{j < i, x_j \in (0, k_j c_j]} \epsilon_j / c_j \right). \end{aligned}$$

Case 3 ($i > M$):

Since $k'_i = 0$ by inductive hypothesis, we must have $y'_i = 0$ as desired. Further, observe that as observed in the beginning of the proof, $k_i = 0$ for all $i > M$ as well so that we have $y_i = 0$. Finally, if $x_i > 0$, we have $x_i + \epsilon_i \geq x_i/2 > 0$ since $|\epsilon_i| \leq q \leq x_i/2$ so that $\text{sign}(x'_i) = \text{sign}(x_i)$. Therefore, we can conclude

$$|y'_i - x'_i| = |x'_i| = \begin{cases} |x_i| - \epsilon_i & x_i \leq 0 \\ |x_i| + \epsilon_i & x_i > 0 \end{cases}.$$

Since $y_i = 0$, $|x_i| = |y_i - x_i|$ and this is the desired form for $|y'_i - x'_i|$.

This concludes the induction.

From the expression for $|y'_i - x'_i|$ we see that if g is given by

$$g_i = \begin{cases} -1 & x_i \leq 0 \\ 1 & x_i > k_i c_i \\ \text{sign}(x_M - y_M) \frac{c_M}{c_i} & x_i \in (0, k_i c_i], x_M \neq k_M c_M \\ \frac{c_M}{c_i} & x_i \in (0, k_i c_i], x_M = k_M c_M \end{cases}$$

then $S_W(x + \epsilon) = S_W(x) + \langle g, \epsilon \rangle$. Finally, observe that our perturbation x_δ has the property $\text{sign}((x_\delta)_M - y_M) = 1$ if $x_M = k_M y_M$ to prove the Proposition. \square

Chapter 7

Losses With Curvature

In the previous chapters we have exclusively considered the problem of online linear optimization, in which all loss functions ℓ_t must be linear. This is an extremely important problem because any online linear optimization algorithms can be directly applied to online convex optimization problems through the use of subgradients. Since subgradients are often relatively efficient to compute, this results in fast and effective algorithms. However, one might wonder if we could obtain tighter regret bounds by leveraging some non-linearity of the losses.

In this chapter we will exploit the notions of *smoothness* and *strong convexity* to obtain asymptotic improvements in our regret bounds. In particular, while our bounds for online linear optimization are all $\tilde{O}(\sqrt{T})$, we will be able to obtain $\tilde{O}(1)$ regret for smooth or strongly convex losses. An interesting property of all the results in this section is that we will continue to only make use of subgradients, so that even though we obtain smaller regret on these special classes of curved losses, our algorithms do not actually need to be provided with any parameters characterizing this curvature.

Before describing the results, we briefly recall the basic facts about smoothness and strong convexity. A convex function ℓ is L -smooth if it is differentiable $\ell(x + \delta) \leq \ell(x) + \langle \nabla \ell(x), \delta \rangle + \frac{L}{2} \|\delta\|^2$ for any x and δ . This implies that

$$\inf_z \ell(z) \leq \ell(w) - \frac{1}{2L} \|\nabla \ell(w)\|_*^2 \quad (7.1)$$

by choosing $\delta = \operatorname{argmin} \langle \nabla \ell(w), \delta \rangle + \frac{L}{2} \|\delta\|^2$.

Recall that a convex function ℓ is μ -strongly convex if $\ell(x + \delta) \geq \ell(x) + \langle g, \delta \rangle + \frac{\mu}{2} \|\delta\|^2$ for any x, δ and $g \in \partial \ell(x)$ (strongly-convex functions do *not* need to be differentiable). Rearranging this definition implies

$$\ell(w) - \ell(\hat{w}) \leq \langle g, w - \hat{w} \rangle - \frac{\mu}{2} \|w - \hat{w}\|^2 \quad (7.2)$$

Previously (in Section 3.3) we saw that strongly convex losses are much easier to optimize than the worst-case linear losses: we can achieve $O(\log(T)/\mu)$ regret for μ -strongly convex losses, while the best we can guarantee with linear losses is $O(\sqrt{T})$. In this chapter we will demonstrate how to achieve both bounds simultaneously without knowledge of μ .

7.1 Adapting to Smoothness

In this section we will show that our prior algorithms, without further modification, actually obtain improved regret on smooth losses. The analysis in this subsection is based on well-known prior results (e.g. [56]), but we restate the basic facts here for completeness.

Our first result states that, so long as the benchmark point \hat{w} is close to the minimum of each individual ℓ_t , we obtain better than \sqrt{T} regret:

Theorem 7.1. *Suppose each ℓ_t is L -smooth with respect to some norm $\|\cdot\|$, and let*

$$Z_T(\hat{w}) = \sum_{t=1}^T \ell_t(\hat{w}) - \inf_{w \in W} \ell_t(w)$$

Suppose an online learning algorithm obtains regret

$$R_T(\hat{w}) \leq \zeta_T(\hat{w}) + \psi_T(\hat{w}) \sqrt{\sum_{t=1}^T \|g_t\|_*^2}$$

for some functions ζ_T and ψ_T . Then the same algorithm also guarantees

$$\bar{R}_T(\hat{w}) \leq 8L\psi_T(\hat{w})^2 + 2\psi_T(\hat{w})\sqrt{2LZ_T(\hat{w})} + 2\zeta_T(\hat{w})$$

This Theorem is stated in a rather general form, but we can immediately apply it to our algorithm from the previous chapter (see Section 5.2), which obtains

$$R_T(\hat{w}) \leq \epsilon + \tilde{O} \left(\|\hat{w}\| \sqrt{\sum_{t=1}^T \|g_t\|_*^2} \right)$$

and so for smooth losses guarantees

$$R_T(\hat{w}) \leq \epsilon + \tilde{O} \left(L\|\hat{w}\|^2 + \|\hat{w}\| \sqrt{LZ_T(\hat{w})} \right)$$

Notice that in the non-online case in which each ℓ_t is fixed to some constant loss ℓ , we can take $\hat{w} = \operatorname{argmin} \ell$ to get $Z_T(\hat{w}) = 0$. Thus in this case we actually obtain logarithmic regret. Further, observe that the algorithm at not point actually requires any information about the smoothness parameter L , so that it is in some sense

adapting to this second-order parameter using only first-order information!

Before we can prove Theorem 7.1, we need a small technical proposition:

Proposition 7.2. *If a, b, c and d are non-negative constants such that*

$$x \leq a\sqrt{bx + c} + d$$

Then

$$x \leq 4a^2b + 2a\sqrt{c} + 2d$$

Proof. Suppose $x \geq 2d$. Then we have

$$\begin{aligned} \frac{x}{2} &\leq a\sqrt{bx + c} \\ x^2 &\leq 4a^2bx + 4a^2c \end{aligned}$$

Now we use the quadratic formula to obtain

$$\begin{aligned} x &\leq \frac{4a^2b}{2} + \frac{\sqrt{16a^4b^2 + 16a^2c}}{2} \\ &\leq 4a^2b + 2a\sqrt{c} \end{aligned}$$

Since we assumed $x \geq 2d$ to obtain this bound, we conclude that x is at most the maximum of $4a^2b + 2a\sqrt{c}$ and $2d$, which is bounded by their sum. \square

With this in hand, we can prove the actual Theorem:

Proof of Theorem 7.1. By equation (7.1) we have

$$\inf_z \ell_t(z) \leq \ell_t(w_t) - \frac{1}{2L} \|g_t\|_\star^2$$

so that

$$\sum_{t=1}^T \|g_t\|_\star^2 \leq 2L \sum_{t=1}^T \ell_t(w_t) - \inf_z \ell_t(z) = 2L \left(Z_T(\hat{w}) + \sum_{t=1}^T \ell_t(\hat{w}) - \ell_t(\hat{w}) \right) = 2L(Z_T(\hat{w}) + R_T(\hat{w}))$$

Therefore we have

$$R_T(\hat{w}) \leq \zeta_T(\hat{w}) + \psi_T(\hat{w}) \sqrt{2LZ_T(\hat{w}) + 2LR_T(\hat{w})}$$

Now apply Proposition 7.2 to obtain

$$R_T(\hat{w}) \leq 8L\psi_T(\hat{w})^2 + 2\psi_T(\hat{w})\sqrt{2LZ_T(\hat{w})} + 2\zeta_T(\hat{w})$$

as desired. \square

Next we will prove a potentially finer-grained versions of Theorem 7.1 that applies in a stochastic setting when the gradients g_t are random variables with $\mathbb{E}[g_t|w_1, \dots, w_t] = \nabla\mathcal{L}$ for some fixed loss function \mathcal{L} :

Theorem 7.3. *Suppose ℓ_t is an random L -smooth function with $\mathbb{E}[\ell_t|H_t] = \mathcal{L}$ for some fixed L -smooth and convex loss function \mathcal{L} , where H_t is the history of an online linear optimization algorithm up to round t . Then if the online linear optimization algorithm guarantees*

$$R_T(\hat{w}) \leq \zeta_T(\hat{w}) + \psi_T(\hat{w})\sqrt{\sum_{t=1}^T \|g_t\|_*^2}$$

Then if $\hat{w} \in \operatorname{argmin} \mathcal{L}$, the algorithm also guarantees

$$\mathbb{E}[R_T(\hat{w})] \leq 2\zeta_T(\hat{w}) + 16L\psi_T(\hat{w})^2 + 2\psi_T(\hat{w})\sqrt{2\sum_{t=1}^T \sigma_t^2}$$

where $\sigma_t = \mathbb{E}[\|\nabla\ell_t(\hat{w})\|_*^2|H_t]$ is the variance of the gradient of ℓ_t at \hat{w} .

We can directly apply this result to our algorithm from the previous chapter to obtain

$$\mathbb{E}[R_T(\hat{w})] \leq \epsilon + \tilde{O}\left(L\|\hat{w}\|^2 + \|\hat{w}\|\sqrt{\sum_{t=1}^T \sigma_t^2}\right)$$

Again, the algorithm at not point requires knowledge of L .

Proof. All expectations of quantities involving subscripts in this proof are conditioned on the past history of the algorithm.

First, observe that the function $\hat{\ell}_t(w) = \ell_t(w) - \langle \nabla\ell_t(\hat{w}), w \rangle$ is convex, L -smooth, and has the property that $\nabla\hat{\ell}_t(\hat{w}) = 0$ so that $\hat{w} \in \operatorname{argmin} \hat{\ell}_t(\hat{w})$. Therefore we have

$$\begin{aligned} \mathbb{E}[\|\nabla\hat{\ell}_t(w_t)\|_*^2] &\leq \mathbb{E}[2L(\hat{\ell}_t(w_t) - \hat{\ell}_t(\hat{w}))] \\ \|\nabla\ell_t(w_t) - \nabla\ell_t(\hat{w})\|_*^2 &\leq 2L\mathbb{E}[\ell_t(w_t) - \ell_t(\hat{w}) + \langle \nabla\ell_t(\hat{w}), \hat{w} - w_t \rangle] \\ &= 2L\mathbb{E}[\ell_t(w_t) - \ell_t(\hat{w})] \end{aligned}$$

where in the last line we have used $\mathbb{E}[\nabla\ell_t(\hat{w})] = 0$. Then we have

$$\begin{aligned}\|g_t\|_*^2 &= \|g_t - \nabla\ell_t(\hat{w}) + \nabla\ell_t(\hat{w})\|_*^2 \\ &\leq 2\|g_t - \nabla\ell_t(\hat{w})\|_*^2 + 2\|\nabla\ell_t(\hat{w})\|_*^2\end{aligned}$$

where in the second line we have used triangle inequality and the fact that $(A + B)^2 \leq 2A^2 + 2B^2$. Then we can conclude

$$\begin{aligned}\mathbb{E}[R_T(\hat{w})] &\leq \mathbb{E}[\zeta_T(\hat{w}) + \psi_T(\hat{w})] \sqrt{\sum_{t=1}^T 2\|g_t - \nabla\ell_t(\hat{w})\|_*^2 + 2\|\nabla\ell_t(\hat{w})\|_*^2} \\ &\leq \zeta_T(\hat{w}) + \psi_T(\hat{w}) \sqrt{2\mathbb{E}\left[\sum_{t=1}^T \|g_t - \nabla\ell_t(\hat{w})\|_*^2 + \|\nabla\ell_t(\hat{w})\|_*^2\right]} \\ &\leq \zeta_T(\hat{w}) + \psi_T(\hat{w}) \sqrt{4L\mathbb{E}[R_T(\hat{w})] + 2\sum_{t=1}^T \sigma_t^2}\end{aligned}$$

where the second line follows from Jensen inequality. Now we again apply Proposition 7.2 to obtain

$$\mathbb{E}[R_T(\hat{w})] \leq 2\zeta_T(\hat{w}) + 16L\psi_T(\hat{w})^2 + 2\psi_T(\hat{w}) \sqrt{2\sum_{t=1}^T \sigma_t^2}$$

as desired. \square

7.1.1 Variance Reduction

In this section we will show how to use variance reduction techniques in concert with adaptive regret bounds to achieve faster convergence for stochastic smooth problems. The variance reduction technique, first pioneered in [26], is a way to produce a stochastic gradient estimate g_t with $\mathbb{E}[g_t] = \mathbb{E}[\nabla\ell_t(w_t)] = \nabla\mathcal{L}(w_t)$ for some L -smooth function \mathcal{L} such that the variance at the optimal point $\sigma_t^2 = \mathbb{E}[\|\nabla\ell_t(\hat{w}) - \nabla\mathcal{L}(\hat{w})\|^2]$ is upper bounded by $L(\mathcal{L}(v) - \mathcal{L}(\hat{w}))$ for some fixed ‘‘anchor point’’ v . This phenomenon has sparked a wealth of recent interest, with many different algorithms designed to take advantage of it [2; 54; 21; 29; 3; 4]. However, to our knowledge, ours is the first algorithm to do so via a black-box reduction, and so allows improvements to online learning algorithms to immediately imply improvements to the algorithms in this section. The results in this subsection are taken from my paper with Robert Busa-Fekete [15].

To begin, we first describe the setting and the variance reduction technique. Variance reduction requires two main assumptions:

1. We assume access to a stream of i.i.d. random loss functions ℓ_1, ℓ_2, \dots such that $\mathbb{E}[\ell_t] = \mathcal{L}$ for some function \mathcal{L} , and that each ℓ_t is a convex L -smooth function.

2. We assume that it is possible (although potentially computationally very expensive) to compute a *true* gradient $\nabla\mathcal{L}(v)$ at any desired point v .

Variance reduction is classically applied to *finite-sum* optimization problems. That is, we are interested in finding a minimizer of a function

$$\mathcal{L}(w) = \sum_{i=1}^N f_i(w) \quad (7.3)$$

This type of problem is frequently encountered in practical machine learning contexts. In particular, when performing *empirical risk minimization*, one uses the average loss on a training dataset to approximate the true population loss. In this case, training the model involves simply minimizing the loss on the training data. Since the training data is finite, this is a problem of the form (7.3).

The focus on finite-sum problems is motivated by the two assumptions required for variance reduction. In a finite-sum problem it is indeed possible to generate a stream of i.i.d. random losses by repeatedly sampling an index i at random and setting $\ell_t = f_i$. Further, we can compute $\nabla\mathcal{L}(v)$ in an (expensive) $O(N)$ operation by simply summing $\sum_{i=1}^N \nabla f_i(v)$.

Using these two assumptions, we consider the following modified functions:

$$\ell_t^v(w) = \ell_t(w) + \langle \nabla\mathcal{L}(v) - \nabla\ell_t(v), w \rangle$$

Observe that since $\mathbb{E}[\nabla\ell_t(w)] = \nabla\mathcal{L}(v)$, we still have $\mathbb{E}[\ell_t^v(w)] = \mathcal{L}(w)$. However, we now also have the following key Proposition:

Proposition 7.4. [See [26]] *Let $\hat{w} \in \operatorname{argmin} \mathcal{L}$ and let $(\sigma_t^v)^2 = \mathbb{E}[\|\nabla\ell_t^v(\hat{w})\|_*^2]$ be the variance of $\nabla\ell_t^v(\hat{w})$. Suppose $\mathbb{E}[\|\nabla\ell_t(w) - \nabla\mathcal{L}(w)\|_*^2] \leq \sigma^2$ for all t and w . Then*

$$(\sigma_t^v)^2 \leq \min(2L(\mathcal{L}(v) - \mathcal{L}(\hat{w})), 2\sigma^2)$$

Proof. The proof is a surprisingly straightforward calculation:

$$\begin{aligned} \mathbb{E}[\|\nabla\ell_t^v(\hat{w})\|_*^2] &= \mathbb{E}[\|\nabla\ell_t(\hat{w}) - \nabla\ell_t(v) + \nabla\mathcal{L}(v)\|_*^2] \\ &= \mathbb{E}[\|\nabla\ell_t(\hat{w}) - \nabla\ell_t(v) + (\nabla\mathcal{L}(\hat{w}) - \nabla\mathcal{L}(v))\|_*^2] \\ &\leq \mathbb{E}[\|\nabla\ell_t(\hat{w}) - \nabla\ell_t(v)\|_*^2] \end{aligned}$$

where here we have observed that $\mathcal{L}(\hat{w}) = 0$ and $\mathbb{E}[\nabla\ell_t(\hat{w}) - \nabla\ell_t(v)] = \nabla\mathcal{L}(\hat{w}) - \nabla\mathcal{L}(v)$, and then used the fact that $\mathbb{E}[\|A - \mathbb{E}[A]\|_*^2] \leq \mathbb{E}[\|A\|_*^2]$ for any random variable A . Then we continue as in the proof of

Theorem 7.3

$$\begin{aligned} \mathbb{E}[\|\nabla \ell_t^v(\hat{w})\|_*^2] &\leq \mathbb{E}[\|\nabla \ell_t(\hat{w}) - \nabla \ell_t(v)\|_*^2] \\ &\leq 2L\mathbb{E}[\ell_t(v) - \ell_t(\hat{w}) + \langle \nabla \ell_t(\hat{w}), \hat{w} - v \rangle] \\ &= 2L(\mathcal{L}(v) - \mathcal{L}(\hat{w})) \end{aligned}$$

Finally, observe that since $\|a + b\|_*^2 \leq 2\|a\|_*^2 + 2\|b\|_*^2$, we have

$$\begin{aligned} \mathbb{E}[\|\nabla \ell_t^v(\hat{w})\|_*^2] &= \mathbb{E}[\|\nabla \ell_t(\hat{w}) - \nabla \mathcal{L}(\hat{w}) - \nabla \ell_t(v) + \nabla \mathcal{L}(v)\|_*^2] \\ &\leq 2\mathbb{E}[\|\nabla \ell_t(\hat{w}) - \nabla \mathcal{L}(\hat{w})\|_*^2] + 2\mathbb{E}[\|\nabla \ell_t(v) - \nabla \mathcal{L}(v)\|_*^2] \\ &\leq 2\sigma^2 \end{aligned}$$

□

Algorithm 12 Variance Reduction with Online Learning

Require: Online learning algorithm \mathcal{A}

- 1: **Initialize:** Epoch lengths $0 = T_0, T_1, T_2, \dots, T_K$. Initial point v_1 .
 - 2: **for** $k = 1$ **to** K **do**
 - 3: Compute $\nabla \mathcal{L}(v_k)$.
 - 4: **for** $t = T_{0:k-1} + 1$ **to** $T_{1:k}$ **do**
 - 5: Get point w_t from \mathcal{A} .
 - 6: Get loss ℓ_t .
 - 7: Set $\hat{\ell}_t^v(w) = \ell_t(w) + \langle \nabla \mathcal{L}(v_k) - \nabla \ell_t(v_k), w \rangle$.
 - 8: Compute $g_t = \nabla \hat{\ell}_t^v(w_t)$.
 - 9: Send g_t to \mathcal{A} as the t th loss.
 - 10: **end for**
 - 11: Set $v_{k+1} = \frac{1}{T_k} \sum_{t=T_{0:k-1}+1}^{T_{1:k}} w_t$.
 - 12: **end for**
-

Theorem 7.5. Set $T = T_{0:K}$. Suppose \mathcal{A} guarantees

$$R_T(\hat{w}) \leq \zeta_T(\hat{w}) + \psi_T(\hat{w}) \sqrt{\sum_{t=1}^T \|g_t\|_*^2}$$

Then if $\hat{w} \in \operatorname{argmin} \mathcal{L}$, Algorithm 12 guarantees

$$\mathbb{E}[R_T(\hat{w})] \leq 4\zeta_T(\hat{w}) + 32L \left(1 + 2 \max_{k>1} \frac{T_k}{T_{k-1}}\right) \psi_T(\hat{w})^2 + 8\psi_T(\hat{w}) \sqrt{T_1 \sigma^2}$$

Proof. Observe that $\mathbb{E}[\ell_t^v | H_t] = \mathcal{L}$, and so by Theorem 7.3, we have:

$$\begin{aligned} \mathbb{E}[R_T(\hat{w})] &\leq 2\zeta_T(\hat{w}) + 16L\psi_T(\hat{w})^2 + 2\psi_T(\hat{w})\sqrt{2\sum_{t=1}^T\sigma_t^2} \\ &\quad \zeta_T(\hat{w}) + 16L\psi_T(\hat{w})^2 + 2\psi_T(\hat{w})\sqrt{2\sum_{k=1}^K\sum_{t=T_{0:k-1}+1}^{T_{0:k}}(\sigma_t^{v_k})^2} \end{aligned}$$

Now we apply Theorem 7.4 to control $(\sigma_t^{v_k})^2$, and then use Jensen inequality:

$$\begin{aligned} (\sigma_t^{v_k})^2 &\leq 2L(\mathcal{L}(v_k) - \mathcal{L}(\hat{w})) \\ &\leq 2L\left(\frac{1}{T_{k-1}}\sum_{t=T_{0:k-2}+1}^{T_{0:k-1}}\mathcal{L}(w_t) - \mathcal{L}(\hat{w})\right) \end{aligned}$$

Now we sum over all the $(\sigma_t^{v_k})^2$:

$$\begin{aligned} \sum_{k=1}^K\sum_{t=T_{0:k-1}+1}^{T_{0:k}}(\sigma_t^{v_k})^2 &\leq \sum_{t=1}^{T_1}(\sigma_t^{v_1})^2 + 2L\sum_{k=2}^K\sum_{t=T_{0:k-1}+1}^{T_{0:k}}\left(\frac{1}{T_{k-1}}\sum_{t=T_{0:k-2}+1}^{T_{0:k-1}}\mathcal{L}(w_t) - \mathcal{L}(\hat{w})\right) \\ &\leq \sum_{t=1}^{T_1}(\sigma_t^{v_1})^2 + 2L\sum_{k=2}^{K+1}\sum_{t=T_{0:k-1}+1}^{T_{0:k}}\left(\frac{1}{T_{k-1}}\sum_{t=T_{0:k-2}+1}^{T_{0:k-1}}\mathcal{L}(w_t) - \mathcal{L}(\hat{w})\right) \\ &\leq \sum_{t=1}^{T_1}(\sigma_t^{v_1})^2 + 2L\max_{k>1}\frac{T_k}{T_{k-1}}R_T(\hat{w}) \end{aligned}$$

From this we conclude (using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$):

$$\sqrt{2\sum_{t=1}^T\sigma_t^2} \leq \sqrt{2\sum_{t=1}^{T_1}(\sigma_t^{v_1})^2} + 2\sqrt{L\max_{k>1}\frac{T_k}{T_{k-1}}R_T(\hat{w})}$$

so now we combine our inequalities to obtain:

$$\begin{aligned} \mathbb{E}[R_T(\hat{w})] &\leq 2\zeta_T(\hat{w}) + 16L\psi_T(\hat{w})^2 + 2\psi_T(\hat{w})\sqrt{2\sum_{t=1}^T\sigma_t^2} \\ &\leq \zeta_T(\hat{w}) + 16L\psi_T(\hat{w})^2 + 2\psi_T(\hat{w})\sqrt{2\sum_{t=1}^{T_1}(\sigma_t^{v_1})^2 + 4L\max_{k>1}\frac{T_k}{T_{k-1}}R_T(\hat{w})} \end{aligned}$$

Finally, we apply Proposition 7.2:

$$\begin{aligned}\mathbb{E}[R_T(\hat{w})] &\leq 4\zeta_T(\hat{w}) + 32L \left(1 + 2 \max_{k>1} \frac{T_k}{T_{k-1}}\right) \psi_T(\hat{w})^2 + 4\psi_T(\hat{w}) \sqrt{2 \sum_{t=1}^{T_1} (\sigma_t^{v_1})^2} \\ &\leq 4\zeta_T(\hat{w}) + 32L \left(1 + 2 \max_{k>1} \frac{T_k}{T_{k-1}}\right) \psi_T(\hat{w})^2 + 8\psi_T(\hat{w}) \sqrt{T_1 \sigma^2}\end{aligned}$$

□

In order to use this theorem, we need to specify a particular algorithm, as well as choices for T_1, T_2, \dots . Observe that the total number of times we compute a “true gradient” $\nabla \mathcal{L}(v_k)$ is K , so we might want T_k to increase rapidly so as to maximize the number of cheap “stochastic gradient steps” inside the inner loop of Algorithm 12 we make for every expensive true gradient computation. However, since Theorem 7.5 makes use of the quantity $\max_{k>1} \frac{T_k}{T_{k-1}}$, we don’t want the T_k to grow super-exponentially. To satisfy both requirements, we set $T_k = 2^k$, and obtain the following Corollary:

Corollary 7.6. *Set $T_k = 2^k$, and set $T = T_{0:K}$. Suppose \mathcal{A} guarantees*

$$R_T(\hat{w}) \leq \zeta_T(\hat{w}) + \psi_T(\hat{w}) \sqrt{\sum_{t=1}^T \|g_t\|_*^2}$$

and we set $T_k = 2^k$. Suppose ζ_T and ψ_T are at most logarithmic in T , and σ is an upper bound on the variance of $\nabla \ell_t(w)$. Finally, suppose we are solving a finite sum problem $\mathcal{L} = \sum_{i=1}^N f_t$, and set $\bar{w} = \frac{1}{T} \sum_{t=1}^T w_t$. then Algorithm 12 guarantees

$$\mathbb{E}[\mathcal{L}(\bar{w}) - \mathcal{L}(\hat{w})] \leq \text{err}$$

in $\tilde{O}\left(N + \frac{1}{\text{err}}\right)$ evaluations of gradients $\nabla f_i(w)$, where \tilde{O} hides factors depending on $\|\hat{w}\|$ and σ , and logarithmic factors in $\frac{1}{\text{err}}$.

This corollary (possibly up to log factors), matches the best non-accelerated rates for variance-reduced methods. However, it has the advantage of not requiring any knowledge of L in order to achieve the desired performance. Previous algorithms that achieve this convergence rate (e.g. [4]), require particular settings that depend on L , and will fail to converge if these are set incorrectly.

Proof. First, by Jensen’s inequality we have

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\bar{w}) - \mathcal{L}(\hat{w})] &\leq \frac{1}{T} \mathbb{E} \left[\sum_{t=1}^T \mathcal{L}(w_t) - \mathcal{L}(\hat{w}) \right] \\ &\leq \frac{1}{T} \mathbb{E}[R_T(\hat{w})]\end{aligned}$$

Now by Theorem 7.5, we have

$$R_T(\hat{w}) \leq 4\zeta_T(\hat{w}) + 160L\psi_T(\hat{w})^2 + 8\psi_T(\hat{w})\sqrt{2\sigma^2}$$

Thus since ζ_T and ψ_T are logarithmic in T , we need $T = \tilde{O}\left(\frac{1}{\text{err}}\right)$ in order to guarantee $\mathbb{E}[\mathcal{L}(\bar{w}) - \mathcal{L}(\hat{w})] \leq \text{err}$.

Next, observe that we can compute $\nabla\mathcal{L}(v_k)$ in N gradient evaluations. Then, since $T_k = 2^k$, we have $K = O(\log(T))$. Thus we have a total of $O(N \log(T))$ gradient evaluations needed to compute $\nabla\mathcal{L}(v_k)$ for all k . Since each computation of g_t requires an additional 2 gradient evaluations, we have a total of $O(N \log(T) + T)$ gradient evaluations, which is $\tilde{O}\left(N + \frac{1}{\text{err}}\right)$ as desired. \square

7.2 Adapting to Strong Convexity

In this section, we present a black-box reduction to make a generic online learning algorithm over a Banach space adaptive to strong convexity of the losses. Given a set W of diameter $D = \sup_{x,y \in W} \|x - y\|$, our reduction obtains $O(\log(TD)^2/\mu)$ regret on online μ -strongly convex optimization problems, but still guarantees $O(\log(TD)^2 D \sqrt{T})$ regret for online linear optimization problems, both of which are only log factors away from the optimal guarantees. Critically, our algorithm again makes use *only of first order information*, never requiring any knowledge of μ ! The material in this section is taken from my two papers [14; 16].

The reduction operates by achieving a regret bound like

$$\sum_{t=1}^T \langle g_t, w_t - \hat{w} \rangle \leq \sqrt{\sum_{t=1}^T \|w_t - \hat{w}\|^2 \|g_t\|_*^2} \quad (7.4)$$

This can be viewed a weaker version of the regret bound $R_T(\hat{w}) \leq \sqrt{\sum_{t=1}^T \langle g_t, w_t - \hat{w} \rangle^2}$ achieved by the MetaGrad algorithm [59]. Although this regret bound is weaker, MetaGrad requires $O(d^2)$ time per update to achieve its bound in a d -dimensional space, while our algorithm requires the same $O(d)$ runtime as gradient descent.

Before we describe how to achieve the regret bound (7.4), we show how it implies the desired adaptivity:

Proposition 7.7. *Suppose an online linear optimization algorithm guarantees the regret bound*

$$\sum_{t=1}^T \langle g_t, w_t - \hat{w} \rangle \leq A + B \sqrt{\sum_{t=1}^T \|w_t - \hat{w}\|^2 \|g_t\|_*^2}$$

for some A and B . Then when each g_t is a subgradient of a μ -strongly convex loss ℓ_t , we have

$$R_T(\hat{w}) = \sum_{t=1}^T \ell_t(w_t) - \ell_t(\hat{w}) \leq A + \frac{B^2 G_{\max}^2}{2\mu}$$

Proof. Recall that μ -strong convexity implies $\ell_t(w_t) - \ell_t(\hat{w}) \leq \langle g_t, w_t - \hat{w} \rangle - \frac{\mu}{2} \|w_t - \hat{w}\|^2$. Therefore we have

$$\begin{aligned} R_T(\hat{w}) &\leq \sum_{t=1}^T \langle g_t, w_t - \hat{w} \rangle - \frac{\mu}{2} \|w_t - \hat{w}\|^2 \\ &\leq A + B \sqrt{\sum_{t=1}^T \|w_t - \hat{w}\|^2 \|g_t\|_*^2} - \sum_{t=1}^T \frac{\mu}{2} \|w_t - \hat{w}\|^2 \\ &\leq A + B G_{\max} \sqrt{\sum_{t=1}^T \|w_t - \hat{w}\|^2} - \sum_{t=1}^T \frac{\mu}{2} \|w_t - \hat{w}\|^2 \\ &\leq \sup_X A + B G_{\max} \sqrt{X} - \frac{\mu}{2} X \\ &\leq A + \frac{B^2 G_{\max}^2}{2\mu} \end{aligned}$$

□

Thus if we could bound A by $O(\log(T))$ and B by $O(\log^2(T))$, we would obtain our target results for adapting to μ -strong convexity while maintaining \sqrt{T} regret in the non-strongly convex case.

In order to come up with an algorithm that achieves the bound (7.4), we interpret it as the square root of $\mathbb{E}[\|w - \hat{w}\|^2]$, where w is a random variable that takes on value w_t with probability proportional to $\|g_t\|_*^2$. This allows us to use the bias-variance decomposition to write (7.4) as:

$$R_T(\hat{w}) \leq \tilde{O} \left(\|\hat{w} - \bar{w}\| \sqrt{\|g\|_{*1:T}^2} + \sqrt{\sum_{t=1}^T \|g_t\|_*^2 \|w_t - \bar{w}\|^2} \right) \quad (7.5)$$

where $\bar{w} = \frac{\sum_{t=1}^T \|g_t\|_*^2 w_t}{\|g\|_{*1:T}^2}$. Now, our previous algorithm for unconstrained online linear optimization obtained regret $R_T(\hat{w}) = \tilde{O}(\|\hat{w}\| \sqrt{\|g\|_{*1:T}^2})$ simultaneously for all $\hat{w} \in W$. Thus, if we know \bar{w} ahead of time, we could translate the predictions of this algorithm by \bar{w} to obtain $R_T(\hat{w}) \leq \tilde{O}(\|\hat{w} - \bar{w}\| \sqrt{\|g\|_{*1:T}^2})$, the bias term of (7.5). We do not actually know \bar{w} , but we can estimate it over time. Errors in the estimation procedure will cause us to incur the variance term of (7.5).

Our overall strategy is very simple: we set $w_t = \hat{w}_t + \bar{w}_{t-1}$ where \hat{w}_t is the t^{th} output of an online learning algorithm, and \bar{w}_{t-1} is (approximately) a weighted average of the previous vectors w_1, \dots, w_{t-1} with the weight of w_t equal to $\|g_t\|_*^2$. This \bar{w}_t offset can be viewed as a kind of momentum term that accelerates us towards optimal points when the losses strongly convex, but has very little effect when the losses are general

convex functions. The full psuedocode is given in Algorithm 13, and the analysis is provided in Theorem 7.8.

Algorithm 13 Adapting to Curvature

Require: Online learning algorithm \mathcal{A}

- 1: **Initialize:** W , a convex closed set in a reflexive Banach space, \bar{x}_0 an arbitrary point in W
 - 2: **for** $t = 1$ **to** T **do**
 - 3: Get point w_t from \mathcal{A}
 - 4: Set $z_t = w_t + \bar{x}_{t-1}$
 - 5: Pick $x_t \in \Pi_W(z_t)$
 - 6: Play x_t , receive subgradient $g_t \in \partial \ell_t(x_t)$
 - 7: Set $\tilde{g}_t \in g_t + \|g_t\|_* \partial S_W(z_t)$
 - 8: Set $\bar{x}_t = \frac{\bar{x}_0 + \sum_{i=1}^t \|\tilde{g}_i\|_*^2 x_i}{1 + \sum_{i=1}^t \|\tilde{g}_i\|_*^2}$
 - 9: Send \tilde{g}_t so \mathcal{A} as the t th subgradient
 - 10: **end for**
-

Theorem 7.8. *Let \mathcal{A} be an online linear optimization algorithm that outputs w_t in response to g_t . Suppose W is a convex closed set of diameter D . Suppose \mathcal{A} guarantees for all t and \hat{v} :*

$$\sum_{i=1}^t \langle \tilde{g}_i, w_i - \hat{v} \rangle \leq \epsilon + \|\hat{v}\| A \sqrt{\sum_{i=1}^t \|\tilde{g}_i\|_*^2 \left(1 + \ln \left(\frac{\|\hat{v}\|^2 t^C}{\epsilon^2} + 1\right)\right)} + B \|\hat{v}\| \ln \left(\frac{\|\hat{v}\| t^C}{\epsilon} + 1\right),$$

for constants A , B and C and ϵ independent of t . Then for all $\hat{w} \in W$, Algorithm 13 guarantees

$$R_T(\hat{w}) \leq \sum_{t=1}^T \langle g_t, x_t - \hat{w} \rangle \leq O \left(\sqrt{V_T(\hat{w}) \ln \frac{TD}{\epsilon} \ln(T)} + \ln \frac{DT}{\epsilon} \ln(T) + \epsilon \right),$$

where $V_T(\hat{w}) := \|\bar{x}_0 - \hat{w}\|^2 + \sum_{t=1}^T \|\tilde{g}_t\|_*^2 \|x_t - \hat{w}\|^2 \leq D^2 + \sum_{t=1}^T \|g_t\|_*^2 \|x_t - \hat{w}\|^2$.

7.3 Conclusions

In this chapter we applied online linear optimization algorithms to losses with some curvature structure. We showed that even though online linear optimization algorithms only have access to first order information, it is nevertheless possible to adapt to second-order information in a number of ways. We can adapt to smoothness when the benchmark point always has small loss, and in the case of finite-sum problems we can actually equal the convergence rate of the SVRG algorithm without knowing any smoothness parameters. When the losses are strongly convex, we can adapt to the strong convexity parameter to obtain logarithmic regret, without sacrificing $\tilde{O}(\sqrt{T})$ regret in the non-strongly convex case. In all cases, we obtain results that adapt to second-order parameters using only first-order information.

Appendix

7.A Proof of Theorem 7.8

We re-state Theorem 7.8 below for reference:

Theorem 7.8. *Let \mathcal{A} be an online linear optimization algorithm that outputs w_t in response to g_t . Suppose W is a convex closed set of diameter D . Suppose \mathcal{A} guarantees for all t and \hat{v} :*

$$\sum_{i=1}^t \langle \tilde{g}_i, w_i - \hat{v} \rangle \leq \epsilon + \|\hat{v}\| A \sqrt{\sum_{i=1}^t \|\tilde{g}_i\|_*^2 \left(1 + \ln \left(\frac{\|\hat{v}\|^2 t^C}{\epsilon^2} + 1\right)\right)} + B \|\hat{v}\| \ln \left(\frac{\|\hat{v}\| t^C}{\epsilon} + 1\right),$$

for constants A , B and C and ϵ independent of t . Then for all $\hat{w} \in W$, Algorithm 13 guarantees

$$R_T(\hat{w}) \leq \sum_{t=1}^T \langle g_t, x_t - \hat{w} \rangle \leq O \left(\sqrt{V_T(\hat{w}) \ln \frac{TD}{\epsilon} \ln(T)} + \ln \frac{DT}{\epsilon} \ln(T) + \epsilon \right),$$

where $V_T(\hat{w}) := \|\bar{x}_0 - \hat{w}\|^2 + \sum_{t=1}^T \|\tilde{g}_t\|_*^2 \|x_t - \hat{w}\|^2 \leq D^2 + \sum_{t=1}^T \|g_t\|_*^2 \|x_t - \hat{w}\|^2$.

Proof. For any t , consider the random vector X_t that takes value x_i for $i \leq t$ with probability proportional to $\|\tilde{g}_i\|_*^2$ and value \bar{x}_0 with probability proportional to 1. Make the following definitions/observations:

1. $Z_t := 1 + \sum_{i=1}^t \|\tilde{g}_i\|_*^2$ for all t , so that

$$V_T(\hat{w}) = \|\bar{x}_0 - \hat{w}\|^2 + \sum_{t=1}^T \|\tilde{g}_t\|_*^2 \|x_t - \hat{w}\|^2 = Z_T \mathbb{E}[\|X_T - \hat{w}\|^2].$$

2. $\bar{x}_T = \mathbb{E}[X_T] = \frac{\bar{x}_0 + \sum_{i=1}^T \|\tilde{g}_i\|_*^2 x_i}{1 + \sum_{i=1}^T \|\tilde{g}_i\|_*^2}$.

3. $\sigma_t^2 := \frac{\|\bar{x}_t - \bar{x}_0\|^2 + \sum_{i=1}^t \|\tilde{g}_i\|_*^2 \|x_i - \bar{x}_t\|^2}{Z_t}$ so that $\sigma_t^2 = \mathbb{E}[\|X_t - \bar{x}_t\|^2]$, and $\sigma_T^2 Z_T = \|\bar{x}_0 - \bar{x}_T\|^2 + \sum_{t=1}^T \|\tilde{g}_t\|_*^2 \|x_t - \bar{x}_T\|^2$.

To prove the theorem, we are going to show for any $\dot{w} \in W$,

$$R_T(\dot{w}) \leq O \left[\sqrt{Z_T \|\dot{w} - \bar{x}_T\|^2 \ln \frac{TD}{\epsilon^2} + \ln \frac{DT}{\epsilon} \ln(T)} + \sqrt{Z_T \sigma_T^2 \ln \frac{TD}{\epsilon} \log(T)} \right], \quad (7.6)$$

which implies the desired bound by a bias-variance decomposition: $Z_T \|\dot{w} - \bar{x}_T\|^2 + Z_T \sigma_T^2 = Z_T \mathbb{E}[\|X_T - \dot{w}\|^2] = V_T(\dot{w})$.

Observe that, by triangle inequality and the definition of dual norm, $\langle g_t, z \rangle + \|g_t\|_* S_W(z) \geq \langle g_t, x \rangle$ for all z and $x \in \Pi_W(z)$, with equality when $z \in W$. Hence, we have

$$\langle g_t, x_t - \dot{w} \rangle \leq \langle g_t, z_t - \dot{w} \rangle + \|g_t\|_* S_W(z_t) - \|g_t\|_* S_W(\dot{w}) \leq \langle \tilde{g}_t, z_t - \dot{w} \rangle, \quad (7.7)$$

for all $\dot{w} \in W$, where in the last inequality we used Proposition 5.5. Using this inequality with the regret guarantee of \mathcal{A} , we have

$$\begin{aligned} R_T(\dot{w}) &\leq \sum_{t=1}^T \langle g_t, x_t - \dot{w} \rangle \leq \sum_{t=1}^T \langle \tilde{g}_t, z_t - \dot{w} \rangle = \sum_{t=1}^T \langle \tilde{g}_t, w_t - (\dot{w} - \bar{x}_T) \rangle + \sum_{t=1}^T \langle \tilde{g}_t, \bar{x}_{t-1} - \bar{x}_T \rangle \\ &\leq O \left(\|\dot{w} - \bar{x}_T\| \sqrt{\sum_{t=1}^T \|\tilde{g}_t\|_*^2 \ln \frac{\|\dot{w} - \bar{x}_T\| T}{\epsilon^2}} + \|\dot{w} - \bar{x}_T\| \ln \frac{\|\dot{w} - \bar{x}_T\| T}{\epsilon} \right) + \epsilon + \sum_{t=1}^T \langle \tilde{g}_t, \bar{x}_{t-1} - \bar{x}_T \rangle \\ &= O \left(\sqrt{Z_T \|\dot{w} - \bar{x}_T\|^2 \ln \frac{DT}{\epsilon^2}} + D \ln \frac{DT}{\epsilon} \right) + \epsilon + \sum_{t=1}^T \langle \tilde{g}_t, \bar{x}_{t-1} - \bar{x}_T \rangle. \end{aligned}$$

Note that the first term is exactly what we want, so we only have to upper bound the second one. This is readily done through Lemma 7.9 that immediately gives us the stated result. \square

Lemma 7.9. *Under the hypotheses of Theorem 7.8, we have*

$$\sum_{t=1}^T \langle \tilde{g}_t, \bar{x}_{t-1} - \bar{x}_T \rangle \leq M \sqrt{Z_T} \sigma_T \sqrt{1 + \ln Z_T} + K(1 + \ln Z_T),$$

where $M = A \sqrt{1 + \ln \left(\frac{2D^2 T^C}{\epsilon^2} + 3T^C \right)}$ and $K = 1 + B \ln \left(\frac{\sum_{i=1}^T \|g_i\|_* D T^C}{\epsilon} + 2T^C \right)$.

Proof. We have that

$$\sum_{i=1}^t \langle \tilde{g}_i, \bar{x}_{i-1} - \bar{x}_t \rangle - \sum_{i=1}^{t-1} \langle \tilde{g}_i, \bar{x}_{i-1} - \bar{x}_{t-1} \rangle = \left\langle \sum_{i=1}^t \tilde{g}_i, \bar{x}_{t-1} - \bar{x}_t \right\rangle.$$

The telescoping sum gives us

$$\sum_{t=1}^T \langle \tilde{g}_t, \bar{x}_{t-1} - \bar{x}_T \rangle = \sum_{t=1}^T \left\langle \sum_{i=1}^t \tilde{g}_i, \bar{x}_{t-1} - \bar{x}_t \right\rangle \leq \sum_{t=1}^T \left\| \sum_{i=1}^t \tilde{g}_i \right\|_{\star} \|\bar{x}_{t-1} - \bar{x}_t\|.$$

So in order to bound $\sum_{t=1}^T \langle \tilde{g}_t, \bar{x}_{t-1} - \bar{x}_T \rangle$, it suffices to bound $\left\| \sum_{i=1}^t \tilde{g}_i \right\|_{\star} \|\bar{x}_{t-1} - \bar{x}_t\|$ by a sufficiently small value. First we will tackle $\left\| \sum_{i=1}^t \tilde{g}_i \right\|_{\star}$. To do this we recall our regret bound for \mathcal{A} . Analogous to (7.7), we have

$$\begin{aligned} \langle g_t, x_t \rangle &\geq \langle g_t, z_t \rangle + \|g_t\|_{\star} S_W(z_t) + \langle \tilde{g}_t, x_t - z_t \rangle \\ \langle \tilde{g}_t, z_t \rangle &\geq \langle g_t, z_t - x_t \rangle + \|g_t\|_{\star} \|z_t - x_t\| + \langle \tilde{g}_t, x_t \rangle \\ &\geq \langle \tilde{g}_t, x_t \rangle. \end{aligned}$$

Therefore, for any $X \in \mathbb{R}$ we have:

$$\begin{aligned} &\sum_{i=1}^t -\|\tilde{g}_i\|_{\star} D + \left\| \sum_{i=1}^t \tilde{g}_i \right\|_{\star} X \\ &\leq \sum_{i=1}^t \langle \tilde{g}_i, x_i - \bar{x}_{i-1} \rangle + \left\| \sum_{i=1}^t \tilde{g}_i \right\|_{\star} X \\ &\leq \sum_{i=1}^t \langle \tilde{g}_i, z_i - \bar{x}_{i-1} \rangle + \left\| \sum_{i=1}^t \tilde{g}_i \right\|_{\star} X \\ &= \sum_{i=1}^t \langle \tilde{g}_i, w_i \rangle + \left\| \sum_{i=1}^t \tilde{g}_i \right\|_{\star} X \\ &\leq \epsilon + |X|A \sqrt{\sum_{i=1}^t \|\tilde{g}_i\|_{\star}^2 \left(1 + \ln \left(\frac{|X|^{2t^C}}{\epsilon^2} + 1\right)\right)} + B|X| \ln \left(\frac{|X|t^C}{\epsilon} + 1\right), \end{aligned}$$

where in the first inequality we have used the fact that the domain is bounded.

Dividing by X and solving for $\left\| \sum_{i=1}^t \tilde{g}_i \right\|_{\star}$, we have

$$\left\| \sum_{i=1}^t \tilde{g}_i \right\|_{\star} \leq \frac{\epsilon}{X} + A \sqrt{\sum_{i=1}^t \|\tilde{g}_i\|_{\star}^2 \left(1 + \ln \left(\frac{|X|^{2t^C}}{\epsilon^2} + 1\right)\right)} + B \ln \left(\frac{|X|t^C}{\epsilon} + 1\right) + \frac{\sum_{i=1}^t \|\tilde{g}_i\|_{\star} D}{X}.$$

Set $X = \epsilon + \sum_{i=1}^t \|\tilde{g}_i\|_* D$ and overapproximate to conclude:

$$\begin{aligned} \left\| \sum_{i=1}^t \tilde{g}_i \right\|_* &\leq 1 + A \sqrt{\sum_{i=1}^t \|\tilde{g}_i\|_*^2 \left(1 + \ln \left(\frac{2D^2 \left(\sum_{i=1}^t \|\tilde{g}_i\|_* \right)^2 t^C}{\epsilon^2} + 3t^C \right) \right)} \\ &\quad + B \ln \left(\frac{\sum_{i=1}^t \|\tilde{g}_i\|_* D t^C}{\epsilon} + 2t^C \right) \\ &\leq M \sqrt{\sum_{i=1}^t \|\tilde{g}_i\|_*^2} + K. \end{aligned}$$

With this in hand, we have

$$\sum_{t=1}^T \langle \tilde{g}_t, \bar{x}_{t-1} - \bar{x}_T \rangle \leq \sum_{t=1}^T \left\| \sum_{i=1}^t \tilde{g}_i \right\|_* \|\bar{x}_{t-1} - \bar{x}_t\| \leq M \sum_{t=1}^T \sqrt{\sum_{i=1}^t \|\tilde{g}_i\|_*^2} \|\bar{x}_{t-1} - \bar{x}_t\| + K \sum_{t=1}^T \|\bar{x}_{t-1} - \bar{x}_t\|. \quad (7.8)$$

Now, we relate $\|\bar{x}_t - \bar{x}_{t-1}\|$ to $\|x_t - \bar{x}_t\|$:

$$\bar{x}_{t-1} - \bar{x}_t = \bar{x}_{t-1} - \frac{Z_{t-1} \bar{x}_{t-1} + \|\tilde{g}_t\|_*^2 x_t}{Z_t} = \frac{\|\tilde{g}_t\|_*^2}{Z_t} (\bar{x}_{t-1} - x_t) = \frac{\|\tilde{g}_t\|_*^2}{Z_t} (\bar{x}_t - x_t) + \frac{\|\tilde{g}_t\|_*^2}{Z_t} (\bar{x}_{t-1} - \bar{x}_t),$$

that implies

$$Z_t (\bar{x}_{t-1} - \bar{x}_t) = \|\tilde{g}_t\|_*^2 (x_t - \bar{x}_t) + \|\tilde{g}_t\|_*^2 (\bar{x}_{t-1} - \bar{x}_t),$$

that is

$$\bar{x}_{t-1} - \bar{x}_t = \frac{\|\tilde{g}_t\|_*^2}{Z_{t-1}} (x_t - \bar{x}_t). \quad (7.9)$$

Hence, we have

$$M \sum_{t=1}^T \sqrt{\sum_{i=1}^t \|\tilde{g}_i\|_*^2} \|\bar{x}_t - \bar{x}_{t-1}\| \leq M \sum_{t=1}^T \sqrt{Z_t} \frac{\|\tilde{g}_t\|_*^2}{Z_{t-1}} \|x_t - \bar{x}_t\|,$$

and

$$K \sum_{t=1}^T \|\bar{x}_t - \bar{x}_{t-1}\| \leq K \sum_{t=1}^T \frac{\|\tilde{g}_t\|_*^2}{Z_{t-1}} \|x_t - \bar{x}_t\| \leq KD \sum_{t=1}^T \frac{\|\tilde{g}_t\|_*^2}{Z_{t-1}}.$$

Using CauchySchwarz inequality, we have

$$M \sum_{t=1}^T \sqrt{Z_t} \frac{\|\tilde{g}_t\|_*^2}{Z_{t-1}} \|x_t - \bar{x}_t\| \leq M \sqrt{\sum_{t=1}^T \frac{\|\tilde{g}_t\|_*^2}{Z_{t-1}}} \sqrt{\sum_{t=1}^T \frac{Z_t}{Z_{t-1}} \|\tilde{g}_t\|_*^2 \|x_t - \bar{x}_t\|^2}.$$

So, putting together the last inequalities, we have

$$\sum_{t=1}^T \langle \tilde{g}_t, \bar{x}_{t-1} - \bar{x}_T \rangle \leq M \sqrt{\sum_{t=1}^T \frac{\|\tilde{g}_t\|_*^2}{Z_{t-1}}} \sqrt{\sum_{t=1}^T \frac{Z_t}{Z_{t-1}} \|\tilde{g}_t\|_*^2 \|x_t - \bar{x}_t\|^2} + KD \sum_{t=1}^T \frac{\|g_t\|_*^2}{Z_{t-1}}.$$

We now focus on the the term $\sum_{t=1}^T \frac{\|g_t\|_*^2}{Z_{t-1}}$ that is easily bounded:

$$\begin{aligned} \sum_{t=1}^T \frac{\|g_t\|_*^2}{Z_{t-1}} &= \sum_{t=1}^T \left(\frac{\|\tilde{g}_t\|_*^2}{Z_t} + \frac{\|\tilde{g}_t\|_*^2}{Z_{t-1}} - \frac{\|\tilde{g}_t\|_*^2}{Z_t} \right) \\ &\leq \sum_{t=1}^T \left(\frac{\|\tilde{g}_t\|_*^2}{Z_t} + \frac{1}{Z_{t-1}} - \frac{1}{Z_t} \right) \\ &\leq \frac{1}{Z_0} + \sum_{t=1}^T \frac{\|\tilde{g}_t\|_*^2}{Z_t} \\ &\leq \frac{1}{Z_0} + \log \frac{Z_T}{Z_0} \\ &= 1 + \ln Z_T, \end{aligned}$$

where in the last inequality we used the well-known inequality $\sum_{t=1}^T \frac{a_t}{a_0 + \sum_{i=1}^t a_i} \leq \ln(1 + \frac{\sum_{t=1}^T a_t}{a_0})$, $\forall a_t \geq 0$.

To upper bound the term $\sum_{t=1}^T \frac{Z_t}{Z_{t-1}} \|\tilde{g}_t\|_*^2 \|x_t - \bar{x}_t\|^2$, observe that

$$\begin{aligned} \sigma_T^2 Z_T &= \|\bar{x}_0 - \bar{x}_T\|^2 + \sum_{t=1}^T \|\tilde{g}_t\|_*^2 \|x_t - \bar{x}_T\|^2 \\ &= \|\bar{x}_0 - \bar{x}_T\|^2 + \sum_{t=1}^{T-1} \|\tilde{g}_t\|_*^2 \|x_t - \bar{x}_T\|^2 + \|\tilde{g}_T\|_*^2 \|x_T - \bar{x}_T\|^2 \\ &= Z_{T-1}(\sigma_{T-1}^2 + \|\bar{x}_T - \bar{x}_{T-1}\|^2) + \|\tilde{g}_T\|_*^2 \|x_T - \bar{x}_T\|^2 \\ &= Z_{T-1} \sigma_{T-1}^2 + \|\tilde{g}_T\|_*^2 \left(1 + \frac{\|\tilde{g}_T\|_*^2}{Z_{T-1}} \right) \|x_T - \bar{x}_T\|^2 \\ &= Z_{T-1} \sigma_{T-1}^2 + \|\tilde{g}_T\|_*^2 \frac{Z_T}{Z_{T-1}} \|x_T - \bar{x}_T\|^2, \end{aligned}$$

where the third equality comes from bias-variance decomposition and the fourth one comes from (7.9).

Hence, we have

$$\sum_{t=1}^T \frac{Z_t}{Z_{t-1}} \|\tilde{g}_t\|_*^2 \|x_t - \bar{x}_t\|^2 = \sum_{t=1}^T (\sigma_t^2 Z_t - \sigma_{t-1}^2 Z_{t-1}) \leq \sigma_T^2 Z_T.$$

Putting all together, we have the stated bound. \square

Bibliography

- [1] Jacob Abernethy, Peter L Bartlett, Alexander Rakhlin, and Ambuj Tewari. Optimal strategies and minimax lower bounds for online convex games. In *Proc. of the nineteenth annual conference on computational learning theory*, 2008.
- [2] Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1200–1205. ACM, 2017.
- [3] Zeyuan Allen-Zhu. Natasha 2: Faster non-convex optimization than sgd. *arXiv preprint arXiv:1708.08694*, 2017.
- [4] Zeyuan Allen-Zhu and Yang Yuan. Improved svrg for non-strongly-convex or sum-of-non-convex objectives. In *International conference on machine learning*, pages 1080–1089, 2016.
- [5] Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.
- [6] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [7] Sébastien Bubeck, Nikhil R Devanur, Zhiyi Huang, and Rad Niazadeh. Online auctions and multi-scale online learning. *arXiv preprint arXiv:1705.09700*, 2017.
- [8] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.
- [9] Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *Information Theory, IEEE Transactions on*, 50(9):2050–2057, 2004.
- [10] Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. A second-order Perceptron algorithm. *SIAM Journal on Computing*, 34(3):640–668, 2005.
- [11] Thomas M Cover. Behavior of sequential predictors of binary sequences. Technical report, Stanford Electronics Lab, 1966.
- [12] Ashok Cutkosky and Kwabena Boahen. Online learning without prior information. In Satyen Kale and Ohad Shamir, editors, *Proc. of the 2017 Conference on Learning Theory*, volume 65 of *Proc. of Machine Learning Research*, pages 643–677, Amsterdam, Netherlands, 07–10 Jul 2017. PMLR.
- [13] Ashok Cutkosky and Kwabena A Boahen. Online convex optimization with unconstrained domains and losses. In *Advances in Neural Information Processing Systems 29*, pages 748–756, 2016.

- [14] Ashok Cutkosky and Kwabena A Boahen. Stochastic and adversarial online learning without hyperparameters. In *Advances in Neural Information Processing Systems*, pages 5066–5074, 2017.
- [15] Ashok Cutkosky and Robert Busa-Fekete. Distributed stochastic optimization via adaptive stochastic gradient descent. *arXiv preprint arXiv:1802.05811*, 2018.
- [16] Ashok Cutkosky and Francesco Orabona. Black-box reductions for parameter-free online learning in banach spaces. *arXiv preprint arXiv:1802.06293*, 2018.
- [17] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *Conference on Learning Theory (COLT)*, 2010.
- [18] John Duchi. Lecture notes for statistics 311/electrical engineering 377, February 2016. URL https://stanford.edu/class/stats311/Lectures/full_notes.pdf.
- [19] Dylan J Foster, Alexander Rakhlin, and Karthik Sridharan. Adaptive online learning. In *Advances in Neural Information Processing Systems 28*, pages 3375–3383. 2015.
- [20] Dylan J Foster, Satyen Kale, Mehryar Mohri, and Karthik Sridharan. Parameter-free online learning via model selection. In *Advances in Neural Information Processing Systems*, pages 6022–6032, 2017.
- [21] Roy Frostig, Rong Ge, Sham M Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. In *Conference on learning theory*, pages 728–763, 2015.
- [22] Petr Hájek, Vicente Montesinos Santalucía, Jon Vanderwerff, and Václav Zizler. *Biorthogonal systems in Banach spaces*. Springer Science & Business Media, 2007.
- [23] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.
- [24] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.
- [25] Elad Hazan, Alexander Rakhlin, and Peter L Bartlett. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems*, pages 65–72, 2008.
- [26] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in neural information processing systems*, pages 315–323, 2013.
- [27] Tomer Koren and Roi Livni. Affine-invariant online optimization and the low-rank experts problem. In *Advances in Neural Information Processing Systems 30*, pages 4750–4758. Curran Associates, Inc., 2017.
- [28] Wojciech Kotłowski. Scale-invariant unconstrained online learning. In *Proc. of ALT*, 2017.
- [29] Lihua Lei, Cheng Ju, Jianbo Chen, and Michael I Jordan. Non-convex finite-sum optimization via scsg methods. In *Advances in Neural Information Processing Systems*, pages 2345–2355, 2017.
- [30] Roberto Lucchetti. *Convexity and well-posed problems*. Springer Science & Business Media, 2006.

- [31] Haipeng Luo, Alekh Agarwal, Nicolo Cesa-Bianchi, and John Langford. Efficient second order online learning by sketching. In *Advances in Neural Information Processing Systems*, pages 902–910, 2016.
- [32] Brendan McMahan and Matthew Streeter. No-regret algorithms for unconstrained online convex optimization. In *Advances in neural information processing systems*, pages 2402–2410, 2012.
- [33] H. Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. *arXiv preprint arXiv:1403.3465*, 2014.
- [34] H Brendan McMahan. A survey of algorithms and analysis for adaptive online learning. *Journal of Machine Learning Research (JMLR)*, 18(90):1–50, 2017.
- [35] H Brendan McMahan and Francesco Orabona. Unconstrained online linear learning in hilbert spaces: Minimax algorithms and normal approximations. In *Conference on Learning Theory (COLT)*, pages 1020–1039, 2014.
- [36] H. Brendan McMahan and Matthew Streeter. Adaptive bound optimization for online convex optimization. In *Conference on Learning Theory (COLT)*, 2010.
- [37] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.
- [38] Francesco Orabona. Dimension-free exponentiated gradient. In *Advances in Neural Information Processing Systems*, pages 1806–1814, 2013.
- [39] Francesco Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Advances in Neural Information Processing Systems*, pages 1116–1124, 2014.
- [40] Francesco Orabona and Dávid Pál. Coin betting and parameter-free online learning. In *Advances in Neural Information Processing Systems 29*, pages 577–585, 2016.
- [41] Francesco Orabona and Dávid Pál. Open problem: Parameter-free and scale-free online algorithms. In *Conference on Learning Theory*, 2016.
- [42] Francesco Orabona and Dávid Pál. Scale-free online learning. *arXiv preprint arXiv:1601.01974*, 2016.
- [43] Francesco Orabona and Tatiana Tommasi. Backprop without learning rates through coin betting. *CoRR*, abs/1705.07795, 2017. URL <http://arxiv.org/abs/1705.07795>.
- [44] Francesco Orabona and Tatiana Tommasi. Training deep networks without learning rates through coin betting. In *Advances in Neural Information Processing Systems*, pages 2157–2167, 2017.
- [45] Francesco Orabona, Koby Crammer, and Nicolò Cesa-Bianchi. A generalized online mirror descent with applications to classification and regression. *Machine Learning*, 99(3):411–435, 2014.
- [46] Erik Ordentlich and Thomas M Cover. On-line portfolio selection. In *Proceedings of the ninth annual conference on Computational learning theory*, pages 310–313. ACM, 1996.
- [47] Iosif Pinelis. Rosenthal-type inequalities for martingales in 2-smooth banach spaces. *Theory Probab. Appl.*, 59(4): 699–706, 2015.

- [48] Gilles Pisier. Probabilistic methods in the geometry of banach spaces. In *Probability and analysis*, pages 167–241. Springer, 1986.
- [49] Gilles Pisier. Martingales in banach spaces (in connection with type and cotype). course ihp, feb. 2–8, 2011. *Manuscript*, <http://www.math.jussieu.fr/~pisier/ihp-pisier.pdf>, 2011.
- [50] Sasha Rakhlin, Ohad Shamir, and Karthik Sridharan. Relax and randomize: From value to algorithms. In *Advances in Neural Information Processing Systems*, pages 2141–2149, 2012.
- [51] Stephane Ross, Paul Mineiro, and John Langford. Normalized online learning. In *Proc. of the 29th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2013.
- [52] Shai Shalev-Shwartz. *Online learning: Theory, algorithms, and applications*. PhD thesis, Hebrew University, 2007.
- [53] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2011.
- [54] Shai Shalev-Shwartz. Sdca without duality, regularization, and individual convexity. In *International Conference on Machine Learning*, pages 747–754, 2016.
- [55] Shai Shalev-Shwartz and Yoram Singer. Convex repeated games and fenchel duality. In *Advances in neural information processing systems*, pages 1265–1272, 2007.
- [56] Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. Smoothness, low noise and fast rates. In *Advances in Neural Information Processing Systems 23*, pages 2199–2207, 2010.
- [57] Nathan Srebro, Karthik Sridharan, and Ambuj Tewari. On the universality of online mirror descent. In *Advances in neural information processing systems*, pages 2645–2653, 2011.
- [58] Matthew Streeter and H Brendan McMahan. Less regret via online conditioning. *arXiv preprint arXiv:1002.4862*, 2010.
- [59] Tim van Erven and Wouter M Koolen. MetaGrad: Multiple learning rates in online learning. In *Advances in Neural Information Processing Systems 29*, pages 3666–3674. 2016.
- [60] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proc. of the 20th International Conference on Machine Learning (ICML-03)*, pages 928–936, 2003.