A Modern Introduction to Online Learning

Francesco Orabona
Boston University
francesco@orabona.com

May 30, 2023

Contents

List of Definitions

List of Algorithms

Abstract

Disclaimer: This is work in progress, I plan to add more material and/or change/reorganize the content.

In this monograph, I introduce the basic concepts of Online Learning through a modern view of Online Convex Optimization. Here, online learning refers to the framework of regret minimization under worst-case assumptions. I present first-order and second-order algorithms for online learning with convex losses, in Euclidean and non-Euclidean settings. All the algorithms are clearly presented as instantiation of Online Mirror Descent or Follow-The-Regularized-Leader and their variants. Particular attention is given to the issue of tuning the parameters of the algorithms and learning in unbounded domains, through adaptive and parameter-free online learning algorithms. Non-convex losses are dealt through convex surrogate losses and through randomization. The bandit setting is also briefly discussed, touching on the problem of adversarial and stochastic multi-armed bandits. These notes do not require prior knowledge of convex analysis and all the required mathematical tools are rigorously explained. Moreover, all the included proofs have been carefully chosen to be as simple and as short as possible.

I want to thank all the people that checked the proofs and reasonings in these notes. In particular, the students in my first class that mercilessly pointed out my mistakes, Nicolò Campolongo that found all the typos in my formulas, and Jake Abernethy for the brainstorming on presentation strategies. Other people that helped me with comments, feedback, references, and/or hunting typos (in alphabetical order): Andreas Argyriou, Param Kishor Budhraja, Nicolò Cesa-Bianchi, Keyi Chen, Mingyu Chen, Peiqing Chen, Ryan D'Orazio, Alon Gonen, Daniel Hsu, Gergely Imreh, Christian Kroer, Kwang-Sung Jun, Michał Kempka, Pierre Laforgue, Chuang-Chieh Lin, Shashank Manjunath, Aryan Mokhtari, Gergely Neu, Ankit Pensia, Daniel Roy, Guanghui Wang, and Jiujia Zhang.

This material is based upon work supported by the National Science Foundation under grant no. 1925930 "Collaborative Research: TRIPODS Institute for Optimization and Learning".

A note on citations: it is customary in the computer science literature to only cite the journal version of a result that first appeared in a conference. The rationale is that the conference version is only a preliminary version, while the journal one is often more complete and sometimes more correct. In these notes, I will not use this custom. Instead, in the presence of the conference and journal version of the same paper, I will cite both. The reason is that I want to clearly delineate the history of the ideas, their first inventors, and the unavoidable rediscoveries. Hence, I need the exact year when some ideas were first proposed. Moreover, in some rare cases the authors changed from the conference to the journal version, so citing only the latter would erase the contribution of some key people from the history of Science.

Chapter 1

What is Online Learning?

Imagine the following repeated game:
In each round $t = 1, \dots, T$

- An adversary choose a real number in $y_t \in [0, 1]$ and he keeps it secret;
- You try to guess the real number, choosing $x_t \in [0, 1]$;
- The adversary's number is revealed and you pay the squared difference $(x_t - y_t)^2$.

Basically, we want to guess a sequence of numbers as precisely as possible. To be a game, we now have to decide what is the "winning condition". Let's see what makes sense to consider as a winning condition.

First, let's simplify a bit the game. Let's assume that the adversary is drawing the numbers i.i.d. from some fixed distribution over $[0, 1]$. However, he is still free to decide which distribution at the beginning of the game. If we knew the distribution, we could just predict each round the mean of the distribution and in expectation we would pay $\sigma^2 T$, where σ^2 is the variance of the distribution. We cannot do better than that! However, given that we do not know the distribution, it is natural to benchmark our strategy with respect to the optimal one. That is, it is natural to measure the quantity

$$\mathbb{E}_Y \left[\sum_{t=1}^{T} (x_t - Y)^2 \right] - \sigma^2 T, \tag{1.1}$$

or equivalently considering the average

$$\frac{1}{T} \mathbb{E}_Y \left[\sum_{t=1}^{T} (x_t - Y)^2 \right] - \sigma^2 \ . \tag{1.2}$$

Clearly these quantities are positive and they seem to be a good measure, because they are somehow normalized with respect to the "difficulty" of the numbers generated by the adversary, through the variance of the distribution. It is not the only possible choice to measure our "success", but for sure it is a reasonable one. It would make sense to consider a strategy "successful" if the difference in (1.1) grows sublinearly over time and, equivalently, if the difference in (1.2) goes to zero as the number of rounds T goes to infinity. That is, on average on the number of rounds, we would like our algorithm to be able to approach the optimal performance.

Minimizing Regret. Given that we have converged to what it seems a good measure of success of the algorithm. Let's now rewrite (1.1) in an equivalent way

$$\mathbb{E} \left[\sum_{t=1}^{T} (x_t - Y)^2 \right] - \min_{x \in [0,1]} \mathbb{E} \left[\sum_{t=1}^{T} (x - Y)^2 \right] \ .$$

Now, the last step: let's remove the assumption on how the data is generated, consider any arbitrary sequence of y_t, and keep using the same measure of success. Of course, we can remove the expectation because there is no stochasticity anymore. Note that in the stochastic case the optimal strategy is given by a single best prediction, so it was natural to compare against it. Instead, with arbitrary sequences it is not clear anymore that this is a good competitor. For example, we might consider a sequence of competitors instead of a single one. Indeed, it can be done, but the single competitor is still interesting in a variety of settings and simpler to explain. So, most of the time we will use a single competitor.

Now, we get that we will win the game if

$$\mathrm{Regret}_T := \sum_{t=1}^{T}(x_t - y_t)^2 - \min_{x\in[0,1]} \sum_{t=1}^{T}(x - y_t)^2$$

grows sublinearly with T. The quantity above is called the *Regret*, because it measures how much the algorithm regrets for not sticking on all the rounds to the optimal choice in hindsight. We will denote it by Regret_T. Our reasoning should provide sufficient justification for this metric, however in the following we will see that this also makes sense from both a convex optimization and machine learning point of view.

Let's now generalize the online game a bit more, considering that the algorithm outputs a vector in $\boldsymbol{x}_t \in V \subseteq \mathbb{R}^d$ and it pays a loss $\ell_t : V \to \mathbb{R}$ that measures how good was the prediction of the algorithm in each round. The set V is called the *feasible set*. Also, let's consider an arbitrary predictor $\boldsymbol{u}$ in[1] $V \subseteq \mathbb{R}^d$ and let's parameterize the regret with respect to it: $\mathrm{Regret}_T(\boldsymbol{u})$. So, to summarize, Online Learning is nothing else than designing and analyzing algorithms to minimize the Regret over a sequence of loss functions with respect to an arbitrary competitor $\boldsymbol{u} \in V \subseteq \mathbb{R}^d$:

$$\mathrm{Regret}_T(\boldsymbol{u}) := \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \,.$$

Remark 1.1. *Strictly speaking, the regret is also a function of the losses $\ell_1, \dots, \ell_T$. However, we will suppress this dependency for simplicity of notation.*

This framework is pretty powerful, and it allows to reformulate a bunch of different problems in machine learning and optimization as similar games. More in general, with the regret framework we can analyze situations in which the data are not independent and identically distributed from a distribution, yet I would like to guarantee that the algorithm is "learning" something. For example, online learning can be used to analyze

- Prediction of clicks on banners on webpages;
- Routing on a network;
- Convergence to equilibrium of repeated games.

It can *also* be used to analyze stochastic algorithms, e.g., Stochastic Gradient Descent, but the adversarial nature of the analysis might give you suboptimal results. For example, it can be used to analyze momentum algorithms, but the adversarial nature of the losses might force you to prove a convergence guarantee that treats the momentum term as a vanishing disturbance that does not help the algorithm in any way.

Let's now go back to our number guessing game and let's try a strategy to win it. Of course, this is one of the simplest example of online learning, without a real application. Yet, going through it we will uncover most of the key ingredients in online learning algorithms and their analysis.

A Winning Strategy. Can we win the number guessing game? Note that we did not assume anything on how the adversary is deciding the numbers. In fact, the numbers can be generated in any way, even in an adaptive way based on our strategy. Indeed, they can be chosen *adversarially*, that is explicitly trying to make us lose the game. This is why we call the mechanism generating the number the *adversary*.

[1] In same cases, we can make the game easier for the algorithm letting it choose the prediction from a set $W \supset V$.

The fact that the numbers are adversarially chosen means that we can immediately rule out any strategy based on any statistical modeling of the data. In fact, it cannot work because the moment we estimate something and act on our estimate, the adversary can immediately change the way he is generating the data, ruining us. So, we have to think about something else. Yet, many times online learning algorithms will look like classic ones from statistical estimation, even if they work for different reasons.

Now, let's try to design a strategy to make the regret provably sublinear in time, *regardless of how the adversary chooses the numbers*. The first thing to do is to take a look at the best strategy in hindsight, that is argmin of the second term of the regret. It should be immediate to see that

$$x_T^\star := \operatorname*{argmin}_{x\in[0,1]} \sum_{t=1}^{T}(x-y_t)^2 = \frac{1}{T}\sum_{t=1}^{T} y_t \ .$$

Now, given that we do not know the future, for sure we cannot use $x_T^\star$ as our guess in each round. However, we do know the past, so a reasonable strategy in each round could be to output the best number over the past. Why such strategy would work? For sure, the reason why it could work is not because we expect the future to be like the past, because it is not true! Instead, we want to leverage the fact that the optimal guess over time cannot change too much between rounds, so we can try to "track" it over time.

Hence, on each round t our strategy is to guess $x_t = x_{t-1}^\star = \frac{1}{t-1}\sum_{i=1}^{t-1} y_i$. Such strategy is usually called *Follow-the-Leader* (FTL), because you are following what would have been the optimal thing to do on the past rounds (the Leader).

Let's now try to show that indeed this strategy will allow us to win the game. Given that this is a simple example, we will prove its regret guarantee using first principles. In the following, we will introduce and use very general proof methods. First, we will need a small lemma.

Lemma 1.2. *Let $V \subseteq \mathbb{R}^d$ and $\ell_t : V \to \mathbb{R}$ an arbitrary sequence of loss functions. Denote by $x_t^\star$ a minimizer of the cumulative losses over the previous t rounds in V. Then, we have*

$$\sum_{t=1}^{T}\ell_t(x_t^\star) \le \sum_{t=1}^{T}\ell_t(x_T^\star) \ .$$

Proof. We prove it by induction over T. The base case is

$$\ell_1(x_1^\star) \le \ell_1(x_1^\star),$$

that is trivially true. Now, for $T \ge 2$, we assume that $\sum_{t=1}^{T-1}\ell_t(x_t^\star) \le \sum_{t=1}^{T-1}\ell_t(x_{T-1}^\star)$ is true and we must prove the stated inequality, that is

$$\sum_{t=1}^{T}\ell_t(x_t^\star) \le \sum_{t=1}^{T}\ell_t(x_T^\star) \ .$$

This inequality is equivalent to

$$\sum_{t=1}^{T-1}\ell_t(x_t^\star) \le \sum_{t=1}^{T-1}\ell_t(x_T^\star), \tag{1.3}$$

where we removed the last element of the sums because they are the same. Now observe that

$$\sum_{t=1}^{T-1}\ell_t(x_t^\star) \le \sum_{t=1}^{T-1}\ell_t(x_{T-1}^\star),$$

by induction hypothesis, and

$$\sum_{t=1}^{T-1}\ell_t(x_{T-1}^\star) \le \sum_{t=1}^{T-1}\ell_t(x_T^\star)$$

because $x_{T-1}^\star$ is a minimizer of the left hand side in V and $x_T^\star \in V$. Chaining these two inequalities, we have that (1.3) is true, and so the theorem is proven. □

Basically, the above lemma quantifies the idea the knowing the future and being adaptive to it is typically better than not being adaptive to it.

With this lemma, we can now prove that the regret will grow sublinearly, in particular it will be *logarithmic* in time. Note that we will not prove that our strategy is minimax optimal, even if it is possible to show that the logarithmic dependency on time is unavoidable for this problem.

Theorem 1.3. *Let $y_t \in [0,1]$ for $t = 1, \dots, T$ an arbitrary sequence of numbers. Let the algorithm's output $x_t = x^\star_{t-1} := \frac{1}{t-1}\sum_{i=1}^{t-1} y_i$. Then, we have*

$$\mathrm{Regret}_T = \sum_{t=1}^{T}(x_t - y_t)^2 - \min_{x\in[0,1]} \sum_{t=1}^{T}(x - y_t)^2 \le 4 + 4\ln T \,.$$

Proof. We use Lemma 1.2 to upper bound the regret:

$$\sum_{t=1}^{T}(x_t - y_t)^2 - \min_{x\in[0,1]} \sum_{t=1}^{T}(x - y_t)^2 = \sum_{t=1}^{T}(x^\star_{t-1} - y_t)^2 - \sum_{t=1}^{T}(x^\star_T - y_t)^2 \le \sum_{t=1}^{T}(x^\star_{t-1} - y_t)^2 - \sum_{t=1}^{T}(x^\star_t - y_t)^2 \,.$$

Now, let's take a look at each difference in the sum in the last equation. We have that

$$\begin{aligned}
(x^\star_{t-1} - y_t)^2 - (x^\star_t - y_t)^2 &= (x^\star_{t-1})^2 - 2y_t x^\star_{t-1} - (x^\star_t)^2 + 2y_t x^\star_t \\
&= (x^\star_{t-1} + x^\star_t - 2y_t)(x^\star_{t-1} - x^\star_t) \\
&\le |x^\star_{t-1} + x^\star_t - 2y_t|\,|x^\star_{t-1} - x^\star_t| \\
&\le 2|x^\star_{t-1} - x^\star_t| \\
&= 2\left|\frac{1}{t-1}\sum_{i=1}^{t-1} y_i - \frac{1}{t}\sum_{i=1}^{t} y_i\right| \\
&= 2\left|\left(\frac{1}{t-1} - \frac{1}{t}\right)\sum_{i=1}^{t-1} y_i - \frac{y_t}{t}\right| \\
&\le 2\left|\frac{1}{t(t-1)}\sum_{i=1}^{t-1} y_i\right| + \frac{2|y_t|}{t} \\
&\le \frac{2}{t} + \frac{2|y_t|}{t} \\
&\le \frac{4}{t} \,.
\end{aligned}$$

Hence, overall we have

$$\sum_{t=1}^{T}(x_t - y_t)^2 - \min_{x\in[0,1]} \sum_{t=1}^{T}(x - y_t)^2 \le 4\sum_{t=1}^{T}\frac{1}{t} \,.$$

To upper bound the last sum, observe that we are trying to find an upper bound to the green area in Figure 1.1. As you can see from the picture, it can be upper bounded by 1 plus the integral of $\frac{1}{t-1}$ from 2 to $T+1$. So, we have

$$\sum_{t=1}^{T}\frac{1}{t} \le 1 + \int_2^{T+1}\frac{1}{t-1}\,\mathrm{d}t = 1 + \ln T \,. \qquad \square$$

Let's write in words the steps of the proof: Lemma 1.2 allows us to upper bound the regret against the single best guess with the regret against the sequence $x^\star_1, \dots, x^\star_T$. In turn, given that $|x^\star_t - x^\star_{t-1}|$ goes to zero very fast, the total regret is sublinear in time.

There are a few things to stress on this strategy. The strategy does not have parameters to tune (e.g., learning rates, regularizers). Note that the presence of parameters does not make sense in online learning: We have only one

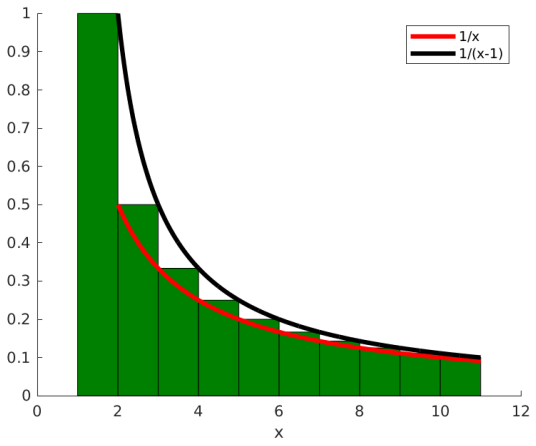

Figure 1.1: Upper bounding the sum with an integral.

stream of data and we cannot run our algorithm over it multiple times to select the best parameter! Also, it does not need to maintain a complete record of the past, but only a "summary" of it, through the running average. This gives a computationally efficient algorithm. When we design online learning algorithms, we will strive to achieve all these characteristics. Last thing that I want to stress is that the algorithm does not use gradients: Gradients are useful and we will use them a lot, but they do not constitute the entire world of online learning.

Before going on I want to remind you that, as seen above, this is different from the classic setting in statistical machine learning. So, for example, "overfitting" has absolutely no meaning here. Same for "generalization gap" and similar ideas linked to a training/testing scenario.

In the next chapters, we will introduce several algorithms for online optimization and one of them will be a strict generalization of the strategy we used in the example above.

1.1 History Bits

The concept of "regret" seems to have been proposed by Savage [1951], an exposition and review of the book by Wald [1950] on a foundation of statistical decision problems based on zero-sum two-person games. Savage [1951] introduces the idea of considering the difference between the utility of the best action in a given state and the utility incurred by any action under the same state. The proposed optimal strategy was the one minimizing such regret over the worst possible state. Savage [1951] called this concept "loss" and did not like the word "regret" because " that term seems to me charged with emotion and liable to lead to such misinterpretation as that the loss necessarily becomes known" [Savage, 1954, page 163]. The name of "regret" instead seems to have suggested in Milnor [1951].

However, Savage's definition is a modification of the one proposed by Wald [1950], who instead proposed to maximize just the utility, under the assumption that the utility of the best action for any state is 0. While minimizing the regret or the minimizing the negative utility under the assumption of Wald [1950] are mathematically equivalent, Savage [pages 169–170 1954] explains that Wald considered the regret formulation different from what he proposed, while Savage attributed to his idea "little or no originality".

Extending the definition of Savage [1951, 1954] to a sequence of games, Hannan [1957] designed a randomized algorithm for zero-sum repeated games with fixed loss matrix with a vanishing expected average regret. Hence, the concept of regret seems to originate from game theory, but strangely enough passing through the ideas of two mathematical statisticians.

Lemma 1.2 is due to Hannan [1957].

Exercises

Problem 1.1. *Extend the previous algorithm and analysis to the case when the adversary selects a vector $\boldsymbol{y}_t \in \mathbb{R}^d$ such that $\|\boldsymbol{y}_t\|_2 \leq 1$, the algorithm guesses a vector $\boldsymbol{x}_t \in \mathbb{R}^d$, and the loss function is $\|\boldsymbol{x}_t - \boldsymbol{y}_t\|_2^2$. Show an upper bound to the regret logarithmic in T and that does not depend on d. Among the other things, you will probably need the Cauchy-Schwarz inequality: $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \leq \|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2$.*

Chapter 2

Online Subgradient Descent

In this chapter, we will introduce the Online Subgradient Descent algorithm: a generic online algorithm to solve online problems with convex losses. First, we will introduce Online Gradient Descent for convex differentiable functions, then we will extend it to non-differentiable functions.

2.1 Online Learning with Convex Differentiable Losses

To summarize what we said in the first chapter, let's define an online learning as the following general game

- For $t = 1, \dots, T$
  - Outputs $\boldsymbol{x}_t \in V \subseteq \mathbb{R}^d$
  - Receive $\ell_t : V \to \mathbb{R}$
  - Pay $\ell_t(\boldsymbol{x}_t)$
- End for

The aim of this game is to minimize the regret with respect to any competitor $\boldsymbol{u} \in V$:

$$\text{Regret}_T(\boldsymbol{u}) := \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \, .$$

We also said that the way the losses ℓ_t are decided is adversarial. Now, without additional assumptions we cannot hope to solve this problem. Hence, we have to understand what are the *reasonable* assumptions we can make. Typically, we will try to restrict the choice of the loss functions in some way. This is considered reasonable because most of the time we have some say in deciding the set from which the loss functions are picked. So, for example, we will consider only *convex* loss functions. However, convexity might not be enough, so we might restrict the class a bit more to, for example, Lipschitz convex functions. On the other hand, assuming to know something about the future is not considered a reasonable assumption, because we very rarely have any control on the future. In general, the stronger the assumptions the better will be guarantee on the regret we can prove. The best algorithms we will see will guarantee a sublinear regret against the weakest assumption we can make, guaranteeing *at the same time* a smaller regret for *easy* adversaries.

It is also important to remember why minimizing the regret is a good objective: Given that we do not assume anything on how the adversary generates the loss functions, minimizing the regret is a good metric that takes into account the difficulty of the problem. If an online learning algorithm is able to guarantee a sublinear regret, it means that its performance on average will approach the performance of any fixed strategy. As said, we will see that in many situations if the adversary is "weak", for example it is a fixed stochastic distribution over the loss functions, being prepared for the worst-case scenario will not preclude us to get the best guarantee anyway.

For a while, we will focus on the case that ℓ_t are convex, and this problem will be called **Online Convex Optimization** (OCO). Later, we will later see how to *convexify* some specific non-convex online problems.

Remark 2.1. *I will now introduce some math concepts. If you have a background in Convex Analysis, this will be easy stuff for you. On the other hand, if you never saw these things before they might look a bit scary. Let me tell you the right way to look at them:* these are tools that will make our job easier. *Without these tools, it would be basically impossible to design any online learning algorithm. And, no, it is not enough to test random algorithms on some machine learning dataset, because fixed datasets are not adversarial. Without a correct proof, you might not realize that your online algorithm fail on particular sequences of losses, as it happened to Adam [Reddi et al., 2018]. I promise you that once you understand the key mathematical concepts, online learning is actually easy.*

2.1.1 Convex Analysis Bits: Convexity

Figure 2.1: Convex (left) and non-convex (right) sets.

Definition 2.2 (Convex Set)**.** *$V \subset \mathbb{R}^d$ is **convex** if for any $\boldsymbol{x}, \boldsymbol{y} \in V$ and any $\lambda \in (0,1)$, we have $\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y} \in V$.*

In words, this means that *the set V has no holes*, see Figure 2.1.

We will make use of **extended-real-valued functions**, that is function that take value in $\mathbb{R} \cup \{-\infty, +\infty\}$. For f an extended-real-valued function on $\mathbb{R}^d$, its **domain** is the set $\operatorname{dom} f = \{\boldsymbol{x} \in \mathbb{R}^d : f(x) < +\infty\}$.

Extended-real-valued functions allow us to easily consider constrained set and are a standard notation in Convex Optimization [see, e.g., Boyd and Vandenberghe, 2004]. For example, if I want the predictions of the algorithm x_t and the competitor $\boldsymbol{u}$ to be in a set $V \subset \mathbb{R}^d$, I can just add $i_V(\boldsymbol{x})$ to all the losses, where $i_V : \mathbb{R}^d \to (-\infty, +\infty]$ is the **indicator function of the set** V defined as

$$i_V(\boldsymbol{x}) = \begin{cases} 0, & \boldsymbol{x} \in V, \\ +\infty, & \text{otherwise.} \end{cases}$$

In this way, the only way for the algorithm and for the competitor to suffer finite loss is to predict inside the set V. Also, extended-real-valued functions will make the use of Fenchel conjugates more direct, see Section 5.2.1.

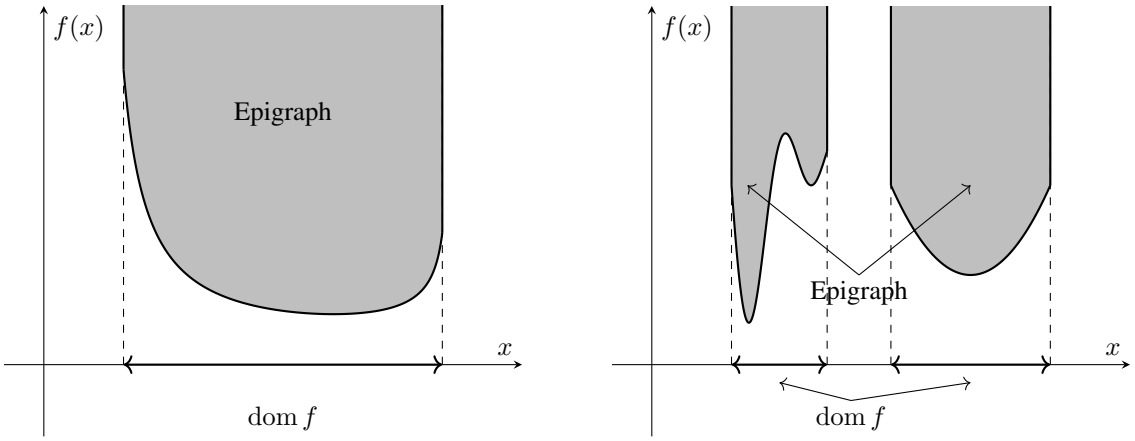

Figure 2.2: Convex (left) and nonconvex (right) functions.

Convex functions will be an essential ingredient in online learning.

Definition 2.3 (Convex Function)**.** *Let $f : \mathbb{R}^d \to [-\infty, +\infty]$. f is **convex** if the epigraph of the function, $\{(\boldsymbol{x}, y) \in \mathbb{R}^{d+1} : y \geq f(x)\}$, is convex.*

We can see a visualization of this definition in Figure 2.2. Note that the definition implies that the domain of a convex function is convex. Also, observe that if $f : V \subseteq \mathbb{R}^d \to \mathbb{R}$ is convex, $f + i_V : \mathbb{R}^d \to (-\infty, +\infty]$ is also convex. Note that $i_V(\boldsymbol{x})$ is convex iff V is convex, so each convex set is associated with a convex function.

The definition above gives rise to the following characterization for convex functions that do not assume the value $-\infty$.

Theorem 2.4 ([Rockafellar, 1970, Theorem 4.1])**.** *Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ and $\operatorname{dom} f$ is a convex set. Then f is convex iff, for any $0 < \lambda < 1$, we have*

$$f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1-\lambda) f(\boldsymbol{y}),\ \forall \boldsymbol{x}, \boldsymbol{y} \in \operatorname{dom} f\ .$$

Example 2.5. *The simplest example of convex functions are the* affine functions*: $f(\boldsymbol{x}) = \langle \boldsymbol{z}, \boldsymbol{x} \rangle + b$.*

Example 2.6. *Norms are always convex, the proof is left as exercise.*

How to recognize a convex function? In the most general case, you have to rely on the definition. However, most of the time we will recognize them as composed by operations that preserve the convexity. For example:

- f and g convex, then the linear combination with non-negative weights is also convex.
- The composition with an affine transformation preserves the convexity.
- Pointwise supremum of convex functions is convex.

The proofs are left as exercises.

A very important property of differentiable convex functions is that we can construct linear lower bound to the function.

Theorem 2.7 ([Rockafellar, 1970, Theorem 25.1 and Corollary 25.1.1])**.** *Suppose $f : \mathbb{R}^d \to (-\infty, +\infty]$ a convex function and let $\boldsymbol{x} \in \operatorname{int} \operatorname{dom} f$. If f is differentiable at $\boldsymbol{x}$ then*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle,\ \forall \boldsymbol{y} \in \mathbb{R}^d\ .$$

We will also use the first-order optimality condition for differentiable convex functions:

Theorem 2.8. *Let V a convex non-empty set, $\boldsymbol{x}^\star \in V$, and f a convex function, differentiable over an open set that contains V. Then $\boldsymbol{x}^\star \in \operatorname{argmin}_{\boldsymbol{x} \in V}\ f(\boldsymbol{x})$ iff $\langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{y} - \boldsymbol{x}^\star \rangle \geq 0,\ \forall \boldsymbol{y} \in V$.*

Proof. Let first assume that $\boldsymbol{x}^\star$ satisfies $\langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{y} - \boldsymbol{x}^\star \rangle \geq 0,\ \forall \boldsymbol{y} \in V$. Then, by Theorem 2.7, for any $\boldsymbol{y} \in V$, we have that

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}^\star) + \langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{y} - \boldsymbol{x}^\star \rangle \geq f(\boldsymbol{x}^\star),$$

that is $\boldsymbol{x}^\star$ is the minimizer of f over V.

Now, assume that $\boldsymbol{x}^\star$ is the minimizer of f over V and assume that there exists $\boldsymbol{y} \in V$ such that $\langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{y} - \boldsymbol{x}^\star \rangle < 0$. Consider $\boldsymbol{z}(\alpha) = \alpha \boldsymbol{y} + (1-\alpha)\boldsymbol{x}^\star$ where $\alpha \in (0, 1)$. Note that $\boldsymbol{z}(\alpha) \in V$ and denote by $h(\alpha) = f(\boldsymbol{z}(\alpha))$. We have that $h'(0) = \langle \nabla f(\boldsymbol{x}^\star), \boldsymbol{y} - \boldsymbol{x}^\star \rangle < 0$. Given that f is differentiable and so continuous, there exists $\alpha^\star$ sufficiently small such that $f(\boldsymbol{z}(\alpha^\star)) < f(\boldsymbol{z}(0)) = f(\boldsymbol{x}^\star)$ that contradicts that fact that $\boldsymbol{x}^\star$ is the minimizer over V. □

In words, at the constrained minimum, the gradient makes an angle of 90° or less with all the feasible variations $\boldsymbol{y} - \boldsymbol{x}^\star$, hence we cannot minimize more the function moving inside V. Moreover, if $\boldsymbol{x}^\star \in \operatorname{int} V$, by choosing ϵ small enough such that $\boldsymbol{y} = \boldsymbol{x}^\star - \epsilon \nabla f(\boldsymbol{x}^\star) \in V$, we obtain that $\boldsymbol{x}^\star$ is a minimum iff $\nabla f(\boldsymbol{x}^\star) = \boldsymbol{0}$.

Another critical property of convex functions is Jensen's inequality.

Theorem 2.9. *Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ be a measurable convex function and $\boldsymbol{x}$ be an $\mathbb{R}^d$-valued random element on some probability space such that $\mathbb{E}[\boldsymbol{x}]$ exists and $\boldsymbol{x} \in \operatorname{dom} f$ with probability 1. Then*

$$E[f(\boldsymbol{x})] \geq f(E[\boldsymbol{x}])\ .$$

We can now see our first OCO algorithm in the case that the functions are convex and differentiable.

2.1.2 Online Gradient Descent

In the first chapter, we saw a simple strategy to obtain a logarithmic regret in the guessing game. The strategy was to use the best over the past, that is the *Follow-the-Leader* strategy. In formulas,

$$\boldsymbol{x}_t = \operatorname*{argmin}_{\boldsymbol{x}} \sum_{i=1}^{t-1} \ell_i(\boldsymbol{x}),$$

and in the first round we can play any admissible point. One might wonder if this strategy always works, but the answer is negative!

Example 2.10 (Failure of FTL)**.** *Let $V = [-1, 1]$ and consider the sequence of losses $\ell_t(x) = z_t x$, where*

$$\begin{aligned} z_1 &= -0.5, \\ z_t &= 1, \ t = 2, 4, \dots \\ z_t &= -1, \ t = 3, 5, \dots \end{aligned}$$

Then, a part from the first round where the prediction of FTL is arbitrary in $[-1, 1]$, the predictions of FTL will be $x_t = 1$ for t even and $x_t = -1$ for t odd. The cumulative loss of the FTL algorithm after T rounds will therefore be $T - 1 - \frac{x_1}{2}$ while the cumulative loss of the fixed solution $u = 0$ is 0. Thus, the regret of FTL with respect to $u = 0$ is $T - 1 - \frac{x_1}{2} \geq T - \frac{3}{2}$.

Hence, we will show an alternative strategy that guarantees sublinear regret for convex Lipschitz functions. Later, we will also prove that the dependency on T is optimal. The strategy is called Projected Online Gradient Descent, or just Online Gradient Descent, see Algorithm 2.1. It consists in updating the prediction of the algorithm at each time step moving in the negative direction of the gradient of the loss received and projecting back onto the feasible set. Some might see that this algorithm is similar to Stochastic Gradient Descent, but it is not the same thing: here the loss functions are different at each step and they are not drawn from a fixed distribution but adversarially chosen. We will later see that Online Gradient Descent can *also* be used as Stochastic Gradient Descent.

Algorithm 2.1 Projected Online Gradient Descent

Require: Non-empty closed convex set $V \subseteq \mathbb{R}^d$, $\boldsymbol{x}_1 \in V$, $\eta_1, \cdots, \eta_T > 0$
1: **for** $t = 1$ **to** T **do**
2: Output $\boldsymbol{x}_t \in V$
3: Receive $\ell_t : V \to \mathbb{R}$ differentiable in an open set containing V and pay $\ell_t(\boldsymbol{x}_t)$
4: Set $\boldsymbol{g}_t = \nabla \ell_t(\boldsymbol{x}_t)$
5: $\boldsymbol{x}_{t+1} = \Pi_V(\boldsymbol{x}_t - \eta_t \boldsymbol{g}_t) = \operatorname{argmin}_{\boldsymbol{y} \in V} \ \|\boldsymbol{x}_t - \eta_t \boldsymbol{g}_t - \boldsymbol{y}\|_2$
6: **end for**

First, we show the following two Lemmas. The first lemma proves that Euclidean projections always decrease the distance with points inside the set.

Proposition 2.11. *Let $\boldsymbol{x} \in \mathbb{R}^d$ and $\boldsymbol{y} \in V$, where $V \subseteq \mathbb{R}^d$ is a non-empty closed convex set and define $\Pi_V(\boldsymbol{x}) := \operatorname{argmin}_{\boldsymbol{y} \in V} \ \|\boldsymbol{x} - \boldsymbol{y}\|_2$. Then, $\|\Pi_V(\boldsymbol{x}) - \boldsymbol{y}\|_2 \leq \|\boldsymbol{x} - \boldsymbol{y}\|_2$.*

Proof. First of all, observe that $\operatorname{argmin}_{\boldsymbol{y} \in V} \ \|\boldsymbol{x} - \boldsymbol{y}\|_2 = \operatorname{argmin}_{\boldsymbol{y} \in V} \ \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$. So, from the optimality condition of Theorem 2.8 on the function $f(\boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$, we obtain

$$\langle \Pi_V(\boldsymbol{x}) - \boldsymbol{x}, \boldsymbol{y} - \Pi_V(\boldsymbol{x}) \rangle \geq 0 \, .$$

Therefore,

$$\begin{aligned} \|\boldsymbol{y} - \boldsymbol{x}\|_2^2 &= \|\boldsymbol{y} - \Pi_V(\boldsymbol{x}) + \Pi_V(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 \\ &= \|\boldsymbol{y} - \Pi_V(\boldsymbol{x})\|_2^2 + 2\langle \boldsymbol{y} - \Pi_V(\boldsymbol{x}), \Pi_V(\boldsymbol{x}) - \boldsymbol{x} \rangle + \|\Pi_V(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 \\ &\geq \|\boldsymbol{y} - \Pi_V(\boldsymbol{x})\|_2^2 \, . \end{aligned}$$

□

The next lemma upper bounds the regret in one iteration of the algorithm.

Lemma 2.12. *Let $V \subseteq \mathbb{R}^d$ a non-empty closed convex set and $\ell_t : V \to \mathbb{R}$ a convex function differentiable in an open set that contains V. Set $\boldsymbol{g}_t = \nabla \ell_t(\boldsymbol{x}_t)$. Then, $\forall \boldsymbol{u} \in V$, the following inequality holds*

$$\eta_t(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \eta_t \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq \frac{1}{2}\|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 - \frac{1}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2 + \frac{\eta_t^2}{2}\|\boldsymbol{g}_t\|_2^2 \,.$$

Proof. From Proposition 2.11 and Theorem 2.7, we have that

$$\begin{aligned}
\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2 - \|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 &\leq \|\boldsymbol{x}_t - \eta_t \boldsymbol{g}_t - \boldsymbol{u}\|_2^2 - \|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 \\
&= -2\eta_t \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle + \eta_t^2 \|\boldsymbol{g}_t\|_2^2 \\
&\leq -2\eta_t(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) + \eta_t^2 \|\boldsymbol{g}_t\|_2^2 \,.
\end{aligned}$$

Reordering, we have the stated bound. □

We can prove the following regret guarantee.

Theorem 2.13. *Let $V \subseteq \mathbb{R}^d$ a non-empty closed convex set with diameter D, i.e., $\max_{\boldsymbol{x},\boldsymbol{y} \in V} \|\boldsymbol{x} - \boldsymbol{y}\|_2 \leq D$. Let $\ell_1, \cdots, \ell_T$ an arbitrary sequence of convex functions $\ell_t : V \to \mathbb{R}$ differentiable in open sets containing V. Pick any $\boldsymbol{x}_1 \in V$ and assume $\eta_{t+1} \leq \eta_t,\ t = 1, \ldots, T$. Then, $\forall \boldsymbol{u} \in V$, the following regret bound holds*

$$\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \frac{D^2}{2\eta_T} + \sum_{t=1}^{T} \frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \,.$$

Moreover, if η_t is constant, i.e., $\eta_t = \eta\ \forall t = 1, \cdots, T$, we have

$$\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \frac{\|\boldsymbol{u} - \boldsymbol{x}_1\|_2^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2 \,.$$

Proof. Dividing the inequality in Lemma 2.12 by η_t and summing over $t = 1, \cdots, T$, we have

$$\begin{aligned}
\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\leq \sum_{t=1}^{T}\left(\frac{1}{2\eta_t}\|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 - \frac{1}{2\eta_t}\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2\right) + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \\
&= \frac{1}{2\eta_1}\|\boldsymbol{x}_1 - \boldsymbol{u}\|_2^2 - \frac{1}{2\eta_T}\|\boldsymbol{x}_{T+1} - \boldsymbol{u}\|_2^2 + \sum_{t=1}^{T-1}\left(\frac{1}{2\eta_{t+1}} - \frac{1}{2\eta_t}\right)\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2 + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \\
&\leq \frac{1}{2\eta_1}D^2 + D^2\sum_{t=1}^{T-1}\left(\frac{1}{2\eta_{t+1}} - \frac{1}{2\eta_t}\right) + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \\
&= \frac{1}{2\eta_1}D^2 + D^2\left(\frac{1}{2\eta_T} - \frac{1}{2\eta_1}\right) + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \\
&= \frac{D^2}{2\eta_T} + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\boldsymbol{g}_t\|_2^2 \,.
\end{aligned}$$

In the same way, when the η_t is constant, we have

$$\begin{aligned}
\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\leq \sum_{t=1}^{T}\left(\frac{1}{2\eta}\|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 - \frac{1}{2\eta}\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2\right) + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2 \\
&= \frac{1}{2\eta}\|\boldsymbol{x}_1 - \boldsymbol{u}\|_2^2 - \frac{1}{2\eta}\|\boldsymbol{x}_{T+1} - \boldsymbol{u}\|_2^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2 \\
&\leq \frac{1}{2\eta}\|\boldsymbol{x}_1 - \boldsymbol{u}\|_2^2 + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2 \,.
\end{aligned}$$

□

We can immediately observe a few things.

- If we want to use time-varying learning rates, you need a bounded domain V for the proof to work. However, this assumption is false in most of the machine learning applications. However, in the stochastic setting you can still use a time-varying learning rate in SGD with an unbounded domain if you use a non-uniform averaging. We will see this in Chapter 3.

- Another important observation is that the regret bound helps us choosing the learning rates η_t. Indeed, it is the only guideline we have. Any other choice that is not justified by a regret analysis it is not justified at all.

As we said, the presence of parameters like the learning rates make no sense in online learning. So, we have to decide a strategy to set them. A simple choice is to find the constant learning rate that minimizes the bounds for a fixed number of iterations. We have to consider the expression

$$\frac{\|\boldsymbol{u} - \boldsymbol{x}_1\|_2^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2$$

and minimize with respect to η. It is easy to see that the optimal η is $\frac{\|\boldsymbol{u}-\boldsymbol{x}_1\|_2}{\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2}}$, that would give the regret bound

$$\|\boldsymbol{u} - \boldsymbol{x}_1\|_2 \sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2} \,.$$

However, we have a problem: in order to use this stepsize, we should know all the future gradients and the distance between the optimal solution and the initial point! This is clearly impossible: Remember that the adversary can choose the sequence of functions. Hence, it can observe your choice of learning rates and decide the sequence so that your learning rate is now the wrong one!

Indeed, it turns out that this kind of rate is completely impossible because it is ruled out by a lower bound. Yet, we will see that it is indeed possible to achieve very similar rates using *adaptive* (Section 4.2) and *parameter-free* algorithms (Chapter 9). For the moment, we can observe that we might be happy to minimize a loose upper bound. In particular, assume that the norm of the gradients is bounded by L, that is $\|\boldsymbol{g}_t\|_2 \leq L$. Also, assuming a bounded diameter, we can upper bound $\|\boldsymbol{u} - \boldsymbol{x}_1\|_2$ by D. Hence, we have

$$\eta^\star = \operatorname*{argmin}_{\eta} \; \frac{D^2}{2\eta} + \frac{\eta L^2 T}{2} = \frac{D}{L\sqrt{T}},$$

that gives a regret bound of

$$DL\sqrt{T} \,. \tag{2.1}$$

So, indeed the regret is sublinear in time.

In the next Section, we will see how to remove the differentiability assumption through the use of *subgradients*.

2.2 Online Subgradient Descent

In the previous section, we have introduced Projected Online Gradient Descent. However, the differentiability assumption for the ℓ_t is quite strong. What happens when the losses are convex but not differentiable? For example $\ell_t(x) = |x - 10|$. Note that this situation is more common than one would think. For example, the hinge loss, $\ell_t(\boldsymbol{w}) = \max(1 - y\langle \boldsymbol{w}, \boldsymbol{x}\rangle, 0)$, and the ReLU activation function used in neural networks, $\ell_t(x) = \max(x, 0)$, are not differentiable. It turns out that we can just use Online Gradient Descent, substituting the *subgradients* to the gradients. For this, we need some more convex analysis!

2.2.1 Convex Analysis Bits: Subgradients

First, we need a technical definition.

Definition 2.14 (Proper Function)**.** *If a function f is nowhere $-\infty$ and finite somewhere, then f is called **proper**.*

In this book, we are mainly interested in convex proper functions, that basically better conform to our intuition of what a convex function looks like.

Example 2.15. *The indicator function of a set $V \subset \mathbb{R}^d$ is proper iff V is non-empty.*

Let's first define formally what is a subgradient.

Definition 2.16 (Subgradient)**.** *For a proper function $f : \mathbb{R}^d \to (-\infty, +\infty]$, we define a **subgradient** of f in $\boldsymbol{x} \in \mathbb{R}^d$ as a vector $\boldsymbol{g} \in \mathbb{R}^d$ that satisfies*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x} \rangle, \ \forall \boldsymbol{y} \in \mathbb{R}^d \, .$$

Basically, a subgradient of f in $\boldsymbol{x}$ is any vector $\boldsymbol{g}$ that allows us to construct a linear lower bound to f. Note that the subgradient is not unique, so we denote the *set* of subgradients of f in $\boldsymbol{x}$ by $\partial f(\boldsymbol{x})$, called **subdifferential of f at $\boldsymbol{x}$**.

Observe that if f is proper and convex, then $\partial f(\boldsymbol{x})$ is empty for $\boldsymbol{x} \notin \operatorname{dom} f$, because the inequality cannot be satisfied when $f(\boldsymbol{x}) = +\infty$. Also, the domain of ∂f, denoted by $\operatorname{dom} \partial f$, is the set of all $\boldsymbol{x} \in \mathbb{R}^d$ such that $\partial f(x)$ is nonempty; it is a subset of $\operatorname{dom} f$. A proper convex function f is always subdifferentiable in $\operatorname{int} \operatorname{dom} f$ [Rockafellar, 1970, Theorem 23.4].

The unique subgradient of a differentiable function is just the gradient, as quantified in the next Theorem.

Theorem 2.17 ([Rockafellar, 1970, Theorem 25.1])**.** *If the function $f : \mathbb{R}^d \to [-\infty, +\infty]$ is convex and finite in $\boldsymbol{x}$, it is differentiable in $\boldsymbol{x}$ iff the subdifferential is composed by a unique element, that turns out to be $\nabla f(\boldsymbol{x})$.*

Also, we can also calculate subgradients of sum of functions.

Theorem 2.18. *Let $f_1, \ldots, f_m$ be proper functions on $\mathbb{R}^d$, and $f = f_1 + \cdots + f_m$. Then, $\partial f(\boldsymbol{x}) \supseteq \partial f_1(\boldsymbol{x}) + \cdots + \partial f_m(\boldsymbol{x}), \forall \boldsymbol{x}$. Moreover, if $f_1, \ldots, f_m$ are also convex, closed, and $\operatorname{dom} f_m \cap \bigcap_{i=1}^{m-1} \operatorname{int} \operatorname{dom} f_i \neq \{\}$, then actually $\partial f(\boldsymbol{x}) = \partial f_1(\boldsymbol{x}) + \cdots + \partial f_m(\boldsymbol{x}), \forall \boldsymbol{x}$.*

Proof. For any $\boldsymbol{z}$, define $\boldsymbol{g}_i \in \partial f_i(\boldsymbol{z})$ for $i = 1, \ldots, m$. From the definition of subgradient, we have

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} f_i(\boldsymbol{x}) \geq \sum_{i=1}^{m} (f_i(\boldsymbol{z}) + \langle \boldsymbol{g}_i, \boldsymbol{x} - \boldsymbol{z} \rangle) = f(\boldsymbol{z}) + \left\langle \sum_{i=1}^{m} \boldsymbol{g}_i, \boldsymbol{x} - \boldsymbol{z} \right\rangle \, .$$

Hence, $\sum_{i=1}^{m} \boldsymbol{g}_i \in \partial f(\boldsymbol{z})$.

For the second statement, see Bauschke and Combettes [2011, Corollary 16.39]. □

Example 2.19. *Let $f(x) = |x|$, then the subdifferential set $\partial f(x)$ is*

$$\partial f(x) = \begin{cases} \{1\}, & x > 0, \\ [-1, 1], & x = 0, \\ \{-1\}, & x < 0 \, . \end{cases}$$

Example 2.20. *Let's calculate the subgradient of the indicator function for a non-empty convex set $V \subset \mathbb{R}^d$. By definition, $\boldsymbol{g} \in \partial i_V(\boldsymbol{x})$ if*

$$i_V(\boldsymbol{y}) \geq i_V(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x} \rangle, \ \forall \boldsymbol{y} \in \mathbb{R}^d \, .$$

*This condition implies that $\boldsymbol{x} \in V$ and $0 \geq \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x} \rangle, \forall \boldsymbol{y} \in V$ (because for $\boldsymbol{y} \notin V$ the inequality is always verified). The set of all $\boldsymbol{g}$ that satisfies the above inequality is called the **normal cone of V at $\boldsymbol{x}$**. Note that the normal cone for any $\boldsymbol{x} \in \operatorname{int} V = \{\mathbf{0}\}$ (Hint: take $\boldsymbol{y} = \boldsymbol{x} + \epsilon \boldsymbol{g}$). For example, for $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 \leq 1\}$, $\partial i_V(\boldsymbol{x}) = \{\alpha \boldsymbol{x} | \alpha \geq 0\}$ for all $\boldsymbol{x} : \|\boldsymbol{x}\|_2 = 1$.*

Another useful theorem is to calculate the subdifferential of the pointwise maximum of convex functions.

Theorem 2.21 ([Bauschke and Combettes, 2011, Theorem 18.5])**.** *Let $(f_i)_{i\in I}$ be a finite set of convex functions from $\mathbb{R}^d$ to $(-\infty,+\infty]$ and suppose $\boldsymbol{x}\in\bigcap_{i\in I}\operatorname{dom} f_i$ and f_i continuous at $\boldsymbol{x}$. Set $F=\max_{i\in I} f_i$ and let $A(\boldsymbol{x})=\{i\in I|f_i(\boldsymbol{x})=F(\boldsymbol{x})\}$ the set of the active functions. Then*

$$\partial F(\boldsymbol{x})=\operatorname{conv}\bigcup_{i\in A(\boldsymbol{x})}\partial f_i(\boldsymbol{x}),$$

where conv *is the convex hull.*

Example 2.22 (Subgradients of the Hinge loss)**.** *Consider the loss $\ell(\boldsymbol{x})=\max(1-\langle \boldsymbol{z},\boldsymbol{x}\rangle,0)$ for $\boldsymbol{z}\in\mathbb{R}^d$. The subdifferential set is*

$$\partial\ell(\boldsymbol{x})=\begin{cases}\{\boldsymbol{0}\}, & 1-\langle \boldsymbol{z},\boldsymbol{x}\rangle<0\\ \{-\alpha\boldsymbol{z}|\alpha\in[0,1]\}, & 1-\langle \boldsymbol{z},\boldsymbol{x}\rangle=0\\ \{-\boldsymbol{z}\}, & \textit{otherwise}\end{cases}.$$

Definition 2.23 (Lipschitz Function)**.** *Let $f:\mathbb{R}^d\to(-\infty,+\infty]$ is **L-Lipschitz** over a set V w.r.t a norm $\|\cdot\|$ if $|f(\boldsymbol{x})-f(\boldsymbol{y})|\le L\|\boldsymbol{x}-\boldsymbol{y}\|,\ \forall \boldsymbol{x},\boldsymbol{y}\in V$.*

We also have this handy result that upper bounds the norm of subgradients of convex Lipschitz functions.

Theorem 2.24. *Let $f:\mathbb{R}^d\to(-\infty,+\infty]$ proper and convex. Then, f is L-Lipschitz in* $\operatorname{int}\operatorname{dom} f$ *w.r.t. the L_2 norm iff for all $\boldsymbol{x}\in\operatorname{int}\operatorname{dom} f$ and $\boldsymbol{g}\in\partial f(\boldsymbol{x})$ we have $\|\boldsymbol{g}\|_2\le L$.*

Proof. Assume f L-Lipschitz, then $|f(\boldsymbol{x})-f(\boldsymbol{y})|\le L\|\boldsymbol{x}-\boldsymbol{y}\|_2,\ \forall \boldsymbol{x},\boldsymbol{y}\in\operatorname{int}\operatorname{dom} f$. For $\epsilon>0$ small enough $\boldsymbol{y}=\boldsymbol{x}+\epsilon\frac{\boldsymbol{g}}{\|\boldsymbol{g}\|_2}\in\operatorname{int}\operatorname{dom} f$, then

$$L\epsilon=L\|\boldsymbol{x}-\boldsymbol{y}\|_2\ge|f(\boldsymbol{y})-f(\boldsymbol{x})|\ge f(\boldsymbol{y})-f(\boldsymbol{x})\ge\langle \boldsymbol{g},\boldsymbol{y}-\boldsymbol{x}\rangle=\epsilon\|\boldsymbol{g}\|_2,$$

that implies that $\|\boldsymbol{g}\|_2\le L$.

For the other implication, the definition of subgradient and Cauchy-Schwarz inequalities gives us

$$f(\boldsymbol{x})-f(\boldsymbol{y})\le\|\boldsymbol{g}\|_2\|\boldsymbol{x}-\boldsymbol{y}\|_2\le L\|\boldsymbol{x}-\boldsymbol{y}\|_2,$$

for any $\boldsymbol{x},\boldsymbol{y}\in\operatorname{int}\operatorname{dom} f$. Taking $\boldsymbol{g}\in\partial f(\boldsymbol{y})$, we also get

$$f(\boldsymbol{y})-f(\boldsymbol{x})\le L\|\boldsymbol{x}-\boldsymbol{y}\|_2,$$

that completes the proof. □

Example 2.25. *Consider the guessing game of the first chapter, we can solve easily it with Online Gradient Descent. Indeed, we just need to calculate the gradients, prove that they are bounded, and find a way to calculate the projection of a real number in $[0,1]$ So, $\ell_t'(x)=2(x-y_t)$, that is bounded for $x,y_t\in[0,1]$. The projection on $[0,1]$ is just $\Pi_{[0,1]}(x)=\min(\max(x,0),1)$. With the optimal learning rate, the resulting regret would be $O(\sqrt{T})$, that is worse than the one we found in the first chapter.*

2.2.2 Analysis with Subgradients

As I promised you, with the proper mathematical tools, the analyzing online algorithms becomes easy. Indeed, switching from gradient to subgradient comes for free! In fact, our analysis of OGD with differentiable losses holds as is using subgradients instead of gradients. The reason is that the only property of the gradients that we used in the proof of Theorem 2.13 was that

$$\ell_t(\boldsymbol{x})-\ell_t(\boldsymbol{u})\le\langle \boldsymbol{g}_t,\boldsymbol{x}_t-\boldsymbol{u}\rangle,$$

where $\boldsymbol{g}_t=\nabla\ell_t(\boldsymbol{x}_t)$. However, the exact same property holds when $\boldsymbol{g}_t\in\partial\ell_t(\boldsymbol{x}_t)$. So, we can state the Online Subgradient descent algorithm in the following way, where the only difference is line 4.

Also, the regret bounds we proved hold as well, just changing differentiability with subdifferentiability and gradients with subgradients. In particular, we have the following Lemma.

Algorithm 2.2 Projected Online Subgradient Descent

Require: Non-empty closed convex set $V \subseteq \mathbb{R}^d$, $\boldsymbol{x}_1 \in V$, $\eta_1, \ldots, \eta_T > 0$

1: **for** $t = 1$ **to** T **do**
2: Output $\boldsymbol{x}_t \in V$
3: Receive $\ell_t : V \to \mathbb{R}$ subdifferentiable in V and pay $\ell_t(\boldsymbol{x}_t)$
4: Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
5: $\boldsymbol{x}_{t+1} = \Pi_V(\boldsymbol{x}_t - \eta_t \boldsymbol{g}_t) = \operatorname{argmin}_{\boldsymbol{y} \in V} \|\boldsymbol{x}_t - \eta_t \boldsymbol{g}_t - \boldsymbol{y}\|_2$
6: **end for**

Lemma 2.26. *Let $V \subseteq \mathbb{R}^d$ a non-empty closed convex set and $\ell_t : V \to \mathbb{R}$ a convex function subdifferentiable in V. Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$. Then, $\forall \boldsymbol{u} \in V$, the following inequality holds*

$$\eta_t(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \eta_t \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq \frac{1}{2}\|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 - \frac{1}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2 + \frac{\eta_t^2}{2}\|\boldsymbol{g}_t\|_2^2 \,.$$

Example 2.27. *Consider again the guessing game of the first class, but now change the loss function to the absolute loss of the difference: $\ell_t(x) = |x - y_t|$. Now we will need to use Online Subgradient Descent, because the functions are non-differentiable. We can easily see that*

$$\partial \ell_t(x) = \begin{cases} \{1\}, & x > y_t \\ [-1, 1], & x = y_t \\ \{-1\}, & x < y_t \,. \end{cases}$$

Again, running Online Subgradient Descent with the optimal learning rate on this problem will give us immediately a regret of $O(\sqrt{T})$, without having to design a specific strategy for it.

2.3 From Convex Losses to Linear Losses

Let's take a deeper look at this step

$$\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u}) \leq \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle, \ \forall \boldsymbol{u} \in \mathbb{R}^d \,.$$

Summing over time, we have

$$\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u}) \leq \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle, \ \forall \boldsymbol{u} \in \mathbb{R}^d \,.$$

Now, define the linear (and convex) losses $\tilde{\ell}_t(\boldsymbol{x}) := \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle$, so we have

$$\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u}) \leq \sum_{t=1}^T \tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_t(\boldsymbol{u}) \,.$$

This is more powerful that what it seems: We upper bounded the regret with respect to the convex losses ℓ_t with a regret with respect to another sequence of linear losses. This is important because it implies that we can build online algorithms that deal only with linear losses, and through the reduction above they can be seamlessly used as OCO algorithms! Note that this not imply that this reduction is always optimal, as we saw in Example 2.25. But, it allows us to easily construct optimal OCO algorithms in many interesting cases.

So, we will often consider just the problem of minimizing the linear regret

$$\operatorname{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle, \ \forall \boldsymbol{u} \in V \subseteq \mathbb{R}^d \,.$$

This problem is called **Online Linear Optimization** (OLO).

2.4 History Bits

The concept of subgradients and the possibility to build a calculus for them appear for the first time in Rockafellar [1963]. The Projected Online Subgradient Descent with time-varying learning rate and the name "Online Convex Optimization" was introduced by Zinkevich [2003], but the framework was introduced earlier by Gordon [1999a,b]. Before the online convex optimization framework, the online learning community focused on specific losses and mostly linear predictors, see Cesa-Bianchi and Lugosi [2006]. Moreover, the concept of subgradient is also a recent addition to the online learning literature, even if it was implicitly used in previous analysis, see for example Cesa-Bianchi [1999, Theorem 4]. Note that, if we consider the optimization literature, the optimization people never restricted themselves to the bounded case.

2.5 Exercises

Problem 2.1. *Prove that* $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 2\sqrt{T} - 1$.

Problem 2.2. *Using the inequality in the previous exercise, prove that a learning rate* $\eta_t \propto \frac{1}{\sqrt{t}}$ *gives rise to a regret only a constant multiplicative factor worse than the one in* (2.1)*.*

Problem 2.3. *Calculate the subdifferential set of the* ϵ*-insensitive loss:* $f(x) = \max(|x - y| - \epsilon, 0)$*. It is a loss used in regression problems where we do not want to penalize predictions* x *within* $\pm\epsilon$ *of the correct value* y*.*

Problem 2.4. *Using the definition of subgradient, find the subdifferential set of* $f(\boldsymbol{x}) = \|\boldsymbol{x}\|_2$*,* $\boldsymbol{x} \in \mathbb{R}^d$*.*

Problem 2.5. *Consider Projected Online Subgradient Descent for the Example 2.10 on the failure of Follow-the-Leader: Can we use it on that problem? Would it guarantee sublinear regret? How the behaviour of the algorithm would differ from FTL?*

Chapter 3

Online-to-Batch Conversion

It is a good moment to take a break from online learning theory and see some application of online learning to other domains. For example, we may wonder what is the connection between online learning and stochastic optimization. Given that Projected Online (Sub)Gradient Descent looks basically the same as Projected Stochastic (Sub)Gradient Descent, they must have something in common. Indeed, we can show that, for example, we can reduce stochastic optimization of convex functions to OCO. Let's see how.

Theorem 3.1. *Let V a non-empty closed convex set of $\mathbb{R}^d$, $F(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x}, \boldsymbol{\xi})]$ where the expectation is w.r.t. $\boldsymbol{\xi}$ drawn from ρ over some vector space X, and $f : V \times X \to \mathbb{R}$ is convex and subdifferentiable in the first argument in V. Draw T samples $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_T$ i.i.d. from ρ and construct the sequence of losses $\ell_t(\boldsymbol{x}) = \alpha_t f(\boldsymbol{x}, \boldsymbol{\xi}_t)$, where $\alpha_t > 0$ are deterministic. Run any OCO algorithm over the losses ℓ_t, to construct the sequence of predictions $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{T+1}$. Then, we have*

$$\mathbb{E}\left[F\left(\frac{1}{\sum_{t=1}^T \alpha_t}\sum_{t=1}^T \alpha_t \boldsymbol{x}_t\right)\right] \leq F(\boldsymbol{u}) + \frac{\mathbb{E}[\mathrm{Regret}_T(\boldsymbol{u})]}{\sum_{t=1}^T \alpha_t}, \ \forall \boldsymbol{u} \in \mathbb{R}^d,$$

where the expectation is with respect to $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_T$.

Proof. We first show that

$$\mathbb{E}\left[\sum_{t=1}^T \alpha_t F(\boldsymbol{x}_t)\right] = \mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{x}_t)\right] . \tag{3.1}$$

In fact, from the linearity of the expectation we have

$$\mathbb{E}\left[\sum_{t=1}^T \ell_t(\boldsymbol{x}_t)\right] = \sum_{t=1}^T \mathbb{E}\left[\ell_t(\boldsymbol{x}_t)\right] .$$

Then, from the law of total expectation, we have

$$\mathbb{E}\left[\ell_t(\boldsymbol{x}_t)\right] = \mathbb{E}\left[\mathbb{E}[\ell_t(\boldsymbol{x}_t)|\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_{t-1}\right] = \mathbb{E}\left[\mathbb{E}[\alpha_t f(\boldsymbol{x}_t, \boldsymbol{\xi}_t)|\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_{t-1}]\right] = \mathbb{E}\left[\alpha_t F(\boldsymbol{x}_t)\right],$$

where we used the fact that $\boldsymbol{x}_t$ depends only on $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_{t-1}$. Hence, (3.1) is proved.

It remains only to use Jensen's inequality, using the fact that F is convex, to have

$$F\left(\frac{1}{\sum_{t=1}^T \alpha_t}\sum_{t=1}^T \alpha_t \boldsymbol{x}_t\right) \leq \frac{1}{\sum_{t=1}^T \alpha_t}\sum_{t=1}^T \alpha_t F(\boldsymbol{x}_t) .$$

Dividing the regret by $\sum_{t=1}^T \alpha_t$ and using the above inequalities gives the stated theorem. □

Let's see now some applications of this result: Let's see how to use the above theorem to transform Online Subgradient Descent in Stochastic Subgradient Descent to minimize the training error of a classifier.

Example 3.2. *Consider a problem of binary classification, with inputs $\boldsymbol{z}_i \in \mathbb{R}^d$ and outputs $y_i \in \{-1, 1\}$. The loss function is the hinge loss: $f(\boldsymbol{x}, (\boldsymbol{z}, y)) = \max(1 - y\langle \boldsymbol{z}, \boldsymbol{x}\rangle, 0)$. Suppose that you want to minimize the training error over a training set of N samples, $\{(\boldsymbol{z}_i, y_i)\}_{i=1}^N$. Also, assume the maximum L_2 norm of the samples is R. That is, we want to minimize*

$$\min_{\boldsymbol{x}} \; F(\boldsymbol{x}) := \frac{1}{N} \sum_{i=1}^{N} \max(1 - y_i \langle \boldsymbol{z}_i, \boldsymbol{x}\rangle, 0) \; .$$

Run the reduction described in Theorem 3.1 for T iterations using OGD. In each iteration, construct $\ell_t(\boldsymbol{x}) = \max(1 - y_t\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle, 0)$ sampling a training point uniformly at random from 1 to N. Set $\boldsymbol{x}_1 = \boldsymbol{0}$ and $\eta = \frac{1}{R\sqrt{T}}$. We have that

$$\mathbb{E}\left[F\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t\right)\right] - F(\boldsymbol{x}^\star) \leq R\frac{\|\boldsymbol{x}^\star\|_2^2 + 1}{2\sqrt{T}} \; .$$

In words, we used an OCO algorithm to stochastically optimize a function, transforming the regret guarantee into a convergence rate guarantee.

In this last example, we have to use a constant learning rate to be able to minimize the training error over the entire space $\mathbb{R}^d$. In the next one, we will see a different approach, that allows to use a varying learning rate without the need of a bounded feasible set.

Example 3.3. *Consider the same setting of the previous example, and let's change the way in which we construct the online losses. Now use $\ell_t(\boldsymbol{x}) = \frac{1}{R\sqrt{t}} \max(1 - y_t\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle, 0)$ and step size $\eta = 1$. Hence, we have*

$$\mathbb{E}\left[F\left(\sum_{t=1}^{T} \frac{1}{R\sqrt{t}}\boldsymbol{x}_t\right)\right] - F(\boldsymbol{x}^\star) \leq \frac{\|\boldsymbol{x}^\star\|_2^2}{2\sum_{t=1}^{T}\frac{1}{R\sqrt{t}}} + \frac{1}{2\sum_{t=1}^{T}\frac{1}{R\sqrt{t}}}\sum_{t=1}^{T}\frac{1}{t} \leq R\frac{\|\boldsymbol{x}^\star\|_2^2 + 1 + \ln T}{4\sqrt{T+1} - 4},$$

where we used $\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \geq 2\sqrt{T+1} - 2$.

I stressed the fact that the only meaningful way to define a regret is with respect to an arbitrary point in the feasible set. This is obvious in the case we consider unconstrained OLO, because the optimal competitor is unbounded. But, it is also true in unconstrained OCO. Let's see an example of this.

Example 3.4. *Consider a problem of binary classification, with inputs $\boldsymbol{z}_i \in \mathbb{R}^d$ and outputs $y_i \in \{-1, 1\}$. The loss function is the logistic loss: $f(\boldsymbol{x}, (\boldsymbol{z}, y)) = \ln(1 + \exp(-y\langle \boldsymbol{z}, \boldsymbol{x}\rangle))$. Suppose that you want to minimize the training error over a training set of N samples, $\{(\boldsymbol{z}_i, y_i)\}_{i=1}^N$. Also, assume the maximum L_2 norm of the samples is R. That is, we want to minimize*

$$\min_{\boldsymbol{x}} \; F(\boldsymbol{x}) := \frac{1}{N} \sum_{i=1}^{N} \ln(1 + \exp(-y_i \langle \boldsymbol{z}_i, \boldsymbol{x}\rangle)) \; .$$

So, run the reduction described in Theorem 3.1 for T iterations using OSD. In each iteration, construct $\ell_t(\boldsymbol{x}) = \ln(1 + \exp(-y_t\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle))$ sampling a training point uniformly at random from 1 to N. Set $\boldsymbol{x}_1 = \boldsymbol{0}$ and $\eta = \frac{1}{R\sqrt{T}}$. We have that

$$\mathbb{E}\left[F\left(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t\right)\right] \leq \frac{R}{2\sqrt{T}} + \min_{\boldsymbol{u}\in\mathbb{R}^d} F(\boldsymbol{u}) + R\frac{\|\boldsymbol{u}\|^2}{2\sqrt{T}} \; .$$

In words, we will be $\frac{R}{2\sqrt{T}}$ away from the optimal value of regularized empirical risk minimization problem, where the weight of the regularization is $\frac{R}{2\sqrt{T}}$. Now, let's consider the case that the training set is linearly separable, this means that the infimum of F is 0 and the optimal solution does not exist, i.e., it has norm equal to infinity. So, any convergence guarantee that depends on $\boldsymbol{x}^\star$ would be vacuous. On the other hand, our guarantee above still makes perfectly sense.

Note that the above examples only deal with training error. However, in the next section we show a more interesting application of the online-to-batch conversion, that is to directly minimize the generalization error. Moreover, we will show guarantees in high probability, rather than just in expectation.

3.1 Agnostic PAC Learning

In this section, we show another application of online-to-batch methods to obtain statistical learning guarantees. So, we now consider a different setting from what we have seen till now. We assume that we have a prediction strategy $\phi_{\boldsymbol{x}}$ parametrized by a vector $\boldsymbol{x}$ and we want to learn the relationship between an input $\boldsymbol{z}$ and its associated label y. Moreover, we will assume that $(\boldsymbol{z}, y)$ is drawn from a joint probability distribution ρ. Also, we are equipped with a loss function that measures how good is our prediction $\hat{y} = \phi_{\boldsymbol{x}}(\boldsymbol{z})$ compared to the true label y, that is $\ell(\hat{y}, y)$. So, learning the relationship can be cast as minimizing the expected loss of our predictor

$$\min_{\boldsymbol{x} \in V} \mathbb{E}_{(\boldsymbol{z},y)\sim\rho}[\ell(\phi_{\boldsymbol{x}}(\boldsymbol{z}), y)] \, .$$

In machine learning terms, the object above is nothing else than the *test error* of our predictor.

Note that the above setting assumes labeled samples, but we can generalize it even more considering the *Vapnik's general setting of learning*, where we collapse the prediction function and the loss in a unique function. This allows, for example, to treat supervised and unsupervised learning in the same unified way. So, we want to minimize the *risk*

$$\min_{\boldsymbol{x} \in V} \left(\mathrm{Risk}(\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{\xi}\sim\rho}[f(\boldsymbol{x}, \boldsymbol{\xi})]\right),$$

where ρ is an unknown distribution over D and $f : \mathbb{R}^d \times D \to \mathbb{R}$ is measurable w.r.t. the second argument. Also, the set $\mathbb{F}$ of all predictors that can be expressed by vectors $\boldsymbol{x}$ in V is called the *hypothesis class*.

Example 3.5. *In a linear regression task where the loss is the square loss, we have $\boldsymbol{\xi} = (\boldsymbol{z}, y) \in \mathbb{R}^d \times \mathbb{R}$ and $\phi_{\boldsymbol{x}}(\boldsymbol{z}) = \langle \boldsymbol{z}, \boldsymbol{x} \rangle$. Hence, $f(\boldsymbol{x}, \boldsymbol{\xi}) = (\langle \boldsymbol{z}, \boldsymbol{x} \rangle - y)^2$.*

Example 3.6. *In linear binary classification where the loss is the hinge loss, we have $\boldsymbol{\xi} = (\boldsymbol{z}, y) \in \mathbb{R}^d \times \{-1, 1\}$ and $\phi_{\boldsymbol{x}}(\boldsymbol{z}) = \langle \boldsymbol{z}, \boldsymbol{x} \rangle$. Hence, $f(\boldsymbol{x}, \boldsymbol{\xi}) = \max(1 - y\langle \boldsymbol{z}, \boldsymbol{x} \rangle, 0)$.*

Example 3.7. *In binary classification with a neural network with the logistic loss, we have $\boldsymbol{\xi} = (\boldsymbol{x}, y) \in \mathbb{R}^d \times \{-1, 1\}$ and $\phi_{\boldsymbol{x}}$ is the network corresponding to the weights $\boldsymbol{x}$. Hence, $f(\boldsymbol{x}, \boldsymbol{\xi}) = \ln(1 + \exp(-y\phi_{\boldsymbol{x}}(\boldsymbol{z})))$.*

The key difficulty of the above problem is that we do not know the distribution ρ. Hence, there is no hope to exactly solve this problem. Instead, we are interested in understanding *what is the best we can do if we have access to T samples drawn i.i.d. from ρ*. More in details, we want to upper bound the *excess risk*

$$\mathrm{Risk}(\boldsymbol{x}_T) - \min_{\boldsymbol{x}} \mathrm{Risk}(\boldsymbol{x}),$$

where $\boldsymbol{x}_T$ is a predictor that was *learned* using T samples.

It should be clear that this is just an optimization problem and the one above is just the suboptimality gap. In this view, the objective of machine learning can be considered as a particular optimization problem.

Remark 3.8. *Note that this is not the only way to approach the problem of learning. Indeed, the regret minimization model is an alternative model to learning. Moreover, another approach would be to try to estimate the distribution ρ and then solve the risk minimization problem. No approach is superior to the other and each of them has its pros and cons.*

Given that we have access to the distribution ρ through samples drawn from it, any procedure we might think to use to minimize the risk will be stochastic in nature. This means that we cannot assure a deterministic guarantee. Instead, *we can try to prove that with high probability our minimization procedure will return a solution that is close to the minimizer of the risk*. It is also intuitive that the precision and probability we can guarantee must depend on how many samples we draw from ρ.

Quantifying the dependency of precision and probability of failure on the number of samples used is the objective of the **Agnostic Probably Approximately Correct** (PAC) framework, where the keyword "agnostic" refers to the fact that we do not assume anything on the best possible predictor. More in details, given a precision parameter ϵ and a probability of failure δ, we are interested in characterizing the *sample complexity of the hypothesis class* $\mathbb{F}$ that is defined as the number of samples T necessary to guarantee with probability at least $1 - \delta$ that the best learning

algorithm using the hypothesis class $\mathbb{F}$ outputs a solution $\boldsymbol{x}_T$ that has an excess risk upper bounded by ϵ. Note that the sample complexity does not depend on ρ, so it is a worst-case measure w.r.t. all the possible distributions. This makes sense if you think that we know nothing about the distribution ρ, so if your guarantee holds for the worst distribution it will also hold for any other distribution. Mathematically, we will say that the hypothesis class is agnostic PAC-learnable is such sample complexity function exists.

Definition 3.9 (Agnostic-PAC-learnable)**.** *We will say that a function class $F = \{f(\boldsymbol{x}, \cdot) : \boldsymbol{x} \in \mathbb{R}^d\}$ is* Agnostic-PAC-learnable *if there exists an algorithm $\mathcal{A}$ and a function $T(\epsilon, \delta) : \mathbb{R} \times [0, 1] \to \mathbb{N}$ such that when $\mathcal{A}$ is used with $T \geq T(\epsilon, \delta)$ samples drawn from ρ, with probability at least $1 - \delta$ the solution $\boldsymbol{x}_T$ returned by the algorithm has excess risk at most ϵ.*

Note that the Agnostic PAC learning setting does not say what is the procedure we should follow to find such sample complexity. The approach most commonly used in machine learning to solve the learning problem is the so-called *Empirical Risk Minimization (ERM) problem*. It consist of drawing T samples i.i.d. from ρ and minimizing the *empirical risk* defined as

$$\widehat{\mathrm{Risk}}(\boldsymbol{x}) := \min_{\boldsymbol{x} \in V} \frac{1}{T} \sum_{t=1}^{T} f(\boldsymbol{x}; \boldsymbol{\xi}_t) \, .$$

The minimizer $\hat{\boldsymbol{x}}_T$ is called the *empirical risk minimizer*. In words, ERM is nothing else than the minimization of the some loss function on a training set. However, in many interesting cases we can have that $\mathrm{argmin}_{\boldsymbol{x} \in V} \frac{1}{T} \sum_{t=1}^{T} f(\boldsymbol{x}; \boldsymbol{\xi}_t)$ can be very far from the true optimum $\mathrm{argmin}_{\boldsymbol{x} \in V} \mathbb{E}[f(\boldsymbol{x}; \boldsymbol{\xi})]$, even with an infinite number of samples! So, we need to modify the ERM formulation in some way, e.g., using a *regularization* term or a Bayesian prior of $\boldsymbol{x}$, or find conditions under which ERM works.

It is worth stressing that sometimes people are concerned with the difference between the training error and the test error of the trained predictor, i.e., $\widehat{\mathrm{Risk}}(\hat{\boldsymbol{x}}_T) - \mathrm{Risk}(\hat{\boldsymbol{x}}_T)$. However, this gap can be large without implying anything on the risk of the trained predictor.

The ERM approach is so widespread that machine learning itself is often wrongly identified with some kind of minimization of the training error. We now show that ERM is not the entire world of ML, showing that *the existence of a no-regret algorithm, that is an online learning algorithm with sublinear regret, guarantee Agnostic-PAC learnability*. More in details, we will show that an online algorithm with sublinear regret can be used to solve machine learning problems. This is not just a curiosity, for example this gives rise to computationally efficient parameter-free algorithms, that can be achieved through ERM only running a two-step procedure, i.e., running ERM with different parameters and selecting the best solution among them.

We already mentioned this possibility when we talked about the online-to-batch conversion, but this time we will strengthen it proving high probability guarantees rather than expectation ones.

So, we need some more bits on concentration inequalities.

3.2 Bits on Concentration Inequalities

We will use a concentration inequality to prove the high probability guarantee, but we will need to go beyond the sum of i.i.d.random variables. In particular, we will use the concept of *martingales*.

Definition 3.10 (Martingale)**.** *A sequence of random variables $Z_1, Z_2, \ldots$ is called a **martingale** if for all $t \geq 1$ it satisfies:*

$$\mathbb{E}[|Z_t|] < \infty, \qquad \mathbb{E}[Z_{t+1} | Z_1, \ldots, Z_t] = Z_t \, .$$

Definition 3.11 (Supermartingale)**.** *A sequence of random variables $Z_1, Z_2, \ldots$ is called a **supermartingale** if for all $t \geq 1$ it satisfies:*

$$\mathbb{E}[|Z_t|] < \infty, \qquad \mathbb{E}[Z_{t+1} | Z_1, \ldots, Z_t] \leq Z_t \, .$$

Example 3.12. *Consider a fair coin c_t and a betting algorithm that bets $|x_t|$ money on each round on the side of the coin equal to* $\mathrm{sign}(x_t)$*. We win or lose money 1:1, so the total money we won up to round t is $Z_t = \sum_{i=1}^t c_i x_i$. $Z_1, \dots, Z_t$ is a martingale. Indeed, we have*

$$\mathbb{E}[Z_t | Z_1, \dots, Z_{t-1}] = \mathbb{E}[Z_{t-1} + x_t c_t | Z_1, \dots, Z_{t-1}] = Z_{t-1} + \mathbb{E}[x_t c_t | Z_1, \dots, Z_{t-1}] = 0 \, .$$

If we throw away part of the wealth in each round, we obtain a supermartingale.

For bounded martingales we can prove high probability guarantees as for bounded i.i.d. random variables. The following Theorem will be the key result we will need.

Theorem 3.13 (Hoeffding-Azuma inequality)**.** *Let $Z_1, \dots, Z_T$ be a martingale of T random variables that satisfy $|Z_t - Z_{t+1}| \leq B, t = 1, \dots, T-1$ almost surely. Then, we have*

$$\mathbb{P}\{Z_T - Z_0 \geq \epsilon\} \leq \exp\left(-\frac{\epsilon^2}{2B^2 T}\right) \, .$$

Also, the same upper bounds hold on $\mathbb{P}\{Z_0 - Z_T \geq \epsilon\}$.

3.3 From Regret to Agnostic PAC

We now show how the online-to-batch conversion we introduced before gives us high probability guarantee for our machine learning problem.

Theorem 3.14. *Let $V \subseteq \mathbb{R}^d$, $\mathrm{Risk}(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x}, \boldsymbol{\xi})]$, where the expectation is w.r.t. $\boldsymbol{\xi}$ drawn from ρ with support over some vector space D, and $f : V \times D \to [0, 1]$. Draw T samples $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_T$ i.i.d. from ρ and construct the sequence of losses $\ell_t(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{\xi}_t)$. Let $\mathcal{A}$ any online learning algorithm over the losses ℓ_t that outputs the sequence of predictions $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{T+1}$ and guarantees $\mathrm{Regret}_T(\boldsymbol{u}) \leq R(\boldsymbol{u}, T)$ for all $\boldsymbol{u} \in V$, for a function $R : V \times \mathbb{N} \to \mathbb{R}$. Then, we have with probability at least $1 - \delta$, it holds that*

$$\frac{1}{T}\sum_{t=1}^T \mathrm{Risk}(\boldsymbol{x}_t) \leq \min_{\boldsymbol{u} \in V} \mathrm{Risk}(\boldsymbol{u}) + \frac{R(\boldsymbol{u}, T)}{T} + 2\sqrt{\frac{2 \ln \frac{2}{\delta}}{T}} \, .$$

Proof. Define $Z_t = \sum_{i=1}^t (\mathrm{Risk}(\boldsymbol{x}_i) - \ell_i(\boldsymbol{x}_i))$. We claim that Z_t is a martingale. In fact, we have

$$\mathbb{E}[\ell_t(\boldsymbol{x}_t) | \boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_{t-1}] = \mathbb{E}[f(\boldsymbol{x}_t, \boldsymbol{\xi}_t) | \boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_{t-1}] = \mathrm{Risk}(\boldsymbol{x}_t),$$

where we used the fact that $\boldsymbol{x}_t$ depends only on $\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_{t-1}$ Hence, we have

$$\mathbb{E}[Z_{t+1} | Z_1, \dots, Z_t] = Z_t + \mathbb{E}[\mathrm{Risk}(\boldsymbol{x}_{t+1}) - \ell_{t+1}(\boldsymbol{x}_{t+1}) | Z_1, \dots, Z_t] = Z_t,$$

that proves our claim.

Hence, using Theorem 3.13, we have

$$\mathbb{P}\left\{\sum_{t=1}^T (\mathrm{Risk}(\boldsymbol{x}_t) - \ell_t(\boldsymbol{x}_t)) \geq \epsilon\right\} = \mathbb{P}\left\{Z_T - Z_0 \geq \epsilon\right\} \leq \exp\left(-\frac{\epsilon^2}{2T}\right) \, .$$

This implies that, with probability at least $1 - \delta/2$, we have

$$\sum_{t=1}^T \mathrm{Risk}(\boldsymbol{x}_t) \leq \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) + \sqrt{2T \ln \frac{2}{\delta}} \, .$$

or equivalently

$$\frac{1}{T}\sum_{t=1}^T \mathrm{Risk}(\boldsymbol{x}_t) \leq \frac{1}{T}\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) + \sqrt{\frac{2 \ln \frac{2}{\delta}}{T}} \, .$$

We now use the definition of regret w.r.t. any $\boldsymbol{u}$, to have

$$\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{x}_t) = \frac{\text{Regret}_T(\boldsymbol{u})}{T} + \frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{u}) \leq \frac{R(\boldsymbol{u},T)}{T} + \frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{u})\ .$$

The last step is to upper bound with high probability $\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{u})$ with $\text{Risk}(\boldsymbol{u})$. This is easier than the previous upper bound because we set $\boldsymbol{u}$ to be the fixed vector that minimizes $\text{Risk}(\boldsymbol{x})+\frac{R(\boldsymbol{x},T)}{T}$ in V. So, $\ell_t(\boldsymbol{u})$ are i.i.d. random variables and for sure $Z_t = \sum_{i=1}^{t}\left(\text{Risk}(\boldsymbol{u}) - \ell_i(\boldsymbol{u})\right)$ forms a martingale. So, reasoning as above, we have that with probability at least $1-\delta/2$ it holds that

$$\frac{1}{T}\sum_{t=1}^{T}\ell_t(\boldsymbol{u}) \leq \text{Risk}(\boldsymbol{u}) + \sqrt{\frac{2\ln\frac{2}{\delta}}{T}}\ .$$

Putting all together and using the union bound, we have the stated bound. □

The theorem above upper bounds the average risk of the T predictors, while we are interested in producing a single predictor. If the risk is a convex function and V is convex, than we can lower bound the l.h.s. of the inequalities in the theorem with the risk evaluated on the average of the $\boldsymbol{x}_t$. That is

$$\text{Risk}\left(\frac{1}{T}\sum_{t=1}^{T}\boldsymbol{x}_t\right) \leq \frac{1}{T}\sum_{t=1}^{T}\text{Risk}(\boldsymbol{x}_t)\ .$$

If the risk is not a convex function, we need a way to generate a single solution with small risk. One possibility is to construct a *stochastic classifier* that samples one of the $\boldsymbol{x}_t$ with uniform probability and predicts with it. For this classifier, we immediately have

$$\text{Risk}(\{\boldsymbol{x}_1,\dots,\boldsymbol{x}_T\}) = \frac{1}{T}\sum_{t=1}^{T}\text{Risk}(\boldsymbol{x}_t),$$

where the expectation in the definition of the risk of the stochastic classifier is also with respect to the random index.

Yet another way, is to select among the T predictors, the one with the smallest risk. This works because the average is lower bounded by the minimum. This is easily achieved using $T/2$ samples for the online learning procedure and $T/2$ samples to generate a validation set to evaluate the solution and pick the best one. The following Theorem shows that selecting the predictor with the smallest empirical risk on a validation set will give us a predictor close to the best one with high probability.

Theorem 3.15. *Let $V \subseteq \mathbb{R}^d$, $\text{Risk}(\boldsymbol{x}) = \mathbb{E}[f(\boldsymbol{x},\boldsymbol{\xi})]$, where the expectation is w.r.t. $\boldsymbol{\xi}$ drawn from ρ with support over some vector space D, and $f : V \times D \to [0,1]$. We have a finite set of vectors $S = \{\boldsymbol{x}_1,\dots,\boldsymbol{x}_{|S|}\}$ and a T random vectors $\boldsymbol{\xi}_1,\dots,\boldsymbol{\xi}_T$ drawn i.i.d. from ρ. Denote by $\hat{\boldsymbol{x}} = \text{argmin}_{\boldsymbol{x}\in S}\ \widehat{\text{Risk}}(\boldsymbol{x})$, where $\widehat{\text{Risk}}(\boldsymbol{x}) = \frac{1}{T}\sum_{t=1}^{T}f(\boldsymbol{x},\boldsymbol{\xi}_t)$. Then, with probability at least $1-\delta$, we have*

$$\text{Risk}(\hat{\boldsymbol{x}}) \leq \min_{\boldsymbol{x}\in S}\ \text{Risk}(\boldsymbol{x}) + 2\sqrt{\frac{2\ln(2|S|/\delta)}{T}}\ .$$

Proof. We want to calculate the probability that the hypothesis that minimizes the validation error is far from the best hypothesis in the set. We cannot do it directly because we do not have the required independence to use a concentration. Instead, *we will upper bound the probability that there exists at least one function whose empirical risk is far from the risk.* So, using the union bound, we have

$$\mathbb{P}\left\{\exists\boldsymbol{x}\in S : |\,\text{Risk}(\boldsymbol{x}) - \widehat{\text{Risk}}(\boldsymbol{x})| > \frac{\epsilon}{2}\right\} \leq \sum_{i=1}^{|S|}\mathbb{P}\left\{|\,\text{Risk}(\boldsymbol{x}_i) - \widehat{\text{Risk}}(\boldsymbol{x}_i)| > \frac{\epsilon}{2}\right\} \leq 2|S|\exp\left(-\frac{\epsilon^2 T}{8}\right)\ .$$

Hence, with probability at least $1-\delta$, we have that

$$|\operatorname{Risk}(\boldsymbol{x}) - \widehat{\operatorname{Risk}}(\boldsymbol{x})| \leq \frac{\epsilon}{2}, \ \forall \boldsymbol{x} \in S,$$

where $\epsilon = 2\sqrt{\frac{2\ln(2|S|/\delta)}{T}}$.

We are now able to upper bound the risk of $\hat{\boldsymbol{x}}$, just using the fact that the above applies to $\hat{\boldsymbol{x}}$ too. Defining $\boldsymbol{x}^\star = \operatorname{argmin}_{\boldsymbol{x}\in S} \operatorname{Risk}(\boldsymbol{x})$, we have

$$\operatorname{Risk}(\hat{\boldsymbol{x}}) \leq \widehat{\operatorname{Risk}}(\hat{\boldsymbol{x}}) + \epsilon/2 \leq \widehat{\operatorname{Risk}}(\boldsymbol{x}^\star) + \epsilon/2 \leq \operatorname{Risk}(\boldsymbol{x}^\star) + \epsilon,$$

where in the last inequality we used the fact that $\hat{\boldsymbol{x}}$ minimizes the empirical risk. □

Using this theorem, we can use $T/2$ samples for the training and $T/2$ samples for the validation, where $T \geq 2$. Denoting by $\hat{x}_T$ the predictor with the best empirical risk on the validation set among the $T/2$ generated during the online procedure, we have with probability at least $1-2\delta$ that

$$\operatorname{Risk}(\hat{\boldsymbol{x}}_T) \leq \min_{\boldsymbol{u}\in V} \operatorname{Risk}(\boldsymbol{u}) + \frac{2R(\boldsymbol{u}, T/2)}{T} + 8\sqrt{\frac{\ln(T/\delta)}{T}}\,.$$

It is important to note that with any of the above three methods to select one $\boldsymbol{x}_t$ among the T generated by the online learning procedure, the sample complexity guarantee we get matches the one we would have obtained by ERM, up to polylogarithmic factors. In other words, there is nothing special about ERM compared to the online learning approach to statistical learning. Moreover, ERM implies the existence of a hypothetical procedure that perfectly minimizes the training error. In reality, we should take into account the optimization error in the analysis of ERM. On the other hand, in the online learning approach we have a guarantee directly for the computed solution.

Another important point is that the above guarantee does not imply the existence of online learning algorithms with sublinear regret for any learning problem. It just says that, if it exists, it can be used in the statistical setting too.

3.4 History Bits

The specific shape of Theorem 3.1 is new, but I would not be surprised if it appeared somewhere in the literature. In particular, the uniform averaging is from Cesa-Bianchi et al. [2004], but was proposed for the absolute loss in Blum et al. [1999]. The non-uniform averaging of Example 3.3 is from Zhang [2004], even if there it is not proposed explicitly as an online-to-batch conversion.

A more recent method to do online-to-batch conversion has been introduced in Cutkosky [2019a], that independently rediscovered and generalized the averaging method in Nesterov and Shikhman [2015]. This new method allows to prove the convergence of the last iterate rather than the one of the weighted average, with a small change in any online learning algorithm.

Theorem 3.14 is from Cesa-Bianchi et al. [2004], but here I used a second concentration to state it in terms of the competitor's true risk rather than its empirical risk. Theorem 3.15 is nothing else than the Agnostic PAC learning guarantee for ERM for hypothesis classes with finite cardinality. Cesa-Bianchi et al. [2004] gives also an alternative procedure to select a single hypothesis among the T generated during the online procedure that does not require splitting the data in training and validation. However, the obtained guarantee matches the one we have proved.

3.5 Exercises

Problem 3.1. *Implement the algorithm in Example 3.2 in any language you like: implementing an algorithm is the perfect way to see if you understood all the details of the algorithm.*

Chapter 4

Beyond $\sqrt{T}$ Regret

4.1 Strong Convexity and Online Subgradient Descent

Let's now go back to online convex optimization theory. The example in the first chapter showed us that it is possible to get logarithmic regret in time. However, we saw that we get only $\sqrt{T}$-regret with Online Subgradient Descent (OSD) on the same game. What is the reason? It turns out that the losses in the first game, $\ell_t(x) = (x - y_t)^2$ on $[0, 1]$, are not just Lipschitz. They also posses some *curvature* that can be exploited to achieve a better regret. In a moment we will see that the only change we will need to OSD is a different learning rate, dictated as usual by the regret analysis.

The key concept we will need is the one of *strong convexity*.

4.1.1 Convex Analysis Bits: Strong Convexity

Here, we introduce a stronger concept of convexity, that allows to build better lower bound to a function. Instead of the linear lower bound achievable through the use of subgradients, we will make use of *quadratic* lower bound.

Definition 4.1 (Strongly Convex Function)**.** *Let $\lambda \geq 0$. A proper function $f : \mathcal{X} \to (-\infty, +\infty]$ is λ-**strongly convex w.r.t.** $\|\cdot\|$ over a convex set $V \subseteq \operatorname{dom} f$ if*

$$f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}) \leq \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{y}) - \frac{1}{2}\lambda\alpha(1-\alpha)\|\boldsymbol{x} - \boldsymbol{y}\|^2,$$

for all $\boldsymbol{x}, \boldsymbol{y} \in V$ and all $\alpha \in (0, 1)$.

*We will also say that f is **strongly convex** in V, if there exists $\lambda > 0$ and a norm such that the above holds.*

From the definition, it is clear that if a function is λ-strongly convex, it is also λ'-strongly convex for any $0 \leq \lambda' < \lambda$. Moreover, 0-strong convexity is just the definition of convex function.

We can also obtain an equivalent characterization in terms of subgradients.

Lemma 4.2. *Let $\lambda \geq 0$. We have that $f : \mathcal{X} \to (-\infty, +\infty]$ is λ-strongly convex over a convex set $V \subseteq \operatorname{dom} \partial f$ w.r.t. $\|\cdot\|$ iff*

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

Proof. Let's first assume that f is λ-strongly convex over V w.r.t. $\|\cdot\|$. Then, for any $\alpha \in (0, 1)$ and any $\boldsymbol{g} \in \partial f(\boldsymbol{y})$, we have

$$\langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{y} \rangle \leq \frac{f(\alpha \boldsymbol{x} + (1-\alpha)\boldsymbol{y}) - f(\boldsymbol{y})}{\alpha} \leq f(\boldsymbol{x}) - f(\boldsymbol{y}) - \frac{1}{2}\lambda(1-\alpha)\|\boldsymbol{x} - \boldsymbol{y}\|^2 .$$

Taking the limit for α to 0, we obtain the statement.

Let's now assume that the inequality in the lemma holds and let's prove that f is λ-strongly convex. Setting $\boldsymbol{v} = \alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}$, for any $\boldsymbol{g} \in \partial f(\boldsymbol{v})$, we have

$$\langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{v} \rangle \leq f(\boldsymbol{x}) - f(\boldsymbol{v}) - \frac{\lambda}{2}\|\boldsymbol{x} - \boldsymbol{v}\|^2,$$
$$\langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{v} \rangle \leq f(\boldsymbol{y}) - f(\boldsymbol{v}) - \frac{\lambda}{2}\|\boldsymbol{y} - \boldsymbol{v}\|^2 .$$

Summing these two inequalities with coefficients α and $1-\alpha$, we have

$$\begin{aligned}
0 &= \langle \boldsymbol{g}, \alpha\boldsymbol{x} - \alpha\boldsymbol{v} + (1-\alpha)\boldsymbol{y} - (1-\alpha)\boldsymbol{v} \rangle \\
&\leq \alpha f(\boldsymbol{x}) - \alpha f(\boldsymbol{v}) - (1-\alpha) f(\boldsymbol{y}) - (1-\alpha) f(\boldsymbol{v}) - \alpha\frac{\lambda}{2}\|\boldsymbol{x} - \boldsymbol{v}\|^2 - (1-\alpha)\frac{\lambda}{2}\|\boldsymbol{y} - \boldsymbol{v}\|^2 \\
&= \alpha f(\boldsymbol{x}) - f(\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}) - (1-\alpha) f(\boldsymbol{y}) - \alpha\frac{\lambda}{2}\|\boldsymbol{x} - \alpha\boldsymbol{x} - (1-\alpha)\boldsymbol{y}\|^2 - (1-\alpha)\frac{\lambda}{2}\|\boldsymbol{y} - \alpha\boldsymbol{x} - (1-\alpha)\boldsymbol{y}\|^2 \\
&= \alpha f(\boldsymbol{x}) - f(\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}) - (1-\alpha) f(\boldsymbol{y}) - \alpha(1-\alpha)\frac{\lambda}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 .
\end{aligned}$$

□

In words, the lemma above tells us that a strongly convex function can be lower bounded by a quadratic, where the linear term is the usual one constructed through the subgradient, and the quadratic term depends on the strong convexity. Hence, we have a tighter lower bound to the function w.r.t. simply using convexity. This is what we would expect using a Taylor expansion on a twice-differentiable convex function and lower bounding the smallest eigenvalue of the Hessian. Indeed, we have the following Theorem.

Theorem 4.3. *Let $V \subseteq \mathbb{R}^d$ convex and $f : V \to \mathbb{R}$ twice differentiable in an open set containing V. Then f is λ-strongly convex with respect to $\|\cdot\|$ iff*

$$\langle \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \geq \lambda\|\boldsymbol{x} - \boldsymbol{y}\|^2 . \tag{4.1}$$

Moreover, a sufficient condition for λ-strong convexity in V w.r.t. $\|\cdot\|$ is that for all $\boldsymbol{x}, \boldsymbol{y}$ we have $\langle \nabla^2 f(\boldsymbol{x})\boldsymbol{y}, \boldsymbol{y} \rangle \geq \lambda\|\boldsymbol{y}\|^2$, where $\nabla^2 f(\boldsymbol{x})$ is the Hessian matrix of f at $\boldsymbol{x}$.

Proof. Assume that f is λ-strongly convex with respect to $\|\cdot\|$. Then, from Lemma 4.2, we have

$$\langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle \leq f(\boldsymbol{x}) - f(\boldsymbol{y}) - \frac{\lambda}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2,$$
$$\langle \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \leq f(\boldsymbol{y}) - f(\boldsymbol{x}) - \frac{\lambda}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2 .$$

Summing the two inequalities, we have the stated bound.

Now, assume that the (4.1) holds. Define $h(\alpha) = f(\boldsymbol{y} + \alpha(\boldsymbol{x} - \boldsymbol{y}))$, and denote $\boldsymbol{w}_\alpha = \boldsymbol{y} + \alpha(\boldsymbol{x} - \boldsymbol{y})$. Then,

$$h'(\alpha) - h'(0) = \langle \nabla f(\boldsymbol{w}_\alpha) - \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle = \frac{1}{\alpha}\langle \nabla f(\boldsymbol{w}_\alpha) - \nabla f(\boldsymbol{y}), \boldsymbol{w}_\alpha - \boldsymbol{y} \rangle .$$

Using (4.1), we obtain that

$$h'(\alpha) - h'(0) \geq \frac{\lambda}{\alpha}\|\boldsymbol{w} - \boldsymbol{y}\|^2 = \lambda\alpha\|\boldsymbol{x} - \boldsymbol{y}\|^2 .$$

Therefore, we have

$$f(\boldsymbol{x}) - f(\boldsymbol{y}) - \langle \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle = h(1) - h(0) - h'(0) = \int_0^1 (h'(\alpha) - h'(0))\,\mathrm{d}\alpha \geq \frac{\lambda}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2,$$

where in the second equality we used the fact that ∇f is continuous because f is twice differentiable. Hence, by Lemma 4.2, f is strongly convex. □

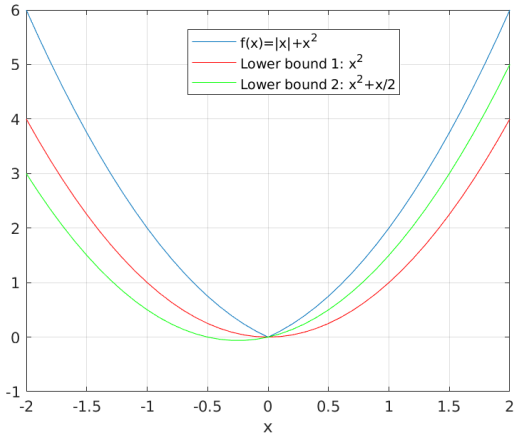

Figure 4.1: Possible lower bounds to the strongly convex non-differentiable function $f(x) = |x| + x^2$.

However, for a strongly convex function does not need to be twice differentiable. Indeed, we do not even need plain differentiability. Hence, the use of the subgradient implies that the quadratic lower bound does not have to be uniquely determined, as in the next Example.

Example 4.4. *Consider the strongly convex function $f(x) = |x| + x^2$. In Figure 4.1, we show two possible quadratic lower bounds to the function in $x = 0$.*

We also have the following useful property on the sum of strongly convex functions.

Theorem 4.5. *Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ be μ_1-strongly convex and $g : \mathbb{R}^d \to (-\infty, +\infty]$ a μ_2-strongly convex function in a non-empty convex set $V \subseteq \operatorname{int\,dom} f \cap \operatorname{int\,dom} g$ w.r.t. $\|\cdot\|$. Then, $f + g$ is $\mu_1 + \mu_2$-strongly convex in V w.r.t. $\|\cdot\|$.*

Proof. From the assumption on V and Theorem 2.18, we have that the subdifferential set of the sum is equal to the sum of the subdifferential sets. Hence, the proof is immediate from Lemma 4.2. □

Example 4.6. *Let $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$. Using Theorem 4.3, we have that f is 1-strongly convex w.r.t. $\|\cdot\|_2$ in $\mathbb{R}^d$.*

4.1.2 Online Subgradient Descent for Strongly Convex Losses

Theorem 4.7. *Let V a non-empty closed convex set in $\mathbb{R}^d$. Assume that the functions $\ell_t : V \to \mathbb{R}$ are μ_t-strongly convex w.r.t $\|\cdot\|_2$ and subdifferentiable in V, where $\mu_t > 0$. Use OSD in Algorithm 2.2 with stepsizes equal to $\eta_t = \frac{1}{\sum_{i=1}^t \mu_i}$. Then, for any $\boldsymbol{u} \in V$, we have the following regret guarantee*

$$\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \leq \frac{1}{2} \sum_{t=1}^T \frac{\|\boldsymbol{g}_t\|_2^2}{\sum_{i=1}^t \mu_i} \,.$$

Proof. From the assumption of μ_t-strong convexity of the functions ℓ_t, we have that

$$\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u}) \leq \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle - \frac{\mu_t}{2}\|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 \,.$$

From the fact that $\eta_t = \frac{1}{\sum_{i=1}^t \mu_i}$, we have

$$\begin{aligned} \frac{1}{2\eta_1} - \frac{\mu_1}{2} &= 0, \\ \frac{1}{2\eta_t} - \frac{\mu_t}{2} &= \frac{1}{2\eta_{t-1}}, \; t = 2, \dots, T \,. \end{aligned}$$

Hence, use Lemma 2.26 and sum from $t = 1, \dots, T$, to obtain

$$\begin{aligned}\sum_{t=1}^{T} (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\leq \sum_{t=1}^{T} \left(\frac{1}{2\eta_t} \|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 - \frac{1}{2\eta_t} \|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2 - \frac{\mu_t}{2} \|\boldsymbol{x}_t - \boldsymbol{u}\|^2 + \frac{\eta_t}{2} \|\boldsymbol{g}_t\|_2^2 \right) \\ &= -\frac{1}{2\eta_1} \|\boldsymbol{x}_2 - \boldsymbol{u}\|_2^2 + \sum_{t=2}^{T} \left(\frac{1}{2\eta_{t-1}} \|\boldsymbol{x}_t - \boldsymbol{u}\|_2^2 - \frac{1}{2\eta_t} \|\boldsymbol{x}_{t+1} - \boldsymbol{u}\|_2^2 \right) + \sum_{t=1}^{T} \frac{\eta_t}{2} \|\boldsymbol{g}_t\|_2^2 \,.\end{aligned}$$

Observing that the first sum on the left hand side is a telescopic sum, we have the stated bound. □

Remark 4.8. *Notice that the theorem requires a bounded domain, otherwise the loss functions will not be Lipschitz given that they are also strongly convex.*

Corollary 4.9. *Under the assumptions of Theorem 4.7, if in addiction we have $\mu_t = \mu > 0$ and ℓ_t is L-Lipschitz w.r.t. $\|\cdot\|_2$, for $t = 1, \dots, T$, then we have*

$$\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \leq \frac{L^2}{2\mu} (1 + \ln T) \,.$$

Remark 4.10. *Corollary 4.9* does not *imply that for any finite T the regret will be smaller than using learning rates $\propto \frac{1}{L\sqrt{t}}$. Instead, asymptotically the regret in Corollary 4.9 is always better than to one of OSD with Lipschitz losses.*

Example 4.11. *Consider once again the example in the first chapter: $\ell_t(x) = (x - y_t)^2$. Note that the loss functions are 2-strongly convex w.r.t. $|\cdot|$. Hence, setting $\eta_t = \frac{1}{2t}$ and $\ell_t'(x) = 2(x - y_t)$ gives a regret of $\ln(T) + 1$.*

Let's now use again the online-to-batch conversion on strongly convex stochastic problems.

Example 4.12. *As done before, we can use the online-to-batch conversion to use Corollary 4.9 to obtain stochastic subgradient descent algorithms for strongly convex stochastic functions. For example, consider the classic Support Vector Machine objective*

$$\min_{\boldsymbol{x}} \; F(\boldsymbol{x}) := \frac{\lambda}{2} \|\boldsymbol{x}\|_2^2 + \frac{1}{N} \sum_{i=1}^{N} \max(1 - y_i \langle \boldsymbol{z}_i, \boldsymbol{x} \rangle, 0),$$

or any other regularized formulation like regularized logistic regression:

$$\min_{\boldsymbol{x}} \; F(\boldsymbol{x}) := \frac{\lambda}{2} \|\boldsymbol{x}\|_2^2 + \frac{1}{N} \sum_{i=1}^{N} \ln(1 + \exp(-y_i \langle \boldsymbol{z}_i, \boldsymbol{x} \rangle)),$$

where $\boldsymbol{z}_i \in \mathbb{R}^d$, $\|\boldsymbol{z}_i\|_2 \leq R$, and $y_i \in \{-1, 1\}$. First, notice that the minimizer of both expressions has to be in the L_2 ball of radius proportional to $\sqrt{\frac{1}{\lambda}}$ (proof left as exercise). Hence, we can set V equal to this set. Then, setting $\ell_t(\boldsymbol{x}) = \frac{\lambda}{2} \|\boldsymbol{x}\|_2^2 + \max(1 - y_i \langle \boldsymbol{z}_i, \boldsymbol{x} \rangle, 0)$ or $\ell_t(\boldsymbol{x}) = \frac{\lambda}{2} \|\boldsymbol{x}\|_2^2 + \ln(1 + \exp(-y_i \langle \boldsymbol{z}_i, \boldsymbol{x} \rangle))$ results in λ-strongly convex loss functions. Using Corollary 4.9 and Theorem 3.1 gives immediately

$$\mathbb{E}\left[F\left(\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{x}_t \right) \right] - \min_{\boldsymbol{x}} \; F(\boldsymbol{x}) \leq O\left(\frac{\ln T}{\lambda T} \right) \,.$$

However, we can do better! Indeed, $\ell_t(\boldsymbol{x}) = \frac{\lambda t}{2} \|\boldsymbol{x}\|_2^2 + t \max(1 - y_i \langle \boldsymbol{z}_i, \boldsymbol{x} \rangle, 0)$ or $\ell_t(\boldsymbol{x}) = \frac{\lambda t}{2} \|\boldsymbol{x}\|_2^2 + t \ln(1 + \exp(-y_i \langle \boldsymbol{z}_i, \boldsymbol{x} \rangle))$ results in λt-strongly convex loss functions. Using Corollary 4.9, we have that $\eta_t = \frac{2}{\lambda t(t+1)}$ and Theorem 3.1 gives immediately

$$\mathbb{E}\left[F\left(\frac{2}{T(T+1)} \sum_{t=1}^{T} t \, \boldsymbol{x}_t \right) \right] - \min_{\boldsymbol{x}} \; F(\boldsymbol{x}) \leq O\left(\frac{1}{\lambda T} \right),$$

that is asymptotically better because it does not have the logarithmic term.

4.2 Adaptive Algorithms: $L^\star$ bounds and AdaGrad

In this section, we will explore a bit more under which conditions we can get better regret upper bounds than $O(DL\sqrt{T})$ as $T \to \infty$. Also, we will obtain this improved guarantees in an *automatic* way. That is, the algorithm will be *adaptive* to characteristics of the sequence of loss functions, without having to rely on information about the future.

4.2.1 Adaptive Learning Rates for Online Subgradient Descent

Consider the minimization of the linear regret

$$\text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \ .$$

Using Online Subgradient Descent (OSD), we said that the regret for bounded domains can be upper bounded by

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \leq \frac{D^2}{2\eta_T} + \frac{1}{2} \sum_{t=1}^{T} \eta_t \|\boldsymbol{g}_t\|_2^2 \ .$$

With a fixed learning rate, the learning rate that minimizes this upper bound on the regret is

$$\eta_T^\star = \frac{D}{\sqrt{\sum_{t=1}^{T} \|g_t\|_2^2}} \ .$$

Unfortunately, as we said, this learning rate cannot be used because it assumes the knowledge of the future rounds. However, we might be lucky and we might try to just approximate it in each round using the knowledge up to time t. That is, we might try to use

$$\eta_t = \frac{D}{\sqrt{\sum_{i=1}^{t} \|\boldsymbol{g}_i\|_2^2}}, \tag{4.2}$$

and just skip the rounds in which $\boldsymbol{g}_i = \boldsymbol{0}$ to avoid possible divisions by 0. Observe that $\eta_T = \eta_T^\star$, so the first term of the regret would be exactly what we need! For the other term, the optimal learning rate would give us

$$\frac{1}{2} \sum_{t=1}^{T} \eta_T^\star \|\boldsymbol{g}_t\|_2^2 = \frac{1}{2} D \sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2} \ .$$

Now let's see what we obtain with our approximation.

$$\frac{1}{2} \sum_{t=1}^{T} \eta_t \|\boldsymbol{g}_t\|^2 = \frac{1}{2} D \sum_{t=1}^{T} \frac{\|\boldsymbol{g}_t\|_2^2}{\sqrt{\sum_{i=1}^{t} \|\boldsymbol{g}_i\|_2^2}} \ .$$

We need a way to upper bound that sum. The way to treat these sums, as we did in other cases, is to try to approximate them with integrals. So, we can use the following very handy Lemma that generalizes a lot of similar specific ones.

Lemma 4.13. *Let $a_0 \geq 0$ and $f : [0, +\infty) \to [0, +\infty)$ a nonincreasing function. Then*

$$\sum_{t=1}^{T} a_t f\left(a_0 + \sum_{i=1}^{t} a_i\right) \leq \int_{a_0}^{\sum_{t=0}^{T} a_t} f(x) \, \mathrm{d}x \ .$$

Proof. Denote by $s_t = \sum_{i=0}^{t} a_i$.

$$a_t f\left(a_0 + \sum_{i=1}^{t} a_i\right) = a_t f(s_t) = \int_{s_{t-1}}^{s_t} f(s_t) dx \leq \int_{s_{t-1}}^{s_t} f(x) dx \ .$$

Summing over $t = 1, \cdots, T$, we have the stated bound. □

Using this Lemma, we have that

$$\frac{1}{2}D\sum_{t=1}^{T}\frac{\|\boldsymbol{g}_t\|_2^2}{\sqrt{\sum_{i=1}^{t}\|\boldsymbol{g}_i\|_2^2}} \leq D\sqrt{\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2}\,.$$

Surprisingly, this term is only a factor of 2 worse than what we would have got from the optimal choice of $\eta_T^\star$. However, this learning rate can be computed without knowledge of the future and it can actually be used! Overall, with this choice we get

$$\text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T}\ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T}\ell_t(\boldsymbol{u}) \leq \frac{3}{2}D\sqrt{\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2}\,. \tag{4.3}$$

Note that it is possible to improve the constant in front of the bound to $\sqrt{2}$ by multiplying the learning rates by $\frac{\sqrt{2}}{2}$. So, putting all together we have the following theorem.

Theorem 4.14. *Let $V \subseteq \mathbb{R}^d$ a closed non-empty convex set with diameter D, i.e., $\max_{\boldsymbol{x},\boldsymbol{y}\in V}\|\boldsymbol{x}-\boldsymbol{y}\|_2 \leq D$. Let $\ell_1,\dots,\ell_T$ an arbitrary sequence of convex functions $\ell_t : V \to \mathbb{R}$ subdifferentiable in V for $t = 1,\dots,T$. Pick any $\boldsymbol{x}_1 \in V$, set $\eta_t = \frac{\sqrt{2}D}{2\sqrt{\sum_{i=1}^{t}\|\boldsymbol{g}_i\|_2^2}}$, $t = 1,\dots,T$, and do not update on rounds when $\boldsymbol{g}_t = \boldsymbol{0}$. Then, $\forall \boldsymbol{u} \in V$, the following regret bound holds*

$$\text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \sqrt{2}D\sqrt{\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2} = \sqrt{2}\min_{\eta>0}\ \frac{D^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2\,.$$

The second equality in the theorem clearly show the advantage of this learning rates: We obtain (almost) the same guarantee we would have got knowing the future gradients!

This is an interesting result on its own: it gives a principled way to set the learning rates with an almost optimal guarantee. Moreover, it shows that the sum of the squared gradients act as *intrinsic notion of time*, better suited than T to capture the dependency on time. There are also other consequences of this simple regret: We will now specialize this result to the case that the losses are *smooth*.

4.2.2 Convex Analysis Bits: Dual Norms and Smooth Functions

We now consider a family of loss functions that have the characteristic of being lower bounded by the squared norm of the subgradient. We will also introduce the concept of *dual norms*. While dual norms are not strictly needed for this topic, they give more generality and at the same time they allows me to slowly introduce some of the concepts that will be needed for the chapter on Online Mirror Descent.

Definition 4.15 (Dual Norm)**.** *The **dual norm** $\|\cdot\|_\star$ of a norm $\|\cdot\|$ is defined as $\|\boldsymbol{\theta}\|_\star = \max_{\boldsymbol{x}:\|\boldsymbol{x}\|\leq 1}\langle\boldsymbol{\theta},\boldsymbol{x}\rangle$.*

Example 4.16. *The dual norm of the L_2 norm is the L_2 norm. We can easily prove it. First of all, if $\boldsymbol{\theta} = \boldsymbol{0}$, then the dual norm is 0 too. Hence, let's assume that $\boldsymbol{\theta} \neq \boldsymbol{0}$. Indeed, $\|\boldsymbol{\theta}\|_\star = \max_{\boldsymbol{x}:\|\boldsymbol{x}\|_2\leq 1}\langle\boldsymbol{\theta},\boldsymbol{x}\rangle \leq \|\boldsymbol{\theta}\|_2$ by Cauchy-Schwarz inequality. Also, set $\boldsymbol{v} = \frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|_2}$, so $\max_{\boldsymbol{x}:\|\boldsymbol{x}\|_2\leq 1}\langle\boldsymbol{\theta},\boldsymbol{x}\rangle \geq \langle\boldsymbol{\theta},\boldsymbol{v}\rangle = \|\boldsymbol{\theta}\|_2$.*

Example 4.17. *Let $p \geq 1$. The L_p norm of a vector $\boldsymbol{x} \in \mathbb{R}^d$ is defined as $\|\boldsymbol{x}\|_p = (\sum_{i=1}^{d}|x_i|^p)^{1/p}$. The dual norm is the q-norm where $\frac{1}{p}+\frac{1}{q}=1$. Note that the dual of the L_1 norm is the L_∞ norm, defined as $\|\boldsymbol{x}\|_\infty = \max_{i=1,\dots,d}|x_i|$. The proof is left to the reader.*

Example 4.18. *Let $\boldsymbol{A}$ be a positive definite matrix, then it is possible to show that $\|\boldsymbol{x}\|_{\boldsymbol{A}} := \sqrt{\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}}$ is a norm. The dual norm is $\|\boldsymbol{x}\|_{\boldsymbol{A}^{-1}} = \sqrt{\boldsymbol{x}^\top\boldsymbol{A}^{-1}\boldsymbol{x}}$. In fact, we have that the dual norm of $\|\cdot\|_{\boldsymbol{A}}$ is defined as*

$$\begin{aligned}\|\boldsymbol{\theta}\|_\star &= \max_{\boldsymbol{x}:\|\boldsymbol{x}\|_{\boldsymbol{A}}\leq 1}\langle\boldsymbol{\theta},\boldsymbol{x}\rangle = \max_{\boldsymbol{x}:\boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}\leq 1}\boldsymbol{\theta}^\top\boldsymbol{x} = \max_{\boldsymbol{y}:\boldsymbol{y}^\top\boldsymbol{y}\leq 1}\boldsymbol{\theta}^\top\boldsymbol{A}^{-1/2}\boldsymbol{y} = \max_{\boldsymbol{y}:\|\boldsymbol{y}\|_2\leq 1}(\boldsymbol{A}^{-1/2}\boldsymbol{\theta})^\top\boldsymbol{y} = \|\boldsymbol{A}^{-1/2}\boldsymbol{\theta}\|_2 \\ &= \sqrt{\boldsymbol{\theta}^\top\boldsymbol{A}^{-1}\boldsymbol{\theta}},\end{aligned}$$

where we have used the change of variable $\boldsymbol{y} = \boldsymbol{A}^{1/2}\boldsymbol{x}$ in the third equality and the dual norm norm of the L_2 norm from Example 4.16 in the second to last equality.

If you do not know the concept of *operator norms*, the concept of dual norm can be a bit weird at first. One way to understand it is that it is a way to measure how "big" are linear functionals. For example, consider the linear function $f(\boldsymbol{x}) = \langle \boldsymbol{z}, \boldsymbol{x} \rangle$, we want to try to understand how big it is. So, we can measure $\max_{\boldsymbol{x} \neq 0} \frac{\langle \boldsymbol{z}, \boldsymbol{x} \rangle}{\|\boldsymbol{x}\|}$ that is we measure how big is the output of the linear functional compared to its input $\boldsymbol{x}$, where $\boldsymbol{x}$ is measured with some norm. Now, you can show that the above is equivalent to the dual norm of $\boldsymbol{z}$.

Remark 4.19. *The definition of dual norm immediately implies $\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle \leq \|\boldsymbol{\theta}\|_\star \|\boldsymbol{x}\|$.*

Now we can introduce smooth functions, using the dual norms defined above.

Definition 4.20 (Smooth Function)**.** *Let $f : V \to \mathbb{R}$ differentiable in an open set containing V. We say that f is M-**smooth** w.r.t. $\|\cdot\|$ if $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_\star \leq M\|\boldsymbol{x} - \boldsymbol{y}\|$ for all $\boldsymbol{x}, \boldsymbol{y} \in V$.*

Keeping in mind the intuition above on dual norms, taking the dual norm of a gradient makes sense if you associate each gradient with the linear functional $\langle \nabla f(\boldsymbol{y}), \boldsymbol{x} \rangle$, that is the one needed to create a linear approximation of f.

Remark 4.21. *Note that smoothness does not imply convexity.*

Smooth functions have many properties, for example a smooth function can be upper and lower bounded by a quadratic.

Lemma 4.22. *Let $f : V \to \mathbb{R}$ be M-smooth. Then, for any $\boldsymbol{x}, \boldsymbol{y} \in V$, we have*

$$|f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle| \leq \frac{M}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 .$$

Proof. First, notice that by the definition of smoothness, $\nabla f : V \to \mathbb{R}^d$ is Lipschitz and so continuous. Hence, by the fundamental theorem of calculus, we have

$$\begin{aligned} f(\boldsymbol{y}) &= f(\boldsymbol{x}) + \int_0^1 \langle \nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})), \boldsymbol{y} - \boldsymbol{x} \rangle d\tau \\ &= f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle + \int_0^1 \langle \nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle d\tau . \end{aligned}$$

Therefore,

$$\begin{aligned} |f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle| &= \left| \int_0^1 \langle \nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle d\tau \right| \\ &\leq \int_0^1 |\langle \nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle| d\tau \\ &\leq \int_0^1 \|\nabla f(\boldsymbol{x} + \tau(\boldsymbol{y} - \boldsymbol{x})) - \nabla f(\boldsymbol{x})\|_\star \|\boldsymbol{y} - \boldsymbol{x}\| d\tau \\ &\leq \int_0^1 \tau M \|\boldsymbol{y} - \boldsymbol{x}\|^2 d\tau = \frac{M}{2}\|\boldsymbol{y} - \boldsymbol{x}\|^2 . \end{aligned}$$

□

In the following, we will need the following property.

Theorem 4.23. *Let $f : \mathbb{R}^d \to \mathbb{R}$ be M-smooth and bounded from below, then for all $\boldsymbol{x} \in \mathbb{R}^d$*

$$\|\nabla f(\boldsymbol{x})\|_\star^2 \leq 2M(f(\boldsymbol{x}) - \inf_{\boldsymbol{y} \in \mathbb{R}^d} f(\boldsymbol{y})) .$$

Proof. From Lemma 4.22, for any $\boldsymbol{x}, \boldsymbol{v} \in \mathbb{R}^d$, we have

$$\langle -\nabla f(\boldsymbol{x}), \boldsymbol{v} \rangle - \frac{M}{2}\|\boldsymbol{v}\|^2 \leq f(\boldsymbol{x}) - f(\boldsymbol{x}+\boldsymbol{v}) \leq f(\boldsymbol{x}) - \inf_{\boldsymbol{y}\in\mathbb{R}^d} f(\boldsymbol{y}) \,.$$

Given that this holds for any $\boldsymbol{v}$, we can take the supremum of the left hand side with respect to $\boldsymbol{v}$. Using Example 5.11, we have

$$\frac{1}{2M}\|\nabla f(\boldsymbol{x})\|_\star^2 = \sup_{\boldsymbol{v}} \langle -\nabla f(\boldsymbol{x}), \boldsymbol{v} \rangle - \frac{M}{2}\|\boldsymbol{v}\|^2 \leq f(\boldsymbol{x}) - \inf_{\boldsymbol{y}\in\mathbb{R}^d} f(\boldsymbol{y}) \,. \qquad \square$$

4.2.3 $L^\star$ bounds

We now introduce the $L^\star$ bounds, that depend on the cumulative competitor loss that is usually denoted by $L^\star$.

Assume now that the loss functions $\ell_1, \dots, \ell_T$ are bounded from below and M-smooth on an open sets that includes V. Without loss of generality, we can assume that each of them is bounded from below by 0. Under these assumptions, we can obtain bounds that depends on the cumulative loss of the competitor rather than time.

From the regret of OGD in Theorem 2.13 and Theorem 4.23 for a constant learning rate η, we obtain for any $\boldsymbol{u} \in V$

$$\sum_{t=1}^T (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \frac{\|\boldsymbol{u}-\boldsymbol{x}_1\|_2^2}{2\eta} + \eta \sum_{t=1}^T M\ell_t(\boldsymbol{x}_t),$$

that reordering implies

$$\sum_{t=1}^T (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \frac{\eta M}{1-\eta M} \sum_{t=1}^T \ell_t(\boldsymbol{u}) + \frac{\|\boldsymbol{u}-\boldsymbol{x}_1\|_2^2}{2\eta(1-\eta M)} \,.$$

Assuming $\eta \leq \frac{1}{2M}$, we simplify this regret upper bound in

$$\sum_{t=1}^T (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq 2\eta M \sum_{t=1}^T \ell_t(\boldsymbol{u}) + \frac{\|\boldsymbol{u}-\boldsymbol{x}_1\|_2^2}{\eta} \,.$$

This is already an interesting result because it guarantees that a fixed learning rate that depends only on M can achieve a vanishing average regret if there exists a competitor $\boldsymbol{u} \in V$ whose cumulative loss grows sublinearly. However, we could do better. In fact, for a fixed $\boldsymbol{u} \in V$, setting $\eta = \min\left(\frac{1}{\sqrt{2M\sum_{t=1}^T \ell_t(\boldsymbol{u})}}, \frac{1}{2M}\right)$, we obtain

$$\sum_{t=1}^T (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq (\|\boldsymbol{u}-\boldsymbol{x}_1\|_2^2 + 1) \max\left(\sqrt{2M\sum_{t=1}^T \ell_t(\boldsymbol{u})}, 2M\right) \,.$$

Comparing this bound to the one of OGD with Lipschitz losses, we see that here the dependency is on $\sqrt{\sum_{t=1}^T \ell_t(\boldsymbol{u})}$ instead of $\sqrt{T}$. The cumulative loss of the competitor can be must smaller than T and in particular can be even 0. In this case, the regret is upper bounded by a constant. Moreover, in this latter case we can afford to use a learning rate η that depends only on the smoothness constant. However, this result is partially interesting because it requires the knowledge of the future through the cumulative loss of the competitor. In the following, we show how to easily get rid of this limitation with a different choice of the learning rate.

In fact, under the same assumptions on the losses, from the regret in Theorem 4.14 and Theorem 4.23 we immediately obtain

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \leq 2D\sqrt{M\sum_{t=1}^T \ell_t(\boldsymbol{x}_t)},$$

where D is the diameter of V, assumed to be bounded. This is an implicit bound, in the sense that $\sum_{t=1}^T \ell_t(\boldsymbol{x}_t)$ appears on both sides of the inequality. To makes it explicit, we will use the following simple Lemma (proof left as an exercise).

Lemma 4.24. *Let $a, c > 0$, $b \geq 0$, and $x \geq 0$ such that $x - \sqrt{ax + b} \leq c$. Then $x \leq a + c + 2\sqrt{b + ac}$.*

So, we have the following theorem.

Theorem 4.25. *Let $V \subseteq \mathbb{R}^d$ a closed non-empty convex set with diameter D, i.e., $\max_{\boldsymbol{x},\boldsymbol{y} \in V} \|\boldsymbol{x} - \boldsymbol{y}\|_2 \leq D$. Let $\ell_1, \dots, \ell_T$ an arbitrary sequence of non-negative convex functions $\ell_t : V \to \mathbb{R}$ M-smooth in open sets containing V for $t = 1, \dots, T$. Pick any $\boldsymbol{x}_1 \in V$, set $\eta_t = \frac{\sqrt{2}D}{2\sqrt{\sum_{i=1}^t \|\boldsymbol{g}_i\|_2^2}}$, $t = 1, \dots, T$, and do not update on rounds in which $\boldsymbol{g}_t = \boldsymbol{0}$. Then, $\forall \boldsymbol{u} \in V$, the following regret bound holds*

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \leq 4MD^2 + 4D\sqrt{M \sum_{t=1}^T \ell_t(\boldsymbol{u})}\,.$$

This regret guarantee is very interesting because in the worst case it is still of the order $O(\sqrt{T})$, but in the best case scenario it becomes a constant! In fact, if there exists a $\boldsymbol{u} \in V$ such that $\sum_{t=1}^T \ell_t(\boldsymbol{u}) = 0$ we get a constant regret. Basically, if the losses are "easy", the algorithm *adapts* to this situation and gives us a better regret.

4.2.4 AdaGrad

We now present another application of the regret bound in (4.3). **AdaGrad**, that stands for Adaptive Gradient, is an Online Convex Optimization algorithm proposed independently by McMahan and Streeter [2010] and Duchi et al. [2010]. It aims at being adaptive to the sequence of gradients. It is usually known as a stochastic optimization algorithm, but in reality it was proposed for Online Convex Optimization (OCO). To use it as a stochastic algorithm, you should use an online-to-batch conversion, otherwise you do not have any guarantee of convergence.

We will present a proof that only allows hyperrectangles as feasible sets V, on the other hand the restriction makes the proof almost trivial. Let's see how it works.

AdaGrad has key ingredients:

- A coordinate-wise learning process;
- The adaptive learning rates in (4.2).

For the first ingredient, as we said, the regret of any OCO problem can be upper bounded by the regret of the Online Linear Optimization (OLO) problem. That is,

$$\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \leq \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle\,.$$

Now, the essential observation is to explicitly write the inner product as a sum of product over the single coordinates:

$$\sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle = \sum_{t=1}^T \sum_{i=1}^d g_{t,i} x_{t,i} - \sum_{t=1}^T \sum_{i=1}^d g_{t,i} u_i = \sum_{i=1}^d \left(\sum_{t=1}^T g_{t,i} x_{t,i} - \sum_{t=1}^T g_{t,i} u_i \right) = \sum_{i=1}^d \mathrm{Regret}_{T,i}(u_i),$$

where we denoted by $\mathrm{Regret}_{T,i}(u_i)$ the regret of the 1-dimensional OLO problem over coordinate i, that is $\sum_{t=1}^T g_{t,i} x_{t,i} - \sum_{t=1}^T g_{t,i} u_i$. In words, *we can decompose the original regret as the sum of d OLO regret minimization problems and we can try to focus on each one of them separately*.

A good candidate for the 1-dimensional problems is OSD with the learning rates in (4.2). We can specialize the regret in (4.3) to the 1-dimensional case for linear losses, so we get for each coordinate i

$$\sum_{t=1}^T g_{t,i} x_{t,i} - \sum_{t=1}^T g_{t,i} u_i \leq \sqrt{2} D_i \sqrt{\sum_{t=1}^T g_{t,i}^2}\,.$$

This choice gives us the AdaGrad algorithm in Algorithm 4.1.

Putting all together, we have immediately the following regret guarantee.

Algorithm 4.1 AdaGrad for Hyperrectangles

Require: $V = \{\boldsymbol{x} : a_i \leq x_i \leq b_i\}$, $\boldsymbol{x}_1 \in V$
for $t = 1$ **to** T **do**
  Output $\boldsymbol{x}_t$
  Receive $\ell_t : V \to \mathbb{R}$ subdifferentiable in V pay $\ell_t(\boldsymbol{x}_t)$
  Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
  for $i = 1$ **to** d **do**
    if $g_{t,i} \neq 0$ **then**
      Set $\eta_{t,i} = \frac{\sqrt{2}D_i}{2\sqrt{\sum_{j=1}^{t} g_{j,i}^2}}$
      $x_{t+1,i} = \max(\min(x_{t,i} - \eta_{t,i} g_{t,i}, b_i), a_i)$
    else
      $x_{t+1,i} = x_{t,i}$
    end if
  end for
end for

Theorem 4.26. *Let $V = \{\boldsymbol{x} : a_i \leq x_i \leq b_i\}$ with diameters along each coordinate equal to $D_i = b_i - a_i$. Let $\ell_1, \cdots, \ell_T$ an arbitrary sequence of convex functions $\ell_t : V \to \mathbb{R}$ subdifferentiable in V for $t = 1, \dots, T$. Pick any $\boldsymbol{x}_1 \in V$ and $\eta_{t,i} = \frac{\sqrt{2}D_i}{2\sqrt{\sum_{j=1}^{t} g_{j,i}^2}}$, $t = 1, \dots, T$. Then, $\forall \boldsymbol{u} \in V$, the following regret bound holds*

$$\text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \leq \sqrt{2} \sum_{i=1}^{d} D_i \sqrt{\sum_{t=1}^{T} g_{t,i}^2}\,.$$

Is this a better regret bound compared to the one in Theorem 4.14? It depends! To compare the two we have to consider that V is a hyperrectangle because the analysis of AdaGrad above only works for hyperrectangle. Then, we have to compare

$$D\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2} \qquad \text{versus} \qquad \sum_{i=1}^{d} D_i \sqrt{\sum_{t=1}^{T} g_{t,i}^2}\,.$$

From Cauchy-Schwarz, we have that $\sum_{i=1}^{d} D_i \sqrt{\sum_{t=1}^{T} g_{t,i}^2} \leq D\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2}$. So, *assuming the same sequence of subgradients*, AdaGrad has always a better regret on hyperrectangles. Also, note that

$$\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2} \leq \sum_{i=1}^{d} \sqrt{\sum_{t=1}^{T} g_{t,i}^2} \leq \sqrt{d}\sqrt{\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_2^2},$$

where the lower bound is by the fact that the L_1 norm is bigger than the L_2, and the upper bound is given by Cauchy-Schwarz. So, in the case that V is a hypercube we have $D_i = D_\infty = \max_{\boldsymbol{x},\boldsymbol{y}} \|\boldsymbol{x} - \boldsymbol{y}\|_\infty$ and $D = \sqrt{d} D_\infty$, the bound of AdaGrad is between $1/\sqrt{d}$ and 1 times the bound of Theorem 4.14. In other words, if we are lucky with the subgradients, the particular **shape of the domain** might save us a factor of $\sqrt{d}$ in the guarantee.

Note that the more general analysis of AdaGrad allows to consider arbitrary domains, but it does not change the general message that the best domains for AdaGrad are hyperrectangles. We will explore this issue of choosing the online algorithm based on the shape of the feasible set V when we will introduce Online Mirror Descent.

Hidden in the guarantee is that the biggest advantage of AdaGrad is the property of being coordinate-wise *scale-free*. That is, if each coordinate of the gradients are multiplied by different constants, the learning rate will automatically scale them back. This is hidden by the fact that the optimal solution $\boldsymbol{u}$ would also scale accordingly, but the fixed diameters of the feasible set hide it. This might be useful in the case the ranges of coordinates of the gradients are vastly different one from the other. Indeed, this does happen in the stochastic optimization of deep neural networks, where the first layers have smaller magnitude of the gradients compared to the last layers.

4.3 History Bits

The concept of strong convexity is defined for the first time in Polyak [1966].

The logarithmic regret in Corollary 4.9 was shown for the first time in the seminal paper Hazan et al. [2006]. The general statement in Theorem 4.7 was proven by Hazan et al. [2008].

The non-uniform averaging for the online-to-batch conversion of Example 4.12 is from Lacoste-Julien et al. [2012], but there is not proposed as an online-to-batch conversion. The basic idea of solving the problem of SVM with OSD and online-to-batch conversion of Example 4.12 was the Pegasos algorithm [Shalev-Shwartz et al., 2007], for many years the most used optimizer for SVMs.

The adaptive learning rate in (4.2) first appeared in Streeter and McMahan [2010]. However, similar methods were used long time before. Indeed, the key observation to approximate oracle quantities with estimates up to time t was first proposed in the self-confident algorithms [Auer et al., 2002c], where the learning rate is inversely proportional to the square root of the cumulative loss of the algorithm, and for smooth losses it implies the $L^\star$ bounds similar to the one in Theorem 4.25. The $L^\star$ bound for the square loss and linear predictors was introduced by Cesa-Bianchi et al. [1996]. In the past, several work focused on obtaining regret upper bounds depending on constant times $L^\star$ [see, e.g., Kivinen and Warmuth, 1997], however these guarantees are meaningful only if L^* is sublinear in T. Zhang [2004] explored the use of $L^\star$ bounds in stochastic optimization. The observation that the cumulative sum of the squared gradients act as an intrinsic notion of time comes from the statistics literature, see the discussion in Blackwell and Freedman [1973].

AdaGrad was proposed in basically identically form independently by two groups at the same conference: McMahan and Streeter [2010] and Duchi et al. [2010]. The analysis presented here is the one in Streeter and McMahan [2010] that does not handle generic feasible sets and does not support "full-matrices", i.e., full-matrix learning rates instead of diagonal ones. However, in machine learning applications AdaGrad is rarely used with a projection step (even if doing so provably destroys the worst-case performance [Orabona and Pál, 2018]). Also, in the adversarial setting full-matrices do not seem to offer advantages in terms of regret compared to diagonal ones, see the discussion in Cutkosky [2020, Section 5].

Note that the AdaGrad learning rate is usually written as

$$\eta_{t,i} = \frac{D_i}{\epsilon + \sqrt{\sum_{j=1}^t g_{j,i}^2}},$$

where $\epsilon > 0$ is a small constant used to prevent division by zero. In reality, ϵ is not necessary: there should be no update when the coordinate of the gradient is 0 [Orabona and Pál, 2015, 2018, Agarwal et al., 2020]. Moreover, removing ϵ makes the updates scale-free, as stressed in Orabona and Pál [2015, 2018]. Scale-freeness in online learning has been introduced in Cesa-Bianchi et al. [2005, 2007] for the setting of learning with expert advice and in Orabona and Pál [2015, 2018] for OCO.

AdaGrad inspired an incredible number of clones, most of them with similar, worse, or no regret guarantees. The keyword "adaptive" itself has shifted its meaning over time. It used to denote the ability of the algorithm to obtain the same guarantee as it knew in advance a particular property of the data (i.e., adaptive to the gradients/noise/scale = (almost) same performance as it knew the gradients/noise/scale in advance). Indeed, in Statistics this keyword is used with the same meaning. However, "adaptive" has changed its meaning over time. Nowadays, it seems to denote any kind of coordinate-wise learning rates that does not guarantee anything in particular.

4.4 Exercises

Problem 4.1. *Prove that OSD in Example 4.11 with $x_1 = 0$ is exactly the Follow-the-Leader strategy for that particular problem.*

Problem 4.2. *Prove that $\ell_t(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{z}_t\|_2^2$ is 2-strongly convex w.r.t. $\|\cdot\|_2$ and derive the OSD update for it and its regret guarantee.*

Problem 4.3. *Prove that $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_p^2$ is $p-1$-strongly convex w.r.t. $\|\cdot\|_p$ where $1 \leq p \leq 2$.*

Problem 4.4. *Prove that the dual norm of $\|\cdot\|_p$ is $\|\cdot\|_q$, where $\frac{1}{p}+\frac{1}{q}=1$ and $p,q\geq 1$.*

Problem 4.5. *Show that using online subgradient descent on a bounded domain V with the learning rates $\eta_t = O(1/t)$ with Lipschitz, smooth, and strongly convex functions you can get $O(\ln(1+L^\star))$ bounds.*

Problem 4.6. *Prove that the logistic loss $\ell(\boldsymbol{x}) = \ln(1+\exp(-y\langle \boldsymbol{z}, \boldsymbol{x}\rangle))$, where $\|\boldsymbol{z}\|_2 \leq 1$ and $y \in \{-1,1\}$ is $\frac{1}{4}$-smooth w.r.t. $\|\cdot\|_2$.*

Chapter 5

Lower bounds for Online Linear Optimization

In this chapter we will present some lower bounds for online linear optimization (OLO). Remembering that linear losses are convex, this immediately gives us lower bounds for online convex optimization (OCO). We will consider both the constrained and the unconstrained case. The lower bounds are important because they inform us on what are the optimal algorithms and where are the gaps in our knowledge.

5.1 Lower bounds for Bounded OLO

We will first consider the bounded constrained case. Finding a lower bound accounts to find a strategy for the adversary that forces a certain regret onto the algorithm, *no matter what the algorithm does*. We will use the probabilistic method to construct our lower bound.

The basic method relies on the fact that for any random variable $\boldsymbol{z}$ with domain V, and any function f

$$\sup_{\boldsymbol{x} \in V} f(\boldsymbol{x}) \geq \mathbb{E}[f(\boldsymbol{z})] \,.$$

For us, it means that we can lower bound the effect of the worst-case sequence of functions with an expectation over any distribution over the functions. If the distribution is chosen wisely, we can expect that gap in the inequality to be small. Why do you rely on expectations rather than actually constructing an adversarial sequence? Because the use of a stochastic loss functions makes very easy to deal with arbitrary algorithms. In particular, we will choose a distribution over stochastic loss functions that makes the expected loss of the algorithm 0, independently from the strategy of the algorithm.

Theorem 5.1. *Let $V \subset \mathbb{R}^d$ be any non-empty bounded closed convex subset. Let $D = \sup_{\boldsymbol{v},\boldsymbol{w} \in V} \|\boldsymbol{v} - \boldsymbol{w}\|_2$ be the diameter of V. Let $\mathcal{A}$ be any (possibly randomized) algorithm for OLO on V. Let T be any non-negative integer. There exists a sequence of vectors $\boldsymbol{g}_1, \dots, \boldsymbol{g}_T$ with $\|\boldsymbol{g}_t\|_2 \leq L$ and $\boldsymbol{u} \in V$ such that the regret of algorithm $\mathcal{A}$ satisfies*

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \geq \frac{LD\sqrt{T}}{2} \,.$$

Proof. Let's denote by $\mathrm{Regret}_T = \max_{\boldsymbol{u} \in V} \mathrm{Regret}_T(\boldsymbol{u})$. Let $\boldsymbol{v}, \boldsymbol{w} \in V$ such that $\|\boldsymbol{v} - \boldsymbol{w}\|_2 = D$. Let $\boldsymbol{z} = \frac{\boldsymbol{v} - \boldsymbol{w}}{\|\boldsymbol{v} - \boldsymbol{w}\|_2}$, so that $\langle \boldsymbol{z}, \boldsymbol{v} - \boldsymbol{w} \rangle = D$. Let $\epsilon_1, \dots, \epsilon_T$ be i.i.d. Rademacher random variables, that is $\mathbb{P}\{\epsilon_t = 1\} = \mathbb{P}\{\epsilon_t = -1\} = 1/2$ and set the vector of the stochastic linear losses $\tilde{\boldsymbol{g}}_t = L\epsilon_t \boldsymbol{z}$.

So, we have

$$\begin{aligned}
\sup_{\boldsymbol{g}_1,\dots,\boldsymbol{g}_T} \text{Regret}_T &\geq \mathbb{E}_{\tilde{\boldsymbol{g}}_1,\dots,\tilde{\boldsymbol{g}}_T}\left[\sum_{t=1}^T \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{u}\in V}\sum_{t=1}^T \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{u}\rangle\right] \\
&= \mathbb{E}_{\epsilon_1,\dots,\epsilon_T}\left[\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{x}_t\rangle - \min_{\boldsymbol{u}\in V}\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{u}\rangle\right] = \mathbb{E}_{\epsilon_1,\dots,\epsilon_T}\left[-\min_{\boldsymbol{u}\in V}\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{u}\rangle\right] \\
&= \mathbb{E}_{\epsilon_1,\dots,\epsilon_T}\left[\max_{\boldsymbol{u}\in V}\sum_{t=1}^T -L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{u}\rangle\right] = \mathbb{E}_{\epsilon_1,\dots,\epsilon_T}\left[\max_{\boldsymbol{u}\in V}\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{u}\rangle\right] \\
&\geq \mathbb{E}_{\epsilon_1,\dots,\epsilon_T}\left[\max_{\boldsymbol{u}\in \{\boldsymbol{v},\boldsymbol{w}\}}\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{u}\rangle\right] = \mathbb{E}_{\epsilon_1,\dots,\epsilon_T}\left[\frac{1}{2}\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{v}+\boldsymbol{w}\rangle + \frac{1}{2}\left|\sum_{t=1}^T L\epsilon_t\langle \boldsymbol{z}, \boldsymbol{v}-\boldsymbol{w}\rangle\right|\right] \\
&= \frac{L}{2}\mathbb{E}_{\epsilon_1,\dots,\epsilon_T}\left[\left|\sum_{t=1}^T \epsilon_t\langle \boldsymbol{z}, \boldsymbol{v}-\boldsymbol{w}\rangle\right|\right] = \frac{LD}{2}\mathbb{E}_{\epsilon_1,\dots,\epsilon_T}\left[\left|\sum_{t=1}^T \epsilon_t\right|\right] \geq \frac{LD\sqrt{T}}{2}\,.
\end{aligned}$$

where we used $\mathbb{E}[\epsilon_t] = 0$ and the ϵ_t are independent in the first equality, the fact that ϵ_t and $-\epsilon_t$ follow the same distribution in the fourth equality, $\max(a,b) = \frac{a+b}{2} + \frac{|a-b|}{2}$ in the sixth equality, and Khintchine inequality in the last inequality. □

Remark 5.2. *Differently from similar proofs, in the above proof we do not assume that V is symmetric with respect to* $\mathbf{0}$.

We see that the lower bound is a constant multiplicative factor from the upper bound we proved for Online Subgradient Descent (OSD) with learning rates $\eta_t = \frac{D}{L\sqrt{t}}$ or $\eta = \frac{D}{L\sqrt{T}}$. This means that OSD is asymptotically optimal with both settings of the learning rate.

At this point there is an important consideration to do: How can this be the optimal regret when we managed to prove a better regret, for example with adaptive learning rates? The subtlety is that, constraining the adversary to play L-Lipschitz losses, the adversary could always force on the algorithm at least the regret in Theorem 5.1. However, we can design algorithms that take advantage of *suboptimal plays of the adversary*. Indeed, for example, if the adversary plays in a way that all the subgradients have the same norm equal to L, there is nothing to adapt to!

5.2 Unconstrained Online Linear Optimization

We now move to the unconstrained case, however first we have to enrich our math toolbox with another essential tool, *Fenchel conjugates*.

5.2.1 Convex Analysis Bits: Fenchel Conjugate

Definition 5.3 (Closed Function)**.** *A function $f : V \subseteq \mathbb{R}^d \to [-\infty, +\infty]$ is **closed** iff $\{\boldsymbol{x} : f(\boldsymbol{x}) \leq \alpha\}$ is closed for every $\alpha \in \mathbb{R}$.*

Note that in a Hausdorff space a function is closed iff it is lower semicontinuous [Bauschke and Combettes, 2011, Lemma 1.24].

Example 5.4. *The indicator function of a set $V \subset \mathbb{R}^d$, is closed iff V is closed.*

Definition 5.5 (Fenchel Conjugate)**.** *For a function $f : \mathbb{R}^d \to [-\infty, \infty]$, we define the **Fenchel conjugate** $f^\star : \mathbb{R}^d \to [-\infty, \infty]$ as*

$$f^\star(\boldsymbol{\theta}) = \sup_{\boldsymbol{x}\in\mathbb{R}^d} \langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - f(\boldsymbol{x})\,.$$

From the definition we immediately obtain the Fenchel-Young's inequality for proper functions:

$$\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle \leq f(\boldsymbol{x}) + f^\star(\boldsymbol{\theta}),\ \forall \boldsymbol{x}, \boldsymbol{\theta} \in \mathbb{R}^d\ .$$

Moreover, $f^\star$ is always convex and closed, regardless of the convexity of f [Bauschke and Combettes, 2011, Proposition 13.11].

We have the following useful properties for the Fenchel conjugate.

Theorem 5.6 ([Rockafellar, 1970, Theorem 12.2])**.** *Let f a convex function. Then, $f^\star$ is a closed convex function, proper iff f is proper. Moreover, if f is also closed then $f^{\star\star} = f$.*

Theorem 5.7. *Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ be proper. Then, the following conditions are equivalent:*

(a) $\boldsymbol{\theta} \in \partial f(\boldsymbol{x})$.

(b) $\langle \boldsymbol{\theta}, \boldsymbol{y} \rangle - f(\boldsymbol{y})$ achieves its supremum in $\boldsymbol{y}$ at $\boldsymbol{y} = \boldsymbol{x}$.

(c) $f(\boldsymbol{x}) + f^\star(\boldsymbol{\theta}) = \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle$.

Moreover, if f is also convex and closed, we have an additional equivalent condition

(d) $\boldsymbol{x} \in \partial f^\star(\boldsymbol{\theta})$.

Proof. Let's prove (a)⇔(b). From the definition of subgradient, we have

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \langle \boldsymbol{\theta}, \boldsymbol{y} - \boldsymbol{x} \rangle, \forall \boldsymbol{y}$$

that is

$$\langle \boldsymbol{\theta}, \boldsymbol{x} \rangle - f(\boldsymbol{x}) \geq \langle \boldsymbol{\theta}, \boldsymbol{y} \rangle - f(\boldsymbol{y}), \forall \boldsymbol{y}\ .$$

Then, (b)⇔(c) by definition of $f^\star(\boldsymbol{\theta})$.

If f is also convex and closed, then $f^{\star\star} = f$ is proper by Theorem 5.6. Hence, (c) is equivalent to $f^{\star\star}(\boldsymbol{x}) + f^\star(\boldsymbol{\theta}) = \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle$, that is equivalent to (d) by following the same reasoning as above. □

The above theorem implies that for convex, closed, and proper functions $\partial f^\star$ is the inverse of ∂f, i.e., $\boldsymbol{\theta} \in \partial f(\boldsymbol{x})$ iff $\boldsymbol{x} \in \partial f^\star(\boldsymbol{\theta})$.

Example 5.8. *Let $f(x) = \exp(x)$, hence we have $f^\star(\theta) = \max_x\ x\theta - \exp(x)$. Solving the optimization, we have that $x^\star = \ln(\theta)$ if $\theta > 0$. Hence, $f^\star(\theta) = \begin{cases} \theta \ln \theta - \theta, & \text{if } \theta > 0 \\ 0, & \text{if } \theta = 0 \\ +\infty, & \text{if } \theta < 0 \end{cases}$.*

Example 5.9 (Conjugate of inner product)**.** *Let $f(\boldsymbol{x}) = \langle \boldsymbol{z}, \boldsymbol{x} \rangle$ where $\boldsymbol{z} \neq \boldsymbol{0} \in \mathbb{R}^d$. Then*

$$f^\star(\boldsymbol{\theta}) = \sup_{\boldsymbol{x} \in \mathbb{R}^d} \langle \boldsymbol{\theta} - \boldsymbol{z}, \boldsymbol{x} \rangle = \begin{cases} 0, & \boldsymbol{\theta} = \boldsymbol{z} \\ +\infty, & \text{otherwise.} \end{cases}$$

Example 5.10 (Conjugate of hinge loss)**.** *Let $f(\boldsymbol{x}) = \max(1 - \langle \boldsymbol{z}, \boldsymbol{x} \rangle, 0)$ where $\boldsymbol{z} \in \mathbb{R}^d$ and let's calculate $f^\star(\boldsymbol{\theta})$. If $\boldsymbol{\theta}$ has a component orthogonal to $\boldsymbol{z}$, I can choose $\boldsymbol{x}$ along that component and the supremum in the definition of $f^\star$ is $+\infty$. Hence, let's consider the case that $\boldsymbol{\theta} = \alpha \boldsymbol{z}$. In this case, we have*

$$f^\star(\boldsymbol{\theta}) = \sup_{\boldsymbol{x}}\ \alpha \langle \boldsymbol{z}, \boldsymbol{x} \rangle - \max(1 - \langle \boldsymbol{z}, \boldsymbol{x} \rangle, 0) = \sup_{u}\ \alpha u - \max(1 - u, 0)\ .$$

If $\alpha > 0$ or $\alpha < -1$, again the supremum is $+\infty$. Hence, we only need to consider the case that $-1 \leq \alpha \leq 0$. From a case analysis on u, it is easy to see that in this case the supremum is attained in $u = 1$. Putting all together, we have

$$f^\star(\boldsymbol{\theta}) = \begin{cases} \alpha, & \text{if } \boldsymbol{\theta} = \alpha \boldsymbol{z}, \alpha \in [-1, 0], \\ +\infty, & \text{otherwise.} \end{cases}$$

Example 5.11 (Conjugate of squared norms)**.** *Consider the function $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|^2$, where $\|\cdot\|$ is a norm in $\mathbb{R}^d$, with dual norm $\|\cdot\|_\star$. We can show that its conjugate is $f^\star(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|_\star^2$. From*

$$\langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - \frac{1}{2}\|\boldsymbol{x}\|^2 \leq \|\boldsymbol{\theta}\|_\star \|\boldsymbol{x}\| - \frac{1}{2}\|\boldsymbol{x}\|^2$$

for all $\boldsymbol{x}$. The right hand side is a quadratic function of $\|\boldsymbol{x}\|$, which has maximum value $\frac{1}{2}\|\boldsymbol{\theta}\|_\star^2$. Therefore for all $\boldsymbol{x}$, we have

$$\langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - \frac{1}{2}\|\boldsymbol{x}\|^2 \leq \frac{1}{2}\|\boldsymbol{\theta}\|_\star^2,$$

which shows that $f^\star(\boldsymbol{\theta}) \leq \frac{1}{2}\|\boldsymbol{\theta}\|_\star^2$. To show the other inequality, let $\boldsymbol{x}$ be any vector with $\langle \boldsymbol{\theta}, \boldsymbol{x}\rangle = \|\boldsymbol{\theta}\|_\star \|\boldsymbol{x}\|$, scaled so that $\|\boldsymbol{x}\| = \|\boldsymbol{\theta}\|_\star$. Then we have, for this $\boldsymbol{x}$,

$$\langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - \frac{1}{2}\|\boldsymbol{x}\|^2 = \frac{1}{2}\|\boldsymbol{\theta}\|_\star^2,$$

which shows that $f^\star(\boldsymbol{\theta}) \geq \frac{1}{2}\|\boldsymbol{\theta}\|_\star^2$.

Lemma 5.12. *Let f be a function and let $f^\star$ be its Fenchel conjugate. For $a > 0$ and $b \in \mathbb{R}$, the Fenchel conjugate of $g(\boldsymbol{x}) = af(\boldsymbol{x}) + b + \langle \boldsymbol{g}, \boldsymbol{x}\rangle$ is $g^\star(\boldsymbol{\theta}) = af^\star((\boldsymbol{\theta} - \boldsymbol{g})/a) - b$.*

Proof. From the definition of conjugate function, we have

$$g^\star(\boldsymbol{\theta}) = \sup_{\boldsymbol{x}\in\mathbb{R}^d} \langle \boldsymbol{\theta} - \boldsymbol{g}, \boldsymbol{x}\rangle - af(\boldsymbol{x}) - b = -b + a \sup_{\boldsymbol{x}\in\mathbb{R}^d} \left(\left\langle \frac{\boldsymbol{\theta} - \boldsymbol{g}}{a}, \boldsymbol{x} \right\rangle - f(\boldsymbol{x}) \right) = -b + af^\star\left(\frac{\boldsymbol{\theta} - \boldsymbol{g}}{a}\right). \qquad \square$$

Lemma 5.13. *Let f_1 and f_2 such that $f_1(\boldsymbol{x}) \leq f_2(\boldsymbol{x})$ for all $\boldsymbol{x}$. Then, $f_1^\star(\boldsymbol{\theta}) \geq f_2^\star(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$.*

Proof.

$$f_2^\star(\boldsymbol{\theta}) = \sup_{\boldsymbol{x}} \langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - f_2(\boldsymbol{x}) \leq \sup_{\boldsymbol{x}} \langle \boldsymbol{\theta}, \boldsymbol{x}\rangle - f_1(\boldsymbol{x}) = f_1^\star(\boldsymbol{\theta}). \qquad \square$$

Lemma 5.14 ([Bauschke and Combettes, 2011, Example 13.7])**.** *Let $f : \mathbb{R} \to (-\infty, +\infty]$ even, i.e., $f(x) = f(-x)$. Then $(f \circ \|\cdot\|_2)^\star = f^\star \circ \|\cdot\|_2$.*

5.2.2 Lower Bound for the Unconstrained Case

The above lower bound applies only to the constrained setting. In the unconstrained setting, we proved that OSD with $\boldsymbol{x}_1 = \boldsymbol{0}$ and constant learning rate of $\eta = \frac{1}{L\sqrt{T}}$ gives a regret of $\frac{1}{2}L(\|\boldsymbol{u}\|_2^2 + 1)\sqrt{T}$ for any $\boldsymbol{u} \in \mathbb{R}^d$. Is this regret optimal? It is clear that the regret must be at least linear in $\|\boldsymbol{u}\|_2$, but is linear enough?

The approach I will follow is to *reduce the OLO game to the online game of betting on a coin*, where the lower bounds are known. So, let's introduce the coin-betting online game:

- Start with an initial amount of money $\epsilon > 0$.
- In each round, the algorithm bets a fraction of its current wealth on the outcome of a coin.
- The outcome of the coin is revealed and the algorithm wins or loses its bet, 1 to 1.

The aim of this online game is to win as much money as possible. Also, as in all the online games we consider, we do not assume anything on how the outcomes of the coin are decided. Note that this game can also be written as OCO using the log loss.

We will denote by $c_t \in \{-1, 1\}, t = 1, \ldots, T$ the outcomes of the coin. We will use the absolute value of $\beta_t \in [-1, 1]$ to denote the fraction of money to bet and its sign to denote on which side we are betting. The money

the algorithm has won from the beginning of the game till the end of round t will be denoted by r_t and given that the money are won or lost 1 to 1, we have

$$\overbrace{r_t+\epsilon}^{\text{Money at the end of round } t} = \overbrace{r_{t-1}+\epsilon}^{\text{Money at the beginning of round } t} + \overbrace{c_t\beta_t(r_{t-1}+\epsilon)}^{\text{Money won or lost}} = \epsilon\prod_{i=1}^{t}(1+\beta_i c_i),$$

where we used the fact that $r_0 = 0$. We will also denote by $x_t = \beta_t(\epsilon + r_{t-1})$ the bet of the algorithm on round t.

If we got all the outcomes of the coin correct, we would double our money in each round, so that $\epsilon + r_T = \epsilon 2^T$. However, given the adversarial nature of the game, we can actually prove a stronger lower bound to the maximum wealth we can gain.

Theorem 5.15 ([Cesa-Bianchi and Lugosi, 2006, a simplified statement of Theorem 9.2])**.** *Let $T \geq 21$. Then, for any betting strategy with initial money $\epsilon > 0$ that bets fractions of its current wealth, there exists a sequence of coin outcomes c_t, such that*

$$\epsilon + \sum_{t=1}^{T} c_t x_t \leq \frac{1}{\sqrt{T}} \max_{-1\leq\beta\leq 1} \epsilon\prod_{t=1}^{T}(1+\beta c_t) \leq \frac{\epsilon}{\sqrt{T}}\exp\left(\frac{(\sum_{t=1}^{T} c_t)^2}{4(1-\ln 2)T}\right) .$$

Now, let's connect the coin-betting game with OLO. Remember that proving a regret guarantee in OLO consists in showing

$$\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}\rangle \leq \psi_T(\boldsymbol{u}),\ \forall \boldsymbol{u}\in\mathbb{R}^d,$$

for some function ψ_T, where we want the dependency on T to be sublinear. Using our new learned concept of Fenchel conjugate, this is equivalent to prove that

$$\sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \leq \inf_{\boldsymbol{u}} \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}\rangle + \psi_T(\boldsymbol{u}) = -\sup_{\boldsymbol{u}} -\left\langle \sum_{t=1}^{T}\boldsymbol{g}_t, \boldsymbol{u}\right\rangle - \psi_T(\boldsymbol{u}) = -\psi_T^\star\left(-\sum_{t=1}^{T}\boldsymbol{g}_t\right) .$$

Hence, for a given online algorithm we can prove regret bounds proving that there exists a function ψ_T or equivalently finding its conjugate $\psi_T^\star$. Similarly, proving a lower bound in unconstrained OLO means finding a sequence of $\boldsymbol{g}_t$ and a function ψ_T or a function $\psi_T^\star$ that lower bound the regret or the cumulative losses of the algorithm respectively.

Without any other information, it can be challenging to guess what is the slowest increasing function ψ_T. So, we restrict our attention to online algorithms that guarantee a constant regret against the zero vector. This immediately imply the following important consequence.

Theorem 5.16. *Let $\epsilon(t)$ a non-decreasing function of the index of the rounds and $\mathcal{A}$ an OLO algorithm that guarantees $\mathrm{Regret}_t(\mathbf{0}) \leq \epsilon(t)$ for any sequence of $\boldsymbol{g}_1, \dots, \boldsymbol{g}_t \in \mathbb{R}^d$ with $\|\boldsymbol{g}_i\|_2 \leq 1$. Then, there exists β_t such that $\boldsymbol{x}_t = \boldsymbol{\beta}_t(\epsilon(T) - \sum_{i=1}^{t-1}\langle \boldsymbol{g}_i, \boldsymbol{x}_i\rangle)$ and $\|\beta_t\|_2 \leq 1$ for $t = 1, \dots, T$.*

Proof. Define $r_t = -\sum_{i=1}^{t}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle$ the "reward" of the algorithm. So, we have

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}\rangle = -r_T + \left\langle \sum_{t=1}^{T}\boldsymbol{g}_t, \boldsymbol{u}\right\rangle .$$

Since, we assumed that $\mathrm{Regret}_t(\mathbf{0}) \leq \epsilon(t)$, we always have $r_t \geq -\epsilon(t)$. Using this, we claim that $\|\boldsymbol{x}_t\|_2 \leq r_{t-1} + \epsilon(t)$ for all $t = 1, \dots, T$. To see this, assume that there is a sequence $\boldsymbol{g}_1, \dots, \boldsymbol{g}_{t-1}$ that gives $\|\boldsymbol{x}_t\|_2 > r_{t-1} + \epsilon(t)$. We then set $\boldsymbol{g}_t = \frac{\boldsymbol{x}_t}{\|\boldsymbol{x}_t\|_2}$. For this sequence, we would have $r_t = r_{t-1} - \|\boldsymbol{x}_t\|_2 < -\epsilon(t)$, that contradicts the observation that $r_t \geq -\epsilon(t)$.

So, from the fact that $\|\boldsymbol{x}_t\|_2 \leq r_{t-1} + \epsilon(t) \leq r_{t-1} + \epsilon(T)$ we have that there exists β_t such that $\boldsymbol{x}_t = \boldsymbol{\beta}_t(\epsilon(T) + r_{t-1})$ for a $\boldsymbol{\beta}_t$ and $\|\boldsymbol{\beta}_t\|_2 \leq 1$. □

This theorem informs us of something important: *any OLO algorithm that suffer a non-decreasing regret against the null competitor must predict in the form of a "vectorial" coin-betting algorithm*. This immediately implies the following.

Theorem 5.17. *Let $T \geq 21$. For any OLO algorithm, under the assumptions of Theorem 5.16, there exist a sequence of $\boldsymbol{g}_1, \dots, \boldsymbol{g}_T \in \mathbb{R}^d$ with $\|\boldsymbol{g}_t\|_2 \leq 1$ and $\boldsymbol{u}^\star \in \mathbb{R}^d$, such that*

$$\sum_{i=1}^{t} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u}^\star \rangle \geq 0.8 \|\boldsymbol{u}^\star\|_2 \sqrt{T} \left(\sqrt{0.3 \ln \frac{0.8 \|\boldsymbol{u}^\star\|_2 T}{\epsilon(T)}} - 1 \right) + \epsilon(T) \left(1 - \frac{1}{\sqrt{T}} \right) .$$

Proof. The proof works by reducing the OLO game to a coin-betting game and then using the upper bound to the reward for coin-betting games.

First, set $\boldsymbol{g}_t = [-c_t, 0, \dots, 0]$, where $c_t \in \{-1, 1\}$ will be defined in the following, so that $\langle \boldsymbol{g}_t, \boldsymbol{x} \rangle = -c_t x_1$ for any $\boldsymbol{x} \in \mathbb{R}^d$. Given Theorem 5.16, we have that the first coordinate of $\boldsymbol{x}_t$ has to satisfy

$$x_{t,1} = \beta_{t,1} \left(\epsilon(T) - \sum_{i=1}^{t-1} \langle \boldsymbol{g}_i, \boldsymbol{x}_i \rangle \right) = \beta_{t,1} \left(\epsilon(T) + \sum_{i=1}^{t-1} c_i, x_{i,1} \right),$$

for some $\beta_{t,1}$ such that $|\beta_{t,1}| \leq 1$. Hence, the above is nothing else than a coin-betting algorithm that bets $x_{t,1}$ money on the outcome of a coin c_t, with initial money $\epsilon(T)$. This means that the upper bound to its reward in Theorem 5.15 applies: there exists a sequence of c_t such that

$$\begin{aligned} \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \epsilon(T) &= -\sum_{t=1}^{T} c_t x_{t,1} - \epsilon(T) \geq -\frac{\epsilon(T)}{\sqrt{T}} \exp\left(\frac{\ln 2 (\sum_{t=1}^{T} c_t)^2}{T} \right) = -\sum_{t=1}^{T} c_t u_1^\star + f^\star(|u_1^\star|) \\ &= \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle + f^\star(\|\boldsymbol{u}^\star\|_2), \end{aligned}$$

where $f^\star$ is the Fenchel conjugate of the function $f(x) = \frac{\epsilon(T)}{\sqrt{T}} \exp\left(\frac{x^2 \ln 2}{T} \right)$ and $u^\star = f'(\sum_{t=1}^{T} c_t) \in \mathbb{R}$ by Theorem 5.7. Using Theorem A.3 and reordering the terms, we get the stated bound. □

From the above theorem we have that OSD with learning rate $\eta = \frac{\alpha}{L\sqrt{T}}$ does not have the optimal dependency on $\|\boldsymbol{u}\|_2$ for any $\alpha > 0$.

In Chapter 9, we will see that the connection between coin-betting and OLO can also be used to design OLO algorithm. This will give us *optimal unconstrained OLO algorithms with the surprising property of not requiring a learning rate at all*.

5.3 History Bits

The lower bound for OCO is quite standard, the proof presented is a simplified version of the one in Orabona and Pál [2018].

On the other hand, both the online learning literature and the optimization one almost ignored the issue of lower bounds for the unconstrained case. The connection between coin-betting and OLO was first unveiled in Orabona and Pál [2016]. Theorem 5.16 is an unpublished result by Ashok Cutkosky, that proved similar and more general results in his PhD thesis [Cutkosky, 2018]. Theorem 5.17 is new, by me. McMahan and Abernethy [2013] implicitly also proposed using the conjugate function for the lower bound in unconstrained OLO.

There is a caveat in the unconstrained lower bound: A stronger statement would be to choose the norm of $\boldsymbol{u}^\star$ beforehand. To do this, we would have to explicitly construct the sequence of the c_t. One way to do is to use Rademacher coins and then leverage again the hypothesis on the regret against the null competitor. This route was used in Streeter and McMahan [2012], but the proof relied on assuming the value of the global optimum of a non-convex function with an infinite number of local minima. The correct proof avoiding that step was then given in

Orabona [2013]. Yet, the proof presented here, that converts reward upper bounds in regret lower bound, is simpler in spirit and (I hope!) more understandable. Given that, as far as I know, this is the first time that unconstrained OLO lower bounds are taught in a class, I valued simplicity over generality.

5.4 Exercises

Problem 5.1. *Fix $U > 0$. Mimicking the proof of Theorem 5.1, prove that for any OCO algorithm there exists a $\boldsymbol{u}^\star$ and a sequence of loss functions such that $\mathrm{Regret}_T(\boldsymbol{u}^\star) \geq \frac{1}{2}\|\boldsymbol{u}^\star\|_2 L\sqrt{T}$ where $\|\boldsymbol{u}^\star\|_2 = U$ and the loss functions are L-Lipschitz w.r.t. $\|\cdot\|_2$.*

Problem 5.2. *Extend the proof of Theorem 5.1 to an arbitrary norm $\|\cdot\|$ to measure the diameter of V and with $\|\boldsymbol{g}_t\|_\star \leq L$.*

Problem 5.3. *Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ be even. Prove that $f^\star$ is even*

Problem 5.4. *Find the exact expression of the conjugate function of $f(x) = \alpha \exp(\beta x^2)$, for $\alpha, \beta > 0$. Hint: Wolfram Alpha or any other kind of symbolic solver can be very useful for this type of problems.*

Chapter 6

Online Mirror Descent

In this chapter, we will introduce the Online Mirror Descent (OMD) algorithm. To explain its genesis, I think it is essential to understand what subgradients do. In particular, the negative subgradients are not always pointing towards a direction that minimizes the function.

6.1 Subgradients are not Informative

We have seen that in online learning we receive a sequence of loss functions and we have to output a vector before observing the loss function on which we will be evaluated. However, we can gain a lot of intuition if we consider the easy case that the sequence of loss functions is always a fixed function, i.e., $\ell_t(\boldsymbol{x}) = \ell(\boldsymbol{x})$. If our hypothetical online algorithm does not work in this situation, for sure it won't work on the more general case.

Hence, considering the case of fixed loss functions, let's take a look at the key step in the proof of the upper bound to the regret for Online Subgradient Descent (OSD) in Lemma 2.26. We used the following property of the subgradients:

$$\ell(\boldsymbol{x}_t) - \ell(\boldsymbol{u}) \leq \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \ . \tag{6.1}$$

In words, to minimize the left hand side of this equation, it is enough to minimize the right hand side, that is nothing else than the instantaneous linear regret on the linear function $\langle \boldsymbol{g}_t, \cdot \rangle$. This is the only reason why OSD works! However, I am sure you heard a million of times the (wrong) intuition that gradient points towards the minimum, and you might be tempted to think that the same (even more wrong) intuition holds for subgradients. Indeed, I am sure that even if we proved the regret guarantee based on (6.1), in the back of your mind you keep thinking "yeah, sure, it works because the subgradient tells me where to go to minimize the function". Typically this idea is so strong that I have to present explicit counterexamples to fully convince a person.

So, take a look at the following examples that illustrate the fact that a subgradient does not always point in a direction where the function decreases.

Example 6.1. *Let $f(\boldsymbol{x}) = \max[-x_1, x_1 - x_2, x_1 + x_2]$, see Figure 6.1. The vector $\boldsymbol{g} = [1, 1]$ is a subgradient in $\boldsymbol{x} = (1, 0)$ of $f(\boldsymbol{x})$. No matter how we choose the stepsize, moving in the direction of the negative subgradient will not decrease the objective function. An even more extreme example is in Figure 6.2, with the function $f(\boldsymbol{x}) = \max[x_1^2 + (x_2 + 1)^2, x_1^2 + (x_2 - 1)^2]$. Here, in the point $(1, 0)$, any positive step in the direction of the negative subgradient will* increase *the objective function.*

6.2 Reinterpreting the Online Subgradient Descent Algorithm

How Online Subgradient Descent works? It works exactly as I told you before: thanks to (6.1). But, what does that inequality really mean?

A way to understand how the OSD algorithm works is to think that it minimizes a local approximation of the original objective function. This is not unusual for optimization algorithms, for example the Netwon's algorithm constructs

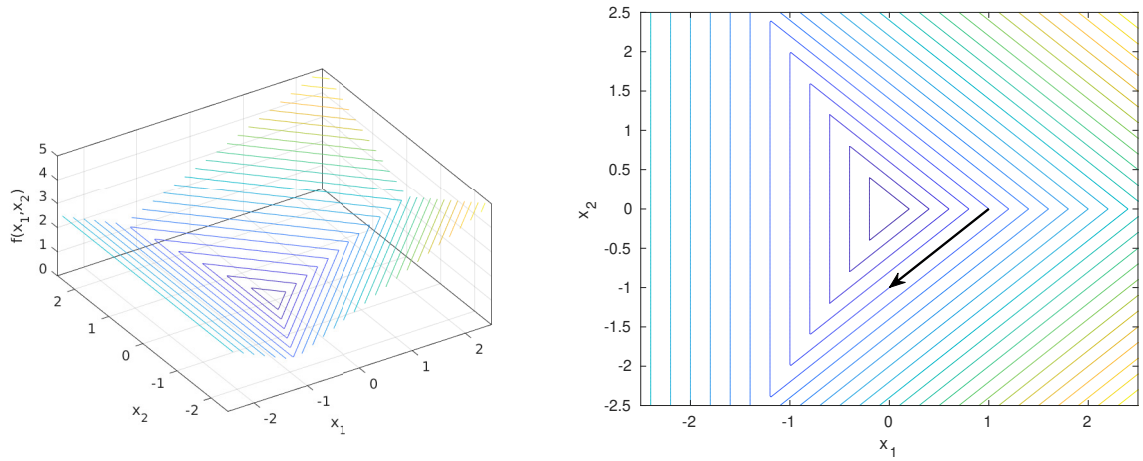

Figure 6.1: 3D plot (left) and level sets (right) of $f(\boldsymbol{x}) = \max[-x_1, x_1 - x_2, x_1 + x_2]$. A negative subgradient is indicated by the black arrow.

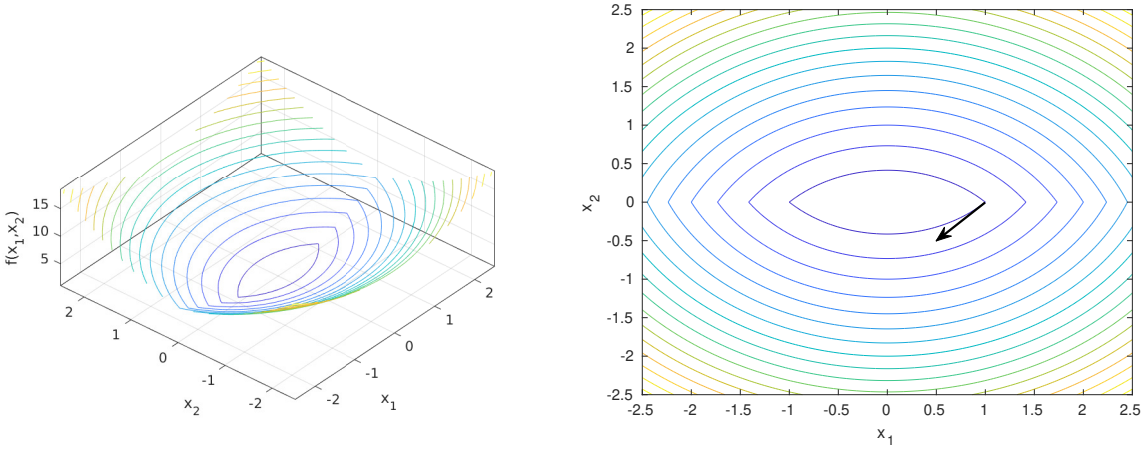

Figure 6.2: 3D plot (left) and level sets (right) of $f(\boldsymbol{x}) = \max[x_1^2 + (x_2+1)^2, x_1^2 + (x_2-1)^2]$. A negative subgradient is indicated by the black arrow.

an approximation with a Taylor expansion truncated to the second term. Thanks to the definition of subgradients, we can immediately build a linear lower bound to a function f around $\boldsymbol{x}_0$:

$$f(\boldsymbol{x}) \geq \tilde{f}(\boldsymbol{x}) := f(\boldsymbol{x}_0) + \langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{x}_0 \rangle, \ \forall \boldsymbol{x} \in V \ .$$

So, in our setting, this would mean that we update the online algorithm with the minimizer of a linear approximation of the loss function you received. Unfortunately, minimizing a linear function is unlikely to give us a good online algorithm. Indeed, over unbounded domains the minimum of a linear function is $-\infty$.

So, let's introduce the other key concept: we constraint the minimization of this lower bound only in a neighborhood of $\boldsymbol{x}_0$, where we have good reason to believe that the approximation is more precise. Moreover, in online learning it makes sense not to go too far from the previous iteration because the losses are different in each step and we do now want to give too much importance to the current loss. Coding the neighborhood constraint with a L_2 squared

distance from $\boldsymbol{x}_0$ less than some positive number h, we might think to use the following update

$$\begin{aligned}\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x}\in V} & f(\boldsymbol{x}_t) + \langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{x}_t\rangle \\ \text{s.t. } & \|\boldsymbol{x}_t - \boldsymbol{x}\|^2 \le h\ .\end{aligned}$$

Equivalently, for some $\eta > 0$, we can consider the unconstrained formulation

$$\operatorname*{argmin}_{\boldsymbol{x}\in V}\ \hat{f}(\boldsymbol{x}) := f(\boldsymbol{x}_0) + \langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{x}_0\rangle + \frac{1}{2\eta}\|\boldsymbol{x}_0 - \boldsymbol{x}\|_2^2\ . \tag{6.2}$$

This is a well-defined update scheme, that hopefully moves $\boldsymbol{x}_t$ closer to the optimum of f. See Figure 6.3 for a graphical representation in one-dimension.

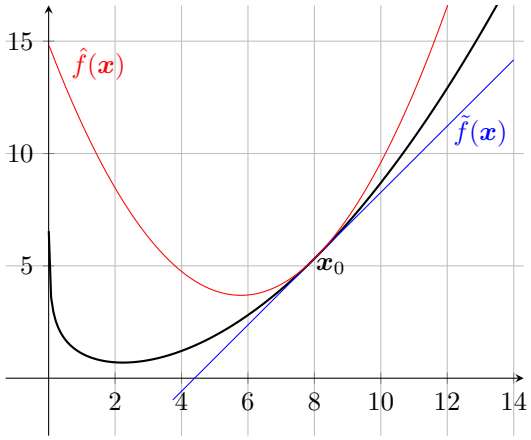

Figure 6.3: Approximations of $f(\boldsymbol{x})$.

And now the final element of our story: the argmin in (6.2) is exactly the update we used in OSD! Indeed, solving the argmin and completing the square, we get

$$\begin{aligned}\operatorname*{argmin}_{\boldsymbol{x}\in V} \langle \boldsymbol{g}_t, \boldsymbol{x}\rangle + \frac{1}{2\eta_t}\|\boldsymbol{x}_t - \boldsymbol{x}\|_2^2 &= \operatorname*{argmin}_{\boldsymbol{x}\in V}\ \|\eta_t \boldsymbol{g}_t\|^2 + 2\eta_t\langle \boldsymbol{g}_t, \boldsymbol{x} - \boldsymbol{x}_t\rangle + \|\boldsymbol{x}_t - \boldsymbol{x}\|_2^2 \\ &= \operatorname*{argmin}_{\boldsymbol{x}\in V}\ \|\boldsymbol{x} - \boldsymbol{x}_t + \eta_t \boldsymbol{g}_t\|_2^2 \\ &= \Pi_V(\boldsymbol{x}_t - \eta_t \boldsymbol{g}_t),\end{aligned} \tag{6.3}$$

where Π_V is the Euclidean projection onto V, i.e., $\Pi_V(\boldsymbol{x}) = \operatorname{argmin}_{\boldsymbol{y}\in V}\ \|\boldsymbol{x} - \boldsymbol{y}\|_2$.

The new way to write the update of OSD in (6.2) will be the core ingredient for designing Online Mirror Descent. In fact, OMD is a strict generalization of that update when we use a different way to measure the locality of $\boldsymbol{x}$ from $\boldsymbol{x}_t$. That is, we measured the distance between to the current point with the squared L_2 norm. What happens if we change the norm? Do we even have to use a norm?

To answer these questions we have to introduce another useful mathematical object: the *Bregman divergence*.

6.3 Convex Analysis Bits: Bregman Divergence

We first give a new definition, a slightly stronger notion of convexity.

Definition 6.2 (Strictly Convex Function)**.** *Let $f : V \subseteq \mathbb{R}^d \to \mathbb{R}$ and V a convex set. f is **strictly convex** if*

$$f(\alpha\boldsymbol{x} + (1-\alpha)\boldsymbol{y}) < \alpha f(\boldsymbol{x}) + (1-\alpha) f(\boldsymbol{y}),\ \forall \boldsymbol{x}, \boldsymbol{y} \in V, \boldsymbol{x} \neq \boldsymbol{y}, 0 < \alpha < 1\ .$$

From the definition, it is immediate to see that strong convexity w.r.t. any norm implies strict convexity. Note that for a differentiable function, strict convexity also implies that $f(\boldsymbol{y}) > f(\boldsymbol{x}) + \langle \nabla f(\boldsymbol{x}), \boldsymbol{y} - \boldsymbol{x} \rangle$ for $\boldsymbol{x} \neq \boldsymbol{y}$ [Bauschke and Combettes, 2011, Proposition 17.13].

We now define our new notion of "distance".

Definition 6.3 (Bregman Divergence)**.** *Let $\psi : X \to \mathbb{R}$ be strictly convex and differentiable on $\operatorname{int} X \neq \{\}$. The **Bregman Divergence** w.r.t. ψ is denoted by $B_\psi : X \times \operatorname{int} X \to \mathbb{R}$ defined as*

$$B_\psi(\boldsymbol{x}; \boldsymbol{y}) = \psi(\boldsymbol{x}) - \psi(\boldsymbol{y}) - \langle \nabla \psi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \, .$$

From the definition, we see that the Bregman divergence is always non-negative for $\boldsymbol{x}, \boldsymbol{y} \in \operatorname{int} X$, from the convexity of ψ. However, something stronger holds. By the strict convexity of ψ, for fixed a point $\boldsymbol{y} \in \operatorname{int} X$ we have that $\psi(\boldsymbol{x}) \geq \psi(\boldsymbol{y}) + \langle \nabla \psi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle, \ \forall \boldsymbol{y} \in X$, with equality only for $\boldsymbol{y} = \boldsymbol{x}$. Hence, the strict convexity allows us to use the Bregman divergence as a similarity measure between $\boldsymbol{x}$ and $\boldsymbol{y}$. Moreover, this similarity measure *changes* with the reference point $\boldsymbol{y}$. This also implies that, as you can see from the definition, the Bregman divergence is not symmetric.

Let me give you some more intuition on the concept of the Bregman divergence. Consider the case that ψ is twice differentiable in an open ball B around $\boldsymbol{y}$ and $\boldsymbol{x} \in B$. So, by the Taylor's theorem, there exists $0 \leq \alpha \leq 1$ such that

$$B_\psi(\boldsymbol{x}; \boldsymbol{y}) = \psi(\boldsymbol{x}) - \psi(\boldsymbol{y}) - \nabla \psi(\boldsymbol{y})^\top (\boldsymbol{x} - \boldsymbol{y}) = \frac{1}{2} (\boldsymbol{x} - \boldsymbol{y})^\top \nabla^2 \psi(\boldsymbol{z}) (\boldsymbol{x} - \boldsymbol{y}),$$

where $\boldsymbol{z} = \alpha \boldsymbol{x} + (1 - \alpha) \boldsymbol{y}$. Hence, we are using a *squared local norm* that depends on the Hessian of ψ. *Different areas of the space will have a different value of the Hessian, and so the Bregman will behave differently*. We will use this exact idea in the local norm analysis of Online Mirror Descent (Section 6.5) and Follow-the-Regularized-Leader (Section 7.4).

We can also lower bound the Bregman divergence if the function ψ is strongly convex. In particular, if ψ is λ-strongly convex w.r.t. a norm $\| \cdot \|$ in $\operatorname{int} X$, then we have

$$B_\psi(\boldsymbol{x}; \boldsymbol{y}) \geq \frac{\lambda}{2} \|\boldsymbol{x} - \boldsymbol{y}\|^2 \, . \tag{6.4}$$

Example 6.4. *If $\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$, then $B_\psi(\boldsymbol{x}; \boldsymbol{y}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2 - \frac{1}{2}\|\boldsymbol{y}\|_2^2 - \langle \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y} \rangle = \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2$.*

Example 6.5. *Let $V = \{\boldsymbol{x} \in \mathbb{R}^d : x_i \geq 0, \|\boldsymbol{x}\|_1 = 1\}$, $X = \mathbb{R}^d_{\geq 0}$, and $\psi(\boldsymbol{x}) = \sum_{i=1}^d x_i \ln x_i$, the negative entropy. Then, for all $\boldsymbol{x} \in X$ and $\boldsymbol{y} \in \operatorname{int} X$ we have*

$$B_\psi(\boldsymbol{x}; \boldsymbol{y}) = \sum_{i=1}^d (x_i \ln x_i - y_i \ln y_i - (\ln(y_i) + 1)(x_i - y_i)) = \sum_{i=1}^d \left(x_i \ln \frac{x_i}{y_i} - x_i + y_i \right) \, .$$

This is called the generalized Kullback-Leibler divergence, where "generalized" is due to the fact that $\boldsymbol{x}$ and $\boldsymbol{y}$ do not have to be discrete probability distributions.

We also have the following immediate lemma that links the Bregman divergences between 3 points.

Lemma 6.6 ([Chen and Teboulle, 1993])**.** *Let B_ψ the Bregman divergence w.r.t. $\psi : X \to \mathbb{R}$. Then, for any three points $\boldsymbol{x}, \boldsymbol{y} \in \operatorname{int} X$ and $\boldsymbol{z} \in X$, the following identity holds*

$$B_\psi(\boldsymbol{z}; \boldsymbol{x}) + B_\psi(\boldsymbol{x}; \boldsymbol{y}) - B_\psi(\boldsymbol{z}; \boldsymbol{y}) = \langle \nabla \psi(\boldsymbol{y}) - \nabla \psi(\boldsymbol{x}), \boldsymbol{z} - \boldsymbol{x} \rangle \, .$$

6.4 Online Mirror Descent

Based on what we said before, we can start from the equivalent formulation of the OSD update

$$\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x} \in V} \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{1}{2\eta_t} \|\boldsymbol{x}_t - \boldsymbol{x}\|_2^2,$$

and we can change the last term with another measure of distance. In particular, using the Bregman divergence w.r.t. a function ψ, we have

$$\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x} \in V} \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x}; \boldsymbol{x}_t),$$

where we assumed that the argmin exists. These two updates are exactly the same when $\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$.

So we get the Online Mirror Descent algorithm in Algorithm 6.1.

Algorithm 6.1 Online Mirror Descent

Require: Non-empty closed convex $V \subseteq X \subseteq \mathbb{R}^d$, $\psi : X \to \mathbb{R}$ strictly convex and differentiable on $\operatorname{int} X$, $\boldsymbol{x}_1 \in \operatorname{int} X$, $\eta_1, \dots, \eta_T > 0$

1: **for** $t = 1$ **to** T **do**
2: Output $\boldsymbol{x}_t \in V$
3: Receive $\ell_t : V \to \mathbb{R}$ subdifferentiable in V and pay $\ell_t(\boldsymbol{x}_t)$
4: Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
5: Set $\boldsymbol{x}_{t+1} \in \operatorname{argmin}_{\boldsymbol{x} \in V} \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x}; \boldsymbol{x}_t)$
6: **end for**

However, without an additional assumption, this algorithm has a problem. Can you see it? The problem is that $\boldsymbol{x}_{t+1}$ might be on the boundary of V and in the next step we would have to evaluate $B_\psi(\boldsymbol{x}; \boldsymbol{x}_{t+1})$ for a point on the boundary of V. Given that $V \subseteq X$, we might end up on the boundary of X where the Bregman is not defined!

To fix this problem, different sufficient conditions can be used. In the following, we will use either one of the following assumptions:

$$\lim_{\lambda \to 0} \langle \nabla \psi(\boldsymbol{x} + \lambda(\boldsymbol{y} - \boldsymbol{x})), \boldsymbol{y} - \boldsymbol{x} \rangle = -\infty, \quad \forall \boldsymbol{x} \in \operatorname{bdry} X, \forall \boldsymbol{y} \in \operatorname{int} X \tag{6.5}$$

$$V \subseteq \operatorname{int} X\ . \tag{6.6}$$

If either of these conditions is true and the argmin exists on each round then the algorithm is well-defined as proved in the following theorem.

Theorem 6.7. *Let B_ψ the Bregman divergence w.r.t. $\psi : X \to \mathbb{R}$. Let $V \subseteq X$ a non-empty closed convex set. Assume* (6.5) *or* (6.6) *hold and, with the notation in Algorithm 6.1, the argmin exists on all rounds. Then, $\boldsymbol{x}_{t+1} \in \operatorname{int} X$.*

Proof. In the case that (6.6) holds, we have that $\boldsymbol{x}_{t+1} \in V$ implies immediately that $\boldsymbol{x}_{t+1} \in \operatorname{int} X$.

Let's now assume that (6.5) holds and let's prove it by induction. The base case is true by the definition of $\boldsymbol{x}_1$. Let's now assume that $\boldsymbol{x}_t \in \operatorname{int} X$ and let's prove that $\boldsymbol{x}_{t+1} \in \operatorname{int} X$. We will prove it by contradiction. So, assume that $\boldsymbol{x}_{t+1} \in \operatorname{bdry} X$. Set $\boldsymbol{z} \in \operatorname{int} X \cap V$ and define $\phi(\lambda) = \langle \eta_t \boldsymbol{g}_t, (1-\lambda)\boldsymbol{x}_{t+1} + \lambda \boldsymbol{z} \rangle + B_\psi((1-\lambda)\boldsymbol{x}_{t+1} + \lambda \boldsymbol{z}; \boldsymbol{x}_t)$ for $\lambda \in (0, 1)$. From (6.5), we have that

$$\lim_{\lambda \to 0} \phi'(\lambda) = \lim_{\lambda \to 0} \langle \eta_t \boldsymbol{g}_t, \boldsymbol{z} - \boldsymbol{x}_{t+1} \rangle + \langle \nabla \psi(\boldsymbol{x}_{t+1} + \lambda(\boldsymbol{z} - \boldsymbol{x}_{t+1})) - \nabla \psi(\boldsymbol{x}_t), \boldsymbol{z} - \boldsymbol{x}_{t+1} \rangle = -\infty\ .$$

Hence, there exists $\epsilon > 0$ such that

$$\langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_\epsilon \rangle + B_\psi(\boldsymbol{x}_\epsilon; \boldsymbol{x}_t) = \phi(\epsilon) < \phi(0) = \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_{t+1} \rangle + B_\psi(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t),$$

where $\boldsymbol{x}_\epsilon := (1-\epsilon)\boldsymbol{x}_{t+1} + \epsilon \boldsymbol{z} \in \operatorname{int} X \cap V$. However, this contradicts, the definition of $\boldsymbol{x}_{t+1}$ as an argmin, proving that $\boldsymbol{x}_{t+1}$ must be in $\operatorname{int} X$. □

When (6.5) holds, this theorem implies that the predictions of the algorithm always stay in the interior of the feasible set without the need for any projection. If in addition $V = X$, the update of the algorithm is the solution of an unconstrained problem because the feasible set is implicit in the Bregman divergence.

Now we have a well-defined algorithm, but does it guarantee a sublinear regret? We know that at least in one case it recovers the OSD algorithm, that does work. So, from an intuitive point of view, how well the algorithm work should depend on some characteristic on ψ. In particular, a key property will be the *strong convexity* of ψ. The strong convexity also takes care of the existence of the argmin in the algorithm, thanks to next Theorem.

Theorem 6.8. *Let $\lambda > 0$ and $f : \mathbb{R}^d \to (-\infty, +\infty]$ proper, closed, and λ-strongly convex w.r.t. $\|\cdot\|$ on its domain. Assume $\operatorname{dom} \partial f \neq \{\}$. Then, f has exactly one minimizer.*

Proof. Let $\boldsymbol{y} \in \operatorname{dom} \partial f$ and $\boldsymbol{g} \in \partial f(\boldsymbol{y})$. From Lemma 4.2, for any $\boldsymbol{x} \in \mathbb{R}^d$, we have

$$\begin{aligned} f(\boldsymbol{x}) &\geq f(\boldsymbol{y}) + \langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{y} \rangle + \frac{\lambda}{2}\|\boldsymbol{x} + \boldsymbol{y}\|^2 \\ &\geq f(\boldsymbol{y}) - \|\boldsymbol{g}\|_\star \|\boldsymbol{x}\| - \langle \boldsymbol{g}, \boldsymbol{y} \rangle + \frac{\lambda}{2}(\|\boldsymbol{x}\| - \|\boldsymbol{y}\|)^2 \\ &= f(\boldsymbol{y}) - \|\boldsymbol{g}\|_\star \|\boldsymbol{x}\| - \langle \boldsymbol{g}, \boldsymbol{y} \rangle + \frac{\lambda}{2}(\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\|\boldsymbol{x}\|\|\boldsymbol{y}\|), \end{aligned}$$

where in the second inequality we used the reverse triangle inequality and the definition of dual norms. From the above, we have that $\lim_{\|\boldsymbol{x}\| \to \infty} f(\boldsymbol{x}) = +\infty$. In turn, this implies that the level sets of f are bounded. From the assumption that f is closed, we get that the level sets are compact. Hence, for any $\boldsymbol{y}$ in $\operatorname{dom} f$, the minimum of f is the same of the minimum of f over the set $\{\boldsymbol{x} : f(\boldsymbol{x}) \leq f(\boldsymbol{y})\}$, that is the minimum over a compact set, that exists by the Weierstrass theorem, Theorem A.11. The uniqueness is given by the fact that strongly convex function are strictly convex. □

To analyze OMD, we first prove a one step relationship, similar to the one we proved for Online Gradient Descent and OSD. Note how in this Lemma, we will use a lot of the concepts we introduced till now: strong convexity, dual norms, subgradients, etc. In a way, over the past sections I slowly prepared you to be able to prove this lemma.

Lemma 6.9. *Let B_ψ the Bregman divergence w.r.t. $\psi : X \to \mathbb{R}$ and assume ψ to be proper, closed, and λ-strongly convex with respect to $\|\cdot\|$ in $V \cap \operatorname{int} X$. Let $V \subseteq X$ a non-empty closed convex set. Assume* (6.5) *or* (6.6) *hold. Then, with the notation in Algorithm 6.1, for all t we have that $\boldsymbol{x}_{t+1}$ exists, it is unique, and it is in the interior of X. Moreover, $\forall \boldsymbol{u} \in V$, the following inequality holds*

$$\begin{aligned} \eta_t(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\leq \eta \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) - B_\psi(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) + \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle \\ &\leq B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) + \frac{\eta_t^2}{2\lambda}\|\boldsymbol{g}_t\|_\star^2 . \end{aligned}$$

Proof. First of all, in each round $\boldsymbol{x}_{t+1}$ exists using Theorem 6.8 and the fact that $B_\psi(\cdot; \boldsymbol{x}_t)$ is proper, closed, and strongly convex. Moreover, from Theorem 6.7, $\boldsymbol{x}_t \in \operatorname{int} X$ for all t.

Now, from the optimality condition in Theorem 2.8 for the update of OMD, we have

$$\langle \eta_t \boldsymbol{g}_t + \nabla\psi(\boldsymbol{x}_{t+1}) - \nabla\psi(\boldsymbol{x}_t), \boldsymbol{u} - \boldsymbol{x}_{t+1} \rangle \geq 0, \ \forall \boldsymbol{u} \in V . \tag{6.7}$$

Hence, we have that

$$\begin{aligned} &\langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \\ &= \langle \nabla\psi(\boldsymbol{x}_t) - \nabla\psi(\boldsymbol{x}_{t+1}) - \eta_t \boldsymbol{g}_t, \boldsymbol{u} - \boldsymbol{x}_{t+1} \rangle + \langle \nabla\psi(\boldsymbol{x}_{t+1}) - \nabla\psi(\boldsymbol{x}_t), \boldsymbol{u} - \boldsymbol{x}_{t+1} \rangle + \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle \\ &\leq \langle \nabla\psi(\boldsymbol{x}_{t+1}) - \nabla\psi(\boldsymbol{x}_t), \boldsymbol{u} - \boldsymbol{x}_{t+1} \rangle + \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle \\ &= B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) - B_\psi(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) + \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle \\ &\leq B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) - \frac{\lambda}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 + \eta_t \|\boldsymbol{g}_t\|_\star \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\| \\ &\leq B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) + \frac{\eta_t^2}{2\lambda}\|\boldsymbol{g}_t\|_\star^2, \end{aligned}$$

where in the second inequality we used (6.7), in the second equality we used Lemma 6.6, in the second inequality we used the definition of dual norm and (6.4) because ψ is λ-strong convex w.r.t. $\|\cdot\|$, finally in the last inequality we used the fact that $ax - \frac{b}{2}x^2 \leq \frac{a^2}{2b}$ for $x \in \mathbb{R}$ and $a, b > 0$.

The lower bound with the function values is due, as usual, to the definition of subgradients. □

We now see how to use this one step relationship to prove a regret bound, that will finally show us if and when this entire construction is a good idea. In fact, it is worth stressing that *the above motivation is not enough in any way to justify the existence of the OMD algorithm*. Also, in the next section we will explain why the algorithm is called Online "Mirror" Descent.

We can now prove a regret bound for OMD.

Theorem 6.10. *Set $\boldsymbol{x}_1 \in V$ such that ψ is differentiable in $\boldsymbol{x}_1$. Assume $\eta_{t+1} \leq \eta_t,\ t = 1, \dots, T$. Then, under the assumptions of Lemma 6.9 and $\forall \boldsymbol{u} \in V$, the following regret bounds hold*

$$\sum_{t=1}^{T} (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \max_{1 \leq t \leq T} \frac{B_\psi(\boldsymbol{u}; \boldsymbol{x}_t)}{\eta_T} + \frac{1}{2\lambda} \sum_{t=1}^{T} \eta_t \|\boldsymbol{g}_t\|_\star^2 \,.$$

Moreover, if η_t is constant, i.e., $\eta_t = \eta \, \forall t = 1, \cdots, T$, we have

$$\sum_{t=1}^{T} (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \frac{B_\psi(\boldsymbol{u}; \boldsymbol{x}_1)}{\eta} + \frac{\eta}{2\lambda} \sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\star^2 \,.$$

Proof. Fix $\boldsymbol{u} \in V$. As in the proof of OGD, dividing the inequality in Lemma 6.9 by η_t and summing from $t = 1, \cdots, T$, we get

$$\begin{aligned}
\sum_{t=1}^{T} (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\leq \sum_{t=1}^{T} \left(\frac{1}{\eta_t} B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - \frac{1}{\eta_t} B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) \right) + \sum_{t=1}^{T} \frac{\eta_t}{2\lambda} \|\boldsymbol{g}_t\|_\star^2 \\
&= \frac{1}{\eta_1} B_\psi(\boldsymbol{u}; \boldsymbol{x}_1) - \frac{1}{\eta_T} B_\psi(\boldsymbol{u}; \boldsymbol{x}_{T+1}) + \sum_{t=1}^{T-1} \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) + \sum_{t=1}^{T} \frac{\eta_t}{2\lambda} \|\boldsymbol{g}_t\|_\star^2 \\
&\leq \frac{1}{\eta_1} D^2 + D^2 \sum_{t=1}^{T-1} \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t} \right) + \sum_{t=1}^{T} \frac{\eta_t}{2\lambda} \|\boldsymbol{g}_t\|_\star^2 \\
&= \frac{1}{\eta_1} D^2 + D^2 \left(\frac{1}{\eta_T} - \frac{1}{\eta_1} \right) + \sum_{t=1}^{T} \frac{\eta_t}{2\lambda} \|\boldsymbol{g}_t\|_\star^2 \\
&= \frac{D^2}{\eta_T} + \sum_{t=1}^{T} \frac{\eta_t}{2\lambda} \|\boldsymbol{g}_t\|_\star^2,
\end{aligned}$$

where we denoted by $D^2 = \max_{1 \leq t \leq T} B_\psi(\boldsymbol{u}; \boldsymbol{x}_t)$.

The second statement is left as an exercise. □

In words, OMD allows us to prove regret guarantees that depend on arbitrary couple of dual norms $\|\cdot\|$ and $\|\cdot\|_\star$. In particular, the primal norm will be used to measure the feasible set V or the distance between the competitor and the initial point, and the dual norm will be used to measure the gradients. If you happen to know something about these quantities, we can choose the most appropriate couple of norm to guarantee a small regret. The only thing you need is a function ψ that is strongly convex with respect to the primal norm you have chosen ψ.

Overall, the regret bound is still of the order of $\sqrt{T}$ for Lipschitz functions, that only difference is that now the Lipschitz constant is measured with a different norm. Also, everything we did for Online Subgradient Descent can be trivially used here. For example, we can slightly generalize the stepsizes we saw in Section 4.2 to

$$\eta_t = \frac{D\sqrt{2\lambda}}{2\sqrt{\sum_{i=1}^{t} \|\boldsymbol{g}_i\|_\star^2}}$$

to achieve a regret upper bound of $\frac{D}{\sqrt{\lambda}} \sqrt{2 \sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\star^2}$.

In Sections 6.6 and 6.7, we will see practical examples of OMD that guarantee strictly better regret than OSD. As we did in the case of AdaGrad, the better guarantee will depend on the shape of the domain and the characteristics of the subgradients.

Next, we see the meaning of the "Mirror".

6.4.1 The "Mirror" Interpretation

Here, we explain the "mirror" interpretation of OMD. First, we need a couple of convex analysis results.

For closed, convex, and proper functions, Theorem 5.7 implies that $\boldsymbol{x} \in \partial f^\star(\boldsymbol{\theta})$ iff $\boldsymbol{\theta} \in \partial f(\boldsymbol{x})$, that in words means that $(\partial f)^{-1} = \partial f^\star$ in the sense of multivalued mappings. Now, we show that for strongly convex functions the Fenchel conjugate is smooth and hence differentiable.

Theorem 6.11 (Duality Strong Convexity/Smoothness)**.** *Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ be a proper, closed, convex function, and $\operatorname{dom} \partial f$ be non-empty. Then, f is $\lambda > 0$ strongly convex w.r.t. $\|\cdot\|$ iff $f^\star$ is $\frac{1}{\lambda}$-smooth w.r.t. $\|\cdot\|_\star$ on $\mathbb{R}^d$.*

Proof. Let's first prove the implication from left to right. First, let's show that $f^\star$ is differentiable. Since f is proper, closed, and strongly convex, the maximizer of $\max_{\boldsymbol{x}} \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle - f(\boldsymbol{x})$ exists and it is unique by Theorem 6.8. Denote by $\boldsymbol{x}^\star$ the argmax of this expression. Hence, from Theorem 5.7, we have $\boldsymbol{x}^\star \in \partial f^\star(\boldsymbol{\theta})$. Let's now show that this is the only element in the subdifferential. Assume there exists $\boldsymbol{x}' \in \partial f^\star(\boldsymbol{\theta})$, then from Theorem 5.7, we have $f^\star(\boldsymbol{\theta}) = \langle \boldsymbol{\theta}, \boldsymbol{x}' \rangle - f(\boldsymbol{x}')$ but from the uniqueness of the maximizer we have that $\boldsymbol{x}^\star = \boldsymbol{x}'$.

Now, let's prove the that gradient of $f^\star$ is $\frac{1}{\lambda}$-Lipschitz w.r.t. $\|\cdot\|_\star$. For any $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, set $\boldsymbol{x}_1 = \nabla f^\star(\boldsymbol{\theta}_1)$ and $\boldsymbol{x}_2 = \nabla f^\star(\boldsymbol{\theta}_2)$. From Theorem 5.7, we have that $\boldsymbol{\theta}_1 \in \partial f(\boldsymbol{x}_1)$ and $\boldsymbol{\theta}_2 \in \partial f(\boldsymbol{x}_2)$. Hence, by Lemma 4.2, we have

$$
\begin{aligned}
f(\boldsymbol{x}_2) &\geq f(\boldsymbol{x}_1) + \langle \boldsymbol{\theta}_1, \boldsymbol{x}_2 - \boldsymbol{x}_1 \rangle + \frac{\lambda}{2} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2, \\
f(\boldsymbol{x}_1) &\geq f(\boldsymbol{x}_2) + \langle \boldsymbol{\theta}_2, \boldsymbol{x}_1 - \boldsymbol{x}_2 \rangle + \frac{\lambda}{2} \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2 \,.
\end{aligned}
$$

Summing these two inequalities, we have

$$
\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_\star \|\boldsymbol{x}_1 - \boldsymbol{x}_2\| \geq \langle \boldsymbol{\theta}_2 - \boldsymbol{\theta}_1, \boldsymbol{x}_1 - \boldsymbol{x}_2 \rangle \geq \lambda \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2,
$$

where in the first inequality we used the definition of dual norms. Solving the inequality we get that

$$
\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_\star \geq \lambda \|\boldsymbol{x}_1 - \boldsymbol{x}_2\| = \lambda \|\nabla f^\star(\boldsymbol{\theta}_1) - \nabla f^\star(\boldsymbol{\theta}_2)\| \,.
$$

Let's now prove the other direction. Assume that $f^\star$ is $\frac{1}{\lambda}$-smooth w.r.t. $\|\cdot\|_\star$ on $\mathbb{R}^d$. Set $\boldsymbol{y} \in \operatorname{dom} \partial f$ and $\boldsymbol{u} \in \partial f(\boldsymbol{y})$. Hence, by Theorem 5.7 and the differentiability of $f^\star$, we also have $\boldsymbol{y} = \nabla f^\star(\boldsymbol{u})$. Define $\phi(\boldsymbol{\theta}) := f^\star(\boldsymbol{\theta} + \boldsymbol{u}) - f^\star(\boldsymbol{u}) - \langle \boldsymbol{\theta}, \nabla f^\star(\boldsymbol{u}) \rangle$. From the $\frac{1}{\lambda}$-smoothness and Lemma 4.22, we have $\phi(\boldsymbol{\theta}) \leq \frac{1}{2\lambda} \|\boldsymbol{\theta}\|_\star^2$. From Lemma 5.13 and Example 5.11, we have that $\phi^\star(\boldsymbol{x}) \geq \frac{\lambda}{2} \|\boldsymbol{x}\|^2$. Let's now calculate $\phi^\star(\boldsymbol{x})$.

$$
\begin{aligned}
\phi^\star(\boldsymbol{x}) &= \sup_{\boldsymbol{\theta}} \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle - f^\star(\boldsymbol{\theta} + \boldsymbol{u}) + f^\star(\boldsymbol{u}) + \langle \boldsymbol{\theta}, \nabla f^\star(\boldsymbol{u}) \rangle \\
&= f^\star(\boldsymbol{u}) - \langle \boldsymbol{u}, \boldsymbol{x} + \nabla f^\star(\boldsymbol{u}) \rangle + \sup_{\boldsymbol{v}} \langle \boldsymbol{v}, \boldsymbol{x} + \nabla f^\star(\boldsymbol{u}) \rangle - f^\star(\boldsymbol{v}) \\
&= f^\star(\boldsymbol{u}) - \langle \boldsymbol{u}, \boldsymbol{x} + \nabla f^\star(\boldsymbol{u}) \rangle + f(\boldsymbol{x} + \nabla f^\star(\boldsymbol{u})) \\
&= -\langle \boldsymbol{u}, \boldsymbol{x} \rangle - f(\nabla f^\star(\boldsymbol{u})) + f(\boldsymbol{x} + \nabla f^\star(\boldsymbol{u})),
\end{aligned}
$$

where we used Theorem 5.6 in the third equality and Theorem 5.7 in the last one. Putting all together, we have

$$
f(\boldsymbol{x} + \boldsymbol{y}) - f(\boldsymbol{y}) - \langle \boldsymbol{u}, \boldsymbol{x} \rangle \geq \frac{\lambda}{2} \|\boldsymbol{x}\|^2, \; \forall \boldsymbol{u} \in \partial f(\boldsymbol{y}) \,. \qquad \square
$$

We will also use the following theorem on the first-order optimality condition.

Theorem 6.12. *Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ proper. Then $\boldsymbol{x}^\star \in \operatorname{argmin}_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x})$ iff $\mathbf{0} \in \partial f(\boldsymbol{x}^\star)$.*

Proof. We have that

$$
\boldsymbol{x}^\star \in \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x}) \Leftrightarrow \forall \boldsymbol{y} \in \mathbb{R}^d, \; f(\boldsymbol{y}) \geq f(\boldsymbol{x}^\star) = f(\boldsymbol{x}^\star) + \langle \mathbf{0}, \boldsymbol{y} - \boldsymbol{x}^\star \rangle \Leftrightarrow \mathbf{0} \in \partial f(\boldsymbol{x}^\star) \,. \qquad \square
$$

Hence, we can state the following theorem.

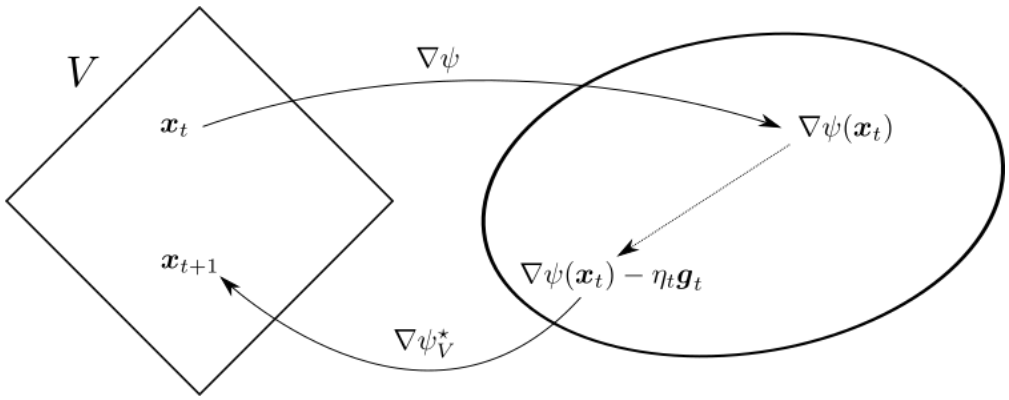

Figure 6.4: OMD update in terms of duality mappings.

Theorem 6.13. *Let B_ψ the Bregman divergence w.r.t. $\psi : X \to \mathbb{R}$, where ψ is $\lambda > 0$ strongly convex and closed. Let $V \subseteq X$ a non-empty closed convex set and $\boldsymbol{x}_t \in V$. Define*

$$\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x} \in V} \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x}; \boldsymbol{x}_t),$$

and assume ψ to be differentiable in $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$. Then, for any $\boldsymbol{g}_t \in \mathbb{R}^d$, we have

$$\boldsymbol{x}_{t+1} = \nabla\psi_V^\star(\nabla\psi(\boldsymbol{x}_t) - \eta_t \boldsymbol{g}_t), \tag{6.8}$$

where ψ_V the restriction of ψ to V, that is $\psi_V := \psi + i_V$.

Proof. We have that

$$\begin{aligned}
\boldsymbol{x}_{t+1} &= \operatorname*{argmin}_{\boldsymbol{x} \in V} \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x}; \boldsymbol{x}_t) \\
&= \operatorname*{argmin}_{\boldsymbol{x} \in V} \ \eta_t \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + B_\psi(\boldsymbol{x}; \boldsymbol{x}_t) \\
&= \operatorname*{argmin}_{\boldsymbol{x} \in V} \ \eta_t \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \psi(\boldsymbol{x}) - \psi(\boldsymbol{x}_t) - \langle \nabla\psi(\boldsymbol{x}_t), \boldsymbol{x} - \boldsymbol{x}_t \rangle \\
&= \operatorname*{argmin}_{\boldsymbol{x} \in V} \langle \eta_t \boldsymbol{g}_t - \nabla\psi(\boldsymbol{x}_t), \boldsymbol{x} \rangle + \psi(\boldsymbol{x}) \ .
\end{aligned}$$

Now, we use the first-order optimality condition in Theorem 6.12, to have

$$\boldsymbol{0} \in \eta_t \boldsymbol{g}_t + \nabla\psi(\boldsymbol{x}_{t+1}) - \nabla\psi(\boldsymbol{x}_t) + \partial i_V(\boldsymbol{x}_{t+1}),$$

that is

$$\nabla\psi(\boldsymbol{x}_t) - \eta_t \boldsymbol{g}_t \in (\nabla\psi + \partial i_V)(\boldsymbol{x}_{t+1}) \subseteq \partial\psi_V(\boldsymbol{x}_{t+1}),$$

where in the last inclusion we used Theorem 2.18. Hence, from Theorem 5.7, we have

$$\boldsymbol{x}_{t+1} \in \partial\psi_V^\star(\nabla\psi(\boldsymbol{x}_t) - \eta_t \boldsymbol{g}_t) \ .$$

Using that fact that $\psi_V := \psi + i_V$ is λ-strongly convex, proper, and closed, from Theorem 6.11 we have that $\partial\psi_V^\star = \{\nabla\psi_V^\star\}$. Hence,

$$\boldsymbol{x}_{t+1} = \nabla\psi_V^\star(\nabla\psi(\boldsymbol{x}_t) - \eta_t \boldsymbol{g}_t) \ . \qquad \square$$

Let's explain what this theorem says. We said that Online Mirror Descent extends the Online Subgradient Descent method to non-euclidean norms. Hence, the regret bound we proved contains dual norms, that measure the iterate and

the gradients. We also said that it makes sense to use a dual norm to measure a gradient, because it is a natural way to measure how "big" is the linear functional $\boldsymbol{x} \to \langle \nabla f(\boldsymbol{y}), \boldsymbol{x} \rangle$. In a more correct way, gradients actually live in the *dual space*, that is in a different space of the predictions $\boldsymbol{x}_t$. Hence, we cannot sum iterates and gradients together, in the same way in which we cannot sum pears and apples together. So, why we were doing it in OSD? The reason is that in that case the dual space coincides with the primal space. But, it is a very particular case due to the fact that we used the L_2 norm. Instead, in the general case, iterates and gradients are in two different spaces.

So, in OMD we need a way to go from one space to the other. And this is exactly the role of $\nabla \psi$ and $\nabla \psi_V^\star$, that are called *duality mappings*. We can now understand that the theorem tells us that OMD takes the primal vector $\boldsymbol{x}_t$, transforms it into a dual vector through $\nabla \psi$, does a subgradient descent step in the dual space, and finally transforms the vector back to the primal space through $\nabla \psi^\star$. This reasoning is summarized in Figure 6.4.

Example 6.14. *Let $\psi : \mathbb{R}^d \to \mathbb{R}$ equal to $\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$ and $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 \leq 1\}$. Define $\psi_V = \psi + i_V$. Then,*

$$\psi_V^\star(\boldsymbol{\theta}) = \sup_{\boldsymbol{x} \in V} \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle - \frac{1}{2}\|\boldsymbol{x}\|_2^2$$

First of all, if $\boldsymbol{\theta} = \boldsymbol{0}$ we have that $\psi_V^\star(\boldsymbol{\theta}) = \boldsymbol{0}$. So, in the following we assume $\boldsymbol{\theta} \neq \boldsymbol{0}$.

Now for any $\boldsymbol{x} \in V$ there exist $\boldsymbol{q}$ and α such that $\boldsymbol{x} = \alpha \frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|_2} + \boldsymbol{q}$ and $\langle \boldsymbol{q}, \boldsymbol{\theta} \rangle = 0$. Hence, we have

$$\sup_{\boldsymbol{x} \in V} \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle - \frac{1}{2}\|\boldsymbol{x}\|_2^2 = \sup_{\alpha, \boldsymbol{q} : \alpha \frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|_2} + \boldsymbol{q} \in V, \langle \boldsymbol{q}, \boldsymbol{\theta} \rangle = 0} \alpha \|\boldsymbol{\theta}\|_2 - \frac{\alpha^2}{2} - \frac{1}{2}\|\boldsymbol{q}\|_2^2 = \sup_{-1 \leq \alpha \leq 1} \alpha \|\boldsymbol{\theta}\|_2 - \frac{\alpha^2}{2} \ .$$

Solving the constrained optimization problem, we have $\alpha^\star = \min(1, \|\boldsymbol{\theta}\|_2)$. Hence, we have

$$\psi_V^\star(\boldsymbol{\theta}) = \begin{cases} \frac{1}{2}\|\boldsymbol{\theta}\|_2^2, & \|\boldsymbol{\theta}\|_2 \leq 1 \\ \|\boldsymbol{\theta}\|_2 - \frac{1}{2}, & \|\boldsymbol{\theta}\|_2 > 1 \end{cases},$$

that is finite everywhere and differentiable. So, we have $\nabla \psi(\boldsymbol{x}) = \boldsymbol{x}$ and

$$\nabla \psi_V^\star(\boldsymbol{\theta}) = \begin{cases} \boldsymbol{\theta}, & \|\boldsymbol{\theta}\|_2 \leq 1 \\ \frac{\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|_2}, & \|\boldsymbol{\theta}\|_2 > 1 \end{cases} = \Pi_V(\boldsymbol{\theta}) \ .$$

So, using (6.8)*, we obtain exactly the update of projected online subgradient descent.*

6.4.2 Yet Another Way to Write the Online Mirror Descent Update

There exists yet another way to write the update of OMD. This third method uses the concept of *Bregman projections*. Extending the definition of Euclidean projections, we can define the projection with respect to a Bregman divergence. Let $\Pi_{V,\psi}$ be defined by

$$\Pi_{V,\psi}(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{y} \in V} \ B_\psi(\boldsymbol{y}; \boldsymbol{x}) \ .$$

In the online learning literature, the OMD algorithm is typically presented with a two step update: first, solving the argmin over the entire space and then projecting back over V with respect to the Bregman divergence. In the following, we show that most of the time the two-step update is equivalent to the one-step update in (6.8).

First, we prove a general theorem that allows to break the constrained minimization of functions in the minimization over the entire space plus and Bregman projection step.

Theorem 6.15. *Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ proper, closed, strictly convex, and differentiable in $\operatorname{int} \operatorname{dom} f$. Also, let $V \subset \mathbb{R}^d$ a non-empty, closed convex set with $V \cap \operatorname{dom} f \neq \{\}$ and assume that $\tilde{\boldsymbol{y}} = \operatorname{argmin}_{\boldsymbol{z} \in \mathbb{R}^d} \ f(\boldsymbol{z})$ exists and $\tilde{\boldsymbol{y}} \in \operatorname{int} \operatorname{dom} f$. Denote by $\boldsymbol{y}' = \operatorname{argmin}_{\boldsymbol{z} \in V} \ B_f(\boldsymbol{z}; \tilde{\boldsymbol{y}})$. Then, the following hold:*

1. *$\boldsymbol{y} = \operatorname{argmin}_{\boldsymbol{z} \in V} \ f(\boldsymbol{z})$ exists and is unique.*
2. *$\boldsymbol{y} = \boldsymbol{y}'$.*

Proof. For the first point, from [Bauschke and Combettes, 2011, Proposition 11.12] and the existence of $\tilde{\boldsymbol{y}}$, we have that f is coercive. So, from Bauschke and Combettes [2011, Proposition 11.14], the minimizer of f in V exists. Given that f is strictly convex, the minimizer must be unique too.

For the second point, from the definition of $\boldsymbol{y}$, we have $f(\boldsymbol{y}) \leq f(\boldsymbol{y}')$. On the other hand, from the first-order optimality condition, we have $\nabla f(\tilde{\boldsymbol{y}}) = \mathbf{0}$. So, we have

$$f(\boldsymbol{y}') - f(\tilde{\boldsymbol{y}}) = B_f(\boldsymbol{y}'; \tilde{\boldsymbol{y}}) \leq B_f(\boldsymbol{y}; \tilde{\boldsymbol{y}}) = f(\boldsymbol{y}) - f(\tilde{\boldsymbol{y}}),$$

that is $f(\boldsymbol{y}') \leq f(\boldsymbol{y})$. Given that f is strictly convex, $\boldsymbol{y}' = \boldsymbol{y}$. □

Now, note that, if $\tilde{\psi}(\boldsymbol{x}) = \psi(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{x} \rangle$, then

$$\begin{aligned} B_{\tilde{\psi}}(\boldsymbol{x}; \boldsymbol{y}) &= \tilde{\psi}(\boldsymbol{x}) - \tilde{\psi}(\boldsymbol{y}) - \langle \nabla \tilde{\psi}(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \\ &= \psi(\boldsymbol{x}) - \psi(\boldsymbol{y}) - \langle \boldsymbol{g} + \nabla \psi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle + \langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{y} \rangle \\ &= \psi(\boldsymbol{x}) - \psi(\boldsymbol{y}) - \langle \nabla \psi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle \\ &= B_\psi(\boldsymbol{x}; \boldsymbol{y}) \, . \end{aligned}$$

Now, define $f(\boldsymbol{x}) = \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x} \rangle + B_\psi(\boldsymbol{x}; \boldsymbol{x}_t)$, so that $f(\boldsymbol{x}) = \psi(\boldsymbol{x}) + \langle \boldsymbol{z}, \boldsymbol{x} \rangle + K$ for some $K \in \mathbb{R}$ and $\boldsymbol{z} \in \mathbb{R}^d$. This implies that $B_f(\boldsymbol{x}; \boldsymbol{y}) = B_\psi(\boldsymbol{x}; \boldsymbol{y})$. Hence, under the assumption of the above theorem, we have that $\boldsymbol{x}_{t+1} = \operatorname{argmin}_{\boldsymbol{x} \in V} \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x}; \boldsymbol{x}_t)$ is equivalent to

$$\begin{aligned} \tilde{\boldsymbol{x}}_{t+1} &= \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{R}^d} \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x} \rangle + B_\psi(\boldsymbol{x}; \boldsymbol{x}_t), \\ \boldsymbol{x}_{t+1} &= \operatorname*{argmin}_{\boldsymbol{x} \in V} \; B_\psi(\boldsymbol{x}; \tilde{\boldsymbol{x}}_{t+1}) \, . \end{aligned}$$

The advantage of this update is that sometimes it gives two easier problems to solve rather than a single difficult one.

6.5 OMD Regret Bound using Local Norms

In Lemma 6.9, strong convexity basically tells us some minimum curvature in all the directions, that allows to upper bound the difference between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$. However, it turns out that we can still get a meaningful regret upper bound without this assumption. In particular, we can get an interesting expression for the regret that involves the use of **local norms**. We will use these ideas in the section on Multi-armed Bandits (Chapter 10).

Lemma 6.16. *Let B_ψ the Bregman divergence w.r.t. $\psi : X \to \mathbb{R}$ and assume ψ twice differentiable and with the Hessian positive definite in the interior of its domain. Let $V \subseteq X$ a non-empty closed convex set. Assume* (6.5) *or* (6.6) *to hold. Define $\|\boldsymbol{x}\|_{\boldsymbol{A}} := \sqrt{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}$. Also, with the notation in Algorithm 6.1, assume $\boldsymbol{x}_{t+1}$ and $\tilde{\boldsymbol{x}}_{t+1} \in \operatorname{argmin}_{\boldsymbol{x} \in X} \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x}; \boldsymbol{x}_t)$ exist. Then, $\forall \boldsymbol{u} \in V$, there exists $\boldsymbol{z}_t$ on the line segments between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$, and $\boldsymbol{z}'_t$ on the line segments between $\boldsymbol{x}_t$ and $\tilde{\boldsymbol{x}}_{t+1}$, such that the following inequality holds*

$$\eta_t(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \eta_t \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) + \frac{\eta_t^2}{2} \min \left(\|\boldsymbol{g}_t\|^2_{(\nabla^2 \psi_t(\boldsymbol{z}_t))^{-1}}, \|\boldsymbol{g}_t\|^2_{(\nabla^2 \psi_t(\boldsymbol{z}'_t))^{-1}} \right) .$$

Proof. First of all, from Theorem 6.7, $\boldsymbol{x}_t$ and $\tilde{\boldsymbol{x}}_t$ are in the interior of X for all $t \geq 1$. Then, from Lemma 6.9, we have

$$\langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq B_\psi(\boldsymbol{u}; \boldsymbol{x}_t) - B_\psi(\boldsymbol{u}; \boldsymbol{x}_{t+1}) - B_\psi(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) + \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle \, . \tag{6.9}$$

From the Taylor's theorem, we have said that $B_\psi(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) = \frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t)^\top \nabla^2 \psi(\boldsymbol{z}_t)(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t)$ for some $\boldsymbol{z}_t$ on the line segment between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$. Observe that this is $\frac{1}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|^2_{\nabla^2 \psi(\boldsymbol{z}_t)}$ and it is indeed a norm because we assumed the Hessian of ψ to be positive definite. Hence, by Fenchel-Young inequality and Examples 4.18 and 5.11, we have

$$\begin{aligned} \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - B_\psi(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) &\leq \frac{\eta_t^2}{2} \|\boldsymbol{g}_t\|^2_{(\nabla^2 \psi(\boldsymbol{z}_t))^{-1}} + \frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t)^\top \nabla^2 \psi(\boldsymbol{z}_t)(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t) - B_\psi(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) \\ &= \frac{\eta_t^2}{2} \|\boldsymbol{g}_t\|^2_{(\nabla^2 \psi(\boldsymbol{z}_t))^{-1}}, \end{aligned}$$

that gives the first term in the minimum.

For the second term in the minimum, we instead observe that

$$\langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - B_\psi(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) \leq \max_{\boldsymbol{x} \in X} \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x} \rangle - B_\psi(\boldsymbol{x}; \boldsymbol{x}_t) = \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \tilde{\boldsymbol{x}}_{t+1} \rangle - B_\psi(\tilde{\boldsymbol{x}}_{t+1}; \boldsymbol{x}_t) \ .$$

Then, we proceed as in the first bound. □

Despite the apparent more difficult formulation, the second term in the minimum is often easier to use, especially in constrained settings because $\tilde{\boldsymbol{x}}_{t+1}$ is defined over X rather than over V. Also, under the assumptions of Theorem 6.15, it is easy to recognize that $\boldsymbol{x}_{t+1}$ is the Bregman projection of $\tilde{\boldsymbol{x}}_{t+1}$ onto V.

6.6 Example of OMD: Exponentiated Gradient

Algorithm 6.2 Exponentiated Gradient

Require: $\eta > 0$

Set $\boldsymbol{x}_1 = [1/d, \dots, 1/d]$
for $t = 1$ **to** T **do**
  Output $\boldsymbol{x}_t \in \Delta^{d-1}$
  Receive $\ell_t : \Delta^{d-1} \to \mathbb{R}$ subdifferentiable in Δ^{d-1} and pay $\ell_t(\boldsymbol{x}_t)$
  Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
  $x_{t+1,j} = \frac{x_{t,j} \exp(-\eta g_{t,j})}{\sum_{i=1}^d x_{t,i} \exp(-\eta g_{t,i})}, \ j = 1, \dots, d$
end for

Let $\Delta^{d-1} = \{\boldsymbol{x} \in \mathbb{R}^d : x_i \geq 0, \|\boldsymbol{x}\|_1 = 1\}$ the probability simplex and set $V = \Delta^{d-1}$. So, in words, we want to output discrete probability distributions over $\mathbb{R}^d$. Also, let $X = \mathbb{R}^d_{\geq 0}$ and $\psi(\boldsymbol{x}) : X \to \mathbb{R}$ defined as $\psi(\boldsymbol{x}) = \sum_{i=1}^d x_i \ln x_i$, where we define $0 \ln(0) = 0$. Note that the restriction of ψ to V is the negative entropy of the discrete distributions in Δ^{d-1}. It is possible to verify that ψ satisfies the first condition in Theorem 6.7, hence the update is well defined.

The Fenchel conjugate $\psi_V^\star(\boldsymbol{\theta})$ is defined as

$$\psi_V^\star(\boldsymbol{\theta}) = \sup_{\boldsymbol{x} \in V} \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle - \psi(\boldsymbol{x}) = \sup_{\boldsymbol{x} \in V} \langle \boldsymbol{\theta}, \boldsymbol{x} \rangle - \sum_{i=1}^d x_i \ln x_i \ .$$

It is a constrained optimization problem, we could solve it using the KKT conditions. However, there is a simpler way to do it: We will remove the probability simplex constraint rephrasing the problem over $d-1$ variables. In fact, the maximization problem is equivalent to

$$\min_{\boldsymbol{x} \in \mathbb{R}^{d-1}} \ \sum_{i=1}^{d-1} x_i \ln x_i + (1 - \sum_{i=1}^{d-1} x_i) \ln \left(1 - \sum_{i=1}^{d-1} x_i \right) - \sum_{i=1}^{d-1} \theta_i x_i - \theta_d \left(1 - \sum_{i=1}^{d-1} x_i \right) \ .$$

Note that the constraint on $x_1, \dots, x_{d-1}$ and $1 - \sum_{i=1}^{d-1} x_i$ to be non-negative is enforced by the domain of the logarithm. This is now an unconstrained concave optimization problem, so we can solve it equating the gradient of the objective function to zero. Hence, we have

$$\ln \frac{x_i}{1 - \sum_{j=1}^{d-1} x_j} = \theta_i - \theta_d, \ i = 1, \dots, d-1 \ .$$

That is

$$x_i = \exp(\theta_i - \theta_d) \left(1 - \sum_{j=1}^{d-1} x_j \right), \ i = 1, \dots, d-1 \ . \tag{6.10}$$

Summing this equality over $i = 1, \dots, d-1$, we obtain

$$\sum_{i=1}^{d-1} x_i = \sum_{i=1}^{d-1} \exp(\theta_i - \theta_d) \left(1 - \sum_{j=1}^{d-1} x_j \right)$$

that can be solved to obtain

$$1 - \sum_{j=1}^{d-1} x_i = \frac{1}{1 + \sum_{j=1}^{d-1} \exp(\theta_j - \theta_d)} .$$

Substituting it back in (6.10), we have

$$x_i = \frac{\exp(\theta_i - \theta_d)}{1 + \sum_{j=1}^{d-1} \exp(\theta_j - \theta_d)} = \frac{\exp(\theta_i)}{\sum_{j=1}^{d} \exp(\theta_j)}, \ i = 1, \dots, d .$$

Denoting with $\alpha = \sum_{i=1}^{d} \exp(\theta_i)$, and substituting in the definition of the conjugate function we get

$$\psi_V^\star(\boldsymbol{\theta}) = \sum_{i=1}^{d} \left(\frac{1}{\alpha} \theta_i \exp(\theta_i) - \frac{1}{\alpha} \exp(\theta_i)(\theta_i - \ln(\alpha)) \right) = \ln(\alpha) \frac{1}{\alpha} \sum_{i=1}^{d} \exp(\theta_i) = \ln(\alpha) = \ln \left(\sum_{i=1}^{d} \exp(\theta_i) \right) .$$

We also have $(\nabla \psi_V^\star(\boldsymbol{\theta}))_j = \frac{\exp(\theta_j)}{\sum_{i=1}^{d} \exp(\theta_i)}$ and $(\nabla \psi(\boldsymbol{x}))_j = \ln(x_j) + 1$ for $\boldsymbol{x} \in \mathbb{R}^d_{>0}$.

Putting all together, we have the online mirror descent update rule for entropic distance generating function.

$$x_{t+1,j} = \frac{\exp(\ln x_{t,j} + 1 - \eta_t g_{t,j})}{\sum_{i=1}^{d} \exp(\ln x_{t,i} + 1 - \eta_t g_{t,i})} = \frac{x_{t,j} \exp(-\eta_t g_{t,j})}{\sum_{i=1}^{d} x_{t,i} \exp(-\eta_t g_{t,i})} .$$

The algorithm is summarized in Algorithm 6.2. This algorithm is called *Exponentiated Gradient* (EG) because in the update rule we take the component-wise exponent of the (sub)gradient vector.

Let's take a look at the regret bound we get. From Example 6.5, for all $\boldsymbol{x} \in \Delta^{d-1}$ and $\boldsymbol{y} \in \{\boldsymbol{x} \in \mathbb{R}^d : x_i > 0, \|\boldsymbol{x}\|_1 = 1\}$, we have $B_\psi(\boldsymbol{x}; \boldsymbol{y}) = \sum_{i=1}^{d} x_i \ln \frac{x_i}{y_i}$. Now, we prove the strong convexity of ψ.

Lemma 6.17. *$\psi(\boldsymbol{x}) = \sum_{i=1}^{d} x_i \ln x_i$ is 1-strongly convex with respect to the L_1 norm over the set $K = \{\boldsymbol{x} \in \mathbb{R}^d : x_i > 0, \|\boldsymbol{x}\|_1 = 1\}$.*

Proof. The statement is implied by Pinsker's inequality, but here we give a short and direct proof from Beck and Teboulle [2003, Proposition 5.1].

Let $\phi(u) = (u-1) \ln u - \frac{2(u-1)^2}{u+1}$ for $u > 0$. Observe that $\phi''(u) > 0$ for $u > 0$ so the function is convex. Moreover, $\phi(1) = \phi'(1) = 0$. So, we have $\phi(u) \geq \phi(1) + \phi'(1)(x-1) = 0$ for all $\boldsymbol{u} > 0$.

Using this inequality, we have

$$\begin{aligned} \langle \nabla \psi(\boldsymbol{x}) - \nabla \psi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle &= \sum_{i=1}^{d} (x_i - y_i) \ln \frac{x_i}{y_i} = \sum_{i=1}^{d} y_i \left(\frac{x_i}{y_i} - 1 \right) \ln \frac{x_i}{y_i} \geq \sum_{i=1}^{d} \frac{2(x_i - y_i)^2}{x_i + y_i} \\ &= \sum_{i=1}^{d} \frac{x_i + y_i}{2} \left(\frac{|x_i - y_i|}{\frac{x_i + y_i}{2}} \right)^2 \geq \left(\sum_{i=1}^{d} \frac{x_i + y_i}{2} \frac{|x_i - y_i|}{\frac{x_i + y_i}{2}} \right)^2 = \|\boldsymbol{x} - \boldsymbol{y}\|_1^2, \end{aligned}$$

where in the last inequality we used Jensen's inequality because $\frac{\boldsymbol{x}+\boldsymbol{y}}{2} \in K$. Using Theorem 4.3 completes the proof. □

Another thing to decide is the initial point $\boldsymbol{x}_1$. We can set $\boldsymbol{x}_1$ to be the minimizer of ψ in V. In this way $B_\psi(\boldsymbol{u}; \boldsymbol{x}_1)$ simplifies to $\psi(\boldsymbol{u}) - \min_{\boldsymbol{x} \in V} \psi(\boldsymbol{x})$. Hence, we set $\boldsymbol{x}_1 = [1/d, \dots, 1/d] \in \mathbb{R}^d$, because the uniform distribution minimizes the negative entropy. So, we have $B_\psi(\boldsymbol{u}; \boldsymbol{x}_1) = \sum_{i=1}^{d} u_i \ln u_i + \ln d \leq \ln d$.

Putting all together, we have

$$\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \frac{\ln d}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_\infty^2 \,.$$

Assuming $\|\boldsymbol{g}_t\|_\infty \leq L_\infty$, we can set $\eta = \sqrt{\frac{2\ln d}{L_\infty^2 T}}$, to obtain that a regret of $\frac{\sqrt{2}}{2}L_\infty\sqrt{T\ln d}$.

Remark 6.18. *Note that the time-varying version of OMD with entropic distance generating function would give rise to a vacuous bound, can you see why? We will see how FTRL overcomes this issue using a time-varying regularizer rather than a time-varying learning rate.*

We can also get a *tighter* bound using the local norms. Let's use the additional assumption that $g_{t,i} \geq 0$, for all $t = 1, \dots, T$ and $i = 1, \dots, d$. Summing the inequality of Lemma 6.16 from $t = 1$ to T, we have for all $\boldsymbol{u} \in V$ that

$$\sum_{t=1}^{T}\ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T}\ell(\boldsymbol{u}) \leq \frac{\ln d}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|^2_{(\nabla^2\psi(\boldsymbol{z}'_t))^{-1}},$$

where $\boldsymbol{z}'_t$ is on the line segment between $\boldsymbol{x}_t$ and $\tilde{\boldsymbol{x}}_{t+1}$. In this case, it is easy to calculate $\tilde{x}_{t+1,i}$ as $x_{t,i}\exp(-\eta g_{t,i})$ for $i = 1, \dots, d$. Moreover, $\nabla^2\psi(\boldsymbol{z}'_t)$ is a diagonal matrix whose elements on the diagonal are $\frac{1}{z'_{t,i}}, i = 1, \dots, d$. Hence, we have that

$$\|\boldsymbol{g}_t\|^2_{(\nabla^2\psi(\boldsymbol{z}'_t))^{-1}} = \sum_{i=1}^{d} g_{t,i}^2 z'_{t,i} \leq \sum_{i=1}^{d} g_{t,i}^2 x_{t,i} \,.$$

Putting all together, the final bound would be

$$\sum_{t=1}^{T}\ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T}\ell(\boldsymbol{u}) \leq \frac{\ln d}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\sum_{i=1}^{d} g_{t,i}^2 x_{t,i} \,.$$

This is indeed a tighter bound because $\sum_{i=1}^{d} g_{t,i}^2 x_{t,i} \leq \|\boldsymbol{g}_t\|_\infty^2$.

How would Online Subgradient Descent (OSD) work on the same problem? First, it is important to realize that nothing prevents us to use OSD on this problem. We just have to implement the euclidean projection onto the probability simplex. The regret bound we would get from OSD is

$$\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \frac{2}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|\boldsymbol{g}_t\|_2^2,$$

where we set $\boldsymbol{x}_1 = [1/d, \dots, 1/d]$ and $\|\boldsymbol{u} - \boldsymbol{x}_1\|_2 \leq 2$ for any $\boldsymbol{u} \in V$. Assuming $\|\boldsymbol{g}_t\|_\infty \leq L_\infty$, we have that in the worst case $\|\boldsymbol{g}_t\|_2^2 \leq dL_\infty^2$. Hence, we can set $\eta = \sqrt{\frac{4}{dL_\infty^2 T}}$, to obtain that a regret of $2L_\infty\sqrt{dT}$. Hence, in a worst case sense, using an entropic distance generating function transforms a dependency on the dimension from $\sqrt{d}$ to $\sqrt{\ln d}$ for Online Convex Optimization (OCO) over the probability simplex.

So, as we already saw analyzing AdaGrad, the shape of the domain is the important ingredient when we change from euclidean norms to other norms.

6.7 Example of OMD: p-norm Algorithms

Consider the distance generating function $\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_p^2$, for $1 < p \leq 2$ over $X = V = \mathbb{R}^d$. Let's remind the reader that the p-norm of a vector $\boldsymbol{x}$ is defined as $(\sum_{i=1}^{d}|x_i|^p)^{\frac{1}{p}}$. From Examples 4.17 and 5.11, we have that $\psi_V^\star(\boldsymbol{\theta}) = \frac{1}{2}\|\boldsymbol{\theta}\|_q^2$, where $\frac{1}{p}+\frac{1}{q} = 1$, so that $q \geq 2$. Let's calculate the dual maps: $(\nabla\psi(\boldsymbol{x}))_j = \operatorname{sign}(x_j)|x_j|^{p-1}\|\boldsymbol{x}\|_p^{2-p}$ and $(\nabla\psi_V^\star(\boldsymbol{x}))_j = \operatorname{sign}(x_j)|x_j|^{q-1}\|\boldsymbol{x}\|_q^{2-q}$. Hence, we can write the update rule as

$$\tilde{x}_{t+1,j} = \operatorname{sign}(x_{t,j})|x_{t,j}|^{p-1}\|\boldsymbol{x}_t\|_p^{2-p} - \eta g_{t,j},\ j = 1, \dots, d,$$
$$x_{t+1,j} = \operatorname{sign}(\tilde{x}_{t+1,j})|\tilde{x}_{t+1,j}|^{q-1}\left\|\tilde{\boldsymbol{x}}_{t+1}\right\|_q^{2-q},\ j = 1, \dots, d,$$

where we broke the update in two steps to simplify the notation (and the implementation). Starting from $\boldsymbol{x}_1 = \mathbf{0}$, we have that

$$B_\psi(\boldsymbol{u}; \boldsymbol{x}_1) = \psi(\boldsymbol{u}) - \psi(\boldsymbol{x}_1) - \langle \nabla\psi(\boldsymbol{x}_1), \boldsymbol{u} - \boldsymbol{x}_1 \rangle = \psi(\boldsymbol{u}) \ .$$

The last ingredient is the fact that $\psi(\boldsymbol{x})$ is $p-1$ strongly convex with respect to $\|\cdot\|_p$.

Lemma 6.19 ([Shalev-Shwartz, 2007, Lemma 17])**.** *$\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_p^2$ is $(p-1)$-strongly convex with respect to $\|\cdot\|_p$, for $1 \le p \le 2$.*

Hence, the regret bound will be

$$\sum_{t=1}^{T} (\ell_t(\boldsymbol{w}_t) - \ell_t(\boldsymbol{u})) \le \frac{\|\boldsymbol{u}\|_p^2}{2\eta} + \frac{\eta}{2(p-1)} \sum_{t=1}^{T} \|\boldsymbol{g}_t\|_q^2 \ .$$

Setting $p = 2$, we get the (unprojected) Online Subgradient Descent. However, we can set p to achieve a logarithmic dependency in the dimension d as in EG. Let's assume again that $\|\boldsymbol{g}_t\|_\infty \le L_\infty$, so we have

$$\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_q^2 \le L_\infty^2 d^{2/q} T \ .$$

Also, note that $\|\boldsymbol{u}\|_p \le \|\boldsymbol{u}\|_1$, so we have an upper bound to the regret of

$$\mathrm{Regret}_T(\boldsymbol{u}) \le \frac{\|\boldsymbol{u}\|_1^2}{2\eta} + \frac{L_\infty^2 d^{2/q} T \eta}{2(p-1)}, \ \forall \boldsymbol{u} \in \mathbb{R}^d \ .$$

Setting $\eta = \frac{\alpha\sqrt{p-1}}{L_\infty d^{1/q}\sqrt{T}}$, we get an upper bound to the regret of

$$\frac{1}{2}\left(\frac{\|\boldsymbol{u}\|_1^2}{\alpha} + \alpha\right) L_\infty \sqrt{T} \frac{d^{1/q}}{\sqrt{p-1}} = \frac{1}{2}\left(\frac{\|\boldsymbol{u}\|_1^2}{\alpha} + \alpha\right) L_\infty \sqrt{T}\sqrt{q-1} d^{1/q} \le \frac{1}{2}\left(\frac{\|\boldsymbol{u}\|_1^2}{\alpha} + \alpha\right) L_\infty \sqrt{T}\sqrt{q} d^{1/q} \ .$$

Assuming $d \ge 3$, the choice of q that minimizes the last term is $q = 2\ln d$ that makes the term $\sqrt{q}d^{1/q} = \sqrt{2e\ln d}$. Hence, we have regret bound of the order of $O(\sqrt{T\ln d})$ as $T \to \infty$.

So, the p-norm allows to interpolate from the behaviour of OSD to the one of EG. Note that here the set V is the entire space, however we could still set $V = \{\boldsymbol{x} \in \mathbb{R}^d : x_i \ge 0, \|\boldsymbol{x}\|_1 = 1\}$. While this would allow us to get the same asymptotic bound of EG, the update would not be in a closed form anymore.

6.8 An Application of Online Mirror Descent: Learning with Expert Advice

Let's introduce a particular Online Convex Optimization game called *Learning with Expert Advice*.

In this setting, we have d experts that gives us some advice on each round. In turn, in each round we have to decide which expert we want to follow. After we made our choice, the losses associated to each expert are revealed and we pay the loss associated to the expert we picked. The aim of the game is to minimize the losses we make compared to cumulative losses of the best expert. This is a general setting that allows to model many interesting cases. For example, we have a number of different online learning algorithms and we would like to close to the best among them.

Is this problem solvable? If we put ourselves in the adversarial setting, unfortunately it cannot be solved! Indeed, even with 2 experts, the adversary can force on us linear regret. Let's see how. In each round we have to pick expert 1 or expert 2. In each round, the adversary can decide that the expert we pick has loss 1 and the other one has loss 0. This means that the cumulative loss of the algorithm over T rounds is T. On the other hand, the best cumulative loss over expert 1 and 2 is less than $T/2$. This means that our regret, no matter what we do, can be as big as $T/2$.

The problem above is due to the fact that the adversary has too much power. One way to reduce its power is using *randomization*. We can allow the algorithm to be randomized *and* force the adversary to decide the losses at time t without knowing the outcome of the randomization of the algorithm at time t (but it can depend on the past

randomization). This is enough to make the problem solvable. Moreover, it will also make the problem convex, allowing us to use any OCO algorithm on it.

First, let's write the problem in the original formulation. We set a discrete feasible set $V = \{\boldsymbol{e}_i\}_{i=1}^d$, where $\boldsymbol{e}_i$ is the vector will all zeros but a 1 in the coordinate i. Our predictions and the competitor are from V. The losses are linear losses: $\ell_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle$, for $t = 1, \dots, T$ and $i = 1, \dots, d$. The regret is

$$\text{Regret}_T(\boldsymbol{e}_i) = \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{e}_i \rangle, \; i = 1, \dots, d \, . \tag{6.11}$$

The only thing that makes this problem non-convex is the feasibility set, that is clearly a non-convex one.

Let's now see how the randomization makes this problem convex. Let's extend the feasible set to $V' = \{\boldsymbol{x} \in \mathbb{R}^d : x_i > 0, \|\boldsymbol{x}\|_1 = 1\}$. Note that $\boldsymbol{e}_i \in V'$. For this problem we can use an OCO algorithm to minimize the regret

$$\text{Regret}'_T(\boldsymbol{u}) = \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle, \forall \boldsymbol{u} \in V' \, .$$

Can we find a way to transform an upper bound to this regret to the one we care in (6.11)? One way is the following one: On each time step, construct a random variable i_t that is equal to i with probability $x_{t,i}$ for $i = 1, \dots, d$. Then, select the expert according to the outcome of i_t. Now, in expectation we have

$$\mathbb{E}[g_{t,i_t}] = \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle,$$

and

$$\mathbb{E}[\text{Regret}_T(\boldsymbol{e}_i)] = \mathbb{E}[\text{Regret}'_T(\boldsymbol{e}_i)] = \mathbb{E}\left[\sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{e}_i \rangle \right] \, .$$

This means that we can minimize in expectation the non-convex regret with a randomized OCO algorithm. We can summarize this reasoning in Algorithm 6.3.

Algorithm 6.3 Learning with Expert Advice through Randomization

Require: $\boldsymbol{x}_1 \in \{\boldsymbol{x} \in \mathbb{R}^d : x_i > 0, \|\boldsymbol{x}\|_1 = 1\}$

1: **for** $t = 1$ **to** T **do**
2: Draw i_t according to $P(i_t = i) = x_{t,i}$
3: Select expert i_t
4: Observe all the experts' losses $\boldsymbol{g}_t$ and pay the loss of the selected expert
5: Update $\boldsymbol{x}_t$ with an OCO algorithm with feasible set $\{\boldsymbol{x} \in \mathbb{R}^d : x_i > 0, \|\boldsymbol{x}\|_1 = 1\}$
6: **end for**

For example, assume that $\|\boldsymbol{g}_t\|_\infty \leq L_\infty$ for all t. Then, using the EG algorithm from Section 6.6, we obtain the following update rule

$$x_{t+1,j} = \frac{x_{t,j} \exp(-\eta g_{t,j})}{\sum_{i=1}^d x_{t,i} \exp(-\eta g_{t,i})}, \; j = 1, \dots, d,$$

where setting $\boldsymbol{x}_1 = [1/d, \dots, 1/d]$ and $\eta = \frac{\sqrt{2 \ln d}}{L_\infty \sqrt{T}}$. For such algorithm, the regret will be

$$\mathbb{E}[\text{Regret}_T(\boldsymbol{e}_i)] \leq \frac{\sqrt{2}}{2} L_\infty \sqrt{T \ln d}, \; \forall \boldsymbol{e}_i \, .$$

It is worth stressing the importance of the result just obtained: We can design an algorithm that in expectation is close to the best expert in a set, *paying only a logarithmic penalty in the size of the set*. The p-norm Algorithm in Section 6.7 would give a similar guarantee.

In Section 9.6.1, we will see algorithms that achieve even the better regret guarantee of $O(\sqrt{T \cdot \text{KL}(\boldsymbol{u}; \boldsymbol{x}_1)})$ as $T \to \infty$, for any $\boldsymbol{u}$ in the probability simplex. You should be able to convince yourself that no setting of η in EG allows to achieve such regret guarantee. Indeed, the algorithm will be based on a very different strategy.

6.9 Optimistic OMD

Till now, we have mainly considered the adversarial model as our model of the environment. This allowed us to design algorithm that work in this setting, as well as in other more benign settings. However, the world is never completely adversarial. So, we might be tempted to model the environment in some way, but that would leave our algorithm vulnerable to attacks. An alternative, is to consider the data as generated by some *predictable process plus an adversarial signal*. In this view, it might be beneficial to try to model the predictable part, without compromising the robustness to the adversarial signal.

In this section, we will explore this possibility through a particular version of OMD, where we *predict* the next gradient. In very intuitive terms, if our predicted gradient is correct, we can expect the regret to decrease. However, if our prediction is wrong we still want to recover the worst case guarantee. Such algorithm is called **Optimistic OMD**.

The core idea of Optimistic OMD is to predict the next gradient and use it in the update rule, as summarized in Algorithm 6.4. Here, at round t the algorithm receives a hint $\tilde{\boldsymbol{g}}_{t+1}$ on the next subgradient $\boldsymbol{g}_{t+1}$ and uses it to construct the update. At the same time, you have to remove the hint you used at the previous time step, $\tilde{\boldsymbol{g}}_t$. Note that for the sake of the analysis, it does not matter how the prediction is generated. It can be even generated by another online learning procedure!

Algorithm 6.4 Optimistic Online Mirror Descent

Require: Non-empty closed convex $V \subseteq X \subseteq \mathbb{R}^d$, $\psi : X \to \mathbb{R}$ strictly convex and differentiable on $\operatorname{int} X$, $\boldsymbol{x}_1 \in \operatorname{int} X$, $\eta_1, \dots, \eta_T > 0$

$\tilde{\boldsymbol{g}}_1 = \boldsymbol{0}$
for $t = 1$ **to** T **do**
  Output $\boldsymbol{x}_t$
  Receive $\ell_t : V \to \mathbb{R}$ subdifferentiable in V and pay $\ell_t(\boldsymbol{x}_t)$
  Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
  Predict next subgradient $\tilde{\boldsymbol{g}}_{t+1} \in \mathbb{R}^d$
  $\boldsymbol{x}_{t+1} \in \operatorname{argmin}_{\boldsymbol{x} \in V} \langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t + \tilde{\boldsymbol{g}}_{t+1}, \boldsymbol{x} \rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x}; \boldsymbol{x}_t)$
end for

To gain some intuition on why this update makes sense, consider the case that $\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$, $\eta_t = \eta$, and $V = \mathbb{R}^d$. In this case, $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \eta\tilde{\boldsymbol{g}}_t - \eta\boldsymbol{g}_t - \eta\tilde{\boldsymbol{g}}_{t+1}$. Unrolling the update, we get $\boldsymbol{x}_{t+1} = \boldsymbol{x}_1 - \eta(\tilde{\boldsymbol{g}}_{t+1} + \sum_{i=1}^t \boldsymbol{g}_i)$. Without hints, that is in plain OMD, under the same assumptions the unrolled update would be $\boldsymbol{x}_{t+1} = \boldsymbol{x}_1 - \eta\sum_{i=1}^t \boldsymbol{g}_i$ and $\boldsymbol{x}_{t+2} = \boldsymbol{x}_1 - \eta\sum_{i=1}^{t+1} \boldsymbol{g}_i$. Hence, $\tilde{\boldsymbol{g}}_{t+1}$ acts as a proxy for the next (unknown) subgradient $\boldsymbol{g}_t$.

Note that one might be tempted to multiply $\tilde{\boldsymbol{g}}_t$ by η_{t-1}, because in the previous iteration we used the learning rate η_{t-1}. However, the OMD proof reveals that the correct way to see the update is to think the learning rate as attached to the Bregman divergence rather than to the subgradients.

One might also be tempted to find a way to study this algorithm with a special proof. However, the one-step lemma we proved for OMD is essentially tight: we only used two inequalities, one to deal with the set V and the other one to linearize the losses. But, but steps can be made tight, considering $V = \mathbb{R}^d$ and linear losses. Hence, if the update is just OMD with a different sequence of subgradients, the proof *must* follow from the one of OMD with a different set of subgradients. This is a general rule: If we have a theorem based on a tight inequality, any other proof of the same theorem, no matter how complex, must be looser or in the best case equivalent.

Note that setting $\tilde{\boldsymbol{g}}_1 = \boldsymbol{0}$ is not a limitation because setting it to any other value would be equivalent to changing the arbitrary initial point $\boldsymbol{x}_1$.

Theorem 6.20. *Let B_ψ the Bregman divergence w.r.t. $\psi : X \to \mathbb{R}$ and assume ψ to be proper, closed, and λ-strongly convex with respect to $\|\cdot\|$ in V. Let $V \subseteq X$ a non-empty closed convex set. With the notation in Algorithm 6.4, assume $\boldsymbol{x}_{t+1}$ exists, and it is in* $\operatorname{int} X$.

Assume $\eta_{t+1} \leq \eta_t,\ t = 1, \dots, T$. Then, and $\forall \boldsymbol{u} \in V$, the following regret bounds hold

$$\begin{aligned}\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\leq \max_{1\leq t\leq T}\frac{B_\psi(\boldsymbol{u};\boldsymbol{x}_t)}{\eta_T} + \sum_{t=1}^{T}\left(\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - \frac{1}{\eta_t}B_\psi(\boldsymbol{x}_{t+1};\boldsymbol{x}_t)\right)\\ &\leq \max_{1\leq t\leq T}\frac{B_\psi(\boldsymbol{u};\boldsymbol{x}_t)}{\eta_T} + \frac{1}{2\lambda}\sum_{t=1}^{T}\eta_t\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_\star^2\,.\end{aligned}$$

Moreover, if η_t is constant, i.e., $\eta_t = \eta\ \forall t = 1, \cdots, T$, we have

$$\begin{aligned}\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\leq \frac{B_\psi(\boldsymbol{u};\boldsymbol{x}_1)}{\eta} + \sum_{t=1}^{T}\left(\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - \frac{1}{\eta}B_\psi(\boldsymbol{x}_{t+1};\boldsymbol{x}_t)\right)\\ &\leq \frac{B_\psi(\boldsymbol{u};\boldsymbol{x}_1)}{\eta} + \frac{\eta}{2\lambda}\sum_{t=1}^{T}\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_\star^2\,.\end{aligned}$$

Proof. We can use Lemma 6.9 with $\boldsymbol{g}_t \to \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t + \tilde{\boldsymbol{g}}_{t+1}$, to have

$$\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t + \tilde{\boldsymbol{g}}_{t+1}, \boldsymbol{x}_t - \boldsymbol{u}\rangle \leq \frac{1}{\eta_t}\left(B_\psi(\boldsymbol{u};\boldsymbol{x}_t) - B_\psi(\boldsymbol{u};\boldsymbol{x}_{t+1}) - B_\psi(\boldsymbol{x}_{t+1};\boldsymbol{x}_t)\right) + \langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t + \tilde{\boldsymbol{g}}_{t+1}, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle\,.$$

Summing over $t = 1, \dots, T$ the l.h.s., we obtain

$$\begin{aligned}\sum_{t=1}^{T}\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t + \tilde{\boldsymbol{g}}_{t+1}, \boldsymbol{x}_t - \boldsymbol{u}\rangle &= \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u}\rangle + \sum_{t=1}^{T}\langle \tilde{\boldsymbol{g}}_{t+1} - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{u}\rangle\\ &= \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u}\rangle + \langle \tilde{\boldsymbol{g}}_1 - \tilde{\boldsymbol{g}}_{T+1}, \boldsymbol{u}\rangle + \sum_{t=1}^{T}\langle \tilde{\boldsymbol{g}}_{t+1} - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t\rangle\,.\end{aligned}$$

Summing the r.h.s., we have that

$$\sum_{t=1}^{T}\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t + \tilde{\boldsymbol{g}}_{t+1}, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle = \sum_{t=1}^{T}\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle + \sum_{t=1}^{T}\langle \tilde{\boldsymbol{g}}_{t+1}, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle\,.$$

Finally, observe that

$$\sum_{t=1}^{T}\langle \tilde{\boldsymbol{g}}_{t+1} - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \tilde{\boldsymbol{g}}_{t+1}, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle = \sum_{t=1}^{T}(\langle \tilde{\boldsymbol{g}}_{t+1}, \boldsymbol{x}_{t+1}\rangle - \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t\rangle) = \langle \tilde{\boldsymbol{g}}_{T+1}, \boldsymbol{x}_{T+1}\rangle - \langle \tilde{\boldsymbol{g}}_1, \boldsymbol{x}_1\rangle\,.$$

Given that the regret on the rounds $t = 1, \dots, T$ does not depend on $\tilde{\boldsymbol{g}}_{T+1}$, we can safely set it to $\mathbf{0}$. □

We defer the discussion of applications of optimistic algorithms to section on Optimistic FTRL in Section 7.12.

6.10 History Bits

The Bregman divergence was introduced by Bregman [1967] as a particular example of a distance-like function satisfying certain properties, to generalize the cyclic projection algorithm to general topological vector spaces. Often people drop the condition on the strict convexity [e.g., Bauschke et al., 2003] but in reality it is part of the original definition by Bregman [1967].

Mirror Descent (MD) was introduced by Nemirovskij and Yudin [1983] in the *offline* setting. The description of MD with Bregman divergence that I described here (with minor changes) was done by Beck and Teboulle [2003]. The minor changes are in decoupling the domain X of ψ from the feasibility set V. This allows to use functions ψ that do

not satisfy the condition (6.5) but they satisfy (6.6). In the online setting, the mirror descent scheme was used for the first time by Warmuth and Jagota [1997].

Most of the online learning literature for OMD assumes ψ to be *Legendre* [see, e.g., Cesa-Bianchi and Lugosi, 2006] that corresponds to assuming (6.5) (or $\lim_{\boldsymbol{x}\to\operatorname{bdry} X} \|\nabla\psi(\boldsymbol{x})\|_2 = +\infty$, see [Rockafellar, 1970, Theorem 26.1 and Lemma 26.2]). This condition allows to prove that $\nabla\psi_V^\star = (\nabla\psi_V)^{-1}$. However, it turns out that the Legendre condition is not necessary and we only need the function ψ to be differentiable on the predictions $\boldsymbol{x}_t$. For example, we only need one of the two conditions in (6.5) or (6.6) to hold. Removing the Legendre assumption makes it easier to use OMD with different combinations of feasibility sets/Bregman divergences. So, I did not introduce the concept of Legendre functions at all, relying instead on (a minor modification of) OMD as described by Beck and Teboulle [2003]. Theorem 6.7 is derived from [Bauschke and Borwein, 1997, Theorem 3.12].

The proof of Theorem 6.11 is based on the one in Kakade et al. [2009].

The local norms were introduced in Abernethy et al. [2008] for Follow-The-Regularized-Leader with self-concordant regularizers.

The EG algorithm was introduced by Kivinen and Warmuth [1997], but not as a specific instantiation of OMD. Beck and Teboulle [2003] rediscover EG for the offline case as an example of Mirror Descent. Later, Cesa-Bianchi and Lugosi [2006] show that EG is just an instantiation of OMD. The p-norm algorithms for online prediction were originally introduced by Grove et al. [1997, 2001]. The trick to set $q = 2\ln d$ is from Gentile and Littlestone [1999], Gentile [2003] (online learning) and apparently rediscovered in Ben-Tal et al. [2001] (optimization). The learning with experts setting was introduced by Littlestone and Warmuth [1994] and Vovk [1990]. The ideas in Algorithm 6.3 are based on the Multiplicative Weights algorithm [Littlestone and Warmuth, 1994] and the Hedge algorithm [Freund and Schapire, 1995, 1997]. As a side note, the weighted majority algorithm was also discovered independently in the game theory literature by Fudenberg and Levine [1995]. For two experts with losses in $[0,1]$, Cover [1965] showed that the minimax regret is $\sqrt{\frac{T}{2\pi}}$ and proposed an algorithm achieving it. Notably, the approach in Cover [1965] is based on online betting. On the other hand, for more than 2 experts and losses in $[0,1]$, the minimax regret is $(1+o(1))\sqrt{\frac{T\ln d}{2}}$, where $o(1)\to 0$ when $d, T\to\infty$ [Cesa-Bianchi et al., 1993, 1997]. By now, the literature on learning with expert is huge, with tons of variations over algorithms and settings.

The idea of "hallucinating" future losses in online learning is originally from Azoury and Warmuth [2001] in the Forward Algorithm. Apparently, this idea was forgotten and rediscovered by Chiang et al. [2012] that used the previous loss function as an estimate of the next one, showing smaller regret in the case that the losses have small temporal variation. Later, Rakhlin and Sridharan [2013b] generalized this idea in the Optimistic OMD algorithm. Surprisingly enough, the procedure using two Optimistic OGD algorithms to solve saddle-point problems was already proposed by Popov [1980], see also Section 11.7. Optimistic OMD was proposed in Rakhlin and Sridharan [2013b] with a two-step update. It was then simplified to the one-step updates I presented here by Joulani et al. [2017]. The proof I present here is based on the one I proposed for Optimistic FTRL in Section 7.12.

6.11 Exercises

Problem 6.1. *Derive a closed form update for OMD when using the ψ of Example 6.5 and $V = X$.*

Problem 6.2. *Prove the three-points equality for Bregman divergences in Lemma 6.6.*

Problem 6.3. *Let $\boldsymbol{A}\in\mathbb{R}^{d\times d}$ a positive definite matrix. Define $\|\boldsymbol{x}\|_{\boldsymbol{A}}^2 = \boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x}$. Prove that $\frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_{\boldsymbol{A}}^2$ is the Bregman divergence $B_\psi(\boldsymbol{x};\boldsymbol{y})$ associated with $\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_{\boldsymbol{A}}^2$.*

Problem 6.4. *Find the conjugate function of $\psi(\boldsymbol{x}) = \sum_{i=1}^d x_i\ln x_i$ defined over $X = \{\boldsymbol{x}\in\mathbb{R}^d : x_i\geq 0, \|\boldsymbol{x}\|_1 = 1\}$.*

Problem 6.5. *We saw the Fenchel-Young inequality: $\langle\boldsymbol{\theta},\boldsymbol{x}\rangle \leq f(\boldsymbol{x}) + f^\star(\boldsymbol{\theta})$. Now, we want to show an equality, quantifying the gap in the inequality with a Bregman divergence term. Assume that f and $f^\star$ are differentiable, f strictly convex, and $\operatorname{dom} f = \mathbb{R}^d$. Prove that*

$$f(\boldsymbol{x}) + f^\star(\boldsymbol{\theta}) = \langle\boldsymbol{\theta},\boldsymbol{x}\rangle + B_f\left(\boldsymbol{x};\nabla f^\star(\boldsymbol{\theta})\right) .$$

Problem 6.6. *In proof of Online Mirror Descent, we have the terms*

$$-B_\psi(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) + \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle \, .$$

Prove that they can be lower bounded by $B_\psi(\boldsymbol{x}_t; \boldsymbol{x}_{t+1})$.

Problem 6.7. *Generalize the concept of strong convexity to Bregman functions, instead of norms, and prove a logarithmic regret guarantee for such functions using OMD.*

Problem 6.8. *Derive the EG update rule and regret bound in the case that the algorithm starts from an arbitrary vector $\boldsymbol{x}_1$ in the probability simplex.*

Problem 6.9. *Consider the regret upper bound of EG in Section 6.6. Show that the terms $\sum_{i=1}^d g_{t,i}^2 x_{t,i}$ for $t = 1, \dots, T$ can be tightened to $\sum_{i=1}^d (g_{t,i} - m_t)^2 x_{t,i}$ for any $m_t \in \mathbb{R}$.*

Problem 6.10. *Extend Theorem 5.1 to arbitrary norms, measuring the diameter with respect to a norm $\|\cdot\|$ and considering losses L-Lipschitz with respect to the dual norm $\|\cdot\|_\star$.*

Problem 6.11. *In this problem, we will tackle Online Non-Convex Optimization. Assume that $V \subset \mathbb{R}^d$ is the feasible set and it is convex and bounded. The losses $\ell_t : \mathbb{R}^d \to [0, 1]$ are non-convex and 1-Lipschitz w.r.t. $\|\cdot\|_2$. Prove that there exists a randomized algorithm that achieves sublinear regret on this problem, assuming knowledge of the total number of rounds T. Hint: Aim for something like $\mathbb{E}[\mathrm{Regret}_T(\boldsymbol{u})] = O(\sqrt{dT \ln T})$ and do not worry about efficiency of the algorithm.*

Chapter 7

Follow-The-Regularized-Leader

Till now, we focused only on Online Subgradient Descent and its generalization, Online Mirror Descent, with a brief ad-hoc analysis of a Follow-The-Leader (FTL) analysis in the first chapter. In this chapter, we will extend FTL to a powerful and generic algorithm to do online convex optimization: **Follow-the-Regularized-Leader** (FTRL).

FTRL is a very intuitive algorithm: At each time step it will play the minimizer of the sum of the past losses *plus* a time-varying regularization. We will see that the regularization is needed to make the algorithm "more stable" with linear losses and avoid the jumping back and forth that we saw in Example 2.10.

7.1 The Follow-the-Regularized-Leader Algorithm

Algorithm 7.1 Follow-the-Regularized-Leader Algorithm

Require: A sequence of regularizers $\psi_1, \dots, \psi_T : X \to \mathbb{R}$, closed non-empty set $V \subseteq X \subseteq \mathbb{R}^d$

1: **for** $t = 1$ **to** T **do**
2: Output $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x} \in V} \ \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \ell_i(\boldsymbol{x})$
3: Receive $\ell_t : V \to \mathbb{R}$ and pay $\ell_t(\boldsymbol{x}_t)$
4: **end for**

As said above, in FTRL we output the minimizer of the regularized cumulative past losses. It should be clear that FTRL is not an algorithm, but rather a family of algorithms, in the same way as OMD is a family of algorithms.

Before analyzing the algorithm, let's get some intuition on it. In OMD, we saw that the "state" of the algorithm is stored in the current iterate $\boldsymbol{x}_t$, in the sense that the next iterate $\boldsymbol{x}_{t+1}$ depends on $\boldsymbol{x}_t$ and the loss received at time t, plus obviously the choice of the learning rate. Instead in FTRL, the next iterate $\boldsymbol{x}_{t+1}$ depends on the entire history of losses received up to time t and on the regularizer used at time t. This has an immediate consequence: In the case that V is bounded, OMD will only "remember" the last $\boldsymbol{x}_t$, and not the iterate before the projection. On the other hand, FTRL keeps in memory the entire history of the past, that in principle allows to recover the iterates before the projection in V.

This difference in behavior might make the reader think that FTRL is more computationally and memory expensive. And indeed it is! But, we will also see that there is a way to consider approximate losses that makes the algorithm as expensive as OMD, yet retaining the same or more information than OMD.

For FTRL, we prove an equality for the regret. This equality factors the regret in three terms that have precise meanings and can be easily upper bounded with some familiar quantities.

Lemma 7.1. *Let $V \subseteq \mathbb{R}^d$ be closed and non-empty. Denote by $F_t(\boldsymbol{x}) = \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \ell_i(\boldsymbol{x})$. Assume that $\operatorname{argmin}_{\boldsymbol{x} \in V} \ F_t(\boldsymbol{x})$ is not empty and set $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x} \in V} \ F_t(\boldsymbol{x})$. Then, for any $\boldsymbol{u} \in \mathbb{R}^d$, we have*

$$\sum_{t=1}^{T} (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) = \psi_{T+1}(\boldsymbol{u}) - \min_{\boldsymbol{x} \in V} \ \psi_1(\boldsymbol{x}) + \sum_{t=1}^{T} [F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t)] + F_{T+1}(\boldsymbol{x}_{T+1}) - F_{T+1}(\boldsymbol{u}) \ .$$

Proof. Given that the terms $\ell_t(\boldsymbol{x}_t)$ are appearing at both side of the equality, we just have to verify that

$$-\sum_{t=1}^{T}\ell_t(\boldsymbol{u}) = \psi_{T+1}(\boldsymbol{u}) - \min_{\boldsymbol{x}\in V}\ \psi_1(\boldsymbol{x}) + \sum_{t=1}^{T}[F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1})] + F_{T+1}(\boldsymbol{x}_{T+1}) - F_{T+1}(\boldsymbol{u})\ .$$

Remembering that $F_1(\boldsymbol{x}_1) = \min_{\boldsymbol{x}\in V}\ \psi_1(\boldsymbol{x})$ and using the fact that the sum with F_t is telescopic, we have

$$-\sum_{t=1}^{T}\ell_t(\boldsymbol{u}) = \psi_{T+1}(\boldsymbol{u}) - F_1(\boldsymbol{x}_1) + F_1(\boldsymbol{x}_1) - F_{T+1}(\boldsymbol{x}_{T+1}) + F_{T+1}(\boldsymbol{x}_{T+1}) - F_{T+1}(\boldsymbol{u}) = \psi_{T+1}(\boldsymbol{u}) - F_{T+1}(\boldsymbol{u}),$$

that is true by the definition of F_{T+1}. □

Remark 7.2. *We basically did not assume anything on ℓ_t nor on ψ_t, hence the above equality holds even for non-convex losses and regularizers. Yet, solving the minimization problem at each step might be computationally infeasible.*

Remark 7.3. *Note that the left hand side of the equality in the theorem does not depend on ψ_{T+1}. So, we will often set ψ_{T+1} equal to ψ_T.*

Remark 7.4. *The FTRL algorithm is invariant to any constant added to the regularizers, hence we can always state the regret guarantee with $\psi_t(\boldsymbol{u}) - \min_{\boldsymbol{x}}\psi_t(\boldsymbol{x})$ instead of $\psi_t(\boldsymbol{u})$. However, sometimes for clarity we will instead explicitly choose the regularizers such that their minimum is 0.*

Remark 7.5. *The terms $\ell_t(\boldsymbol{x}_t)$ appear on the left and right of the equality. Hence, in some proofs we can substitute them with some other terms, for example $\ell_t(\tilde{\boldsymbol{x}}_t)$ where $\tilde{\boldsymbol{x}}_t$ is* not *the FTRL prediction. In other words, we can write the same equality for the prediction of a generic algorithm whose regret will depend on the prediction of the FTRL algorithm.*

However, while surprising, the above equality is not yet a regret bound, because it is "implicit". In fact, the losses are appearing on both sides of the equality.

Let's take a closer look at the equality. If $\boldsymbol{u} \in V$, we have that the sum of the last two terms on the r.h.s. is negative. On the other hand, the first two terms on the r.h.s. are similar to what we got in OMD. The interesting part is the sum of the terms $F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t)$. To give an intuition of what is going on, let's consider that case that the regularizer is constant over time, i.e., $\psi_t = \psi$. Hence, the terms in the sum can be rewritten as

$$F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) = \psi(\boldsymbol{x}_t) + \sum_{i=1}^{t}\ell_i(\boldsymbol{x}_t) - \left(\psi(\boldsymbol{x}_{t+1}) + \sum_{i=1}^{t}\ell_i(\boldsymbol{x}_{t+1})\right)\ .$$

Hence, we are measuring the distance between the minimizer of the regularized losses (with two different regularizers) in two consecutive predictions of the algorithms. Roughly speaking, this term will be small if $\boldsymbol{x}_t \approx \boldsymbol{x}_{t+1}$ and the losses+regularization are "nice". This should remind you exactly the OMD update, where we *constrain* $\boldsymbol{x}_{t+1}$ to be close to $\boldsymbol{x}_{t+1}$. Instead, here the two predictions will be close one to the other if the minimizer of the regularized losses up to time t is close to the minimizer of the losses up to time $t+1$. So, like in OMD, the regularizer here will play the critical role of *stabilizing* the predictions, if the losses do not possess enough curvature.

In the following, we will see different ways to get an explicit upper bound from Lemma 7.1.

7.2 FTRL Regret Bound using Strong Convexity

An easy case to get a regret upper bound for FTRL is when the losses plus regularizer are strongly convex. In fact, the strong convexity guarantees the predictions to be stable. To quantify this intuition, we need a property of strongly convex functions.

7.2.1 Convex Analysis Bits: Properties of Strongly Convex Functions

We will use the following lemma for strongly convex functions.

Lemma 7.6. *Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ be proper and μ-strongly convex with respect to a norm $\|\cdot\|$ over a convex set $V \subseteq \operatorname{dom} \partial f$, where $\mu > 0$. Then, for all $\boldsymbol{x}, \boldsymbol{y} \in V$, $\boldsymbol{g} \in \partial f(\boldsymbol{y})$, and $\boldsymbol{g}' \in \partial f(\boldsymbol{x})$, we have*

$$f(\boldsymbol{x}) - f(\boldsymbol{y}) \leq \langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{y} \rangle + \frac{1}{2\mu} \|\boldsymbol{g} - \boldsymbol{g}'\|_\star^2 .$$

Proof. Define $\phi(\boldsymbol{z}) = f(\boldsymbol{z}) - \langle \boldsymbol{g}, \boldsymbol{z} \rangle$. Observe that $\mathbf{0} \in \partial\phi(\boldsymbol{y})$, hence $\boldsymbol{y}$ is a minimizer of $\phi(\boldsymbol{z})$ by Theorem 6.12. Also, note that $\boldsymbol{g}' - \boldsymbol{g} \in \partial\phi(\boldsymbol{x})$. Hence, by Lemma 4.2, we can write

$$\begin{aligned}
\phi(\boldsymbol{y}) &= \min_{\boldsymbol{z} \in \operatorname{dom} \phi} \phi(\boldsymbol{z}) \\
&\geq \min_{\boldsymbol{z} \in \operatorname{dom} \phi} \left(\phi(\boldsymbol{x}) + \langle \boldsymbol{g}' - \boldsymbol{g}, \boldsymbol{z} - \boldsymbol{x} \rangle + \frac{\mu}{2} \|\boldsymbol{z} - \boldsymbol{x}\|^2 \right) \\
&\geq \min_{\boldsymbol{z} \in \mathbb{R}^d} \left(\phi(\boldsymbol{x}) + \langle \boldsymbol{g}' - \boldsymbol{g}, \boldsymbol{z} - \boldsymbol{x} \rangle + \frac{\mu}{2} \|\boldsymbol{z} - \boldsymbol{x}\|^2 \right) \\
&= \phi(\boldsymbol{x}) - \frac{1}{2\mu} \|\boldsymbol{g}' - \boldsymbol{g}\|_\star^2,
\end{aligned}$$

where the last step comes from the conjugate function of squared norms, see Example 5.11. □

Corollary 7.7. *Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ closed, proper, subdifferentiable, and μ-strongly convex with respect to a norm $\|\cdot\|$ over its domain. Let $\boldsymbol{x}^\star = \operatorname{argmin}_{\boldsymbol{x}} f(\boldsymbol{x})$. Then, for all $\boldsymbol{x} \in \operatorname{dom} \partial f$, and $\boldsymbol{g} \in \partial f(\boldsymbol{x})$, we have*

$$f(\boldsymbol{x}) - f(\boldsymbol{x}^\star) \leq \frac{1}{2\mu} \|\boldsymbol{g}\|_\star^2 .$$

In words, the above lemma says that an upper bound to the suboptimality gap is proportional to the squared norm of the subgradient.

7.2.2 An Explicit Regret Bound

We now state a Lemma quantifying the intuition on the "stability" of the predictions.

Lemma 7.8. *With the notation in Algorithm 7.1, denote by $F_t(\boldsymbol{x}) = \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \ell_i(\boldsymbol{x})$. Assume $V \subseteq X$ and V convex. If F_t is closed, subdifferentiable, and strongly convex in V, then $\boldsymbol{x}_t$ exists and is unique. In addition, assume $\partial\ell_t(\boldsymbol{x}_t)$ to be non-empty and $F_t + \ell_t$ to be closed, subdifferentiable, and λ_t-strongly convex w.r.t. $\|\cdot\|$ in V. Then, we have*

$$F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) \leq \frac{\|\boldsymbol{g}_t\|_\star^2}{2\lambda_t} + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}),\ \forall \boldsymbol{g}_t \in \partial\ell_t(\boldsymbol{x}_t) .$$

Proof. The existence and unicity is given by Theorem 6.8. Then, we have

$$\begin{aligned}
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) &= (F_t(\boldsymbol{x}_t) + \ell_t(\boldsymbol{x}_t)) - (F_t(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_{t+1})) + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}) \\
&\leq (F_t(\boldsymbol{x}_t) + \ell_t(\boldsymbol{x}_t)) - (F_t(\boldsymbol{x}_t^\star) + \ell_t(\boldsymbol{x}_t^\star)) + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}) \\
&\leq \frac{\|\boldsymbol{g}_t\|_\star^2}{2\lambda_t} + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}),
\end{aligned}$$

where in the second inequality we used Corollary 7.7, the definition $\boldsymbol{x}_t^\star := \operatorname{argmin}_{\boldsymbol{x} \in V} F_t(\boldsymbol{x}) + \ell_t(\boldsymbol{x})$, and $\boldsymbol{g}_t \in \partial(F_t + \ell_t + i_V)(\boldsymbol{x}_t)$. Observing that $\boldsymbol{x}_t = \operatorname{argmin}_{\boldsymbol{x} \in V} F_t(\boldsymbol{x})$, we have $\mathbf{0} \in \partial(F_t + i_V)(\boldsymbol{x}_t)$ by Theorem 6.12. Hence, using Theorem 2.18, we have $\partial\ell_t(\boldsymbol{x}_t) \subseteq \partial(F_t + \ell_t + i_V)(\boldsymbol{x}_t)$. □

Let's see some immediate applications of FTRL.

Corollary 7.9. *With the notation in Algorithm 7.1, let V be convex and let $\psi : V \to \mathbb{R}$ be a closed and μ-strongly convex function w.r.t. $\|\cdot\|$. Set the sequence of regularizers as $\psi_t(\boldsymbol{x}) = \frac{1}{\eta_{t-1}}(\psi(\boldsymbol{x}) - \min_{\boldsymbol{z}} \psi(\boldsymbol{z}))$, where $\eta_{t+1} \leq \eta_t,\ t = 1, \dots, T$. Assume ℓ_t convex, closed, and $\partial \ell_t(\boldsymbol{x}_t)$ not empty. Then, FTRL guarantees*

$$\sum_{t=1}^{T} \ell(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \leq \frac{\psi(\boldsymbol{u}) - \min_{\boldsymbol{x} \in V} \psi(\boldsymbol{x})}{\eta_{T-1}} + \frac{1}{2\mu} \sum_{t=1}^{T} \eta_{t-1} \|\boldsymbol{g}_t\|_\star^2,$$

for all $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$. Moreover, if the functions ℓ_t are L-Lipschitz, setting $\eta_{t-1} = \frac{\alpha\sqrt{\mu}}{L\sqrt{t}}$ we get

$$\sum_{t=1}^{T} \ell(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \leq \left(\frac{\psi(\boldsymbol{u}) - \min_{\boldsymbol{x}} \psi(\boldsymbol{x})}{\alpha} + \alpha \right) \frac{L\sqrt{T}}{\sqrt{\mu}} .$$

Proof. The corollary is immediate from Lemma 7.1, Lemma 7.8, and the observation that from the assumptions we have $\psi_t(\boldsymbol{x}) - \psi_{t+1}(\boldsymbol{x}) \leq 0,\ \forall \boldsymbol{x}$. We also set $\psi_{T+1} = \psi_T$, thanks to Remark 7.3. □

This might look like the same regret guarantee of OMD, however here there is a very important difference: The last term contains a time-varying element (η_t) but the domain does not have to be bounded! Also, I used the regularizer $\frac{1}{\eta_{t-1}}\psi(\boldsymbol{x})$ and not $\frac{1}{\eta_t}\psi(\boldsymbol{x})$ to remind you another important difference: In OMD the learning rate η_t is chosen after receiving the subgradient $\boldsymbol{g}_t$ while here you have to choose it before receiving it.

7.3 FTRL with Linearized Losses

An important difference between OMD and FTRL is that here the update rule seems way more expensive than in OMD, because we need to solve an optimization problem at each step. However, it turns out we can use FTRL on *linearized losses* and obtain the same bound with the same computational complexity of OMD.

Consider the case in which the losses are linear, i.e., $\ell_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle,\ t = 1, \dots, T$, we have that the prediction of FTRL is

$$\boldsymbol{x}_{t+1} \in \operatorname*{argmin}_{\boldsymbol{x} \in V}\ \psi_{t+1}(\boldsymbol{x}) + \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x} \rangle = \operatorname*{argmax}_{\boldsymbol{x} \in V} \left\langle -\sum_{i=1}^{t} \boldsymbol{g}_i, \boldsymbol{x} \right\rangle - \psi_{t+1}(\boldsymbol{x}) .$$

Denote by $\psi_{V,t}(\boldsymbol{x}) = \psi_t(\boldsymbol{x}) + i_V(\boldsymbol{x})$. Now, if we assume $\psi_{V,t}$ to be proper, convex, and closed, using the Theorem 5.7, we have that $\boldsymbol{x}_{t+1} \in \partial \psi^\star_{V,t+1}(-\sum_{i=1}^{t} \boldsymbol{g}_i)$. Moreover, if $\psi_{V,t+1}$ is also strongly convex, by Theorem 6.11 we know that $\psi^\star_{V,t+1}$ is differentiable and we get

$$\boldsymbol{x}_{t+1} = \nabla \psi^\star_{V,t+1} \left(-\sum_{i=1}^{t} \boldsymbol{g}_i \right) . \tag{7.1}$$

In turn, this update can be written in the following way

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \boldsymbol{g}_t, \\ \boldsymbol{x}_{t+1} &= \nabla \psi^\star_{V,t+1}(\boldsymbol{\theta}_{t+1}) . \end{aligned}$$

This corresponds to Figure 7.1.

Compare it to the mirror update of OMD, rewritten in a similar way:

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \nabla \psi(\boldsymbol{x}_t) - \eta_t \boldsymbol{g}_t, \\ \boldsymbol{x}_{t+1} &= \nabla \psi^\star_V(\boldsymbol{\theta}_{t+1}) . \end{aligned}$$

They are very similar, but with important differences:

- In OMD, the state is kept in $\boldsymbol{x}_t$, so we need to transform it into a dual variable before making the update and then back to the primal variable.

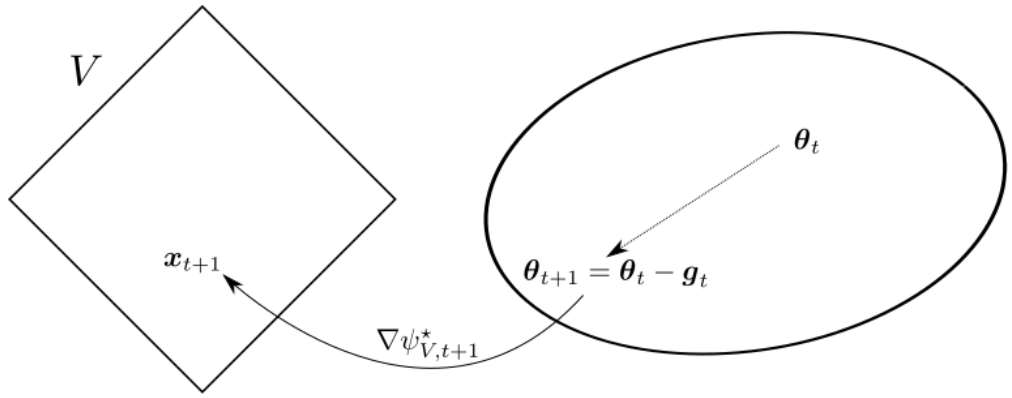

Figure 7.1: Dual mapping for FTRL with linear losses.

- In FTRL with linear losses, the state is kept directly in the dual space, updated and then transformed in the primal variable. The primal variable is only used to predict, but not directly in the update.
- In OMD, the samples are weighted by the learning rates that is typically decreasing
- In FTRL with linear losses, all the subgradients have the same weight, but the regularizer is typically increasing over time.

Also, we will not lose anything in the worst-case regret bound! Indeed, we can run FTRL on the linearized losses $\tilde{\ell}_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle$, where $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$, guaranteeing exactly the same worst-case upper bound on the regret with the losses ℓ_t. The algorithm for such procedure is in Algorithm 7.2.

Algorithm 7.2 Follow-the-Regularized-Leader Algorithm on Linearized Losses

Require: A sequence of regularizers $\psi_1, \dots, \psi_T : X \to \mathbb{R}$, closed non-empty convex set $V \subseteq X \subseteq \mathbb{R}^d$

for $t = 1$ **to** T **do**
  Output $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x} \in V} \ \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \langle \boldsymbol{g}_i, \boldsymbol{x} \rangle$
  Receive $\ell_t : V \to \mathbb{R}$ subdifferentiable in V and pay $\ell_t(\boldsymbol{x}_t)$
  Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
end for

In fact, using the definition of the subgradients and the assumptions of Corollary 7.9, we have

$$\operatorname{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \sum_{t=1}^T (\tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_t(\boldsymbol{u})) \leq \frac{\psi(\boldsymbol{u}) - \min_{\boldsymbol{x} \in V} \psi(\boldsymbol{x})}{\eta_{T-1}} + \frac{1}{2\mu} \sum_{t=1}^T \eta_{t-1} \|\boldsymbol{g}_t\|_\star^2, \ \forall \boldsymbol{u} \in V \, .$$

The only difference with respect to Corollary 7.9 is that here the $\boldsymbol{g}_t$ are the specific ones we use in the algorithm, while in Corollary 7.9 the statement holds for any choice of the $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$. Moreover, FTRL with exact losses can adapt to any "nice" properties of the losses, for example strong convexity, while this information is lost in the linearized case. Moreover, we have to remember that this is just a worst-case guarantee: On real problems FTRL with full losses typically performs better than both FTRL with linearized losses and OMD.

More generally, we can also specialize the FTRL regret equality to this case, obtaining the following Lemma, that is usually considered a "dual analysis" of FTRL for linear losses.

Lemma 7.10. *Under the assumptions of Lemma 7.1, further assume for all t that $\ell_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle$. Then, for any*

$\boldsymbol{u} \in \mathbb{R}^d$, *we have*

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle = \psi_{T+1}(\boldsymbol{u}) - \min_{\boldsymbol{x} \in V} \psi_1(\boldsymbol{x}) + \sum_{t=1}^{T} \left[\psi_{V,t+1}^\star \left(-\sum_{i=1}^{t} \boldsymbol{g}_i \right) - \psi_{V,t}^\star \left(-\sum_{i=1}^{t-1} \boldsymbol{g}_i \right) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \right] + F_{T+1}(\boldsymbol{x}_{T+1}) - F_{T+1}(\boldsymbol{u}),$$

where $\psi_{V,t}^\star$ *is the Fenchel conjugate of* $\psi_{V,t} = \psi_t + i_V$.

Proof. The proof is immediate from Lemma 7.1 and observing that

$$F_t(\boldsymbol{x}_t) = \min_{\boldsymbol{x} \in V} \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \ell_i(\boldsymbol{x}) = -\max_{\boldsymbol{x} \in V} \left\langle -\sum_{i=1}^{t-1} \boldsymbol{g}_i, \boldsymbol{x} \right\rangle - \psi_t(\boldsymbol{x}) = -\psi_{V,t}^\star \left(-\sum_{i=1}^{t-1} \boldsymbol{g}_i \right) . \qquad \square$$

In the next example, we can see the different behavior of FTRL and OMD.

Example 7.11. *Consider* $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 \le 1\}$. *With Online Subgradient Descent (OSD) with learning rate* $\eta_t = \frac{1}{\sqrt{t}}$ *and* $\boldsymbol{x}_1 = \boldsymbol{0}$*, the update is*

$$\tilde{\boldsymbol{x}}_{t+1} = \boldsymbol{x}_t - \frac{1}{\sqrt{t}} \boldsymbol{g}_t,$$
$$\boldsymbol{x}_{t+1} = \tilde{\boldsymbol{x}}_{t+1} \min \left(\frac{1}{\|\tilde{\boldsymbol{x}}_{t+1}\|_2}, 1 \right) .$$

On the other hand in FTRL with linearized losses, we can use $\psi_t(\boldsymbol{x}) = \frac{\sqrt{t}}{2} \|\boldsymbol{x}\|_2^2$ *and it is easy to verify that the update in* (7.1) *becomes*

$$\tilde{\boldsymbol{x}}_{t+1} = \frac{-\sum_{i=1}^{t} \boldsymbol{g}_i}{\sqrt{t}},$$
$$\boldsymbol{x}_{t+1} = \tilde{\boldsymbol{x}}_{t+1} \min \left(\frac{1}{\|\tilde{\boldsymbol{x}}_{t+1}\|_2}, 1 \right) .$$

While the regret guarantee would be the same for these two updates, from an intuitive point of view OMD seems to be losing a lot of potential information due to the fact that we only memorize the projected iterate.

7.3.1 FTRL with Linearized Losses Can Be Equivalent to OMD

Even if FTRL and OMD seem very different, in certain cases they are actually equivalent. Let's consider an example.

Let $\psi : X \to \mathbb{R}$ and consider that case that $V = X = \operatorname{dom} \psi$. The output of OMD is

$$\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x}} \langle \eta \boldsymbol{g}_t, \boldsymbol{x} \rangle + B_\psi(\boldsymbol{x}; \boldsymbol{x}_t) .$$

Assume that $\boldsymbol{x}_{t+1} \in \operatorname{int} \operatorname{dom} \psi$ for all $t = 1, \dots, T$. This implies that $\eta \boldsymbol{g}_t + \nabla \psi(\boldsymbol{x}_{t+1}) - \nabla \psi(\boldsymbol{x}_t) = \boldsymbol{0}$, that is $\nabla \psi(\boldsymbol{x}_{t+1}) = \nabla \psi(\boldsymbol{x}_t) - \eta \boldsymbol{g}_t$. Assuming $\boldsymbol{x}_1 = \min_{\boldsymbol{x} \in V} \psi(\boldsymbol{x})$, we have

$$\nabla \psi(\boldsymbol{x}_{t+1}) = -\eta \sum_{i=1}^{t} \boldsymbol{g}_i .$$

On the other hand, consider FTRL with linearized losses with regularizers $\psi_t = \frac{1}{\eta} \psi$, then

$$\boldsymbol{x}_{t+1} = \operatorname*{argmin}_{\boldsymbol{x}} \frac{1}{\eta} \psi(\boldsymbol{x}) + \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x} \rangle = \operatorname*{argmin}_{\boldsymbol{x}} \psi(\boldsymbol{x}) + \eta \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x} \rangle .$$

Assuming that $\boldsymbol{x}_{t+1} \in \operatorname{int} \operatorname{dom} \psi$, this implies that $\nabla \psi(\boldsymbol{x}_{t+1}) = -\eta \sum_{i=1}^{t} \boldsymbol{g}_i$. Further, assuming that $\nabla \psi$ is invertible, implies that the predictions of FTRL and OMD are the same.

This equivalence immediately gives us some intuition on the role of ψ in both algorithm: The same function is inducing the Bregman divergence, that is our similarity measure, and is the regularizer in FTRL. Moreover, the inverse of the growth rate of the regularizers in FTRL takes the role of the learning rate in OMD.

Example 7.12. *Consider $\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$ and $V = \mathbb{R}^d$, then it satisfies the conditions above to have the predictions of OMD equal to the ones of FTRL.*

Remark 7.13. *Based on the above observation and on the observations in Section 7.3, a common misunderstanding even among experts is that FTRL and OMD differs* only *in the constrained setting. This is clearly false: The regularizer of FTRL can vary over time in arbitrary ways, not just as a scaled version of a fixed regularizer. Indeed, general time-varying regularizers are used, for example, in the Online Newton Step algorithm (Section 7.10) and in Parameter-Free algorithms (Chapter 9).*

7.4 FTRL Regret Bound using Local Norms

In Lemma 7.8, strong convexity basically tells us that the losses plus regularizer have some minimum curvature in all the directions. However, as in the OMD case, it turns out that we can a regret upper bound using again the notion of local norms.

Lemma 7.14. *Under the same assumptions of Lemma 7.1, assume $\psi_1, \dots, \psi_T$ twice differentiable and with the Hessian positive definite in the interior of their domains, and $\psi_{t+1}(\boldsymbol{x}) \geq \psi_t(\boldsymbol{x}), \forall \boldsymbol{x} \in V, t = 1, \dots, T$. Also, assume $\ell_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle,\ t = 1, \dots, T$, for arbitrary vectors $\boldsymbol{g}_t$. Define $\|\boldsymbol{x}\|_{\boldsymbol{A}} := \sqrt{\boldsymbol{x}^\top \boldsymbol{A} \boldsymbol{x}}$. For $t = 1, \dots, T$, assume that $\boldsymbol{x}_t$ is in the interior of the domain of ψ_t and that $\tilde{\boldsymbol{x}}_{t+1} := \operatorname{argmin}_{\boldsymbol{x} \in \mathbb{R}^d} \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + B_{\psi_t}(\boldsymbol{x}; \boldsymbol{x}_t)$ exists. Then, there exists $\boldsymbol{z}_t$ on the line segments between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$ and $\boldsymbol{z}'_t$ on the line segments between $\boldsymbol{x}_t$ and $\tilde{\boldsymbol{x}}_{t+1}$, such that the following inequality holds for any $\boldsymbol{u} \in V$*

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq \psi_{T+1}(\boldsymbol{u}) - \min_{\boldsymbol{x} \in V} \psi_1(\boldsymbol{x}) + \frac{1}{2} \sum_{t=1}^{T} \min \left(\|\boldsymbol{g}_t\|^2_{(\nabla^2 \psi_t(\boldsymbol{z}_t))^{-1}}, \|\boldsymbol{g}_t\|^2_{(\nabla^2 \psi_t(\boldsymbol{z}'_t))^{-1}} \right) .$$

Proof. First of all, observe that ψ_t are strictly convex because the Hessians are positive definite. Hence, they can be used to define Bregman divergences.

From the optimality condition of $\boldsymbol{x}_t$, we have

$$\langle \nabla F_t(\boldsymbol{x}_t), \boldsymbol{v} - \boldsymbol{x}_t \rangle \geq 0,\ \forall \boldsymbol{v} \in V .$$

Hence, in particular we have

$$\langle \nabla F_t(\boldsymbol{x}_t), \boldsymbol{x}_{t+1} - \boldsymbol{x}_t \rangle \geq 0 .$$

Using this inequality, we have

$$B_{F_t}(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) = F_t(\boldsymbol{x}_{t+1}) - F_t(\boldsymbol{x}_t) - \langle \nabla F_t(\boldsymbol{x}_t), \boldsymbol{x}_{t+1} - \boldsymbol{x}_t \rangle \leq F_t(\boldsymbol{x}_{t+1}) - F_t(\boldsymbol{x}_t) .$$

This last inequality implies that

$$\begin{aligned}
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) &= F_t(\boldsymbol{x}_t) - F_t(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{x}_{t+1}) + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}) \\
&\leq F_t(\boldsymbol{x}_t) - F_t(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{x}_{t+1}) \\
&= F_t(\boldsymbol{x}_t) - F_t(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle \\
&\leq \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - B_{F_t}(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) .
\end{aligned}$$

Bregman divergences are independent of linear terms, so $B_{F_t}(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) = B_{\psi_t}(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t)$. From the Taylor's theorem, we have said that $B_{\psi_t}(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) = \frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t)^\top \nabla^2 \psi_t(\boldsymbol{z}_t)(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t)$, where $\boldsymbol{z}_t$ is on the line segment between

$\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$. Observe that this is $\frac{1}{2}\|\boldsymbol{x}_{t+1} - \boldsymbol{x}_t\|^2_{\nabla^2\psi_t(\boldsymbol{z}_t)}$ and it is indeed a norm because we assumed the Hessian of ψ_t to be positive definite. Hence, by Fenchel-Young inequality and Examples 4.18 and 5.11, we have

$$\begin{aligned} F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) &\leq \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - B_{\psi_t}(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) \\ &\leq \frac{1}{2}\|\boldsymbol{g}_t\|^2_{(\nabla^2\psi_t(\boldsymbol{z}_t))^{-1}} + \frac{1}{2}(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t)^\top \nabla^2\psi_t(\boldsymbol{z}_t)(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t) - B_{\psi_t}(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) \\ &= \frac{1}{2}\|\boldsymbol{g}_t\|^2_{(\nabla^2\psi_t(\boldsymbol{z}_t))^{-1}}, \end{aligned} \tag{7.2}$$

that gives the first term in the minimum.

For the second term in the minimum, we start from (7.2) to get

$$\begin{aligned} F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) &\leq \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - B_{\psi_t}(\boldsymbol{x}_{t+1}; \boldsymbol{x}_t) \leq \max_{\boldsymbol{x}\in\mathbb{R}^d} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{x}\rangle - B_{\psi_t}(\boldsymbol{x}; \boldsymbol{x}_t) \\ &= \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \tilde{\boldsymbol{x}}_{t+1}\rangle - B_{\psi_t}(\tilde{\boldsymbol{x}}_{t+1}; \boldsymbol{x}_t)\ . \end{aligned}$$

Then, we proceed as in the first bound. □

Observe as $\boldsymbol{z}'_t$ is defined as a sort of OMD update using ψ_t as distance generating function. Also, differently from the local norm bound for OMD, $\boldsymbol{z}'_t$ does not appear in the FTRL algorithm in any way.

7.5 Example of FTRL: Exponentiated Gradient without Knowing T

As we did in Section 6.6 for OMD, let's see an example of an instantiation of FTRL with linearized losses to have the FTRL version of Exponentiated Gradient.

Let $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_1 = 1, x_i \geq 0\}$ and the sequence of loss functions $\ell_t : V \to \mathbb{R}$ be convex and L_∞-Lipschitz w.r.t. the L_∞ norm. Let $\psi : R^d_{\geq 0} \to \mathbb{R}^d$ be defined as $\psi(\boldsymbol{x}) = \sum_{i=1}^d x_i \ln x_i$, where we define $0 \ln 0 = 0$. Set $\psi_t(\boldsymbol{x}) = \alpha L_\infty \sqrt{t}\psi(\boldsymbol{x})$, that is $\alpha L_\infty \sqrt{t}$-strongly convex w.r.t. the L_1 norm by Lemma 6.17, where $\alpha > 0$ is a parameter of the algorithm.

Given that the regularizers are strongly convex and defining $\psi_{V,t} = \psi_t + i_V$, from (7.1) we have

$$\boldsymbol{x}_t = \nabla\psi^\star_{V,t}\left(-\sum_{i=1}^{t-1} \boldsymbol{g}_i\right)\ .$$

We already saw in Section 6.6 that $\psi^\star_V(\boldsymbol{\theta}) = \ln\left(\sum_{i=1}^d \exp(\theta_i)\right)$, that implies that $\psi^\star_{V,t}(\boldsymbol{\theta}) = \alpha L_\infty \sqrt{t} \ln\left(\sum_{i=1}^d \exp\left(\frac{\theta_i}{\alpha L_\infty \sqrt{t}}\right)\right)$. So, running FTRL with linearized losses, we have that

$$x_{t,j} = \frac{\exp\left(-\frac{\sum_{k=1}^{t-1} g_{k,j}}{\alpha L_\infty \sqrt{t}}\right)}{\sum_{i=1}^d \exp\left(-\frac{\sum_{k=1}^{t-1} g_{k,i}}{\alpha L_\infty \sqrt{t}}\right)},\ j = 1, \dots, d,$$

where $\boldsymbol{g}_t \in \partial\ell_t(\boldsymbol{x}_t)$. Note that this is exactly the same update of EG based on OMD, but here we are effectively using time-varying learning rates.

We also get that the regret guarantee is

$$\begin{aligned} \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell(\boldsymbol{u}) &\leq L_\infty \sqrt{T}\alpha\left(\sum_{i=1}^d u_i \ln u_i + \ln d\right) + \frac{1}{2\alpha L_\infty}\sum_{t=1}^T \frac{\|\boldsymbol{g}_t\|^2_\infty}{\sqrt{t}} \\ &\leq L_\infty \sqrt{T}\left(\alpha\left(\sum_{i=1}^d u_i \ln u_i + \ln d\right) + \frac{1}{\alpha}\right) \\ &\leq L_\infty \sqrt{T}\left(\alpha \ln d + \frac{1}{\alpha}\right),\ \forall \boldsymbol{u} \in V, \end{aligned} \tag{7.3}$$

where we used the fact that using $\psi_t = \alpha L_\infty \sqrt{t}\psi(\boldsymbol{x})$ and $\psi_t = \alpha L_\infty \sqrt{t}(\psi(\boldsymbol{x}) - \min_{\boldsymbol{x}} \psi(\boldsymbol{x}))$ are equivalent. The optimal choice of α to minimize the regret upper bound is $\frac{1}{\sqrt{\ln d}}$. This regret guarantee is similar to the one we proved for OMD, but with an important difference: We do not have to know in advance the number of rounds T. In OMD a similar bound would be vacuous because it would depend on the $\max_{\boldsymbol{u},\boldsymbol{x}\in V} B_\psi(\boldsymbol{u};\boldsymbol{x})$ that is infinite.

As we did in the OMD case, we can also get a bound using the local norms. Let's use the additional assumption that $g_{t,i} \geq 0$, for all $t = 1, \dots, T$ and $i = 1, \dots, d$. Using Lemma 7.14, we have for all $\boldsymbol{u} \in V$ that

$$\begin{aligned}\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell(\boldsymbol{u}) &\leq L_\infty \sqrt{T}\alpha \left(\sum_{i=1}^d u_i \ln u_i + \ln d\right) + \frac{1}{2}\sum_{t=1}^T \|\boldsymbol{g}_t\|^2_{(\nabla^2\psi_t(\boldsymbol{z}'_t))^{-1}} \\ &= L_\infty \sqrt{T}\alpha \left(\sum_{i=1}^d u_i \ln u_i + \ln d\right) + \frac{1}{2\alpha L_\infty}\sum_{t=1}^T \frac{\|\boldsymbol{g}_t\|^2_{(\nabla^2\psi(\boldsymbol{z}'_t))^{-1}}}{\sqrt{t}},\end{aligned}$$

where $\boldsymbol{z}'_t$ is on the line segment between $\boldsymbol{x}_t$ and $\tilde{\boldsymbol{x}}_{t+1}$. In this case, it is easy to calculate $\tilde{x}_{t+1,i}$ as $x_{t,i}\exp(-\frac{g_{t,i}}{\alpha L_\infty \sqrt{t}})$ for $i = 1, \dots, d$. Moreover, $\nabla^2\psi(\boldsymbol{z}'_t)$ is a diagonal matrix whose elements on the diagonal are $\frac{1}{z'_{t,i}}, i = 1, \dots, d$. Hence, we have that

$$\|\boldsymbol{g}_t\|^2_{(\nabla^2\psi(\boldsymbol{z}'_t))^{-1}} = \sum_{i=1}^d g_{t,i}^2 z'_{t,i} \leq \sum_{i=1}^d g_{t,i}^2 x_{t,i}, \tag{7.4}$$

that is less than or equal to the terms $\|\boldsymbol{g}_t\|^2_\infty$ we have in (7.3).

7.6 Example of FTRL: AdaHedge*

In this section, we explain a variation of the EG/Hedge algorithm, called AdaHedge. The basic idea is to design an algorithm that is adaptive to the sum of the squared L_∞ norm of the losses, without any prior information on the range of the losses.

First, consider the case in which we use as constant regularizer the negative entropy $\psi_t(\boldsymbol{x}) = \lambda \sum_{i=1}^d x_i \ln x_i$ defined over $\mathbb{R}^d_{\geq 0}$, where $\lambda > 0$ will be determined in the following. Set V to be the probability simplex in $\mathbb{R}^d$. Using FTRL with linear losses with this regularizer, we immediately obtain

$$\begin{aligned}\text{Regret}_T(\boldsymbol{u}) &\leq \lambda(\ln d + \sum_{i=1}^d u_i \ln u_i) + \sum_{t=1}^T \left(F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle\right) \\ &\leq \lambda \ln d + \sum_{t=1}^T \left(F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle\right),\end{aligned}$$

where we upper bounded the negative entropy of $\boldsymbol{u}$ with 0. Using the strong convexity of the regularizer w.r.t. the L_1 norm and Lemma 7.8, we would further upper bound this as

$$\text{Regret}_T(\boldsymbol{u}) \leq \lambda \ln d + \sum_{t=1}^T \frac{\|\boldsymbol{g}_t\|^2_\infty}{2\lambda}\,.$$

This suggests that the optimal λ should be $\lambda = \sqrt{\frac{\sum_{t=1}^T \|\boldsymbol{g}_t\|^2_\infty}{2\ln d}}$. However, as we have seen in Section 4.2, this choice of any parameter of the algorithm is never feasible. Hence, exactly as we did in Section 4.2, we might think of using an online version of this choice

$$\psi_t(\boldsymbol{x}) = \lambda_t \sum_{i=1}^d x_i \ln x_i \quad \text{where} \quad \lambda_t = \frac{1}{\alpha}\sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|^2_\infty}, \tag{7.5}$$

where $\alpha > 0$ is a constant that will be determined later. An important property of such choice is that it gives rise to an algorithm that is scale-free, that is its predictions $\boldsymbol{x}_t$ are invariant from the scaling of the losses by any constant factor. This is easy to see because

$$x_{t,j} \propto \exp\left(-\frac{\alpha \sum_{i=1}^{t-1} g_{i,j}}{\sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_\infty^2}}\right), \ \forall i = 1, \dots, d\,.$$

Note that this choice makes the regularizer non-decreasing over time and immediately gives us

$$\text{Regret}_T(\boldsymbol{u}) \leq \lambda_T \ln d + \sum_{t=1}^T \frac{\|\boldsymbol{g}_t\|_\infty^2}{2\lambda_t} = \frac{\ln d}{\alpha}\sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|_\infty^2} + \alpha \sum_{t=1}^T \frac{\|\boldsymbol{g}_t\|_\infty^2}{2\sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_\infty^2}}\,.$$

At this point, we might be tempted to use Lemma 4.13 to upper bound the sum in the upper bound, but unfortunately we cannot! Indeed, the denominator does not contain the term $\|\boldsymbol{g}_t\|_\infty^2$. We might add a constant to λ_t, but that would destroy the scale-freeness of the algorithm. However, it turns out that we can still prove our bound without any change to the regularizer. The key observation is that we can bound the term $F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle$ in two different ways. One way is using Lemma 7.8. The other one is

$$\begin{aligned}
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle &\leq F_t(\boldsymbol{x}_{t+1}) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \\
&= \psi_t(\boldsymbol{x}_{t+1}) + \sum_i^{t-1} \langle \boldsymbol{g}_i, \boldsymbol{x}_{t+1}\rangle - \psi_{t+1}(\boldsymbol{x}_{t+1}) - \sum_{i=1}^t \langle \boldsymbol{g}_i, \boldsymbol{x}_{t+1}\rangle + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \\
&\leq -\langle \boldsymbol{g}_t, \boldsymbol{x}_{t+1}\rangle + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \\
&\leq 2\|\boldsymbol{g}_t\|_\infty,
\end{aligned}$$

where we used the definition of $\boldsymbol{x}_{t+1}$ and the fact that the regularizer is non-decreasing over time. So, we can now write

$$\begin{aligned}
\sum_{t=1}^T F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle &\leq \sum_{t=1}^T \min\left(\frac{\alpha\|\boldsymbol{g}_t\|_\infty^2}{2\sqrt{\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_\infty^2}}, 2\|\boldsymbol{g}_t\|_\infty\right) \\
&= 2\sum_{t=1}^T \sqrt{\min\left(\frac{\alpha^2\|\boldsymbol{g}_t\|_\infty^4}{16\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_\infty^2}, \|\boldsymbol{g}_t\|_\infty^2\right)} \\
&\leq 2\sum_{t=1}^T \sqrt{\frac{2}{\frac{16\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_\infty^2}{\alpha^2\|\boldsymbol{g}_t\|_\infty^4} + \frac{1}{\|\boldsymbol{g}_t\|_\infty^2}}} \\
&= 2\sum_{t=1}^T \sqrt{2}\frac{\alpha\|\boldsymbol{g}_t\|_\infty^2}{\sqrt{\alpha^2\|\boldsymbol{g}_t\|_\infty^2 + 16\sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_\infty^2}},
\end{aligned}$$

where we used the fact that the minimum between two numbers is less than their harmonic mean. Assuming $\alpha \geq 4$ and using Lemma 4.13, we have

$$\sum_{t=1}^T F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \leq \frac{\sqrt{2}}{2}\sum_{t=1}^T \frac{\alpha\|\boldsymbol{g}_t\|_\infty^2}{\sqrt{\sum_{i=1}^t \|\boldsymbol{g}_i\|_\infty^2}} \leq \sqrt{2\sum_{t=1}^T \|\boldsymbol{g}_t\|_\infty^2}$$

and

$$\text{Regret}_T(\boldsymbol{u}) \leq \left(\frac{\ln d}{\alpha} + \alpha\sqrt{2}\right)\sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|_\infty^2}\,. \tag{7.6}$$

The bound and the assumption on α suggest to set $\alpha = \max(4, 2^{-1/4}\sqrt{\ln d})$.

We might consider ourselves happy, but there is a clear problem in the above algorithm: the choice of λ_t in the time-varying regularizer strictly depend on our upper bound. So, a loose bound will result in a poor choice of the regularization! In general, every time we use a part of the proof in the design of an algorithm we cannot expect an exciting empirical performance, unless our upper bound was really tight. So, can we design a better regularizer? Well, we need a better upper bound!

Let's consider a generic regularizer $\psi_t(\boldsymbol{x}) = \lambda_t \psi(\boldsymbol{x})$ and its corresponding FTRL with linear losses regret upper bound

$$\mathrm{Regret}_T(\boldsymbol{u}) \leq \lambda_T(\psi(\boldsymbol{u}) - \inf_{\boldsymbol{x}\in V} \psi(\boldsymbol{x})) + \sum_{t=1}^{T} \left(F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle\right),$$

where we assume λ_t to be non-decreasing in time.

Now, observe that the sum is unlikely to disappear for this kind of algorithms, so we could try to make the term $\lambda_T(\psi(\boldsymbol{u}) - \inf_{\boldsymbol{x}\in V} \psi(\boldsymbol{x}))$ of the same order of the sum. So, we would like to set λ_t of the same order of $\sum_{i=1}^{t} \left(F_i(\boldsymbol{x}_i) - F_{i+1}(\boldsymbol{x}_{i+1}) + \langle \boldsymbol{g}_i, \boldsymbol{x}_i\rangle\right)$. However, this approach would cause an annoying recurrence. So, using the fact that λ_t is non-decreasing, let's upper bound the terms in the sum just a little bit:

$$\begin{aligned}
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle &= F_t(\boldsymbol{x}_t) - \lambda_{t+1}\psi(\boldsymbol{x}_{t+1}) - \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x}_{t+1}\rangle + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \\
&\leq F_t(\boldsymbol{x}_t) - \lambda_t\psi(\boldsymbol{x}_{t+1}) - \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x}_{t+1}\rangle + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \\
&\leq F_t(\boldsymbol{x}_t) - \min_{\boldsymbol{x}\in V} \left(\lambda_t\psi(\boldsymbol{x}) + \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x}\rangle\right) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle := \delta_t \,.
\end{aligned}$$

Now, we can set $\lambda_t = \frac{1}{\alpha^2}\sum_{i=1}^{t-1} \delta_i$ for $t \geq 2$, $\lambda_1 = 0$, and $\boldsymbol{x}_1 = \mathrm{argmin}_{\boldsymbol{x}\in V}\ \psi(\boldsymbol{x})$. This immediately implies that

$$\mathrm{Regret}_T(\boldsymbol{u}) \leq \left(\psi(\boldsymbol{u}) - \inf_{\boldsymbol{x}\in V} \psi(\boldsymbol{x}) + \alpha^2\right) \lambda_{T+1} \,.$$

Setting ψ to be equal to the negative entropy, we get an algorithm known as AdaHedge.

With this choice of the regularizer, we can simplify a bit the expression of δ_t. For $t = 1$, we have $\delta_1 = \langle \boldsymbol{g}_1, \boldsymbol{x}_1\rangle - \min_{\boldsymbol{x}\in V} \langle \boldsymbol{g}_1, \boldsymbol{x}\rangle$. Instead, for $t \geq 2$, using the properties of the Fenchel conjugates, we have that

$$\delta_t = \lambda_t \ln \frac{\sum_{j=1}^{d} \exp\left(\theta_{t+1,j}/\lambda_t\right)}{\sum_{j=1}^{d} \exp\left(\theta_{t,j}/\lambda_t\right)} + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle = \lambda_t \ln \left(\sum_{j=1}^{d} x_{t,j} \exp\left(-g_{t,j}/\lambda_t\right)\right) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \,.$$

Overall, we get the pseudo-code of AdaHedge in Algorithm 7.3.

So, now we need an upper bound for λ_T. Observe that $\lambda_{t+1} = \lambda_t + \frac{1}{\alpha^2}\delta_t$. Moreover, as we have done before, we can upper bound δ_t in two different ways. In fact, from Lemma 7.8, we have $\delta_t \leq \frac{\|\boldsymbol{g}_t\|_\infty^2}{2\lambda_t}$ for $t \geq 2$. Also, denoting by $\tilde{\boldsymbol{x}}_{t+1} = \mathrm{argmin}_{\boldsymbol{x}\in V}\ \lambda_t\psi(\boldsymbol{x}) + \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x}\rangle$, we have

$$\begin{aligned}
\delta_t &= F_t(\boldsymbol{x}_t) - \min_{\boldsymbol{x}\in V} \left(\lambda_t\psi(\boldsymbol{x}) + \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \boldsymbol{x}\rangle\right) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle = F_t(\boldsymbol{x}_t) - \lambda_t\psi(\tilde{\boldsymbol{x}}_{t+1}) - \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \tilde{\boldsymbol{x}}_{t+1}\rangle + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \\
&\leq F_t(\tilde{\boldsymbol{x}}_{t+1}) - \lambda_t\psi(\tilde{\boldsymbol{x}}_{t+1}) - \sum_{i=1}^{t} \langle \boldsymbol{g}_i, \tilde{\boldsymbol{x}}_{t+1}\rangle + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle = -\langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_{t+1}\rangle + \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle \leq 2\|\boldsymbol{g}_t\|_\infty \,.
\end{aligned}$$

Algorithm 7.3 AdaHedge Algorithm

Require: $\alpha > 0$

$\lambda_1 = 0$

$\boldsymbol{x}_1 = [1/d, \ldots, 1/d] \in \mathbb{R}^d$

$\boldsymbol{\theta}_1 = \mathbf{0} \in \mathbb{R}^d$

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

  Output $\boldsymbol{x}_t$

  Receive $\boldsymbol{g}_t \in \mathbb{R}^d$ and pay $\langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle$

  Update $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \boldsymbol{g}_t$

  Set $\delta_t = \begin{cases} \langle \boldsymbol{g}_1, \boldsymbol{x}_1 \rangle - \min_{j=1,\ldots,d} \; g_{t,j}, & t = 1 \\ \lambda_t \ln \left(\sum_{j=1}^d x_{t,j} \exp\left(-g_{t,j}/\lambda_t\right) \right) + \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle & \text{otherwise} \end{cases}$

  Update $\lambda_{t+1} = \lambda_t + \frac{1}{\alpha^2}\delta_t$

  Update $x_{t+1,j} \propto \exp\left(\frac{\theta_{t+1,j}}{\lambda_{t+1}}\right), j = 1, \ldots, d$

end for

Hence, we have

$$\begin{aligned} \lambda_1 &= 0, \\ \lambda_2 &\le 2\alpha \|\boldsymbol{g}_1\|_\infty, \\ \lambda_{t+1} = \lambda_t + \frac{\delta_t}{\alpha^2} &\le \lambda_t + \frac{1}{\alpha^2} \min\left(2\|\boldsymbol{g}_t\|_\infty, \frac{\|\boldsymbol{g}_t\|_\infty^2}{2\lambda_t}\right), \; \forall t \ge 2 \, . \end{aligned}$$

We can solve this recurrence using the following Lemma, where $\Delta_t = \lambda_t$ and $a_t = \|\boldsymbol{g}_t\|_\infty$.

Lemma 7.15. *Let $\{a_t\}_{t=1}^\infty$ be any sequence of non-negative real numbers. Suppose that $\{\Delta_t\}_{t=0}^\infty$ is a sequence of non-negative real numbers satisfying*

$$\Delta_0 = 0 \qquad \textit{and} \qquad \Delta_t \le \Delta_{t-1} + \min\left\{ b a_t, \; c\frac{a_t^2}{2\Delta_{t-1}} \right\} \quad \textit{for any } t \ge 1 \, .$$

Then, for any $T \ge 0$, $\Delta_T \le \sqrt{(b^2 + c)\sum_{t=1}^T a_t^2}$.

Proof. Observe that

$$\Delta_T^2 = \sum_{t=1}^T (\Delta_t^2 - \Delta_{t-1}^2) = \sum_{t=1}^T \left[(\Delta_t - \Delta_{t-1})^2 + 2(\Delta_t - \Delta_{t-1})\Delta_{t-1} \right] \, .$$

We bound each term in the sum separately. The left term of the minimum inequality in the definition of Δ_t gives $(\Delta_t - \Delta_{t-1})^2 \le b^2 a_t^2$, while the right term gives $2(\Delta_t - \Delta_{t-1})\Delta_{t-1} \le c a_t^2$. So, we conclude $\Delta_T^2 \le (b^2 + c)\sum_{t=1}^T a_t^2$. □

So, overall we got

$$\text{Regret}_T(\boldsymbol{u}) \le \left(\psi(\boldsymbol{u}) - \inf_{\boldsymbol{x} \in V} \psi(\boldsymbol{x}) + \alpha^2 \right) \lambda_{T+1} \le \left(\frac{\psi(\boldsymbol{u}) - \inf_{\boldsymbol{x} \in V} \psi(\boldsymbol{x})}{\alpha^2} + 1 \right) \sqrt{(4 + \alpha^2) \sum_{t=1}^T \|\boldsymbol{g}_t\|_\infty^2},$$

and setting $\alpha = \sqrt{\ln d}$, we have

$$\text{Regret}_T(\boldsymbol{u}) \le 2\sqrt{(4 + \ln d) \sum_{t=1}^T \|\boldsymbol{g}_t\|_\infty^2} \, .$$

Note that this is roughly the same regret in (7.6), but the very important difference is that this new regret bound depends on the much tighter quantity λ_{T+1}, that we upper bounded with a term proportional to $\sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|_\infty^2}$, but in general it will be much smaller than that. For example, λ_{T+1} could be upper bounded using the tighter local norms, see Section 7.5. Instead, in the first solution, the regret will always be dominated by the term $\sqrt{\sum_{t=1}^T \|\boldsymbol{g}_t\|_\infty^2}$ because we explicitly use it in the regularizer.

There is an important lesson to be learned from AdaHedge: the regret is not the full story and algorithms with the same worst-case guarantee can exhibit vastly different empirical behaviours. Unfortunately, this message is rarely heard and there is a part of the community that focuses too much on the worst-case guarantee rather than on the empirical performance. Even worse, sometimes people favor algorithms with a "more elegant analysis" completely ignoring the likely worse empirical performance.

7.7 Example of FTRL: Group Norms

Suppose you want to want to do linear regression and you have n different groups of features in $\mathbb{R}^d$ associated to each sample. For example, they could represent different modalities (audio, video, etc.) associated to the same sample. You could just concatenate all the features in a long vector and learn with them using FTRL with a squared L_2 regularizer. In alternative, we could use FTRL with p-norm (see Exercise 7.8), and tune p to have a logarithmic dependency on $n\,d$. However, there is something in between: We may want to learn a linear combination of predictors, one on each group, trying to use a very large number of groups and paying a small price for it. This is the equivalent of the learning with experts setting, where each expert now is a linear predictor. In fact, one might be tempted to use one online learning algorithm on each group and a learning with experts algorithm on top that combines the predictions. However, this approach is very clunky and we can do much better. In fact, we can use an L_2 norm over each group and a p-norm over the groups with p very close to 1. In this way, we expect a regret upper bound that depends on the maximum L_2 norm of the subgradients w.r.t. each single group. Let's see how this would work.

First, we introduce the definition of *group norms*.

Definition 7.16 (Group Norm)**.** *Let $\boldsymbol{X} = \begin{bmatrix}\boldsymbol{x}_1 & \boldsymbol{x}_2 & \dots & \boldsymbol{x}_n\end{bmatrix}$ be a $d \times n$ real matrix with columns $\boldsymbol{x}_i \in \mathbb{R}^d$. Given norms Ψ and Φ on $\mathbb{R}^d$ and $\mathbb{R}^n$, we define the **group norm** $\|\boldsymbol{X}\|_{\Psi,\Phi}$ as*

$$\|\boldsymbol{X}\|_{\Psi,\Phi} := \Phi([\Psi(\boldsymbol{x}_1), \dots, \Psi(\boldsymbol{x}_n)]) \,.$$

It is possible to check that this is indeed a norm and we can also calculate the dual norm. We will need an additional definition.

Definition 7.17 (Absolutely Symmetric Function)**.** *We say that $f : \mathbb{R}^d \to \mathbb{R}$ is **absolutely symmetric** if it is invariant under permutations and sign changes of the components of its input.*

Lemma 7.18 ([Kakade et al., 2009, Lemma 17])**.** *Let Φ be an absolutely symmetric norm on $\mathbb{R}^n$. Then,*

$$(\|\cdot\|_{\Psi,\Phi})_\star = \|\cdot\|_{\Psi^\star,\Phi^\star} \,.$$

From this Lemma and Example 5.11, we immediate have that the Fenchel conjugate of $\frac{1}{2}\|\cdot\|_{\Psi,\Phi}^2$ is $\frac{1}{2}\|\cdot\|_{\Psi^\star,\Phi^\star}^2$.

We can also construct strongly convex functions based on group norms.

Theorem 7.19 ([Kakade et al., 2009, Theorem 18])**.** *Let Ψ, Φ be absolutely symmetric norms on $\mathbb{R}^d$, $\mathbb{R}^n$. Let $\Phi^2 \circ \sqrt{} : \mathbb{R}^n \to (-\infty, +\infty]$ denote the following function,*

$$(\Phi^2 \circ \sqrt{})(\boldsymbol{x}) := \Phi^2(\sqrt{x_1}, \dots, \sqrt{x_n}) \,.$$

Suppose, $\Phi^2 \circ \sqrt{}$ is a norm on $\mathbb{R}^n_{\geq 0}$. Further, let the functions Ψ^2 and Φ^2 be λ_1- and λ_2-smooth w.r.t. Ψ and Φ respectively. Then, $\|\cdot\|_{\Psi,\Phi}^2$ is $(\lambda_1 + \lambda_2)$-smooth w.r.t. $\|\cdot\|_{\Psi,\Phi}^2$.

We can now specialize this result to *group p-norms*.

Algorithm 7.4 FTRL with Group Norms for Linear Regression

Require: $q > 0, \alpha > 0$

$\boldsymbol{\theta}_{1,i} = \mathbf{0} \in \mathbb{R}^d,\ i = 1, \dots, n$

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

  Receive matrix of features $\boldsymbol{Z}_t = \begin{bmatrix} \boldsymbol{z}_{t,1} & \dots & \boldsymbol{z}_{t,n} \end{bmatrix}$

  Predict $\hat{y}_t = \sum_{i=1}^n \boldsymbol{z}_{t,i}^\top \boldsymbol{x}_{t,i}$

  Receive label y_t and pay $|\hat{y}_t - y_t|$

  Update $\boldsymbol{\theta}_{t+1,i} = \boldsymbol{\theta}_{t,i} - \boldsymbol{z}_{t,i}\text{sign}(\hat{y}_t - y_t),\ i = 1, \dots, n$

  Update $\boldsymbol{x}_{t+1,i} = \frac{\boldsymbol{\theta}_{t+1,i}\|\boldsymbol{\theta}_{t+1,i}\|_2^{q-2}}{\alpha\sqrt{t}(\sum_{j=1}^n \|\boldsymbol{\theta}_{t+1,j}\|_2^q)^{2/q}},\ i = 1, \dots, n$

end for

Corollary 7.20. *Let $q, s \geq 2$. The function $\frac{1}{2}\|\cdot\|_{q,s}^2$ is $(q+s-2)$-smooth w.r.t. $\|\cdot\|_{q,s}$ on $\mathbb{R}^{d\times n}$.*

Assuming $n \geq 2$, the above suggests to use FTRL with linearized losses and regularizer $\psi_t(\boldsymbol{X}) = \frac{\alpha\sqrt{t}}{2}\|\boldsymbol{X}\|_{2,p}^2$, where $1 < p \leq 2$ and $\alpha > 0$ will be decided in the following. From the above and Example 5.11, the dual is $\psi_t^\star(\boldsymbol{\Theta}) = \frac{1}{2\alpha\sqrt{t}}\|\boldsymbol{\Theta}\|_{2,q}^2$ and it is $\frac{q}{\alpha\sqrt{t}}$-smooth, where $\frac{1}{p} + \frac{1}{q} = 1$. Moreover,

$$[\nabla\psi_t^\star(\boldsymbol{\Theta})]_i = \frac{\boldsymbol{\theta}_i\|\boldsymbol{\theta}_i\|_2^{q-2}}{\alpha\sqrt{t}(\sum_{j=1}^n \|\boldsymbol{\theta}_j\|_2^q)^{1-2/q}},\ i = 1, \dots, n\,.$$

For simplicity, we will use the absolute loss as loss function. We are now ready to introduce Algorithm 7.4.

From the analysis of FTRL with linearized losses, the properties of the group norms above, and Theorem 6.11, it is immediate to state a regret upper bound. With the notation in Algorithm 7.4, for any $\boldsymbol{U} \in \mathbb{R}^{d\times n}$ we have

$$\sum_{t=1}^T |\hat{y}_t - y_t| - \sum_{t=1}^T \left|\sum_{i=1}^n \boldsymbol{z}_t^\top \boldsymbol{u}_i - y_t\right| \leq \frac{\alpha\sqrt{T}}{2}\|\boldsymbol{U}\|_{2,p}^2 + \frac{q}{2\alpha}\sum_{t=1}^T \frac{\|\boldsymbol{Z}_t\|_{2,q}^2}{\sqrt{t}} \leq \frac{\alpha\sqrt{T}}{2}\|\boldsymbol{U}\|_{2,p}^2 + \frac{q\sqrt{T}}{\alpha}\max_{t\leq T}\|\boldsymbol{Z}_t\|_{2,q}^2\,.$$

To obtain a logarithmic dependency in n, we can now proceed as in Section 6.6. Assuming $\|\boldsymbol{z}_{t,i}\|_2 \leq L$, we have that $\|\boldsymbol{Z}_t\|_{2,q}^2 \leq L^2 n^{2/q}$. Hence, setting $\alpha = qLn^{1/q}$ and $q = \max(2\ln n, 2)$ we have that $\frac{q}{\alpha}\|\boldsymbol{Z}_t\|_{2,q}^2 \leq L\max(\sqrt{2e\ln n}, 2)$. Moreover, we can upper bound $\|\boldsymbol{U}\|_{2,p}^2$ with $\|\boldsymbol{U}\|_{2,1}^2$. Hence, the final regret upper bound is $L\max(\sqrt{2e\ln n}, 2)\left(\frac{\|\boldsymbol{U}\|_{2,1}^2}{2} + 1\right)\sqrt{T}$, that is logarithmic in the number of groups.

7.8 Composite Losses and Proximal Operators

Let's now see a variant of the linearization of the losses: *partial linearization of composite losses*.

Suppose that the losses we receive are composed by two terms: one convex function changing over time and another part is fixed and known. These losses are called *composite*. For example, we might have $\ell_t(\boldsymbol{x}) = \tilde{\ell}_t(\boldsymbol{x}) + \lambda\|\boldsymbol{x}\|_1$. Using the linearization, we might just take the subgradient of ℓ_t. However, in this particular case, we might lose the ability of the L_1 norm to produce sparse solutions.

There is a better way to deal with these kind of losses: Move the constant part of the loss inside the regularization term. In this way, that part will not be linearized but used exactly in the argmin of the update. Moreover, the regret of the algorithm will depend only on the subgradient of $\tilde{\ell}_t$ rather than subgradients of the full loss ℓ_t. Assuming that the argmin is still easily computable, you can always expect better performance from this approach. In particular, in the case of adding an L_1 norm to the losses, you will be predicting in each step with the solution of an L_1 regularized optimization problem.

Remark 7.21. *Given that we do not pay for the gradient of the regularizer in the regret upper bound of FTRL, one wonder why we do not put the entire loss function in the regularizer. However, this would require knowledge of the future because when at time t we construct the regularizer ψ_t, we do not have access to the future loss ℓ_t.*

Practically speaking, in the example above, we will define $\psi_t(\boldsymbol{x}) = L\sqrt{t}(\psi(\boldsymbol{x}) - \min_{\boldsymbol{y}} \psi(\boldsymbol{y})) + \lambda t\|\boldsymbol{x}\|_1$, where we assume ψ to be 1-strongly convex and the losses $\tilde{\ell}_t$ be L-Lipschitz. Note that we use at time t a term $\lambda t\|\boldsymbol{x}\|_1$ because we anticipate the next term in the next round. Given that $\psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \ell_t(\boldsymbol{x})$ is $L\sqrt{t}$-strongly convex, using Lemma 7.8, we have

$$
\begin{aligned}
&\sum_{t=1}^{T} \tilde{\ell}_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \tilde{\ell}_t(\boldsymbol{u}) \\
&\quad \le \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \\
&\quad \le L\sqrt{T}\left(\psi(\boldsymbol{u}) - \min_{\boldsymbol{x}} \psi(x)\right) + \lambda T\|\boldsymbol{u}\|_1 - \min_{\boldsymbol{x}}(\lambda\|\boldsymbol{x}\|_1 + \psi(\boldsymbol{x}) - \min_{\boldsymbol{y}} \psi(\boldsymbol{y})) + \frac{1}{2}\sum_{t=1}^{T} \frac{\|\boldsymbol{g}_t\|_\star^2}{L\sqrt{t}} - \lambda \sum_{t=1}^{T-1} \|\boldsymbol{x}_{t+1}\|_1 \\
&\quad \le L\sqrt{T}\left(1 + \psi(\boldsymbol{u}) - \min_{\boldsymbol{x}} \psi(x)\right) + \lambda T\|\boldsymbol{u}\|_1 - \lambda\|\boldsymbol{x}_1\|_1 - \lambda \sum_{t=1}^{T-1} \|\boldsymbol{x}_{t+1}\|_1 \\
&\quad = L\sqrt{T}\left(1 + \psi(\boldsymbol{u}) - \min_{\boldsymbol{x}} \psi(x)\right) + \lambda T\|\boldsymbol{u}\|_1 - \lambda \sum_{t=1}^{T} \|\boldsymbol{x}_t\|_1,
\end{aligned}
$$

where $\boldsymbol{g}_t \in \partial\tilde{\ell}_t(\boldsymbol{x}_t)$. Reordering the terms, we have

$$
\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) = \sum_{t=1}^{T} (\lambda\|\boldsymbol{x}_t\|_1 + \tilde{\ell}_t(\boldsymbol{x}_t)) - \sum_{t=1}^{T} (\lambda\|\boldsymbol{u}\|_1 + \tilde{\ell}_t(\boldsymbol{u})) \le L\sqrt{T}\left(\psi(\boldsymbol{u}) - \min_{\boldsymbol{x}} \psi(\boldsymbol{x}) + 1\right) .
$$

Example 7.22. *Let's also take a look at the update rule in that case that $\psi(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$ and we get composite losses with the L_1 norm. We have*

$$
\boldsymbol{x}_t = \operatorname*{argmin}_{\boldsymbol{x}} \frac{L\sqrt{t}}{2}\|\boldsymbol{x}\|_2^2 + \lambda t\|\boldsymbol{x}\|_1 + \sum_{i=1}^{t-1} \langle \boldsymbol{g}_i, \boldsymbol{x} \rangle .
$$

We can solve this problem observing that the minimization decomposes over each coordinate of $\boldsymbol{x}$. Denote by $\boldsymbol{\theta}_t = \sum_{i=1}^{t-1} \boldsymbol{g}_i$. Hence, we know from first-order optimality condition in Theorem 6.12 that $x_{t,i}$ is the solution for the coordinate i iff there exists $v_i \in \partial|x_{t,i}|$ such that

$$
L\sqrt{t}x_{t,i} + \lambda t v_i + \theta_{t,i} = 0 .
$$

The difficulty comes from the fact that $\boldsymbol{v}$ is a subgradient in the next point $\boldsymbol{x}_t$, that we don't know because we don't know $\boldsymbol{v}$. Unfortunately there is no automatic way to find v_i, so we have to "guess" somehow the correct expression of v_i and verify its correctness in the above expression. In particular, considering 3 different cases, we have

- *$|\theta_{t,i}| \le \lambda t$, then $x_{t,i} = 0$ and $v_i = -\frac{\theta}{\lambda t}$.*
- *$\theta_{t,i} > \lambda t$, then $x_{t,i} = -\frac{\theta_{t,i} - \lambda t}{L\sqrt{t}}$ and $v_i = -1$.*
- *$\theta_{t,i} < -\lambda t$, then $x_{t,i} = -\frac{\theta_{t,i} + \lambda t}{L\sqrt{t}}$ and $v_i = 1$.*

So, overall we have

$$
x_{t,i} = -\frac{\operatorname{sign}(\theta_{t,i}) \max(|\theta_{t,i}| - \lambda t, 0)}{L\sqrt{t}} .
$$

Observe as this update will produce sparse solutions, while just taking the subgradient of the L_1 norm would have never produced sparse predictions.

In the example above, we calculated something like

$$\operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \|\boldsymbol{x}\|_1 - \langle \boldsymbol{v}, \boldsymbol{x}\rangle + \frac{1}{2}\|\boldsymbol{x}\|_2^2 = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \|\boldsymbol{x}\|_1 - \langle \boldsymbol{v}, \boldsymbol{x}\rangle + \frac{1}{2}\|\boldsymbol{x}\|_2^2 + \frac{1}{2}\|\boldsymbol{v}\|_2^2 = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \|\boldsymbol{x}\|_1 + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}\|_2^2 \,.$$

This operation is known in the optimization literature as *Proximal Operator* of the L_1 norm. In general, a proximal operator of a convex, proper, and closed function $f : \mathbb{R}^d \to (-\infty, +\infty]$ is defined as

$$\operatorname{Prox}_f(\boldsymbol{v}) = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{v}\|_2^2 \,. \tag{7.7}$$

Proximal operators are used in optimization in the same way as we used it: They allow to minimize the entire function rather a linear approximation of it. Also, proximal operators generalize the concept of Euclidean projection. Indeed, $\operatorname{Prox}_{i_V}(\boldsymbol{v}) = \Pi_V(\boldsymbol{v})$.

From the definition, we also derive the important property of proximal operators that they minimize the function in each update. In other words, if $\boldsymbol{x}_{t+1} = \operatorname{Prox}_f(\boldsymbol{x}_t)$, then $f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t)$. Another interesting property is the fact that we can write the proximal operator as an *implicit* subgradient descent update. Let $\eta > 0$ and consider the following update

$$\boldsymbol{x}_{t+1} = \operatorname{Prox}_{\eta f}(\boldsymbol{x}_t) = \operatorname*{argmin}_{\boldsymbol{x}\in\mathbb{R}^d} \eta f(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{x}_t\|_2^2 \,.$$

Then, from Theorem 6.12 we have

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta \boldsymbol{g}_t,$$

where $\boldsymbol{g}_t \in \partial f(\boldsymbol{x}_{t+1})$. Hence, we do an update step using the subgradient in the *next* iteration. This also reveals a connection between OMD and proximal updates. Indeed, the update in (7.7) resembles the one in (6.3), where the function is not linearized. We will use this kind of updates in Section 11.6.

7.9 FTRL Regret Bound with Proximal Regularizers

Let's now go back to the FTRL regret bound and let's see if you can strengthen it in the case that the regularizer is *proximal*, that is it satisfies that $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x}\in V} \psi_{t+1}(\boldsymbol{x}) - \psi_t(\boldsymbol{x})$.

Lemma 7.23. *With the notation in Algorithm 7.1, denote by $F_t(\boldsymbol{x}) = \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \ell_i(\boldsymbol{x})$ for all t. Assume $V \subseteq X$ and V convex. Also, assume that F_{t+1} is closed, subdifferentiable, and λ_{t+1}-strongly convex w.r.t. $\|\cdot\|$ in V, the regularizer is such that $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x}\in V} \psi_{t+1}(\boldsymbol{x}) - \psi_t(\boldsymbol{x})$. Then, $\boldsymbol{x}_{t+1}$ exists and is unique. Moreover, if $\partial\ell_t(\boldsymbol{x}_t)$ is non-empty, we have*

$$F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) \leq \frac{\|\boldsymbol{g}_t\|_\star^2}{2\lambda_{t+1}} + \psi_t(\boldsymbol{x}_t) - \psi_{t+1}(\boldsymbol{x}_t),\ \forall \boldsymbol{g}_t \in \partial\ell_t(\boldsymbol{x}_t) \,.$$

Proof. The existence and unicity is given by Theorem 6.8. We have

$$\begin{aligned} F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) &= (F_t(\boldsymbol{x}_t) + \ell_t(\boldsymbol{x}_t) + \psi_{t+1}(\boldsymbol{x}_t) - \psi_t(\boldsymbol{x}_t)) - F_{t+1}(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_t) + \psi_t(\boldsymbol{x}_t) \\ &= F_{t+1}(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_t) + \psi_t(\boldsymbol{x}_t) \\ &\leq \frac{\|\boldsymbol{g}_t\|_\star^2}{2\lambda_{t+1}} - \psi_{t+1}(\boldsymbol{x}_t) + \psi_t(\boldsymbol{x}_t), \end{aligned}$$

where in the inequality we used Corollary 7.7, the fact that $\boldsymbol{x}_{t+1} = \operatorname{argmin}_{\boldsymbol{x}\in V} F_{t+1}(\boldsymbol{x})$, and $\boldsymbol{g}_t \in \partial(F_{t+1} + i_V)(\boldsymbol{x}_t)$. Observing that, from the proximal property, we have that $\boldsymbol{x}_t = \operatorname{argmin}_{\boldsymbol{x}\in V} F_t(\boldsymbol{x}) + \psi_{t+1}(\boldsymbol{x}) - \psi_t(\boldsymbol{x})$, from Theorem 6.12 we have $\boldsymbol{0} \in \partial(F_t + i_V + \psi_{t+1} - \psi_t)(\boldsymbol{x}_t)$. Hence, using Theorem 2.18, and remembering that $F_{t+1}(\boldsymbol{x}) = F_t(\boldsymbol{x}) + \ell_t(\boldsymbol{x}) + \psi_{t+1}(\boldsymbol{x}) - \psi_t(\boldsymbol{x})$, we have that $\partial\ell_t(\boldsymbol{x}_t) \subseteq \partial(F_{t+1} + i_V)(\boldsymbol{x}_t)$. □

Remark 7.24. *Note that a constant regularizer is proximal because any point is the minimizer of the zero function. On the other hand, a constant regularizer makes Lemmas 7.8 and 7.23 the same.*

Algorithm 7.5 Follow-the-Regularized-Leader Algorithm with "Quadratized" Losses

Require: A sequence of regularizers $\psi_1, \dots, \psi_T : X \to \mathbb{R}$, closed non-empty convex set $V \subseteq X \subseteq \mathbb{R}^d$

for $t = 1$ **to** T **do**
  Output $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x} \in V} \ \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \left(\langle \boldsymbol{g}_i, \boldsymbol{x} \rangle + \frac{\mu_t}{2} \|\boldsymbol{x} - \boldsymbol{x}_i\|^2 \right)$
  Receive $\ell_t : V \to \mathbb{R}$ subdifferentiable in V and pay $\ell_t(\boldsymbol{x}_t)$
  Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
  Calculate the strong convexity μ_t of ℓ_t
end for

We will now use the above lemma (or equivalently Lemma 7.8) to prove a logarithmic regret bound for FTL with strongly convex losses.

Corollary 7.25. *Let $\ell_t : V \to \mathbb{R}$ be μ_t-strongly convex w.r.t. $\|\cdot\|$, for $t = 1, \dots, T$. Set the sequence of regularizers to zero. Then, FTL guarantees a regret of*

$$\sum_{t=1}^{T} \ell(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \leq \frac{1}{2} \sum_{t=1}^{T} \frac{\|\boldsymbol{g}_t\|_\star^2}{\sum_{i=1}^{t} \mu_i}, \ \forall \boldsymbol{u} \in \mathbb{R}^d, \ \forall \boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t) \ .$$

The above regret guarantee is the same of OMD over strongly convex losses, but here we do not need to know the strong convexity of the losses. In fact, we just need to output the minimizer over the past losses. However, this might be undesirable because now each update is an optimization problem.

Hence, we can again use the idea of replacing the losses with an easy *surrogate*. In the Lipschitz case, it made sense to use linear losses because they were used to upper bound the regret of the true losses. However, here you can do better and use *quadratic* losses, because the losses are strongly convex. In fact, assuming ℓ_t to be μ_t strongly convex w.r.t. $\|\cdot\|$, we have

$$\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u}) \leq \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle - \frac{\mu_t}{2} \|\boldsymbol{x}_t - \boldsymbol{u}\|^2 = \tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_t(\boldsymbol{u}) \ .$$

So, we could run FTRL on the quadratic losses $\tilde{\ell}_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{\mu_t}{2} \|\boldsymbol{x} - \boldsymbol{x}_t\|^2$, where $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$. Depending on the norm, these quadratic losses might be strongly convex allowing us to use Corollary 7.25. The algorithm is in Algorithm 7.5.

For example, consider the case that the losses are strongly convex w.r.t. the L_2 norm and there is no regularizer. Using Theorem 6.15, the update in Algorithm 7.5 becomes

$$\boldsymbol{x}_t = \operatorname*{argmin}_{\boldsymbol{x} \in V} \sum_{i=1}^{t-1} \left(\langle \boldsymbol{g}_i, \boldsymbol{x} \rangle + \frac{\mu_i}{2} \|\boldsymbol{x} - \boldsymbol{x}_i\|_2^2 \right) = \Pi_V \left(\frac{\sum_{i=1}^{t-1} (\mu_i \boldsymbol{x}_i - \boldsymbol{g}_i)}{\sum_{i=1}^{t-1} \mu_i} \right), \tag{7.8}$$

where the only expensive operation is the projection, that can be computed in polynomial time. Moreover, we will get exactly the same regret bound as in Corollary 7.25, with the only difference that here the guarantee holds for a specific choice of the $\boldsymbol{g}_t$ rather than for any subgradient in $\partial \ell_t(\boldsymbol{x}_t)$.

Example 7.26. *Going back to the example in the first chapter, where $V = [0, 1]$ and $\ell_t(\boldsymbol{x}) = (x - y_t)^2$ are strongly convex, we now see immediately that FTRL without a regularizer, that is Follow the Leader, gives logarithmic regret.*

7.10 Online Newton Step

In the previous section, we saw that the notion of strong convexity allows us to build quadratic surrogate loss functions, on which FTRL has smaller regret. Can we find a more general notion of strong convexity that allows to get a small regret for a larger class of functions? We can start from strong convexity and try to generalize it. So, instead of asking that the function is strongly convex w.r.t. a norm, we might be happy requiring strong convexity holds in a particular point w.r.t. a seminorm that depends on the points itself.

In particular, we can require that for each loss $\ell_t : V \to \mathbb{R}$ the following holds

$$\forall \boldsymbol{x} \in V, \boldsymbol{g} \in \partial \ell_t(\boldsymbol{x}), \exists \boldsymbol{A}_t \in \mathbb{R}^{d \times d} \ : \ \boldsymbol{A}_t \succeq 0, \ \ell_t(\boldsymbol{y}) \geq \ell_t(\boldsymbol{x}) + \langle \boldsymbol{g}, \boldsymbol{y} - \boldsymbol{x} \rangle + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_{\boldsymbol{A}_t}^2, \forall \boldsymbol{y} \in V,$$

where $\|\boldsymbol{x}\|_{\boldsymbol{A}_t}$ is defined as $\sqrt{\boldsymbol{x}^\top \boldsymbol{A}_t \boldsymbol{x}}$. Note that this is a weaker property than strong convexity because $\boldsymbol{A}_t$ depends on $\boldsymbol{x}$. On the other hand, in the characterization of strong convexity in Lemma 4.2 we want the last term to be the same norm everywhere in the space. See also Exercise 6.7 for a similar generalization of strong convexity.

The rationale of this new definition is that it still allows us to build surrogate loss functions, but without requiring to have strong convexity over the entire space. However, we will split the quadratic lower bound in two parts: One part will be the surrogate loss and another part will go into the regularizer. In particular, we can think to use FTRL on the surrogate losses

$$\tilde{\ell}_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle$$

and the proximal regularizers $\psi_t(\boldsymbol{x}) = \frac{\lambda}{2}\|\boldsymbol{x}\|^2 + \frac{1}{2}\sum_{i=1}^{t-1} \|\boldsymbol{x}_i - \boldsymbol{x}\|_{\boldsymbol{A}_i}^2$, where $\lambda > 0$. We will denote by $\boldsymbol{S}_t = \lambda \boldsymbol{I}_d + \sum_{i=1}^t \boldsymbol{A}_i$.

Remark 7.27. *Note that $\|\boldsymbol{x}\|_{\boldsymbol{S}_t} := \sqrt{\boldsymbol{x}^\top \boldsymbol{S}_t \boldsymbol{x}}$ is a norm because $\boldsymbol{S}_t$ is positive definite and $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top \boldsymbol{S}_t \boldsymbol{x}$ is 1-strongly convex w.r.t. $\|\cdot\|_{\boldsymbol{S}_t}$ defined as $\|\boldsymbol{x}\|_{\boldsymbol{S}_t} = \sqrt{\boldsymbol{x}^\top \boldsymbol{S}_t \boldsymbol{x}}$ (from Theorem 4.3 given that the Hessian is $\boldsymbol{S}_t$ and $\boldsymbol{x}^\top \nabla^2 f(\boldsymbol{y}) \boldsymbol{x} = \|\boldsymbol{x}\|_{\boldsymbol{S}_t}^2$). Also, from Example 4.18, the dual norm of $\|\cdot\|_{\boldsymbol{S}_t}$ is $\|\cdot\|_{\boldsymbol{S}_t^{-1}}$.*

From the above remark, we have that the regularizer $\psi_t(\boldsymbol{x})$ is 1-strongly convex w.r.t $\|\cdot\|_{\boldsymbol{S}_{t-1}}$. Hence, using the FTRL regret equality in Lemma 7.1 and Lemma 7.23 for proximal regularizers, we immediately get the following guarantee

$$\begin{aligned}
\sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle &= \sum_{t=1}^T \tilde{\ell}_t(\boldsymbol{x}_t) - \sum_{t=1}^T \tilde{\ell}_t(\boldsymbol{u}) \\
&\leq \psi_{T+1}(\boldsymbol{u}) + \frac{1}{2}\sum_{t=1}^T \|\boldsymbol{g}_t\|_{\boldsymbol{S}_t^{-1}}^2 + \sum_{t=1}^T \left(\psi_t(\boldsymbol{x}_t) - \psi_{t+1}(\boldsymbol{x}_t)\right) \\
&= \frac{\lambda}{2}\|\boldsymbol{u}\|^2 + \frac{1}{2}\sum_{t=1}^T \|\boldsymbol{x}_t - \boldsymbol{u}\|_{\boldsymbol{A}_t}^2 + \frac{1}{2}\sum_{t=1}^T \|\boldsymbol{g}_t\|_{\boldsymbol{S}_t^{-1}}^2 \ .
\end{aligned}$$

So, reordering the terms we have

$$\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \leq \sum_{t=1}^T \left(\langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle - \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{x}_t\|_{\boldsymbol{A}_t}^2 \right) \leq \frac{\lambda}{2}\|\boldsymbol{u}\|^2 + \frac{1}{2}\sum_{t=1}^T \|\boldsymbol{g}_t\|_{\boldsymbol{S}_t^{-1}}^2 \ . \tag{7.9}$$

Note how the proof and the algorithm mirror what we did for FTRL with strongly convex losses in Section 7.9.

Remark 7.28. *It is possible to generalize our Lemma of FTRL for proximal regularizers to hold in this generalized notion of strong convexity. This would allow to get exactly the same bound running FTRL over the original losses with regularizer $\psi(\boldsymbol{x}) = \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2$.*

Let's now see a practical instantiation of this idea. Consider the case that the sequence of loss functions we receive satisfy

$$\ell_t(\boldsymbol{x}) - \ell_t(\boldsymbol{u}) \leq \langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{u} \rangle - \frac{\mu}{2}(\langle \boldsymbol{g}, \boldsymbol{x} - \boldsymbol{u} \rangle)^2, \ \forall \boldsymbol{x}, \boldsymbol{u} \in V, \boldsymbol{g} \in \partial \ell_t(\boldsymbol{x}) \ . \tag{7.10}$$

In words, *we assume to have a class of functions that can be lower bounded by a quadratic that depends on the current subgradient*. Denoting by $A_t = \mu \boldsymbol{g}_t \boldsymbol{g}_t^\top$, we can use the above idea using

$$\psi_t(\boldsymbol{x}) = \frac{\lambda}{2}\|\boldsymbol{x}\|^2 + \frac{1}{2}\sum_{i=1}^{t-1} \|\boldsymbol{x}_i - \boldsymbol{x}\|_{\boldsymbol{A}_i}^2 = \frac{\lambda}{2}\|\boldsymbol{x}\|^2 + \frac{\mu}{2}\sum_{i=1}^{t-1} (\langle \boldsymbol{g}_i, \boldsymbol{x} - \boldsymbol{x}_i \rangle)^2 \ .$$

Hence, the update rule would be

$$\boldsymbol{x}_t = \operatorname*{argmin}_{\boldsymbol{x}\in V} \sum_{i=1}^{t-1}\langle \boldsymbol{g}_t, \boldsymbol{x}\rangle + \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 + \frac{\mu}{2}\sum_{i=1}^{t-1}(\langle \boldsymbol{g}_i, \boldsymbol{x}-\boldsymbol{x}_i\rangle)^2 \ .$$

We obtain the following algorithm, called Online Newton Step (ONS).

Algorithm 7.6 Online Newton Step Algorithm

Require: $V \subset \mathbb{R}^d$ closed non-empty convex set, $\lambda, \mu > 0$
1: **for** $t = 1$ **to** T **do**
2: Set $\boldsymbol{x}_t = \operatorname{argmin}_{\boldsymbol{x}\in V} \sum_{i=1}^{t-1}\langle \boldsymbol{g}_i, \boldsymbol{x}\rangle + \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 + \frac{\mu}{2}\sum_{i=1}^{t-1}(\langle \boldsymbol{g}_i, \boldsymbol{x}-\boldsymbol{x}_i\rangle)^2$
3: Receive $\ell_t : V \to \mathbb{R}$ subdifferentiable in V and pay $\ell_t(\boldsymbol{x}_t)$
4: Set $\boldsymbol{g}_t \in \partial\ell(\boldsymbol{x}_t)$
5: **end for**

Denoting by $S_t = \lambda \boldsymbol{I}_d + \sum_{i=1}^t \boldsymbol{A}_i$ and using (7.9), we have

$$\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{1}{2}\sum_{t=1}^T \|\boldsymbol{g}_t\|^2_{\boldsymbol{S}_t^{-1}} \ .$$

To bound the last term, we will use the following Lemma.

Lemma 7.29 ([Cesa-Bianchi and Lugosi, 2006, Lemma 11.11 and Theorem 11.7])**.** *Let $\boldsymbol{z}_1, \dots, \boldsymbol{z}_T$ a sequence of vectors in $\mathbb{R}^d$ and $\lambda > 0$. Define $H_t = \lambda \boldsymbol{I}_d + \sum_{i=1}^t \boldsymbol{z}_i \boldsymbol{z}_i^\top$. Then, the following holds*

$$\sum_{t=1}^T \boldsymbol{z}_t^\top \boldsymbol{H}_t^{-1} \boldsymbol{z}_t \le \sum_{i=1}^d \ln\left(1 + \frac{\lambda_i}{\lambda}\right),$$

where $\lambda_1, \dots, \lambda_d$ are the eigenvalues of $\boldsymbol{H}_T - \lambda \boldsymbol{I}_d$.

Putting all together and assuming $\|\boldsymbol{g}_t\|_2 \le L$ and (7.10) holds for the losses, then ONS satisfies the following regret

$$\sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \le \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{1}{2\mu}\sum_{i=1}^d \ln\left(1 + \frac{\lambda_i}{\lambda}\right) \le \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{d}{2\mu}\ln\left(1 + \frac{\mu T L^2}{d\lambda}\right),$$

where in the second inequality we used the inequality of arithmetic and geometric means, $(\prod_{i=1}^d x_i)^{1/d} \le \frac{1}{d}\sum_{i=1}^d x_i$, and the fact that $\sum_{i=1}^d \lambda_i \le \mu T L^2$.

Hence, if the losses satisfy (7.10), we can guarantee a logarithmic regret. However, differently from the strongly convex case, here the complexity of the update is at least quadratic in the number of dimensions. Moreover, the regret also depends linearly on the number of dimensions.

Remark 7.30. *Despite the name, the ONS algorithm should not be confused with the Newton algorithm. They are similar in spirit because they both construct quadratic approximation to the function, but the Newton algorithm uses the exact Hessian while the ONS uses an approximation that works only for a restricted class of functions. In this view, the ONS algorithm is more similar to Quasi-Newton methods. However, the best lens to understand the ONS is still through the generalized concept of strong convexity.*

Let's now see an example of functions that satisfy (7.10).

Example 7.31 (Exp-Concave Losses)**.** *Defining $X \subseteq \mathbb{R}^d$ convex, we say that a function $f : X \to \mathbb{R}$ is α-**exp-concave** if $\exp(-\alpha f(\boldsymbol{x}))$ is concave.*

Choose $\beta \le \frac{\alpha}{2}$ such that $|\beta\langle \boldsymbol{g}, \boldsymbol{y}-\boldsymbol{x}\rangle| \le \frac{1}{2}$ for all $\boldsymbol{x}, \boldsymbol{y} \in X$ and $\boldsymbol{g} \in \partial f(\boldsymbol{x})$. Note that we need a bounded domain for β to exist. Then, this class of functions satisfy the property (7.10)*. In fact, given that f is α-exp-concave then it is also 2β-exp-concave. Hence, from the definition we have*

$$\exp(-2\beta f(\boldsymbol{y})) \le \exp(-2\beta f(\boldsymbol{x})) - 2\beta \exp(-2\beta f(\boldsymbol{x}))\langle \boldsymbol{g}, \boldsymbol{y}-\boldsymbol{x}\rangle,$$

that is

$$\exp(-2\beta f(\boldsymbol{y}) + 2\beta f(\boldsymbol{x})) \le 1 - 2\beta\langle \boldsymbol{g}, \boldsymbol{y}-\boldsymbol{x}\rangle,$$

that implies

$$f(\boldsymbol{x}) - f(\boldsymbol{y}) \le \frac{1}{2\beta}\ln\left(1 + 2\beta\langle \boldsymbol{g}, \boldsymbol{x}-\boldsymbol{y}\rangle\right) \le \langle \boldsymbol{g}, \boldsymbol{x}-\boldsymbol{y}\rangle - \frac{\beta}{2}(\langle \boldsymbol{g}, \boldsymbol{y}-\boldsymbol{x}\rangle)^2,$$

where we used the elementary inequality $\ln(1+x) \le x - x^2/4$, for $|x| \le 1$.

Example 7.32. *Let $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| \le U\}$. The logistic loss of a linear predictor $\ell(\boldsymbol{x}) = \ln(1 + \exp(-\langle \boldsymbol{z}, \boldsymbol{x}\rangle))$, where $\|\boldsymbol{z}\|_2 \le 1$ is $\exp(-2U)/2$-exp-concave.*

7.11 Online Linear Regression: Vovk-Azoury-Warmuth Forecaster

Let's now consider the specific case that $\ell_t(\boldsymbol{x}) = \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle - y_t)^2$ and $V = \mathbb{R}^d$, that is the one of *unconstrained online linear regression with square loss*. These losses are not strongly convex w.r.t. $\boldsymbol{x}$, but they are exp-concave when the domain is bounded. We could use the ONS algorithm, but it would not work in the unbounded case. Another possibility would be to run FTRL, but that losses are not strongly convex and we would get only a $O(\sqrt{T})$ regret.

It turns out we can still get a logarithmic regret, if we make an additional assumption! We will assume to have access to $\boldsymbol{z}_t$ before predicting $\boldsymbol{x}_t$. Note that this is a mild assumptions in most of the interesting applications. Then, we will use the same strategy we used for the composite losses in Section 7.8, that is we put in the regularizer the future part of the loss we know we will receive, that is $\frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle)^2$. This has also another equivalent interpretation as running FTRL over the past losses plus the loss on the received $\boldsymbol{z}_t$ hallucinating a label of 0. We will call this algorithm Vovk-Azoury-Warmuth forecaster, from the names of the inventors. The details are in Algorithm 7.7.

Algorithm 7.7 Vovk-Azoury-Warmuth Forecaster

Require: $\lambda > 0$

for $t = 1$ **to** T **do**
  Receive $\boldsymbol{z}_t \in \mathbb{R}^d$
  Set $\boldsymbol{x}_t = \operatorname{argmin}_{\boldsymbol{x}} \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 + \frac{1}{2}\sum_{i=1}^{t-1}(\langle \boldsymbol{z}_i, \boldsymbol{x}\rangle - y_i)^2 + \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle)^2$
  Receive $y_t \in \mathbb{R}$ and pay $\ell(\boldsymbol{x}_t) = \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle - y_t)^2$
end for

As we did for composite losses, we look closely to the loss functions, to see if there are terms that we might move inside the regularizer. The motivation would be the same as in the composite losses case: the bound will depends only on the subgradients of the part of the losses that are outside of the regularizer.

So, observe that

$$\ell_t(\boldsymbol{x}) = \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle)^2 - y_t\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle + \frac{1}{2}y_t^2 \,.$$

From the above, we see that we could think to move the terms $\frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle)^2$ in the regularizer and leaving the linear terms in the loss: $\tilde{\ell}_t(\boldsymbol{x}) = -y_t\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle$. Hence, we will use

$$\psi_t(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top\left(\sum_{i=1}^{t}\boldsymbol{z}_i\boldsymbol{z}_i^\top\right)\boldsymbol{x} + \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 \,.$$

Note that the regularizer at time t contains the $\boldsymbol{z}_t$ that is revealed to the algorithm before it makes its prediction. For simplicity of notation, denote by $S_t = \lambda \boldsymbol{I}_d + \sum_{i=1}^{t}\boldsymbol{z}_i\boldsymbol{z}_i^\top$.

Using such procedure, the prediction can be written in a closed form:

$$\begin{aligned}\boldsymbol{x}_t &= \operatorname*{argmin}_{\boldsymbol{x}} \ \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 + \frac{1}{2}\sum_{i=1}^{t-1}(\langle \boldsymbol{z}_i, \boldsymbol{x}\rangle - y_i)^2 + \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle)^2 \\ &= \left(\lambda \boldsymbol{I}_d + \sum_{i=1}^{t} \boldsymbol{z}_i \boldsymbol{z}_i^\top\right)^{-1} \sum_{i=1}^{t-1} y_i \boldsymbol{z}_i \ .\end{aligned}$$

Using an incremental update of the inverse using the Sherman–Morrison formula, the computational complexity of this update is $O(d^2)$.

Hence, using the regret we proved for FTRL with strongly convex regularizers and $\psi_{T+1} = \psi_T$, we get the following guarantee

$$\begin{aligned}\sum_{t=1}^{T}\left(\tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_t(\boldsymbol{u})\right) &= \sum_{t=1}^{T}(-y_t\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle + y_t\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle) \\ &\le \psi_T(\boldsymbol{u}) - \min_{\boldsymbol{x}} \ \psi_T(\boldsymbol{x}) + \frac{1}{2}\sum_{t=1}^{T} y_t^2 \boldsymbol{z}_t^\top \boldsymbol{S}_t^{-1}\boldsymbol{z}_t + \sum_{t=1}^{T-1}(\psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1})) \\ &= \frac{1}{2}\boldsymbol{u}^\top S_T \boldsymbol{u} + \frac{1}{2}\sum_{t=1}^{T} y_t^2 \boldsymbol{z}_t^\top \boldsymbol{S}_t^{-1}\boldsymbol{z}_t - \frac{1}{2}\sum_{t=1}^{T-1}(\langle \boldsymbol{z}_{t+1}, \boldsymbol{x}_{t+1}\rangle)^2 \ .\end{aligned}$$

Noting that $\boldsymbol{x}_1 = \boldsymbol{0}$ and reordering the terms, we have

$$\begin{aligned}\sum_{t=1}^{T}\frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle - y_t)^2 - \sum_{t=1}^{T}\frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle - y_t)^2 &= \frac{1}{2}\sum_{t=1}^{T}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle)^2 + \sum_{t=1}^{T}(-y_t\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle + y_t\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle) - \frac{1}{2}\sum_{t=1}^{T}(\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle)^2 \\ &\le \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{1}{2}\sum_{t=1}^{T} y_t^2 \boldsymbol{z}_t^\top \boldsymbol{S}_t^{-1}\boldsymbol{z}_t \ .\end{aligned}$$

Remark 7.33. *Note that, differently from the ONS algorithm, the regularizers here are not proximal. Yet, we get $\boldsymbol{S}_t^{-1}$ in the bound because the current sample $\boldsymbol{z}_t$ is used in the regularizer. Without the knowledge of $\boldsymbol{z}_t$ the last term would be $\frac{1}{2}\sum_{t=1}^{T} y_t^2 \boldsymbol{z}_t^\top \boldsymbol{S}_{t-1}^{-1}\boldsymbol{z}_t$ that would have to be controlled using λ and a bound on the norms of $\boldsymbol{z}_t$.*

So, using again Lemma 7.29 and assuming $|y_t| \le Y, \ t = 1, \dots, T$, we have

$$\sum_{t=1}^{T}\frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle - y_t)^2 - \sum_{t=1}^{T}\frac{1}{2}\sum_{t=1}^{T}(\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle - y_t)^2 \le \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{Y^2}{2}\sum_{i=1}^{d}\ln\left(1 + \frac{\lambda_i}{\lambda}\right),$$

where $\lambda_1, \dots, \lambda_d$ are the eigenvalues of $\sum_{i=1}^{T} \boldsymbol{z}_i \boldsymbol{z}_i^\top$.

If we assume that $\|\boldsymbol{z}_t\|_2 \le R$, we can reason as we did for the similar term in ONS, to have

$$\sum_{i=1}^{d}\ln\left(1 + \frac{\lambda_i}{\lambda}\right) \le d\ln\left(1 + \frac{R^2 T}{\lambda d}\right) \ .$$

Putting all together, we have the following theorem.

Theorem 7.34. *Assume $\|\boldsymbol{z}_t\|_2 \le R$ and $|y_t| \le Y$ for $t = 1, \dots, T$. Then, using the prediction strategy in Algorithm 7.7, we have*

$$\sum_{t=1}^{T}\frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t\rangle - y_t)^2 - \sum_{t=1}^{T}\frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{u}\rangle - y_t)^2 \le \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{dY^2}{2}\ln\left(1 + \frac{R^2 T}{\lambda d}\right) \ .$$

Remark 7.35. *It is possible to show that the regret of the Vovk-Azoury-Warmuth forecaster is optimal up to multiplicative factors [Cesa-Bianchi and Lugosi, 2006, Theorem 11.9].*

7.12 Optimistic FTRL

In Section 6.9, we have seen that it is possible to achieve better regret guarantees through so-called Optimistic OMD, that uses a prediction of the next gradient. Here, we will extend the same idea to FTRL, where we predict the next loss function. If our predicted loss is correct, we can expect the regret to decrease. However, if our prediction is wrong we still want to recover the worst case guarantee. Such algorithm is called **Optimistic FTRL**.

Algorithm 7.8 Optimistic Follow-the-Regularized-Leader Algorithm

Require: A sequence of regularizers $\psi_1, \dots, \psi_T : X \to \mathbb{R}$, closed non-empty convex set $V \subseteq X \subseteq \mathbb{R}^d$

1: **for** $t = 1$ **to** T **do**
2: Predict next loss $\tilde{\ell}_t : V \to \mathbb{R}$
3: Output $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x} \in V} \ \psi_t(\boldsymbol{x}) + \tilde{\ell}_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \ell_i(\boldsymbol{x})$
4: Receive $\ell_t : V \to \mathbb{R}$ and pay $\ell_t(\boldsymbol{x}_t)$
5: **end for**

The core idea of Optimistic FTRL is to predict the next loss and use it in the update rule, as summarized in Algorithm 7.8. As in Optimistic OMD, it does not matter how the prediction is generated.

Let's see why this is a good idea. Remember that FTRL simply predicts with the minimizer of the previous losses plus a time-varying regularizer. Let's assume for a moment that instead we have the gift of predicting the future, so we do know the next loss ahead of time. Then, we could predict with its minimizer and suffer a negative regret. However, probably our foresight abilities are not so powerful, so our prediction of the next loss might be inaccurate. In this case, a better idea might be just to add our predicted loss to the previous ones and minimize the regularized sum. We would expect the regret guarantee to improve if our prediction of the future loss is precise. At the same time, if the prediction is wrong, we expect its influence to be limited, given that we use it together with all the past losses.

All these intuitions can be formalized in the following Theorem.

Theorem 7.36. *With the notation in Algorithm 7.8, let V be convex, closed, non-empty. Denote by $F_t(\boldsymbol{x}) = \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \ell_i(\boldsymbol{x})$. For $t = 1, \dots, T$, if F_t is closed, subdifferentiable, and strongly convex in V, then $\boldsymbol{x}_t$ exists and is unique. In addition, for $t = 1, \dots, T$, assume $\partial(\ell_t - \tilde{\ell}_t)(\boldsymbol{x}_t)$ to be non-empty and $F_t + \ell_t$ to be closed, subdifferentiable, and λ_t-strongly convex w.r.t. $\|\cdot\|$ in V. Then, we have*

$$\begin{aligned}
&\sum_{t=1}^{T} \ell(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \\
&\quad \leq \psi_{T+1}(\boldsymbol{u}) - \psi_1(\boldsymbol{x}_1) + \sum_{t=1}^{T} \left[\langle \hat{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}) \right] \\
&\quad \leq \psi_{T+1}(\boldsymbol{u}) - \psi_1(\boldsymbol{x}_1) + \sum_{t=1}^{T} \left[\frac{\|\hat{\boldsymbol{g}}_t\|_\star^2}{2\lambda_t} + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}) \right],
\end{aligned}$$

for all $\boldsymbol{u} \in V$ and all $\hat{\boldsymbol{g}}_t \in \partial(\ell_t - \tilde{\ell}_t)(\boldsymbol{x}_t)$ for $t = 1, \dots, T$,

Proof. We can interpret the Optimistic-FTRL as FTRL with a regularizer $\tilde{\psi}_t(\boldsymbol{x}) = \psi_t(\boldsymbol{x}) + \tilde{\ell}_t(\boldsymbol{x})$. So, as in Lemma 7.8, the existence and unicity of $\boldsymbol{x}_t$ is given by Theorem 6.8. Also, note that $\tilde{\ell}_{T+1}(\boldsymbol{x})$ has no influence on the regret of the algorithm over the T rounds, so we can set it to the null function.

Given that $\boldsymbol{u} \in V$, we can use Lemma 7.1 and discard the last term because negative, obtaining

$$\begin{aligned}
&\sum_{t=1}^{T} \ell(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \\
&\quad \leq \tilde{\ell}_{T+1}(\boldsymbol{u}) + \psi_{T+1}(\boldsymbol{u}) - \min_{\boldsymbol{x} \in V} \left(\tilde{\ell}_1(\boldsymbol{x}) + \psi_1(\boldsymbol{x})\right) + \sum_{t=1}^{T} [F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) + \tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_{t+1}(\boldsymbol{x}_{t+1})] \\
&\quad = \psi_{T+1}(\boldsymbol{u}) - \psi_1(\boldsymbol{x}_1) + \sum_{t=1}^{T} [F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t)],
\end{aligned}$$

where the terms $\tilde{\ell}_t(\boldsymbol{x}_t) - \tilde{\ell}_{t+1}(\boldsymbol{x}_{t+1})$ form a telescopic sum. Now focus on the terms $F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t)$. Observe that $F_t(\boldsymbol{x}) + \ell_t(\boldsymbol{x}) + i_V(\boldsymbol{x})$ is λ_t-strongly convex w.r.t. $\|\cdot\|$, hence from Lemma 4.2 we have

$$\begin{aligned}
F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) &= (F_t(\boldsymbol{x}_t) + \ell_t(\boldsymbol{x}_t)) - (F_t(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_{t+1})) + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}) \\
&\leq \langle \hat{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1}),
\end{aligned}$$

for all $\hat{\boldsymbol{g}}_t \in \partial(F_t + \ell_t + i_V)(\boldsymbol{x}_t)$. Observing that $\boldsymbol{x}_t = \operatorname{argmin}_{\boldsymbol{x} \in V} F_t(\boldsymbol{x}) + \tilde{\ell}_t(\boldsymbol{x})$, we have $\boldsymbol{0} \in \partial(F_t + \tilde{\ell}_t + i_V)(\boldsymbol{x}_t)$. Hence, using Theorem 2.18, $\partial(\ell_t - \tilde{\ell}_t)(\boldsymbol{x}_t) \subseteq \partial(F_t + \ell_t + i_V)(\boldsymbol{x}_t)$.

For the second inequality, by the definition of dual norms, we have that

$$\langle \hat{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \leq \|\hat{\boldsymbol{g}}_t\|_\star \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\| - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \leq \frac{1}{2\lambda_t} \|\hat{\boldsymbol{g}}_t\|_\star^2 . \qquad \square$$

Let's take a look at the second bound in the theorem. Compared to the similar bound for FTRL, we now have the terms $\|\hat{\boldsymbol{g}}_t\|_\star^2$ instead of the ones $\|\boldsymbol{g}_t\|_\star^2$. Consider the simple case that $\tilde{\ell}_t(\boldsymbol{x}) = \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x} \rangle$, then the bound depends on $\|\boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t\|_\star^2$. So, if the prediction of the subgradient of next loss is good, that term can become smaller and possibly even zero! On the other hand, if the predictions are bad, for Lipschitz losses we only lose a constant factor. Overall, in the best case we can gain a lot, in the worst case we do not lose that much.

It is worth mentioning that in the case of linear losses $\ell_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle$ and strongly convex regularizers, the update of optimistic FTRL becomes

$$\boldsymbol{x}_t = \nabla \psi_{V,t}^\star \left(-\tilde{\boldsymbol{g}}_t - \sum_{t=1}^{t-1} \boldsymbol{g}_i \right),$$

where $\psi_{V,t} = \psi_t + i_V$ and we used the hint $\tilde{\ell}_t(\boldsymbol{x}) = \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x} \rangle$.

Despite the simplicity of the algorithm and its analysis, there are many applications of this principle. Here, we will only describe a couple of them, while in Section 11.5 we describe the applications of optimistic algorithms for saddle point optimization.

7.12.1 Regret that Depends on the Variance of the Subgradients

Consider of running Optimistic-FTRL on the linearized losses $\ell_t(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle$. We can gain something out of the Optimistic-FTRL compared to plain FTRL if we are able to predict the next $\boldsymbol{g}_t$. A simple possibility is to predict the average of the past values, $\bar{\boldsymbol{g}}_t = \frac{1}{t-1} \sum_{i=1}^{t-1} \boldsymbol{g}_i$. Indeed, from the first chapter, we know that such strategy is itself an online learning procedure! In particular, it corresponds to a Follow-The-Leader algorithm on the losses $\ell_t(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{g}_t\|_2^2$. Hence, from the strong convexity of these losses and assuming $\|\boldsymbol{g}_t\|_2 \leq 1$, we know that

$$\sum_{t=1}^{T} \|\bar{\boldsymbol{x}}_t - \boldsymbol{g}_t\|_2^2 - \sum_{t=1}^{T} \|\boldsymbol{u} - \boldsymbol{g}_t\|_2^2 \leq 4(1 + \ln T), \ \forall \boldsymbol{u} \in \mathbb{R}^d .$$

This implies

$$\sum_{t=1}^{T} \|\bar{\boldsymbol{x}}_t - \boldsymbol{g}_t\|_2^2 \leq 4(1 + \ln T) + \min_{\boldsymbol{u}} \sum_{t=1}^{T} \|\boldsymbol{u} - \boldsymbol{g}_t\|_2^2 .$$

It is immediate to see that the minimizer is $\boldsymbol{u} = \frac{1}{T}\sum_{t=1}^T \boldsymbol{g}_t$, that results in T times the empirical variance of the subgradients. Plugging it in the Optimistic-FTRL regret, with $\psi_t = \psi$, we have

$$\sum_{t=1}^T \ell(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \leq \psi(\boldsymbol{u}) - \psi(\boldsymbol{x}_1) + \frac{4 + 4\ln T + \min_{\boldsymbol{g}} \sum_{t=1}^T \|\boldsymbol{g}_t - \boldsymbol{g}\|_2^2}{2\lambda} \,.$$

Remark 7.37. *Instead of using the mean of the past subgradients, we could use any other strategy or even a mix of different strategies. For example, assuming the subgradients bounded, we could use an algorithm to solve the Learning with Expert problem, where each expert is a strategy. Then, we would obtain a bound that depends on the predictions of the best strategy, plus the regret of the expert algorithm.*

7.12.2 Online Convex Optimization with Gradual Variations

In this section, we consider the case that the losses we receive have small variations over time. We will show that in this case it is possible to get constant regret in the case that the losses are equal.

In this case, the simple strategy we can use to predict the next subgradient is to use the previous one, that is $\tilde{\ell}_t(\boldsymbol{x}) = \langle \boldsymbol{g}_{t-1}, \boldsymbol{x}\rangle$ for $t \geq 2$ and $\tilde{\ell}_1(\boldsymbol{x}) = 0$.

Corollary 7.38. *Under the assumptions of Theorem 7.36, define $\tilde{\ell}_t(\boldsymbol{x}) = \langle \boldsymbol{g}_{t-1}, \boldsymbol{x}\rangle$ for $t \geq 2$ and $\tilde{\ell}_1(\boldsymbol{x}) = 0$. Set $\psi_t(\boldsymbol{x}) = \lambda_t \psi(\boldsymbol{x})$ where ψ is 1-strongly convex w.r.t. $\|\cdot\|$ and λ_t satisfies $\lambda_t\lambda_{t-1} \geq 8M^2$ for $t = 2, \dots, T$, where M is the smoothness constant of the losses ℓ_t. Then, $\forall \boldsymbol{u} \in V$, we have*

$$\mathrm{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u}\rangle \leq \lambda_{T+1}\psi(\boldsymbol{u}) - \lambda_1\psi(\boldsymbol{x}_1) + \frac{1}{\lambda_1}\|\nabla\ell_1(\boldsymbol{x}_1)\|_\star^2 + \sum_{t=2}^T \frac{2}{\lambda_t}\|\nabla\ell_t(\boldsymbol{x}_{t-1}) - \nabla\ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2 \,.$$

Moreover, assuming $\|\nabla\ell_t(\boldsymbol{x})\|_\star \leq L$ for all $\boldsymbol{x} \in V$, setting $\lambda_t = \sqrt{\max(8M^2, 4L^2) + \sum_{i=2}^{t-1}\|\nabla\ell_t(\boldsymbol{x}_{t-1}) - \nabla\ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2}$, we have

$$\mathrm{Regret}_T(\boldsymbol{u}) \leq \Big(\psi(\boldsymbol{u}) - \min_{\boldsymbol{x}} \psi(\boldsymbol{x}) + 4\Big)\sqrt{\max(8M^2, 4L^2) + \sum_{t=2}^T \|\nabla\ell_t(\boldsymbol{x}_{t-1}) - \nabla\ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2} + \frac{1}{4} \,.$$

Proof. From the Optimistic-FTRL bound with a fixed regularizer, we immediately get

$$\sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u}\rangle \leq \lambda_T\psi(\boldsymbol{u}) - \lambda_1\psi(\boldsymbol{x}_1) + \sum_{t=1}^T \left[\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - \frac{\lambda_t}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2\right] \,.$$

Now, consider the case that the losses ℓ_t are M-smooth. So, for any $\alpha_t > 0$, we have

$$\begin{aligned}\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - \frac{\lambda_t}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 &= \langle \nabla\ell_t(\boldsymbol{x}_t) - \nabla\ell_{t-1}(\boldsymbol{x}_{t-1}), \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - \frac{\lambda_t}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \\ &\leq \frac{\alpha_t}{2}\|\nabla\ell_t(\boldsymbol{x}_t) - \nabla\ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2 + \frac{1}{2\alpha_t}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 - \frac{\lambda_t}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \,.\end{aligned}$$

Focusing on the first term, for $t = 2, \dots, T$, we have

$$\begin{aligned}\frac{\alpha_t}{2}\|\nabla\ell_t(\boldsymbol{x}_t) - \nabla\ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2 &= \frac{\alpha_t}{2}\|\nabla\ell_t(\boldsymbol{x}_t) - \nabla\ell_{t-1}(\boldsymbol{x}_{t-1}) - \nabla\ell_t(\boldsymbol{x}_{t-1}) + \nabla\ell_t(\boldsymbol{x}_{t-1})\|_\star^2 \\ &\leq \alpha_t\|\nabla\ell_t(\boldsymbol{x}_t) - \nabla\ell_t(\boldsymbol{x}_{t-1})\|_\star^2 + \alpha_t\|\nabla\ell_t(\boldsymbol{x}_{t-1}) - \nabla\ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2 \\ &\leq \alpha_t M^2\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|^2 + \alpha_t\|\nabla\ell_t(\boldsymbol{x}_{t-1}) - \nabla\ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2 \,.\end{aligned}$$

Choose $\alpha_t = \frac{2}{\lambda_t}$. We have for $t = 2, \dots, T$

$$\begin{aligned}
&\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \\
&\quad \leq \frac{2M^2}{\lambda_t} \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|^2 + \frac{2}{\lambda_t} \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2 - \frac{\lambda_t}{4} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \,.
\end{aligned}$$

For $t = 1$, we have

$$\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \leq \frac{1}{\lambda_t} \|\nabla \ell_t(\boldsymbol{x}_t) - \tilde{\boldsymbol{g}}_t\|_\star^2 - \frac{\lambda_t}{4} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \,.$$

Now observe the assumption λ_t implies $\frac{2M^2}{\lambda_t} \leq \frac{\lambda_{t-1}}{4}$ for $t = 2, \dots, T$. So, summing for $t = 1, \dots, T$, we have

$$\sum_{t=1}^T \left(\langle \boldsymbol{g}_t - \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{x}_{t+1} \rangle - \frac{\lambda_t}{2} \|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 \right) \leq \frac{1}{\lambda_1} \|\nabla \ell_1(\boldsymbol{x}_1)\|_\star^2 + \sum_{t=2}^T \frac{2}{\lambda_t} \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2 \,.$$

Putting all together, we have the first stated bound.

The second one is obtained observing that

$$\begin{aligned}
\sum_{t=2}^T \frac{\|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2}{\lambda_t} &= \sum_{t=2}^T \frac{\|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2}{\sqrt{\max(8M^2, 4L^2) + \sum_{i=2}^{t-1} \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2}} \\
&\leq \sum_{t=2}^T \frac{\|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2}{\sqrt{\sum_{i=2}^{t} \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2}} \\
&\leq 2 \sqrt{\sum_{i=2}^T \|\nabla \ell_t(\boldsymbol{x}_{t-1}) - \nabla \ell_{t-1}(\boldsymbol{x}_{t-1})\|_\star^2} \,.
\end{aligned}$$

□

Note that if the losses are all the same, the regret becomes a constant! This is not surprising, because the prediction of the next loss is a linear approximation of the previous loss. Indeed, looking back at the proof, the key idea is to use the smoothness to argue that, if even the past subgradient was taken in a different point than the current one, it is still a good prediction of the current subgradient.

Remark 7.39. *Note that the assumption of smoothness is necessary. Indeed, passing always the same function and using the online-to-batch conversion would result in a convergence rate of $O(1/T)$ for a Lipschitz function, that is impossible by the lower bound in offline convex optimization.*

7.13 History Bits

The Follow-the-Regularized-Leader algorithm has a peculiar story. It was introduced by 4 different groups roughly around 1999-2006 following 4 different points of view. The algorithm was officially introduced in Abernethy et al. [2008], Hazan and Kale [2008] where at each step the prediction is computed as the minimizer of a regularization term plus the sum of losses on all past rounds. However, the key ideas were already presented years before. Gordon [1999a,b] appeared to have proposed it for the first time, naming it Maximum a Posteriori with a clear inspiration to Bayesian methods. Also, the analysis in Gordon [1999a,b] is the first one to use concepts from convex analysis. Later, Agarwal and Hazan [2005] proposed to add a regularizer to stabilize the Follow The Leader algorithm [Hannan, 1957] in an algorithm named Smooth Prediction for the specific application of Portfolio Selection. A little bit later, Shai Shalev-Shwartz and Yoram Singer proposed a very general analysis of FTRL using a dual perspective [Shalev-Shwartz and Singer, 2006, 2007b]. In particular, the PhD thesis of Shai Shalev-Shwartz [Shalev-Shwartz, 2007] contains the most precise dual analysis of FTRL, that also allows multiple dual updates and regularizers with a time-varying weight.

However, he called it "online mirror descent" because the name FTRL was only invented later! (I also contributed to the confusion naming a general analysis of FTRL with time-varying regularizer and linear losses "generalized online mirror descent" [Orabona et al., 2015]. So, now I am trying to set the record straight :)) The optimization community also knows FTRL, but only with linear losses, and they call it Dual Averaging [Nesterov, 2009]. Note that the paper Nesterov [2009] is actually from 2005[1], and in the paper Nesterov claims that these ideas are from 2001-2002, but he decided not to publish them for a while because he was convinced that "the importance of black-box approach in Convex Optimization will be irreversibly vanishing, and, finally, this approach will be completely replaced by other ones based on a clever use of problem's structure (interior-point methods, smoothing, etc.)." As Shai Shalev-Shwartz, Nesterov looks at FTRL from the dual point of view but he focuses only on linear losses and only in the offline case.

Nesterov's Dual Averaging was then extended to deal with composite losses by Xiao [2009, 2010] in the stochastic and online learning setting, essentially rediscovering the framework of Shalev-Shwartz, and calling the resulting framework Regularized Dual Averaging. Finally, McMahan [2017] gives the elegant equality result that I presented here (with minor improvements) that holds for general loss functions and regularizers. A similar equality, setting $\boldsymbol{u} = \boldsymbol{x}_{T+1}$ and specialized only to linear losses, was also proven in Abernethy et al. [2014], essentially matching the inequality in Orabona et al. [2015]. Note that Dual Averaging stems from the dual interpretation of FTRL for linear losses, but Lemma 7.1 underlines the fact that FTRL is actually more general. People used to prove the regret bound for FTRL using the so-called be-the-leader-follow-the-leader lemma (see Lemma 1.2 and Exercise 7.3). However, the proof is off by a factor of 2 in the case of fixed regularizers [McMahan, 2017]. Moreover, it seems to fail in the case of generic strongly convex increasing regularizers, while it works for the particular case of proximal regularizers McMahan [2017].

Another source of confusion stems from the fact that some people differentiate among a "lazy" and "greedy" version of OMD. In reality, as proved in McMahan [2017], the lazy algorithm is just FTRL with linearized losses and the greedy one is just OMD. The notation "lazy online mirror descent" was introduced in Zinkevich [2004], where he basically introduced for the first time FTRL with linearized losses in the special case of $\psi_t(\boldsymbol{x}) = \frac{\eta}{2}\|\boldsymbol{x}\|_2^2$.

The second result in Lemma 7.14 is a distillation of the reasoning used in Zimmert and Seldin [2021], but without using duality concepts.

The use of FTRL with the regularizer in (7.5) was proposed in Orabona and Pál [2015, 2018], I presented a simpler version of their proof that does not require Fenchel conjugates. The AdaHedge algorithm was introduced in van Erven et al. [2011] and refined in de Rooij et al. [2014]. The analysis reported here is from Orabona and Pál [2015, 2018], that generalized AdaHedge to arbitrary regularizers in AdaFTRL. Additional properties of AdaHedge for the stochastic case were proven in de Rooij et al. [2014].

The first implicit use of the $(2, 1)$ group norm I could find is in Bakin [1999], where it is used as a constrain in the first proposed formulation of Group Lasso. However, there is no mention of the fact that this constraint is a norm. Ding et al. [2006] propose the use of the $(2, 1)$ group norm for Principal Component Analysis, calling it R_1 norm. Few months later, the same group norm was proposed independently in Argyriou et al. [2006]. Agarwal et al. [2008] extended the $(2, 1)$ group norm to (p, s) and used it in OMD for multitask online learning. Kakade et al. [2009] proved the smoothness and strong convexity of squared group norms. Jie et al. [2010] used the $(2, s)$ group norm FTRL for online multi-kernel learning, while Orabona et al. [2010, 2012b] used the same norm in a Perceptron-like algorithm.

The first analysis of FTRL with composite losses is in Xiao [2009, 2010]. The analysis presented here using the negative terms $\psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1})$ to easily prove regret bounds for FTRL for composite losses is from Orabona et al. [2015].

The first proof of FTRL for strongly convex losses was in Shalev-Shwartz and Singer [2007a] (even if they do not call it FTRL). In the same paper, they also define strong convexity with respect to Bregman divergences, rather than just norms, and prove the same logarithmic regret bound (see Exercise 6.7). Analogously, people later defined the concept of smoothness with respect to a Bregman divergence [Birnbaum et al., 2010, 2011], rediscovered in Bauschke et al. [2017]. Later, both concepts were rebranded as "relative smoothness" and "relative strong convexity" [Lu et al., 2018].

There is an interesting bit about FTRL-Proximal [McMahan, 2011]: FTRL-Proximal is an instantiation of FTRL that uses a particular proximal regularizer. It became very famous in internet companies when Google disclosed in a very influential paper that they were using FTRL-Proximal to train the classifier for click prediction [McMahan

[1] https://papers.ssrn.com/sol3/papers.cfm?abstract_id=912637

et al., 2013]. This generated even more confusion because many people started conflating the term FTRL-Proximal (a specific algorithm) with FTRL (an entire family of algorithms).

The Online Newton Step algorithm was introduced in Hazan et al. [2006] and it is described to the particular case that the loss functions are exp-concave. Exp-concave functions were studied in Kivinen and Warmuth [1999]. Here, I described a slight generalization for any sequence of functions that satisfy (7.10), that in my view better shows the parallel between FTRL over strongly convex functions and ONS. Note that Hazan et al. [2006] also describes a variant of ONS based on OMD, but I find its analysis less interesting from a didactic point of view. The proof presented here through the properties of proximal regularizers might be new, I am not sure. Lemma 7.29 dates back to Lai and Wei [1982, Lemma 2]. Orabona et al. [2012a] obtained a logarithmic L^* bound, i.e., proportional to $\ln(1 + \sum_{t=1}^T \ell_t(\boldsymbol{u}))$, for exp-concave and smooth losses using ONS.

The Vovk-Azoury-Warmuth forecaster was introduced independently by Azoury and Warmuth [2001] and Vovk [2001], and its interpretation as hallucinating the future loss is by Azoury and Warmuth [2001]. The proof presented here is from Orabona et al. [2015].

The Optimistic FTRL version was proposed in Rakhlin and Sridharan [2013b] but analyzed only with fixed self-concordant regularizers. Steinhardt and Liang [2014] proposed optimistic FTRL as we know it, even if it was called "Optimistic Mirror Descent" for the misnaming problem we have explained. The proof of Theorem 7.36 I present here is new. A better bound that shows the robustness of Optimistic FTRL to "bad" hints was proven in Flaspohler et al. [2021], that (implicitly) uses the degree of freedom of the FTRL proof underlined in Remark 7.5.

Corollary 7.38 was proved by Chiang et al. [2012] for Optimistic OMD and presented in a similar form for Optimistic FTRL in Joulani et al. [2017], but only for bounded domains.

7.14 Exercises

Problem 7.1. *Prove that the update of FTRL with linearized loss in Example 7.11 is correct.*

Problem 7.2. *Consider FTRL with linearized losses with regularizers $\psi_t(\boldsymbol{x})$. The regret equality in Lemma 7.1 has a term equal to $F_{T+1}(\boldsymbol{x}_{T+1}) - F_{T+1}(\boldsymbol{u})$. Assume that V convex and non-empty, ψ_{T+1} is strictly convex and differentiable in $\boldsymbol{x}_{T+1}$. Then, show that $F_{T+1}(\boldsymbol{x}_{T+1}) - F_{T+1}(\boldsymbol{u}) \leq -B_{\psi_{T+1}}(\boldsymbol{u}; \boldsymbol{x}_{T+1})$ for any $\boldsymbol{u} \in V$.*

Problem 7.3. *Consider FTRL with losses $\ell_t(\boldsymbol{x})$ and regularizers $\psi_t(\boldsymbol{x})$. Assume that $\psi_{t+1} \geq \psi_t$ for $t = 1, \dots, T-1$. Denote by $F_t(\boldsymbol{x}) := \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t-1} \ell_t(\boldsymbol{x})$ and show that*

$$F_t(\boldsymbol{x}_t) - F_{t+1}(\boldsymbol{x}_{t+1}) + \ell_t(\boldsymbol{x}_t) \leq \ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{x}_{t+1}) \,.$$

Problem 7.4. *We can obtain $L^\star$ bounds for smooth losses with linearized FTRL with the additional assumption that the losses are Lipschitz, even without knowing the Lipschitz constant. In particular, show that under these assumptions the regularizers $\psi_t(\boldsymbol{x}) = \frac{\lambda_t}{2}\|\boldsymbol{x}\|_2^2$, where $\lambda_t = \sqrt{a + \sum_{i=1}^{t-1} \|\boldsymbol{g}_i\|_2^2}$ and $a > 0$, would give an $L^\star$ bound. Hint: Use Gaillard et al. [2014, Lemma 14].*

Problem 7.5. *Define $\lambda_t = a + \sum_{i=1}^{t-1} \frac{b_i}{\lambda_i}$, where $a > 0$. Let $b_t \in [0, m]$ and prove that $\sum_{t=1}^T \frac{b_t}{\lambda_t} \leq \sqrt{2 \sum_{t=1}^T b_t} + \frac{m}{a}$. Use this inequality to design an alternative regularizer that solves the previous exercise. Hint: see Sachs et al. [2022, Proof of Theorem 5].*

Problem 7.6. *Let $V = \mathbb{R}^d$ and prove that the update in (7.8) is equivalent to the one of OSD with learning rate $\eta_t = \frac{1}{\sum_{i=1}^t \mu_t}$, when both algorithms start with $\boldsymbol{x}_1 = 0$.*

Problem 7.7. *Prove the statement in Example 7.32.*

Problem 7.8. *Design and analyze the FTRL version of OMD with p-norms in Section 6.7.*

Problem 7.9. *Consider the learning with expert setting and assume $\|\boldsymbol{g}_t\|_\infty \leq 1$ for all $t = 1, \dots T$. From the analysis of EG with local norms, see (7.4), derive a time-varying regularizer that gives a regret upper bound proportional to $\ln d + \sqrt{L^\star(\boldsymbol{u}) \ln d}$, where $L^\star(\boldsymbol{u}) = \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle$ for any $\boldsymbol{u} \in \Delta^{d-1}$.*

Problem 7.10. *Prove that the losses* $\ell_t(\boldsymbol{x}) = \frac{1}{2}(\langle \boldsymbol{z}_t, \boldsymbol{x}\rangle - y_t)^2$*, where* $\|\boldsymbol{z}_t\|_2 \le 1$*,* $|y_t| \le 1$*, and* $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 \le 1\}$*, are exp-concave and find the exp-concavity constant.*

Problem 7.11. *Prove that ONS update is equivalent to the following two steps:*

$$\tilde{\boldsymbol{x}}_t = S_{t-1}^{-1}\left(\sum_{i=1}^{t-1} A_i \boldsymbol{x}_i - \sum_{i=1}^{t-1} \boldsymbol{g}_i\right)$$
$$\boldsymbol{x}_t = \operatorname*{argmin}_{\boldsymbol{x}\in V} \|\boldsymbol{x} - \tilde{\boldsymbol{x}}_t\|_{S_{t-1}},$$

where $A_t = \sum_{i=1}^{t} \boldsymbol{g}_i \boldsymbol{g}_i^\top$ *and* $S_t = \lambda I + A_t$*.*

Chapter 8

Online Linear Classification

In this chapter, we will consider the problem of *online linear classification*. We consider the following setting:

- At each time step we receive a sample $\boldsymbol{z}_t$
- We output a prediction of the binary label $y_t \in \{-1, 1\}$ of $\boldsymbol{z}_t$
- We receive the true label and we see if we did a mistake or not
- We update our online classifier

The aim of the online algorithm is to minimize the number of mistakes it does compared to some best fixed classifier.

We will focus on linear classifiers, that predicts with the sign of the inner product between a vector $\boldsymbol{x}_t$ and the input features $\boldsymbol{z}_t$. Hence, $\tilde{y}_t = \operatorname{sign}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t \rangle)$. This problem can be written again as a regret minimization problem:

$$\operatorname{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}),$$

where $\ell_t(\boldsymbol{x}) = \mathbf{1}[\operatorname{sign}(\langle \boldsymbol{z}_t, \boldsymbol{x} \rangle) \neq y_t]$. It should be clear that these losses are non-convex. Hence, we need an alternative way to deal with them. In the following, we will see two possible approaches to this problem.

8.1 Online Randomized Classifier

As we did for the Learning with Expert Advice framework, we might think to convexify the losses using randomization. Hence, on each round we can predict a number in $q_t \in [-1, 1]$ and output the label $+1$ according with probability $\frac{1+q_t}{2}$ and the label -1 with probability $\frac{1-q_t}{2}$. So, define the random variable

$$\tilde{y}(p) = \begin{cases} +1, & \text{with probability } p, \\ -1, & \text{with probability } 1-p\,. \end{cases}$$

Now observe that $\mathbb{E}[\tilde{y}(\frac{1+q_t}{2}) \neq y_t] = \frac{1}{2}|q_t - y_t|$. If we consider linear predictors, we can think to have $q_t = \langle \boldsymbol{z}_t, \boldsymbol{x}_t \rangle$ and similarly for the competitor $q_t' = \langle \boldsymbol{z}_t, \boldsymbol{u} \rangle$. Constraining both the algorithm and the competitor to the space of vectors where $|\langle \boldsymbol{z}_t, \boldsymbol{x} \rangle| \leq 1$ for $t = 1, \dots, T$, we can write

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbf{1}\left[\tilde{y}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t \rangle) \neq y_t\right] - \sum_{t=1}^{T} \mathbf{1}\left[\tilde{y}(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle) \neq y_t\right]\right] = \mathbb{E}\left[\sum_{t=1}^{T} \frac{1}{2}\left|\langle \boldsymbol{z}_t, \boldsymbol{x}_t \rangle - y_t\right| - \sum_{t=1}^{T} \frac{1}{2}\left|\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle - y_t\right|\right].$$

Hence, the surrogate convex loss becomes $\tilde{\ell}_t(\boldsymbol{x}) = \frac{1}{2}|\langle \boldsymbol{z}_t, \boldsymbol{x} \rangle - y_t|$ and the feasible set is any convex set where we have the property $|\langle \boldsymbol{z}_t, \boldsymbol{x} \rangle| \leq 1$ for $t = 1, \dots, T$.

Given that this problem is convex, assuming $\boldsymbol{z}_t$ to be bounded w.r.t. some norm, we can use almost any of the algorithms we have seen till now, from Online Mirror Descent to Follow-The-Regularized-Leader. All of them would result in $O(\sqrt{T})$ regret upper bounds, assuming that $\boldsymbol{z}_t$ are bounded in some norm. The only caveat is to restrict $\langle \boldsymbol{z}_t, \boldsymbol{x}_t \rangle$ in $[-1, 1]$. One way to do it might be to consider assuming $\|\boldsymbol{z}_t\|_\star \leq R$ and choose the feasible set $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\| \leq \frac{1}{R}\}$.

Putting all together, for example, we can have the following strategy using FTRL with regularizers $\psi_t(\boldsymbol{x}) = \frac{\sqrt{2t}}{4}\|\boldsymbol{x}\|_2^2$.

Algorithm 8.1 Randomized Online Linear Classifier through FTRL

Require: $R > 0$ such that $\|\boldsymbol{z}_t\|_2 \leq R$ for $t = 1, \dots, T$
1: Set $\boldsymbol{\theta}_1 = \mathbf{0} \in \mathbb{R}^d$
2: **for** $t = 1$ **to** T **do**
3: $\quad \boldsymbol{x}_t = \eta_t \theta_t \min(\frac{1}{R\eta_t \|\theta_t\|_2}, 1)$
4: $\quad$ Receive $\boldsymbol{z}_t \in \mathbb{R}^d$
5: $\quad$ Predict $\tilde{y}_t = \begin{cases} 1, & \text{with probability } \frac{\langle \boldsymbol{z}_t, \boldsymbol{x}_t \rangle + 1}{2} \\ -1, & \text{otherwise} \end{cases}$
6: $\quad$ Receive y_t and pay $\mathbf{1}[y_t \neq \tilde{y}_t]$
7: $\quad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \frac{1}{2}\text{sign}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t \rangle - y_t)\boldsymbol{z}_t$ where $\text{sign}(0) := 0$
8: **end for**

Theorem 8.1. *Let $(\boldsymbol{z}_t, y_t)_{t=1}^T$ an arbitrary sequence of samples/labels couples where $(\boldsymbol{z}_t, y_t) \in X \times \{-1, 1\}$ and $X \subset \mathbb{R}^d$. Assume $\|\boldsymbol{z}_t\|_2 \leq R$, for $t = 1, \dots, T$. Then, running the Randomized Online Linear Classifier algorithm with $\eta_t = \frac{\sqrt{2}}{\sqrt{t}}$ where $\alpha > 0$, for any $\boldsymbol{u} \in \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 \leq \frac{1}{R}\}$ we have the following guarantee*

$$\mathbb{E}\left[\sum_{t=1}^T \mathbf{1}[\tilde{y}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t \rangle) \neq y_t]\right] - \mathbb{E}\left[\sum_{t=1}^T \mathbf{1}[\tilde{y}(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle) \neq y_t]\right] \leq \sqrt{2T}\,.$$

Proof. The proof is straightforward from the FTRL regret bound with the chosen increasing regularizer. □

8.2 The Perceptron Algorithm

The above strategy has the shortcoming of restricting the feasible vectors in a possibly very small set. In turn, this could make the performance of the competitor low. In turn, the performance of the online algorithm is only close to the one of the competitor.

Another way to deal with the non-convexity is to compare the number of mistakes that the algorithm does with a convex cumulative loss of the competitor. That is, we can try to prove a weaker regret guarantee:

$$\sum_{t=1}^T \mathbf{1}[y_t \neq \tilde{y}_t] - \sum_{t=1}^T \ell(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t) = O\left(\sqrt{T}\right)\,. \tag{8.1}$$

In particular, the convex loss we consider is *powers* of the **Hinge Loss**: $\ell_q(\tilde{y}, y) = \max(1 - \tilde{y}y, 0)^q$. The hinge loss is a convex upper bound to the 0/1 loss and it achieves the value of zero when the sign of the prediction is correct *and* the magnitude of the inner product is big enough. Moreover, taking powers of it, we get a family of functions that trade-offs the loss for the wrongly classified samples with the one for the correctly classified samples but with a value of $|\langle \boldsymbol{z}_t, \boldsymbol{x} \rangle| \leq 1$, see Figure 8.1.

The oldest algorithm we have to minimize the modified regret in (8.1) is the **Perceptron** algorithm, in Algorithm 8.2.

The Perceptron algorithm updates the current prediction $\boldsymbol{x}_t$ moving in the direction of the current sample multiplied by its label. Let's see why this is a good idea. Assume that $y_t = 1$ and the algorithm made a mistake. Then, the updated

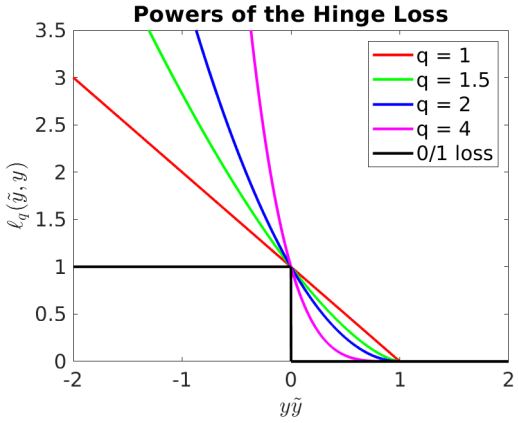

Figure 8.1: Powers of the Hinge Loss.

Algorithm 8.2 Perceptron Algorithm

Set $\boldsymbol{x}_1 = \mathbf{0} \in \mathbb{R}^d$
for $t = 1$ **to** T **do**
  Receive $\boldsymbol{z}_t \in \mathbb{R}^d$
  Predict $\tilde{y}_t = \text{sign}(\langle \boldsymbol{z}_t, \boldsymbol{x}_t \rangle)$
  Receive y_t and pay $\mathbf{1}[y_t \neq \tilde{y}_t]$
  $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \mathbf{1}[y_t \neq \tilde{y}_t] y_t \boldsymbol{z}_t$
end for

prediction $\boldsymbol{x}_{t+1}$ would predict a more positive number on the same sample $\boldsymbol{z}_t$. In fact, we have

$$\langle \boldsymbol{z}_t, \boldsymbol{x}_{t+1} \rangle = \langle \boldsymbol{z}_t, \boldsymbol{x}_t + y_t \boldsymbol{z}_t \rangle = \langle \boldsymbol{z}_t, \boldsymbol{x}_t \rangle + \|\boldsymbol{z}_t\|_2^2 \, .$$

In the same way, if $y_t = -1$ and the algorithm made a mistake, the update would result in a more negative prediction on the same sample.

For the Perceptron algorithm we can prove the following guarantee.

Theorem 8.2. *Let $(\boldsymbol{z}_t, y_t)_{t=1}^T$ an arbitrary sequence of samples/labels couples where $(\boldsymbol{z}_t, y_t) \in X \times \{-1, 1\}$ and $X \subset \mathbb{R}^d$. Assume $\|\boldsymbol{z}_t\|_2 \leq R$, for $t = 1, \ldots, T$. Then, running the Perceptron algorithm we have the following guarantee*

$$\sum_{t=1}^T \mathbf{1}[y_t \neq \tilde{y}_t] - \sum_{t=1}^T \ell_q(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t) \leq \frac{q^2 R^2 \|\boldsymbol{u}\|_2^2}{2} + qR\|\boldsymbol{u}\|_2 \sqrt{\frac{q^2 R^2 \|\boldsymbol{u}\|_2^2}{4} + \sum_{t=1}^T \ell_q(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t)}, \ \forall \boldsymbol{u} \in \mathbb{R}^d, q \geq 1 \, .$$

Before proving the theorem, let's take a look to its meaning. If there exists a $\boldsymbol{u} \in \mathbb{R}^d$ such that $\sum_{t=1}^T \ell_q(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t) = 0$, then the Perceptron algorithm makes a *finite* number of mistakes upper bounded by $R^2 \|\boldsymbol{u}\|_2^2$. In case that are many $\boldsymbol{u}$ that achieves $\sum_{t=1}^T \ell_q(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t) = 0$ we have that the finite number of mistakes is bounded the norm of the smallest $\boldsymbol{u}$ among them. What is the meaning of this quantity?

Remember that a hyperplane represented by its normal vector $\boldsymbol{u}$ divides the space in two half spaces: one with the points $\boldsymbol{z}$ that give a positive value for the inner product $\langle \boldsymbol{z}, \boldsymbol{u} \rangle$ and other one where the same inner product is negative. Now, we have that the distance of a sample $\boldsymbol{z}_t$ from the hyperplane whose normal is $\boldsymbol{u}$ is

$$\frac{|\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle|}{\|\boldsymbol{u}\|_2} \, .$$

Also, given that we are considering a $\boldsymbol{u}$ that gives cumulative hinge loss zero, we have that that quantity is at least $\frac{1}{\|\boldsymbol{u}\|_2}$. So, *the norm of the minimal $\boldsymbol{u}$ that has cumulative hinge loss equal to zero is inversely proportional to the*

minimum distance between the points and the separating hyperplane. This distance is called the **margin** of the samples $(\boldsymbol{z}_t, y_t)_{t=1}^T$. So, if the margin is small, the Perceptron algorithm can do more mistakes than when the margin is big.

If the problem cannot be linearly separable, the Perceptron satisfies a regret of $O(\sqrt{L^\star})$, where $L^\star$ is the loss of the competitor. Moreover, we measure the competitor with a *family of loss functions* and compete with the best $\boldsymbol{u}$ measured with the best loss. This adaptivity is achieved through two basic ingredients:

- *The Perceptron is independent of scaling of the update by a hypothetical learning rate* η, in the sense that the mistakes it does are independent of the scaling. That is, we could update with $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \eta \mathbf{1}[y_t \neq \tilde{y}_t] y_t \boldsymbol{z}_t$ and have the same mistakes and updates because they only depend on the sign of $\tilde{y}_t$. Hence, we can think as it is always using the best possible learning rate η.

- The weakened definition of regret allows to consider a family of loss functions, because *the Perceptron is not using any of them in the update.* In this sense, it is worth stressing that the Perceptron algorithm is *not* online subgradient descent on some sequence of loss functions.

Let's now prove the regret guarantee. For the proof, we will need the two following technical lemmas.

Lemma 8.3. *[Cucker and Zhou, 2007, Lemma 10.17] Let* $p, q > 1$ *be such that* $\frac{1}{p} + \frac{1}{q} = 1$. *Then*

$$ab \leq \frac{1}{q} a^q + \frac{1}{p} b^p, \ \forall a, b \, .$$

Lemma 8.4. *Let* $a, c > 0$, $b \geq 0$, *and* $x \geq 0$ *such that* $x - a\sqrt{x} \leq c$. *Then,* $x \leq c + \frac{a^2}{2} + a\sqrt{\frac{a^2}{4} + c}$.

Proof. Let $y = \sqrt{x}$, then we have $y^2 - ay - c \leq 0$. Solving for y we have $y \leq \frac{a + \sqrt{a^2 + 4c}}{2}$. Hence, $x = y^2 \leq \frac{a^2 + 2a\sqrt{a^2+4c} + a^2 + 4c}{4} = \frac{a^2}{2} + \frac{a}{2}\sqrt{a^2 + 4c} + c$. □

Proof of Theorem 8.2. Denote by the total number of the mistakes of the Perceptron algorithm by $M = \sum_{t=1}^T \mathbf{1}[y_t \neq \tilde{y}_t]$.

First, note that the Perceptron algorithm can be thought as running Online Subgradient Descent (OSD) with a fixed stepsize η over the losses $\tilde{\ell}_t(\boldsymbol{x}) = -\langle \mathbf{1}[y_t \neq \tilde{y}_t] y_t \boldsymbol{z}_t, \boldsymbol{x} \rangle$ over $V = \mathbb{R}^d$. Indeed, OSD over such losses would update

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \eta \mathbf{1}[y_t \neq \tilde{y}_t] y_t \boldsymbol{z}_t \, . \tag{8.2}$$

Now, as said above, η does not affect in any way the sign of the predictions, hence the Perceptron algorithm could be run with (8.2) and its predictions would be exactly the same. Hence, we have

$$\sum_{t=1}^T -\langle \mathbf{1}[y_t \neq \tilde{y}_t] y_t \boldsymbol{z}_t, \boldsymbol{x}_t \rangle + \sum_{t=1}^T \langle \mathbf{1}[y_t \neq \tilde{y}_t] y_t \boldsymbol{z}_t, \boldsymbol{u} \rangle \leq \frac{\|\boldsymbol{u}\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^T \mathbf{1}[y_t \neq \tilde{y}_t] \|\boldsymbol{z}_t\|_2^2, \ \forall \eta > 0 \, .$$

Given that this inequality holds for any η, we can choose the ones that minimizes the r.h.s., to have

$$\sum_{t=1}^T -\langle \mathbf{1}[y_t \neq \tilde{y}_t] y_t \boldsymbol{z}_t, \boldsymbol{x}_t \rangle + \sum_{t=1}^T \langle \mathbf{1}[y_t \neq \tilde{y}_t] y_t \boldsymbol{z}_t, \boldsymbol{u} \rangle \leq \|\boldsymbol{u}\| \sqrt{\sum_{t=1}^T \mathbf{1}[y_t \neq \tilde{y}_t] \|\boldsymbol{z}_t\|_2^2} \leq \|\boldsymbol{u}\| R \sqrt{\sum_{t=1}^T \mathbf{1}[y_t \neq \tilde{y}_t]} \, . \tag{8.3}$$

Note that $-\mathbf{1}[y_t \neq \tilde{y}_t] y_t \langle \boldsymbol{z}_t, \boldsymbol{x}_t \rangle \geq 0$. Also, we have

$$\langle y_t \boldsymbol{z}_t, \boldsymbol{u} \rangle = 1 - (1 - \langle y_t \boldsymbol{z}_t, \boldsymbol{u} \rangle) \geq 1 - \max(1 - \langle y_t \boldsymbol{z}_t, \boldsymbol{u} \rangle, 0) = 1 - \ell(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t) \, .$$

So, denoting by $\tau_t = \mathbf{1}[y_t \neq \tilde{y}_t]$, we can rewrite (8.3) as

$$\begin{aligned}\sum_{t=1}^{T} \tau_t &\leq \|\boldsymbol{u}\| R \sqrt{\sum_{t=1}^{T} \tau_t} + \sum_{t=1}^{T} \tau_t \ell(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t) \\ &\leq \|\boldsymbol{u}\| R \sqrt{\sum_{t=1}^{T} \tau_t} + \left(\sum_{t=1}^{T} \tau_t^p \right)^{1/p} \left(\sum_{t=1}^{T} \ell(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t)^q \right)^{1/q} \\ &= \|\boldsymbol{u}\| R \sqrt{\sum_{t=1}^{T} \tau_t} + \left(\sum_{t=1}^{T} \tau_t \right)^{1/p} \left(\sum_{t=1}^{T} \ell(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t)^q \right)^{1/q},\end{aligned}$$

where we used Holder's inequality and $\frac{1}{p} + \frac{1}{q} = 1$.

Given that $M = \sum_{t=1}^{T} \tau_t$ and denoting by $L_q = \sum_{t=1}^{T} \ell^q(\langle \boldsymbol{z}_t, \boldsymbol{u} \rangle, y_t)$, we have

$$M \leq \|\boldsymbol{u}\| R \sqrt{M} + M^{1/p} L_q^{1/q} .$$

Let's now consider two cases. For $q = 1$, we can use Lemma 8.4 and have the stated bound. Instead, for $q > 1$, using Lemma 8.3 we have

$$M \leq \|\boldsymbol{u}\| R \sqrt{M} + M^{1/p} L_q^{1/q} \leq \|\boldsymbol{u}\| R \sqrt{M} + \frac{1}{p} M + \frac{1}{q} L_q,$$

that implies

$$M \left(1 - \frac{1}{p} \right) \leq \|\boldsymbol{u}\| R \sqrt{M} + \frac{1}{q} L_q .$$

Using the fact that $1 - \frac{1}{p} = \frac{1}{q}$, we have

$$M \leq q \|\boldsymbol{u}\| R \sqrt{M} + L_q .$$

Finally, using Lemma 8.4, we have the stated bound. □

8.3 History Bits

The Perceptron was proposed by Rosenblatt [1958]. To be more precise, he introduced a *family* of algorithms characterized by a certain architecture. Also, he considered what we call now supervised and unsupervised training procedures. The particular class of Perceptron we use nowadays and I described were called α-system [Block, 1962]. I hypothesize that the fact the α-system survived the test of time is exactly due to the simple convergence proof in Block [1962] and Novikoff [1963]. Both proofs are non-stochastic. For the sake of proper credits assignment, it seems that the convergence proof of the Perceptron was proved by many others before Block and Novikoff [see references in Novikoff, 1963]. However, the proof in Novikoff [1963] seems to be the cleanest one. Aizerman et al. [1964] (essentially) described for the first time the Kernel Perceptron and proved a finite mistake bound for it.

The proof of convergence in the non-separable case for $q = 1$ is by Gentile and Littlestone [1999], Gentile [2003] and for $q = 2$ is from Freund and Schapire [1998, 1999a]. The proof presented here is based on the one in Beygelzimer et al. [2017].

8.4 Exercises

Problem 8.1. *in the Perceptron mistake upper bound we let the competitor have any norm and we measure the loss of the competitor with the hinge loss* $\ell^{hinge}(\hat{y}, y) = \max(\gamma - \hat{y}y, 0)$*, where* $\hat{y} \in \mathbb{R}$ *and* $y \in \{-1, 1\}$*. Instead, we can equivalently constrain the competitor to have norm equal to 1 and measure the loss with the* ***hinge loss at margin gamma****:* $\ell_\gamma^{hinge}(\hat{y}, y) = \frac{1}{\gamma} \max(\gamma - \hat{y}y, 0)$*. Prove that* $\ell_\gamma^{hinge}(\hat{y}, y)$ *is still an upper bound on the zero-one loss over the prediction given by* $\operatorname{sign}(\hat{y})$*.*

Chapter 9

Parameter-free Online Linear Optimization

In the previous sections, we have shown that Online Mirror Descent (OMD) and Follow-The-Regularized-Leader (FTRL) achieves a regret of $O(\sqrt{T})$ for convex Lipschitz losses. We have also shown that for bounded domains these bounds are optimal up to constant multiplicative factors. However, in the unbounded case the bounds we get are suboptimal w.r.t. the dependency on the competitor. More in particular, let's consider an example with Online Subgradient Descent with $V = \mathbb{R}^d$ over 1-Lipschitz losses and learning rate $\eta = \frac{\alpha}{\sqrt{T}}$. We get the following regret guarantee

$$\text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \leq \frac{\|\boldsymbol{u}\|_2^2}{2\eta} + \frac{\eta T}{2} = \frac{1}{2}\sqrt{T}\left(\frac{\|\boldsymbol{u}\|_2^2}{\alpha} + \alpha\right), \ \forall \boldsymbol{u} \in \mathbb{R}^d \ .$$

So, in order to get the best possible guarantee, we should know $\|\boldsymbol{u}\|_2$ and set $\alpha = \|\boldsymbol{u}\|_2$. As we said, this strategy does not work for a couple of reasons: i) $\boldsymbol{u}$ is not a fixed vector, rather the regret is a function of $\boldsymbol{u} \in V$; ii) if we guessed any value of $\|\boldsymbol{u}\|_2$ the adversary could easily change the losses to make that value completely wrong.

Far from being a technicality, this is an important issue as shown in the next example.

Example 9.1. *Consider that we want to use OSD with online-to-batch conversion to minimize a function that is 1-Lipschitz, whose minimizer is $\boldsymbol{x}^\star$. The convergence rate will be proportional to $(\frac{\|\boldsymbol{x}^\star\|_2^2}{\alpha} + \alpha)\sqrt{T}$ using a learning rate of $\eta = \frac{\alpha}{\sqrt{T}}$. Consider the case that $\|\boldsymbol{x}^\star\|_2 = 100$, specifying $\alpha = 1$ will result in a convergence rate 100 times slower that specifying the optimal choice in hindsight $\alpha = 100$. Note that this is a real effect not an artifact of the proof. Indeed, it is intuitive that the optimal learning rate should be proportional to the distance between the initial point that algorithm picks and the optimal solution.*

If we could tune the learning rate in the optimal way, we would get a regret of

$$\text{Regret}_T(\boldsymbol{u}) \leq \|\boldsymbol{u}\|_2\sqrt{T}, \forall \boldsymbol{u} \in V \ .$$

However, this is also impossible, because in Chapter 5 we proved a lower bound on the regret of $\Omega(\|\boldsymbol{u}\|_2\sqrt{T\ln\left(\|\boldsymbol{u}\|_2 + 1\right)})$.

In the following, we will show that it is possible to reduce any Online Convex Optimization (OCO) game to betting on a non-stochastic coin. This will allow us to use a radically different way to design OCO algorithms that will enjoy the optimal regret and will not require any parameter (e.g., learning rates, regularization weights) to be tuned. We call these kind of algorithms *parameter-free*.

9.1 Coin-Betting Game

Imagine the following repeated game:

- Set the initial Wealth to ϵ: $\text{Wealth}_0 = \epsilon$.

- In each round $t = 1, \dots, T$
  - You bet $|x_t|$ money on side of the coin equal to $\operatorname{sign}(x_t)$. You cannot bet more money than what you currently have, hence $|x_t| \leq \text{Wealth}_{t-1}$.
  - The adversary reveals the outcome of the coin $c_t \in \{-1, 1\}$.
  - You gain money $x_t c_t$, that is $\text{Wealth}_t = \text{Wealth}_{t-1} + c_t x_t = \epsilon + \sum_{i=1}^{t} c_i x_i$.

Given that we cannot borrow money, we can codify the bets x_t as $\beta_t \text{Wealth}_{t-1}$, with $\beta_t \in [-1, 1]$. So, $|\beta_t|$ is the fraction of money to bet and $\operatorname{sign}(\beta_t)$ the side of the coin on which we bet.

The aim of the game is to make as much money as possible. As usual, given the adversarial nature of the game, we cannot hope to always win money. Instead, we try to gain as much money as the strategy that bets a fixed amount of money $\beta^\star \in [-1, 1]$ for the entire game.

Note that

$$\text{Wealth}_t = \text{Wealth}_{t-1} + c_t x_t = \text{Wealth}_{t-1} + \beta_t \text{Wealth}_{t-1}\, c_t = \text{Wealth}_{t-1}(1 + \beta_t c_t) = \epsilon \prod_{i=1}^{t-1} (1 + \beta_i c_i) \,.$$

So, given the multiplicative nature of the wealth, it is also useful to take the logarithm of the ratio of the wealth of the algorithm and wealth of the optimal betting fraction. Hence, we want to minimize the following regret

$$\begin{aligned} \ln \max_{\beta \in [-1,1]} \epsilon \prod_{t=1}^{T} (1 + \beta c_t) - \ln \text{Wealth}_T &= \ln \max_{\beta \in [-1,1]} \epsilon \prod_{t=1}^{T} (1 + \beta c_t) - \ln \left(\epsilon \prod_{t=1}^{T} (1 + \beta_t c_t) \right) \\ &= \max_{\beta \in [-1,1]} \sum_{t=1}^{T} \ln(1 + \beta c_t) - \sum_{t=1}^{T} \ln(1 + \beta_t c_t) \,. \end{aligned}$$

In words, this is nothing else than the regret of an OCO game where the losses are $\ell_t(x) = -\ln(1 + x c_t)$ and $V = [-1, 1]$. We can also extend a bit the formulation allowing "continuous coins", where $c_t \in [-1, 1]$ rather than in $\{-1, 1\}$.

Remark 9.2. *Note that the constraint to bet a fraction between -1 and 1 is not strictly necessary. We could allow the algorithm to bet more money that what it currently has, lending it some money in each round. However, the restriction makes the analysis easier because it allows the transformation above into an OCO problem, using the non-negativity of $1 + \beta_t c_t$.*

We could just use OMD or FTRL, taking special care of the non-Lipschitzness of the functions, but it turns out that there exists a better strategy specifically for this problem. There exists a very simple strategy to solve the coin-betting game above, that is called **Krichevsky-Trofimov (KT) bettor**. It simply says that on each time step t you bet $\beta_t = \frac{\sum_{i=1}^{t-1} c_i}{t}$. So, the algorithm is the following one.

Algorithm 9.1 Krichevsky-Trofimov Bettor

Require: Initial money $\text{Wealth}_0 = \epsilon > 0$
1: **for** $t = 1$ **to** T **do**
2: Calculate the betting fraction $\beta_t = \frac{\sum_{i=1}^{t-1} c_i}{t}$
3: Bet $x_t = \beta_t \text{Wealth}_{t-1}$, that is $|x_t|$ money on the side $\operatorname{sign}(x_t) = \operatorname{sign}(\beta_t)$
4: Receive the coin outcome $c_t \in [-1, 1]$
5: Win/lose $x_t c_t$, that is $\text{Wealth}_t = \text{Wealth}_{t-1} + c_t x_t$
6: **end for**

For it, we can prove the following theorem.

Theorem 9.3 ([Cesa-Bianchi and Lugosi, 2006, Theorem 9.4]). *Let $c_t \in \{-1, 1\}$ for $t = 1, \dots, T$. Then, the KT bettor in Algorithm 9.1 guarantees*

$$\ln \text{Wealth}_T \geq \ln \max_{\beta \in [-1,1]} \prod_{t=1}^{T} (1 + \beta c_t) - \frac{1}{2} \ln T - K,$$

where K is a universal constant.

Note that if the outcomes of the coin are skewed towards one side, the optimal betting fraction will gain an exponential amount of money, as proved in the next Lemma.

Lemma 9.4. *Let $c_t \in \{-1, 1\}, t = 1, \dots, T$. Then, we have*

$$\max_{\beta \in [-1,1]} \exp\left(\sum_{t=1}^{T} \ln(1 + \beta c_t) \right) \geq \exp\left(\sum_{t=1}^{T} \frac{\left(\sum_{t=1}^{T} c_t \right)^2}{4T} \right) .$$

Proof.

$$\begin{aligned} \max_{\beta \in [-1,1]} \exp\left(\sum_{t=1}^{T} \ln(1 + \beta c_t) \right) &\geq \max_{\beta \in [-1/2,1/2]} \exp\left(\sum_{t=1}^{T} \ln(1 + \beta c_t) \right) \geq \max_{\beta \in [-1/2,1/2]} \exp\left(\beta \sum_{t=1}^{T} c_t - \beta^2 \sum_{t=1}^{T} c_t^2 \right) \\ &= \max_{\beta \in [-1/2,1/2]} \exp\left(\beta \sum_{t=1}^{T} c_t - \beta^2 T \right) = \exp\left(\frac{\left(\sum_{t=1}^{T} c_t \right)^2}{4T} \right) . \end{aligned}$$

where we used the elementary inequality $\ln(1 + x) \geq x - x^2$ for $x \in [-1/2, 1/2]$. □

Hence, KT guarantees an exponential amount of money, paying only a $\sqrt{T}$ penalty. It is possible to prove that the guarantee above for the KT algorithm is optimal to constant additive factors. Moreover, observe that the KT strategy does not require any parameter to be set: no learning rates, nor regularizer. That is, KT is *parameter-free*.

Also, we can extend the guarantee of the KT algorithm to the case in which the coins are "continuous", that is $c_t \in [-1, 1]$. We have the following Theorem.

Theorem 9.5 ([Orabona and Pál, 2016, Lemma 14]). *Let $c_t \in [-1, 1]$ for $t = 1, \dots, T$. Then, the KT bettor in Algorithm 9.1 guarantees*

$$\ln \text{Wealth}_T \geq \sum_{t=1}^{T} \frac{\left(\sum_{t=1}^{T} c_t \right)^2}{4T} - \frac{1}{2} \ln T - K,$$

where K is a universal constant.

So, we have introduced the coin-betting game, extended it to continuous coins and presented a simple and optimal parameter-free strategy. In the next Section, we show *how to use the KT bettor as a parameter-free 1-d OCO algorithm!*

9.2 Parameter-free 1d OCO through Coin-Betting

So, Theorem 9.3 tells us that we can win almost as much money as a strategy betting the optimal fixed fraction of money at each step. We only pay a logarithmic price in the log wealth, that corresponds to a $\frac{1}{\sqrt{T}}$ term in the actual wealth.

Now, let's see why this problem is interesting in OCO. It turns out that *solving the coin-betting game with continuous coins is equivalent to solving a 1-dimensional unconstrained online linear optimization problem*. That is, a

coin-betting algorithm is equivalent to design an online learning algorithm that produces a sequences of $x_t \in \mathbb{R}$ that minimize the 1-dimensional regret with linear losses:

$$\text{Regret}_T(u) := \sum_{t=1}^{T} g_t x_t - \sum_{t=1}^{T} g_t u,$$

where the g_t are adversarial and bounded. Without loss of generality, we will assume $g_t \in [-1, 1]$. Also, remembering that OCO games can be reduced to Online Linear Optimization (OLO) games, such reduction would effectively reduces OCO to coin-betting! Moreover, through online-to-batch conversion, any stochastic 1-d problem could be reduced to a coin-betting game!

The key theorem that allows the conversion between OLO and coin-betting is the following one.

Theorem 9.6. *Let $\phi : \mathbb{R}^d \to (-\infty, +\infty]$ be a proper closed convex function and let $\phi^\star : \mathbb{R}^d \to (-\infty, +\infty]$ be its Fenchel conjugate. An online algorithm that generates $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_T \in \mathbb{R}^d$ guarantees*

$$\forall \boldsymbol{g}_1, \dots, \boldsymbol{g}_T \in \mathbb{R}^d, \quad \epsilon - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle \geq \phi\left(- \sum_{t=1}^{T} \boldsymbol{g}_t \right),$$

where $\epsilon \in \mathbb{R}$, if and only if it guarantees

$$\forall \boldsymbol{u} \in \mathbb{R}^d, \quad \underbrace{\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle}_{\text{Regret}_T(\boldsymbol{u})} \leq \phi^\star(\boldsymbol{u}) + \epsilon \, .$$

Proof. Let's prove the left to right implication.

$$\begin{aligned} \text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle &\leq - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle - \phi\left(- \sum_{t=1}^{T} \boldsymbol{g}_t \right) + \epsilon \\ &\leq \sup_{\boldsymbol{\theta} \in \mathbb{R}^d} \langle \boldsymbol{\theta}, \boldsymbol{u} \rangle - \phi(\boldsymbol{\theta}) + \epsilon = \phi^\star(\boldsymbol{u}) + \epsilon \, . \end{aligned}$$

For the other implication, we have

$$\begin{aligned} - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle = - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle &= \sup_{\boldsymbol{u} \in \mathbb{R}^d} \; - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \\ &\geq \sup_{\boldsymbol{u} \in \mathbb{R}^d} \; - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle - \phi^\star(\boldsymbol{u}) - \epsilon = \phi\left(- \sum_{t=1}^{T} \boldsymbol{g}_t \right) - \epsilon \, . \end{aligned}$$

□

To make sense of the above theorem, assume that we are considering a 1-d problem and $g_t \in [-1, 1]$. Then, guaranteeing a lower bound to

$$\epsilon - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle$$

can be done through a betting strategy that bets x_t money on the coins $c_t = -g_t$. So, the theorem implies that *proving a reward lower bound for the wealth in a coin-betting game implies a regret upper bound for the corresponding 1-dimensional OLO game*. However, proving a reward lower bound is easier because it doesn't depend on the competitor $\boldsymbol{u}$. Indeed, not knowing the norm of the competitor is exactly the reason why tuning the learning rates in OMD is hard!

This consideration immediately gives us the conversion between 1-d OLO and coin-betting: **the outcome of the coin is the negative of the subgradient of the losses on the current prediction.** Indeed, setting $c_t = -g_t$, we have that a coin-betting algorithm that bets x_t would give us

$$\text{Wealth}_T = \epsilon + \sum_{t=1}^{T} x_t c_t = \epsilon - \sum_{t=1}^{T} x_t g_t \, .$$

So, a lower bound on the wealth corresponds to a lower bound that can be used in Theorem 9.6. To obtain a regret guarantee, we only need to calculate the Fenchel conjugate of the reward function, assuming it can be expressed as a function of $\sum_{t=1}^{T} c_t$.

The last step is to reduce 1-d OCO to 1-d OLO. But, this is an easy step that we have done many times. Indeed, we have

$$\mathrm{Regret}_T(u) = \sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(u) \le \sum_{t=1}^{T} x_t g_t - \sum_{t=1}^{T} g_t u,$$

where $g_t \in \partial \ell_t(x_t)$.

So, to summarize, the Fenchel conjugate of the wealth lower bound for the coin-betting game becomes the regret guarantee for the OCO game. In the next section, we specialize all these considerations to the KT algorithm.

9.2.1 KT as a 1d Online Convex Optimization Algorithm

Here, we want to use the considerations in the above section to use KT as a parameter-free 1-d OCO algorithm. First, let's see what such algorithm looks like. KT bets $x_t = \beta_t \,\mathrm{Wealth}_{t-1}$, starting with ϵ money. Now, set $c_t = -g_t$ where $g_t \in \partial \ell_t(x_t)$ and assume the losses ℓ_t 1-Lipschitz. So, we get

$$x_t = -\frac{\sum_{i=1}^{t-1} g_i}{t} \left(\epsilon - \sum_{i=1}^{t-1} g_i x_i \right) .$$

The pseudo-code is in Algorithm 9.2.

Algorithm 9.2 Krichevsky-Trofimov OCO Algorithm

Require: $\epsilon > 0$ (any number between 1 and $\sqrt{T}$)

1: **for** $t = 1$ **to** T **do**
2: Predict $x_t = -\frac{\sum_{i=1}^{t-1} g_i}{t} \left(\epsilon - \sum_{i=1}^{t-1} g_i x_i \right) \in \mathbb{R}$
3: Receive loss $\ell_t : \mathbb{R} \to \mathbb{R}$ and pay $\ell_t(x_t)$
4: Set $g_t \in \partial \ell_t(x_t)$
5: **end for**

Let's now see what kind of regret we get. From Theorem 9.5, we have that the KT bettor guarantees the following lower bound on the wealth when used with $c_t = -g_t$:

$$\epsilon - \sum_{t=1}^{T} x_t g_t \ge \frac{\epsilon}{K\sqrt{T}} \exp\left(\frac{(\sum_{t=1}^{T} g_t)^2}{4T} \right) .$$

So, we found the function ϕ, we just need $\phi^\star$ or an upper bound to it, that can be found with the following Lemma.

Lemma 9.7. *Define $f(x) = \beta \exp \frac{x^2}{2\alpha}$, for $\alpha, \beta > 0$, $x \ge 0$. Then*

$$f^\star(y) = |y| \sqrt{\alpha W\left(\frac{\alpha y^2}{\beta^2} \right)} - \beta \exp\left(\frac{W\left(\frac{\alpha y^2}{\beta^2} \right)}{2} \right) \le |y| \sqrt{\alpha W\left(\frac{\alpha y^2}{\beta^2} \right)} - \beta \le |y| \sqrt{\alpha \ln\left(1 + \frac{\alpha y^2}{\beta^2} \right)} - \beta .$$

where $W(x)$ is the Lambert function, i.e., $W : \mathbb{R}_{\ge 0} \to \mathbb{R}_{\ge 0}$ satisfies $x = W(x)\exp(W(x))$.

Proof. From the definition of Fenchel dual, we have

$$f^\star(y) = \max_x \; x\, y - f(x) = \max_x \; x\, y - \beta \exp \frac{x^2}{2\alpha} \le x^\star\, y - \beta,$$

where $x^\star = \mathrm{argmax}_x \, x\, y - f(x)$. We now use the fact that $x^\star$ satisfies $y = f'(x^\star)$, to have $x^\star = \mathrm{sign}(y) \sqrt{\alpha W\left(\frac{\alpha y^2}{\beta^2} \right)}$, where $W(\cdot)$ is the Lambert function. Using Lemma A.2 in the Appendix, we obtain the stated bound. □

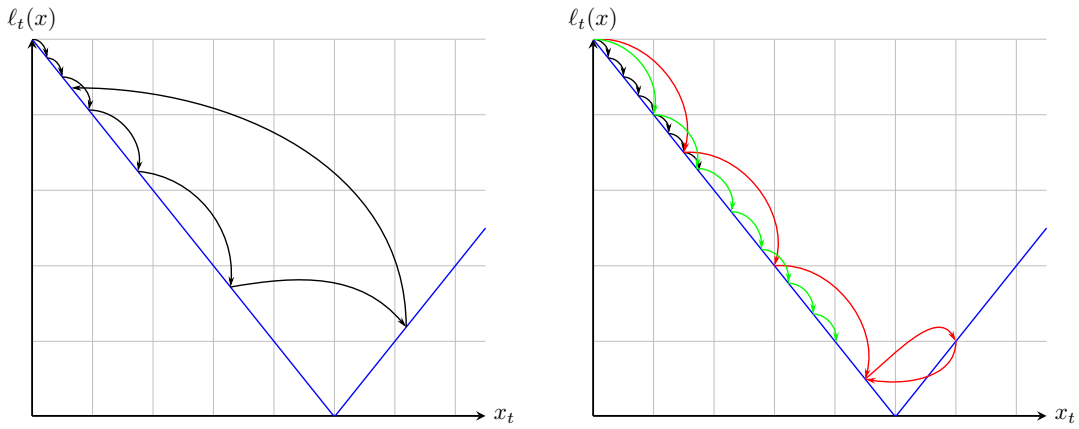

Figure 9.1: Behaviour of KT (left) and online gradient descent with various learning rates and same number of steps (right) when $\ell_t(x) = |x - 10|$ for all t.

So, the regret guarantee of KT used a 1d OLO algorithm is upper bounded by

$$\text{Regret}_T(u) = \sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(u) \leq |u| \sqrt{4T \ln \left(\frac{\sqrt{2}|u|KT}{\epsilon} + 1 \right)} + \epsilon, \ \forall u \in \mathbb{R},$$

where the only assumption was that the first derivatives (or sub-derivatives) of ℓ_t are bounded in absolute value by 1. Also, it is important to note that any setting of ϵ in $[1, \sqrt{T}]$ would not change the asymptotic rate.

To better appreciate this regret, compare this bound to the one of OMD with learning rate $\eta = \frac{\alpha}{\sqrt{T}}$:

$$\text{Regret}_T(u) = \sum_{t=1}^{T} \ell_t(x_t) - \sum_{t=1}^{T} \ell_t(u) \leq \frac{1}{2} \left(\frac{u^2}{\alpha} + \alpha \right) \sqrt{T}, \ \forall u \in \mathbb{R} \, .$$

Hence, the coin-betting approach allows to get almost the optimal bound, without having to guess the correct learning rate! The price that we pay for this parameter-freeness is the log factor, that is optimal from our lower bound.

It is interesting also to look at what the algorithm would do on an easy problem, where $\ell_t(x) = |x - 10|$. In Figure 9.1, we show the different predictions that the KT algorithm and online subgradient descent (OSD) would do. Note how the convergence rate of OSD critically depends on the learning rate: too big will not give convergence and too small will make slow down the convergence. On the other hand, KT will go *exponentially fast* towards the minimum and then it will automatically backtrack. This exponential growth effectively works like a line search procedure that allows to get the optimal regret without tuning learning rates. Later in the iterations, KT will oscillate around the minimum, *automatically shrinking its steps, without any parameter to tune.* Of course, this is a simplified example. In a truly OCO game, the losses are different at each time step and the intuition behind the algorithm becomes more difficult. Yet, the optimality of the regret assures us that the KT strategy is the right strategy.

Next, we will see that we can also reduce OCO in $\mathbb{R}^d$ and learning with experts to coin-betting games.

9.3 Coordinate-wise Parameter-free OCO

We have already seen that it is always possible to decompose an OCO problem over the coordinates and use a different 1-dimensional Online Linear Optimization algorithm on each coordinate. In particular, we saw that

$$\begin{aligned}\text{Regret}_T(\boldsymbol{u}) &= \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \leq \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \\ &= \sum_{t=1}^T \sum_{i=1}^d g_{t,i}(x_{t,i} - u_i) = \sum_{i=1}^d \sum_{t=1}^T g_{t,i}(x_{t,i} - u_i),\end{aligned}$$

where the $\sum_{t=1}^T g_{t,i}(x_{t,i} - u_i)$ is exactly the regret w.r.t. the linear losses constructed by the coordinate i of the subgradient.

Hence, if we have a 1-dimensional OLO algorithm, we can d copies of it, each one fed with the coordinate i of the subgradient. In particular, we might think to use the KT algorithm over each coordinate. The pseudo-code of this procedure is in Algorithm 9.3.

Algorithm 9.3 OCO with Coordinate-Wise Krichevsky-Trofimov

Require: $\epsilon > 0$

for $t = 1$ **to** T **do**
  Output $\boldsymbol{x}_t$ whose coordinates are $x_{t,i} = -\frac{\sum_{j=1}^{t-1} g_{j,i}}{t} \left(\epsilon - \sum_{j=1}^{t-1} g_{j,i} x_{j,i} \right)$ for $i = 1, \dots, d$
  Receive loss $\ell_t : \mathbb{R}^d \to \mathbb{R}$ and pay $\ell_t(\boldsymbol{x}_t)$
  Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
end for

The regret bound we get is immediate: We just have to sum the regret over the coordinates.

Theorem 9.8. *With the notation in Algorithm 9.3, assume that $\|\boldsymbol{g}_t\|_\infty \leq 1$. Then, $\forall \boldsymbol{u} \in \mathbb{R}^d$, the following regret bounds hold*

$$\sum_{t=1}^T (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \sum_{i=1}^d |u_i| \sqrt{4T \ln \left(1 + \frac{\sqrt{2} u_i^2 K T}{\epsilon} \right)} + d\epsilon \leq \|\boldsymbol{u}\|_1 \sqrt{4T \ln \left(1 + \frac{\sqrt{2} \|\boldsymbol{u}\|_\infty^2 K T}{\epsilon} \right)} + d\epsilon,$$

where K is a universal constant.

Note that the Theorem above suggests that in high dimensional settings ϵ should be proportional to $\frac{1}{d}$.

9.4 Parameter-free in Any Norm

The above reductions works only with in a finite dimensional space. Moreover, it gives a dependency on the competitor w.r.t. the L_1 norm that might be undesirable. So, here we present another simple reduction from 1-dimensional OCO to infinite dimensions.

This reduction requires an unconstrained OCO algorithm for the 1-dimensional case and an algorithm for learning in d-dimensional (or infinite dimensional) balls. For the 1-dimensional learner, we could use the KT algorithm, while for learning in d-dimensional balls we can use, for example, Online Mirror Descent (OMD). Given these two learners, we decompose the problem of learning a vector $\boldsymbol{x}_t$ in the problem of learning a *direction* and a *magnitude*. The regret of this procedure turns out to be just the sum of the regret of the two learners.

We can formalize this idea in the following Theorem.

Algorithm 9.4 Learning Magnitude and Direction Separately

Require: 1d Online learning algorithm $\mathcal{A}_{\text{1d}}$, Online learning algorithm $\mathcal{A}_{\mathcal{B}}$ with feasible set equal to the unit ball $B \subset \mathbb{R}^d$ w.r.t. $\|\cdot\|$

for $t = 1$ **to** T **do**
  Get point $z_t \in \mathbb{R}$ from $\mathcal{A}_{\text{1d}}$
  Get point $\tilde{\boldsymbol{x}}_t \in B$ from $\mathcal{A}_B$
  Play $\boldsymbol{x}_t = z_t \tilde{\boldsymbol{x}}_t \in \mathbb{R}^d$
  Receive $\ell_t : \mathbb{R}^d \to \mathbb{R}$ and pay $\ell_t(\boldsymbol{x}_t)$
  Set $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$
  Set $s_t = \langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_t \rangle$
  Send $\ell_t^{\mathcal{A}_{\text{1d}}}(x) = s_t x$ as the t-th linear loss to $\mathcal{A}_{\text{1d}}$
  Send $\ell_t^{\mathcal{A}_B}(\boldsymbol{x}) = \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle$ as the t-th linear loss to $\mathcal{A}_B$
end for

Theorem 9.9. *Denote by* $\operatorname{Regret}_T^{\mathcal{A}_B}(\boldsymbol{u})$ *the linear regret of algorithm* $\mathcal{A}_B$ *for any* $\boldsymbol{u}$ *in the unit ball w.r.t a norm* $\|\cdot\|$*, and* $\operatorname{Regret}_T^{\mathcal{A}_{1d}}(u)$ *the linear regret of algorithm* $\mathcal{A}_{1d}$ *for any competitor* $u \in \mathbb{R}$*. Then, for any* $\boldsymbol{u} \neq \boldsymbol{0} \in \mathbb{R}^d$*, Algorithm 9.4 guarantees regret*

$$\operatorname{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \leq \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle = \operatorname{Regret}_T^{\mathcal{A}_{1d}}(\|\boldsymbol{u}\|) + \|\boldsymbol{u}\| \operatorname{Regret}_T^{\mathcal{A}_B}\left(\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}\right),$$

and

$$\operatorname{Regret}_T(\boldsymbol{0}) \leq \operatorname{Regret}_T^{\mathcal{A}_{1d}}(0)\ .$$

Further, the subgradients s_t *sent to* $\mathcal{A}_{1d}$ *satisfy* $|s_t| \leq \|\boldsymbol{g}_t\|_\star$*.*

Proof. First, observe that $|s_t| \leq \|\boldsymbol{g}_t\|_\star \|\tilde{\boldsymbol{x}}_t\| \leq \|\boldsymbol{g}_t\|_\star$ since $\|\tilde{\boldsymbol{x}}_t\| \leq 1$ for all t. Now, assuming $\boldsymbol{u} \neq \boldsymbol{0}$ compute:

$$\begin{aligned}
\operatorname{Regret}_T(\boldsymbol{u}) &= \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \leq \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle = \sum_{t=1}^T \langle \boldsymbol{g}_t, z_t \tilde{\boldsymbol{x}}_t \rangle - \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \\
&= \underbrace{\sum_{t=1}^T \left(\langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_t \rangle z_t - \langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_t \rangle \|\boldsymbol{u}\| \right)}_{\text{linear regret of } \mathcal{A}_{\text{1d}} \text{ at } \|\boldsymbol{u}\| \in \mathbb{R}} + \sum_{t=1}^T \left(\langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_t \rangle \|\boldsymbol{u}\| - \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \right) \\
&= \operatorname{Regret}_T^{\mathcal{A}_{\text{1d}}}(\|\boldsymbol{u}\|) + \sum_{t=1}^T \left(\langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_t \rangle \|\boldsymbol{u}\| - \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \right) \\
&= \operatorname{Regret}_T^{\mathcal{A}_{\text{1d}}}(\|\boldsymbol{u}\|) + \|\boldsymbol{u}\| \sum_{t=1}^T \left(\langle \boldsymbol{g}_t, \tilde{\boldsymbol{x}}_t \rangle - \left\langle \boldsymbol{g}_t, \frac{\boldsymbol{u}}{\|\boldsymbol{u}\|} \right\rangle \right) \\
&= \operatorname{Regret}_T^{\mathcal{A}_{\text{1d}}}(\|\boldsymbol{u}\|) + \|\boldsymbol{u}\| \operatorname{Regret}_T^{\mathcal{A}_B}\left(\frac{\boldsymbol{u}}{\|\boldsymbol{u}\|}\right)\ .
\end{aligned}$$

The case $\boldsymbol{u} = \boldsymbol{0}$ follows similarly. □

Remark 9.10. *Note that the direction vector is not constrained to have norm equal to 1, yet this does not seem to affect the regret equality.*

We can instantiate the above theorem using the KT betting algorithm for the 1d learner and OMD for the direction learner. Observe that in order to have the coin outcomes in $[-1, 1]$, we need to divide the losses by the Lipschitz constant. We obtain the following examples.

Example 9.11. *Let $\mathcal{A}_B$ be OSD with $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_2 \leq 1\}$ and learning rate $\eta_t = \frac{\sqrt{2}D}{2\sqrt{t}}$. Let $\mathcal{A}_{1d}$ the KT algorithm for 1-dimensional OCO with $\epsilon = 1$. Assume the loss functions are 1-Lipschitz w.r.t. the $\|\cdot\|_2$. Then, using the construction in Algorithm 9.4, we have*

$$\text{Regret}_T(\boldsymbol{u}) \leq O\left(\left(\|\boldsymbol{u}\|_2\sqrt{\ln(\|\boldsymbol{u}\|_2 T + 1)} + \|\boldsymbol{u}\|_2\right)\sqrt{T} + 1\right), \forall \boldsymbol{u} \in \mathbb{R}^d .$$

Using an online-to-batch conversion, this algorithm is a stochastic gradient descent procedure without learning rates to tune.

To better appreciate this kind of guarantee, let's take a look at the one of FTRL (remember that OSD can be used in unbounded domains only with constant learning rates). With the regularizer $\psi_t(\boldsymbol{x}) = \frac{\sqrt{t}}{2\alpha}\|\boldsymbol{x}\|_2$ and 1-Lipschitz losses we get a regret of

$$\text{Regret}_T(\boldsymbol{u}) \leq \sqrt{T}\left(\frac{\|\boldsymbol{u}\|_2^2}{2\alpha} + \alpha\right) .$$

So, to get the right dependency on $\|\boldsymbol{u}\|_2$ we need to tune α, but we saw this is impossible. On the other hand, the regret in Example 9.11 suffers from a logarithmic factor, that is the price to pay not to have to tune parameters.

In the same way, we can even have a parameter-free regret bound for L_p norms.

Example 9.12. *Let $\mathcal{A}_B$ be OMD with $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_p \leq 1\}$ and learning rate $\eta_t = \frac{\sqrt{2(p-1)}D}{2\sqrt{t}}$. Let $\mathcal{A}_{1d}$ the KT algorithm for 1-dimensional OCO with $\epsilon = 1$. Assume the loss functions are 1-Lipschitz w.r.t. the $\|\cdot\|_q$. Then, using the construction in Algorithm 9.4, we have*

$$\text{Regret}_T(\boldsymbol{u}) \leq O\left(\left(\|\boldsymbol{u}\|_p\sqrt{\ln(\|\boldsymbol{u}\|_p T + 1)} + \frac{\|\boldsymbol{u}\|_p}{\sqrt{p-1}}\right)\sqrt{T} + 1\right), \forall \boldsymbol{u} \in \mathbb{R}^d .$$

If we want to measure the competitor w.r.t the L_1 norm, we have to use the same method we saw for OMD in Section 6.7: Set $q = 2\ln d$ and p such that $1/p + 1/q = 1$. Now, assuming that $\|\boldsymbol{g}_t\|_\infty \leq 1$, we have that $\|\boldsymbol{g}_t\|_q \leq d^{1/q}$. Hence, we have to divide all the losses by $d^{1/q}$ and, for all $\boldsymbol{u} \in \mathbb{R}^d$, we obtain

$$\begin{aligned} d^{-1/q}\sum_{t=1}^T(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\leq O\left(\left(\|\boldsymbol{u}\|_p\sqrt{\ln(\|\boldsymbol{u}\|_p T + 1)} + \|\boldsymbol{u}\|_p\sqrt{q-1}\right)\sqrt{T} + 1\right) \\ &\leq O\left(\left(\|\boldsymbol{u}\|_1\sqrt{\ln(\|\boldsymbol{u}\|_1 T + 1)} + \|\boldsymbol{u}\|_1\sqrt{\ln d}\right)\sqrt{T} + 1\right) . \end{aligned}$$

Note that the regret against $\boldsymbol{u} = \boldsymbol{0}$ of the parameter-free construction is *constant*. It is important to understand that there is nothing special in the origin in the unconstrained setting: We could translate the prediction by any offset and get a guarantee that treats the offset as the point with constant regret. This is shown in the next Proposition.

Proposition 9.13. *Let $\mathcal{A}$ an OLO algorithm that predicts $\boldsymbol{x}_t$ and guarantees linear regret $\text{Regret}_T^{OLO}(\boldsymbol{u})$ for any $\boldsymbol{u} \in \mathbb{R}^d$. We have that the regret of the predictions $\hat{\boldsymbol{x}}_t = \boldsymbol{x}_t + \boldsymbol{x}_0$ for OCO is*

$$\sum_{t=1}^T \ell_t(\hat{\boldsymbol{x}}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{u}) \leq \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t + \boldsymbol{x}_0 - \boldsymbol{u}\rangle = \text{Regret}_T^{OLO}(\boldsymbol{u} - \boldsymbol{x}_0) .$$

9.5 Combining Online Convex Optimization Algorithms

Finally, we now show a useful application of the parameter-free OCO algorithms property to have a constant regret against $\boldsymbol{u} = \boldsymbol{0}$.

Theorem 9.14. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ two OLO algorithms that produces the predictions $\boldsymbol{x}_{t,1}$ and $\boldsymbol{x}_{t,2}$ respectively. Then, predicting with $\boldsymbol{x}_t = \boldsymbol{x}_{t,1} + \boldsymbol{x}_{t,2}$, we have for any $\boldsymbol{u} \in \mathbb{R}^d$*

$$\sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u}\rangle = \min_{\boldsymbol{u}=\boldsymbol{u}_1+\boldsymbol{u}_2} \text{Regret}_T^{\mathcal{A}_1}(\boldsymbol{u}_1) + \text{Regret}_T^{\mathcal{A}_2}(\boldsymbol{u}_2) .$$

Moreover, if both algorithm guarantee a constant regret of ϵ against $\boldsymbol{u} = \mathbf{0}$, we have for any $\boldsymbol{u} \in \mathbb{R}^d$

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \leq \epsilon + \min \left(\text{Regret}_T^{\mathcal{A}_1}(\boldsymbol{u}), \text{Regret}_T^{\mathcal{A}_2}(\boldsymbol{u}) \right) .$$

Proof. Set $\boldsymbol{u}_1 + \boldsymbol{u}_2 = \boldsymbol{u}$. Then,

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle = \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_{t,1} \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u}_1 \rangle + \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_{t,2} \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u}_2 \rangle .$$ □

In words, the above theorem allows us to combine online learning algorithm. If the algorithms we combine have constant regret against the null competitor, then we always get the best of the two guarantees.

Example 9.15. *We can combine two parameter-free OCO algorithms, one that gives a bound that depends on the L_2 norm of the competitor and subgradients and another one specialized to the L_1/L_∞ norm of competitor/subgradients. The above theorem assures us that we will also get the best guarantee between the two, paying only an additive constant factor in the regret.*

Of course, upper bounding the OCO regret with the linear regret, the above theorem also upper bounds the OCO regret.

9.6 Reduction to Learning with Experts

In this section, we consider again the learning with expert advice setting from 6.8. First, remember that the regret we got from OMD (and similarly for FTRL) is

$$\text{Regret}_T(\boldsymbol{u}) \leq O\left(\frac{KL(\boldsymbol{u}; \boldsymbol{\pi})}{\eta} + \eta T \right),$$

where $\boldsymbol{\pi}$ is the prior distribution on the experts and $KL(\cdot;\cdot)$ is the KL-divergence. As we reasoned in the OCO case, in order to set the learning rate we should know the value of $KL(\boldsymbol{u}; \boldsymbol{\pi})$. If we could set η to $\sqrt{\frac{KL(\boldsymbol{u};\boldsymbol{\pi})}{T}}$, we would obtain a regret of $\sqrt{T\, KL(\boldsymbol{u}; \boldsymbol{\pi})}$. However, given the adversarial nature of the game, this is impossible. So, as we did in the OCO case, we will show that even this problem can be reduced to betting on a continuous coin, obtaining optimal regret guarantees with a parameter-free algorithm.

Remark 9.16. *There exists a different notion of regret for learning with experts. Order the cumulative losses of all actions from lowest to highest and define the* ϵ-quantile regret *to be the difference between the cumulative loss of the learner and the $\lceil \epsilon d \rceil$-th element in the sorted list. In formulas*

$$\text{Regret}_T(\epsilon) = \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - \sum_{t=1}^{T} g_{t,i_\epsilon},$$

where $\epsilon \in [1/d, 1]$ and i_ϵ is the $\lceil \epsilon d \rceil$-th best performing action.

This definition makes sense when the number of actions is very large and there are many of them that are close to the optimal one. Note the task of guaranteeing a small regret becomes easier with increasing ϵ. So, we would like to design an algorithm whose regret is inversely proportional to ϵ and it also does not depend on d in any way.

We now show that a regret that depends on the KL divergence between a generic competitor $\boldsymbol{u}$ and $\boldsymbol{\pi}$ implies such regret guarantee. Define $\boldsymbol{u}_\epsilon$ as the vector that has the coordinates corresponding to the smallest $\lceil \epsilon d \rceil$ experts equal to $\frac{1}{\lceil \epsilon d \rceil}$ and 0 in the other coordinates. Also, assume to have an algorithm that guarantees an upper bound $\text{Regret}_T(\boldsymbol{u})$ equal to $F_T(KL(\boldsymbol{u}; \boldsymbol{\pi}))$ for any sequence of losses $\boldsymbol{g}_t \in [0,1]^d$, where F is a non-decreasing function. Then, setting $\boldsymbol{\pi} = [1/d, \dots, 1/d]$, we have

$$\text{Regret}_T(\epsilon) = \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - \sum_{t=1}^{T} g_{t,i_\epsilon} \leq \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u}_\epsilon \rangle \leq F_T(KL(\boldsymbol{u}_\epsilon; \boldsymbol{\pi})) = F_T\left(\ln \frac{d}{\lceil \epsilon d \rceil} \right) \leq F_T\left(\ln \frac{1}{\epsilon} \right),$$

where in the first inequality we used the fact that the average of a set of numbers is smaller than the biggest number in the set.

First, let's introduce some notation. Let $d \geq 2$ be the number of experts and Δ^{d-1} be the probability simplex. Let $\boldsymbol{\pi} = [\pi_1, \pi_2, \dots, \pi_d] \in \Delta^{d-1}$ be any *prior* distribution. Let $\mathcal{A}$ be a coin-betting algorithm. We will instantiate d copies of $\mathcal{A}$, each of them with initial wealth π_i for $i = 1, \dots, d$.

Consider any round t. Let $x_{t,i} \in \mathbb{R}$ be the bet of the i-th copy of $\mathcal{A}$. The learning with expert advice algorithm computes $\widehat{\boldsymbol{p}}_t = [\widehat{p}_{t,1}, \widehat{p}_{t,2}, \dots, \widehat{p}_{t,d}] \in \mathbb{R}^d_{\geq 0}$ as

$$\widehat{p}_{t,i} = \max(x_{t,i}, 0) \ . \tag{9.1}$$

Then, the learning with expert advice algorithm predicts $\boldsymbol{p}_t = [p_{t,1}, p_{t,2}, \dots, p_{t,d}] \in \Delta^{d-1}$ as

$$\boldsymbol{p}_t = \begin{cases} \frac{\widehat{\boldsymbol{p}}_t}{\|\widehat{\boldsymbol{p}}_t\|_1}, & \text{if } \widehat{\boldsymbol{p}}_t \neq \boldsymbol{0}, \\ \boldsymbol{\pi}, & \text{otherwise} \ . \end{cases} \tag{9.2}$$

After the prediction, the algorithm receives the vector of the losses for each expert $\boldsymbol{g}_t = [g_{t,1}, g_{t,2}, \dots, g_{t,d}] \in [0,1]^d$. From these losses, we construct the outcome of the continuous coin $c_{t,i} \in [-1, 1]$ for the i-th copy of betting algorithm $\mathcal{A}$, defined as

$$c_{t,i} = \begin{cases} \langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i} & \text{if } x_{t,i} > 0, \\ \max\left(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i}, 0\right) & \text{if } x_{t,i} \leq 0 \ . \end{cases} \tag{9.3}$$

The construction above defines a learning with expert advice algorithm defined by the predictions $\boldsymbol{p}_t$, based on the algorithm $\mathcal{A}$. We can prove the following regret bound for it.

Theorem 9.17 (Regret Bound for Experts)**.** *Let $\mathcal{A}$ be a coin-betting algorithm that guarantees a wealth after t rounds with initial money equal to 1 of $\exp(f_t(\sum_{i=1}^t c'_t))$ for any sequence of continuous coin outcomes $c'_1, \dots, c'_t \in [-1, 1]$. Then, the regret of the learning with expert advice algorithm with prior $\pi \in \Delta^{d-1}$ that predicts at each round with $\boldsymbol{p}_t$ in* (9.2) *satisfies*

$$\forall T \geq 0, \quad \forall \boldsymbol{u} \in \Delta^{d-1}, \qquad \text{Regret}_T(\boldsymbol{u}) = \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{p}_t - \boldsymbol{u} \rangle \leq h\left(KL(\boldsymbol{u}; \boldsymbol{\pi})\right),$$

for any $h : \mathbb{R} \to \mathbb{R}$ concave and non-decreasing such that $x \leq h(f_T(x))$.

Proof. We first prove that $\langle \boldsymbol{c}_t, \boldsymbol{x}_t \rangle \leq 0$. Indeed,

$$\begin{aligned} \langle \boldsymbol{c}_t, \boldsymbol{x}_t \rangle &= \sum_{i=1}^d c_{t,i} x_{t,i} = \sum_{i \,:\, x_{t,i} > 0} \max(x_{t,i}, 0)(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i}) \ + \sum_{i \,:\, x_{t,i} \leq 0} x_{t,i} \max(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i}, 0) \\ &= \|\widehat{\boldsymbol{p}}_t\|_1 \sum_{i=1}^d p_{t,i}(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - \boldsymbol{g}_{t,i}) \ + \sum_{i \,:\, x_{t,i} \leq 0} x_{t,i} \max(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i}, 0) \\ &= 0 \ + \sum_{i \,:\, x_{t,i} \leq 0} x_{t,i} \max(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i}, 0) \ \leq 0 \ . \end{aligned}$$

The first equality follows from definition of $c_{t,i}$. To see the second equality, consider two cases: If $x_{t,i} \leq 0$ for all i then $\|\widehat{p}_t\|_1 = 0$ and therefore both $\|\widehat{\boldsymbol{p}}_t\|_1 \sum_{i=1}^d p_{t,i}(g_{t,i} - \langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle)$ and $\sum_{i \,:\, x_{t,i} > 0} \max(w_{t,i}, 0)(g_{t,i} - \langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle)$ are trivially zero. If $\|\widehat{\boldsymbol{p}}_t\|_1 > 0$ then $\max(x_{t,i}, 0) = \widehat{p}_{t,i} = \|\widehat{\boldsymbol{p}}_t\|_1 p_{t,i}$ for all i.

From the assumption on $\mathcal{A}$, for the arm i, we have for any sequence $\{c'_t\}_{t=1}^T$ such that $c'_t \in [-1, 1]$ that

$$\text{Wealth}_{T,i} = \pi_i + \sum_{t=1}^T c'_t x_t \geq \pi_i \exp\left(f_T\left(\sum_{t=1}^T c'_t\right)\right) \ . \tag{9.4}$$

Algorithm 9.5 Learning with Expert Advice based on KT Bettors

Require: Number of experts d, prior distribution $\pi \in \Delta^{d-1}$, number of rounds T

1: **for** $t = 1, 2, \ldots, T$ **do**

2: Set $x_{t,i} = \frac{\sum_{j=1}^{t-1} c_{j,i}}{t} \left(\pi_i + \sum_{j=1}^{t-1} c_{j,i} x_{j,i} \right)$ for $i = 1, \ldots, d$

3: Set $\widehat{p}_{t,i} = \max(x_{t,i}, 0)$ for $i = 1, \ldots, d$

4: Predict with $\boldsymbol{p}_t = \begin{cases} \widehat{\boldsymbol{p}}_t / \left\| \widehat{\boldsymbol{p}}_t \right\|_1 & \text{if } \left\| \widehat{\boldsymbol{p}}_t \right\|_1 > 0 \\ \pi & \text{if } \left\| \widehat{\boldsymbol{p}}_t \right\|_1 = 0 \end{cases}$

5: Receive loss vector $\boldsymbol{g}_t \in [0,1]^d$

6: Set $c_{t,i} = \begin{cases} \langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i}, & \text{if } x_{t,i} > 0 \\ \max(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i}, 0), & \text{if } x_{t,i} \leq 0 \end{cases}$ for $i = 1, \ldots, d$

7: **end for**

So, inequality $\langle \boldsymbol{c}_t, \boldsymbol{x}_t \rangle \leq 0$ and (9.4) imply

$$\sum_{i=1}^{d} \pi_i \exp\left(f_T \left(\sum_{t=1}^{T} c_{t,i} \right) \right) \leq 1 + \sum_{i=1}^{d} \sum_{t=1}^{T} c_{t,i} x_{t,i} \leq 1 \,. \tag{9.5}$$

Now, for any competitor $\boldsymbol{u} \in \Delta^{d-1}$,

$$\begin{aligned}
\text{Regret}_T(\boldsymbol{u}) &= \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{p}_t - \boldsymbol{u} \rangle \\
&= \sum_{t=1}^{T} \sum_{i=1}^{d} u_i \left(\langle \boldsymbol{g}_t, \boldsymbol{p}_t \rangle - g_{t,i} \right) \\
&\leq \sum_{t=1}^{T} \sum_{i=1}^{d} u_i c_{t,i} \qquad \text{(by definition of } c_{t,i}) \\
&\leq \sum_{i=1}^{d} u_i h\left(f_T \left(\sum_{t=1}^{T} c_{t,i} \right) \right) \qquad \text{(definition of the } h(x)) \\
&\leq h\left[\sum_{i=1}^{d} u_i f_T \left(\sum_{t=1}^{T} c_{t,i} \right) \right] \qquad \text{(by concavity of } h \text{ and Jensen inequality)} \\
&\leq h\left[KL(\boldsymbol{u}; \boldsymbol{\pi}) + \ln \left(\sum_{i=1}^{d} \pi_i \exp\left(f_T \left(\sum_{t=1}^{T} c_{t,i} \right) \right) \right) \right] \qquad \text{(Fenchel-Young inequality)} \\
&\leq h\left(KL(\boldsymbol{u}; \boldsymbol{\pi}) \right) \qquad \text{(by (9.5))} \,.
\end{aligned}$$

□

Now, we could think to use the KT bettor with this theorem, obtaining the Algorithm 9.5. Unfortunately, this algorithm obtains a sub-optimal regret guarantee. In fact, remembering the lower bound on the wealth of KT and setting $h(x) = \sqrt{4T(x + \frac{1}{2} \ln T + K)}$ where K is a universal constant, we have

$$\text{Regret}_T(\boldsymbol{u}) = O(\sqrt{T(KL(\boldsymbol{u}; \boldsymbol{\pi}) + \ln T)}) \,.$$

We might think that the $\ln T$ is the price we have to pay to adapt to the unknown competitor $\boldsymbol{u}$. However, it turns out it can be removed. In the next section, we see how to change the KT strategy to obtain the optimal guarantee.

9.6.1 A Betting Strategy that Loses at Most a Constant Fraction of Money

In the reduction before, if we use the KT betting strategy we would have a $\ln T$ term under the square root. It turns out that we can avoid that term if we know the number of rounds beforehand. Then, in case T is unknown we can just use

a doubling trick, paying only a constant multiplicative factor in the regret.

The logarithmic term in the regret comes from the fact that the lower bound on the wealth is

$$O\left(\frac{1}{\sqrt{T}}\exp\left(\frac{(\sum_{t=1}^{T} c_t)^2}{4T}\right)\right) .$$

Note that in the case in which the number of heads in the sequence is equal to the number of heads, so that $\sum_{t=1}^{T} c_t = 0$, the guaranteed wealth becomes proportional to $\frac{1}{\sqrt{T}}$. So, for T that goes to infinity the bettor will lose all of its money.

Instead, we need a more conservative strategy that guarantees

$$O\left(\alpha\exp\left(\frac{(\sum_{t=1}^{T} c_t)^2}{4T}\right)\right),$$

for α small enough and independent of T. In this case, the betting strategy has to pace its betting, possibly with the knowledge of the duration of the game, so that even in the case that the number of heads is equal to the number of tails it will only lose a fraction of its money. At the same time, it will still gain an exponential amount of money when the coin outcomes are biased towards one side.

We will prove that this is possible, designing a new betting strategy.

Denote by $S_t = \sum_{i=1}^{t} c_i$ for $t = 1, \ldots, T$ and define

$$F_t(x) = \epsilon\exp\left(\frac{x^2}{2(t+T)} - \sum_{i=1}^{t}\frac{1}{2(i+T)}\right), \tag{9.6}$$

$$\beta_t = \frac{F_t(S_{t-1}+1) - F_t(S_{t-1}-1)}{F_t(S_{t-1}+1) + F_t(S_{t-1}-1)} . \tag{9.7}$$

Note that if

$$(1+\beta_t c_t)F_{t-1}\left(\sum_{i=1}^{t-1} c_i\right) \geq F_t\left(\sum_{i=1}^{t} c_i\right), \tag{9.8}$$

then, by induction, $\text{Wealth}_T \geq F_T\left(\sum_{t=1}^{T} c_t\right)$. In fact, we have

$$\text{Wealth}_t = (1+\beta_t c_t)\,\text{Wealth}_{t-1} \geq (1+\beta_t c_t)F_{t-1}\left(\sum_{i=1}^{t-1} c_i\right) \geq F_t\left(\sum_{i=1}^{t} c_i\right) .$$

Hence, we have to prove that (9.8) is true in order to guarantee a minimum wealth of our betting strategy.

First, given that $\ln(1+\beta_t c) - \frac{(x+c)^2}{2(t+T)}$ is a concave function of c, we have

$$\min_{c\in[-1,1]} \ln(1+\beta_t c) - \frac{(x+c)^2}{2(t+T)} = \min\left(\ln(1+\beta_t) - \frac{(x+1)^2}{2(t+T)}, \ln(1-\beta_t) - \frac{(x-1)^2}{2(t+T)}\right) .$$

Also, our choice of β_t makes the two quantities above equal with $x = S_{t-1}$, that is

$$\ln(1+\beta_t) - \ln F_t(S_{t-1}+1) = \ln(1-\beta_t) - \ln F_t(S_{t-1}-1)$$

For other choices of β_t, the two alternatives would be different and the minimum one could always be the one picked by the adversary. Instead, making the two choices worst outcomes equivalent, we minimize the damage of the adversarial

choice of the outcomes of the coin. So, we have that

$$\begin{aligned}
\ln(1+\beta_t c_t) - \ln F_t(S_t) &= \ln(1+\beta_t c_t) + F_t(S_{t-1}+c_t) \\
&\geq \min_{c\in[-1,1]} \ln(1+\beta_t c) + \ln F_t(S_{t-1}+c) \\
&= -\ln \frac{F_t(S_{t-1}+1) + F_t(S_{t-1}-1)}{2} \\
&= -\ln\left[\exp\left(\frac{S_{t-1}^2+1}{2(t+T)} - \sum_{i=1}^{t}\frac{1}{2(i+T)}\right)\frac{1}{2}\left(\exp\left(\frac{S_{t-1}}{t+T}\right) + \exp\left(\frac{-S_{t-1}}{t+T}\right)\right)\right] \\
&= -\frac{S_{t-1}^2+1}{2(t+T)} - \ln\cosh\frac{S_{t-1}}{t+T} + \sum_{i=1}^{t}\frac{1}{2(i+T)} \\
&\geq -\frac{S_{t-1}^2}{2(t+T)} - \frac{S_{t-1}^2}{2(t+T)^2} + \sum_{i=1}^{t-1}\frac{1}{2(i+T)} \\
&\geq -\frac{S_{t-1}^2}{2(t+T)} - \frac{S_{t-1}^2}{2(t+T)(t-1+T)} + \sum_{i=1}^{t-1}\frac{1}{2(i+T)} \\
&= -\frac{S_{t-1}^2}{2(t-1+T)} + \sum_{i=1}^{t-1}\frac{1}{2(i+T)} \\
&= -\ln F_{t-1}(S_{t-1}),
\end{aligned}$$

where in the second equality we used the definition of β_t and in the second inequality we used the fact that $\ln\cosh(x) \leq \frac{x^2}{2}, \forall x$.

Hence, given that (9.8) is true, this strategy guarantees

$$\text{Wealth}_T \geq \exp\left(\frac{\left(\sum_{t=1}^{T} c_t\right)^2}{4T} - \sum_{t=1}^{T}\frac{1}{2(i+T)}\right) \geq \exp\left(\frac{\left(\sum_{t=1}^{T} c_t\right)^2}{4T} - \frac{1}{2}\ln 2\right) = \frac{\sqrt{2}}{2}\exp\left(\frac{\left(\sum_{t=1}^{T} c_t\right)^2}{4T}\right).$$

We can now use this betting strategy in the expert reduction in Theorem 9.17, setting $h(x) = \sqrt{4T(x+\frac{1}{2}\ln 2)}$, to have

$$\text{Regret}_T(\boldsymbol{u}) \leq \sqrt{4T\left(KL(\boldsymbol{u};\boldsymbol{\pi}) + \frac{1}{2}\ln 2\right)}. \tag{9.9}$$

Note that this betting strategy could also be used in the OCO reduction. Given that we removed the logarithmic term in the exponent, in the 1-dimensional case, we would obtain a regret of

$$\text{Regret}_T(u) \leq O\left(|u|\sqrt{T\ln\left(\frac{|u|\sqrt{T}}{\epsilon}+1\right)+\epsilon}\right),$$

where we gained in the $\sqrt{T}$ term inside the logarithmic, instead of the T term of the KT algorithm. This implies that now we can set ϵ to $\sqrt{T}$ and obtain an asymptotic rate of $O(\sqrt{T})$ rather than $O(\sqrt{T\ln T})$.

9.7 History Bits

The keyword "parameter-free" has been introduced in Chaudhuri et al. [2009] for a similar strategy for the learning with expert problem. It is now used as an umbrella term for all online algorithms that *guarantee the optimal regret*

uniformly over the competitor class. Another less used name to denote the exact same property is "comparator-adaptive" [van der Hoeven et al., 2020]. Note that, given the allure of the name "parameter-free", the term has been adopted in many other domains, with many different meanings. However, when used in the online learning literature, it has *only* the meaning we specify above. This means that in this literature "parameter-free" algorithms might still require the knowledge of other characteristics of the problem, for example, the Lipschitz constant of the losses or the number of rounds. This is fine: it is a technical term and we can assign to it any definition we like, as for "smooth" or "universal".

The first algorithm for 1-d parameter-free OCO is from Streeter and McMahan [2012], but the bound was suboptimal. The algorithm was then extended to Hilbert spaces in Orabona [2013], still with a suboptimal bound. The optimal bound in Hilbert space was obtained in McMahan and Orabona [2014] that also obtains the regret upper bound

$$O\left(\|\boldsymbol{u}\|_2\sqrt{T\ln\left(\frac{\sqrt{T}\|\boldsymbol{u}\|_2}{\epsilon}+1\right)}+\epsilon\right),$$

where ϵ is a parameter of the algorithm. This variant allows to remove the factor T in the logarithm by setting $\epsilon = O(\sqrt{T})$, at the price of having a $O(\sqrt{T})$ regret versus $\boldsymbol{u} = \mathbf{0}$. The optimal constant for this kind of guarantees is obtained in Zhang et al. [2022] through a PDE-based analysis, proving also a matching lower bound.

The idea of using a coin-betting to do parameter-free OCO was introduced in Orabona and Pál [2016]. A different reduction from a betting algorithm to the specific case of linear regression with square loss was proposed by Vovk [2006]. The Krichevsky-Trofimov algorithm is from Krichevsky and Trofimov [1981] and its extension to the "continuous coin" is from Orabona and Pál [2016]. The regret-reward duality relationship was proved for the first time in McMahan and Orabona [2014]. Lemma A.2 is from Orabona and Pál [2016]. There are also more refined betting algorithms that allow to obtain parameter-free OCO algorithms that depend on the sum of the squared norm of the subgradients, rather than time [Cutkosky and Orabona, 2018], also in a scale-free way [Mhammedi and Koolen, 2020]. Recently, parameter-free algorithms have been also extended to some non-convex functions in the stochastic setting [Orabona and Pál, 2021].

The approach of using a coordinate-wise version of the coin-betting algorithm was proposed in the first paper on parameter-free OLO in Streeter and McMahan [2012]. Recently, the same approach with a special coin-betting algorithm was also used for optimization of deep neural networks [Orabona and Tommasi, 2017]. Theorem 9.9 is from Cutkosky and Orabona [2018]. Note that the original theorem is more general because it works even in Banach spaces. The idea of combining two parameter-free OLO algorithms to obtain the best of the two guarantees is from Cutkosky [2019b].

Orabona and Pál [2016] proposed a different way to transform a coin-betting algorithm into an OCO algorithm that works in $\mathbb{R}^d$ or even in Hilbert spaces. However, that approach seems to work on for the L_2 norm and it is not a black-box reduction. That said, the reduction in Orabona and Pál [2016] seems to have a better empirical performance compared to the one in Theorem 9.9.

There are also reductions that allow to transform an unconstrained OCO learner into a constrained one [Cutkosky and Orabona, 2018]. They work constructing a Lipschitz barrier function on the domain and passing to the algorithm the original subgradients plus the subgradients of the barrier function.

The first parameter-free algorithm for experts is from Chaudhuri et al. [2009], named NormalHedge, where they introduced the concept of ϵ-quantile regret and obtained a bound of $O(\sqrt{(T+\ln^2 d)(1+\ln\frac{1}{\epsilon})})$ as $T \to \infty$, but without a closed formula update. Then, Chernov and Vovk [2010] removed the spurious dependency on d, again with an update without a closed form. Orabona and Pál [2016] showed that this guarantee can be efficiently obtained through the novel reduction to coin-betting in Theorem 9.17. Later, this kind of regret guarantees were improved to depend on the sum of the squared losses rather than on time, but with an additional $\ln\ln T$ factor, in the Squint algorithm [Koolen and van Erven, 2015]. It is worth noting that the Squint algorithm can be interpreted exactly as a coin-betting algorithm plus the reduction in Theorem 9.17. The first lower bound for the ϵ-quantile regret is proved in Negrea et al. [2021].

The betting strategy in (9.6) and (9.7) are new, and derived from the shifted-KT potentials in Orabona and Pál [2016]. The guarantee is the same obtained by the shifted-KT potentials, but the analysis can be done without knowing the properties of the Gamma function.

Recently, coin-betting algorithms have found numerous applications, as the design of parameter-free online multi-task learning algorithms [Denevi et al., 2020], parameter-free particle-based algorithms [Sharrock and Nemeth, 2023, Sharrock et al., 2023a,b], and the derivation of time-uniform concentration inequalities (see Chapter 12).

9.8 Exercises

Problem 9.1. *Prove that $\ell_t(x) = -\ln(1 + z_t\, x)$ with $V = \{x \in \mathbb{R} : |x| \leq 1/2\}$ and $|z_t| \leq 1,\ t = 1, \dots, T$ are exp-concave. Then, using the Online Newton Step Algorithm, give an algorithm and a regret bound for a game with these losses. Finally, show a wealth guarantee of the corresponding coin-betting strategy.*

Problem 9.2. *Using the same proof technique in the section, find a betting strategy whose wealth depends on $\sum_{t=1}^{T} |c_t|$ rather than on T.*

Chapter 10

Multi-Armed Bandit

The Multi-Armed Bandit setting is similar to the Learning with Expert Advice (LEA) setting: In each round, we select one expert A_t and, differently from the full-information setting, we only observe the loss of that expert $g_{t,i}$. The aim is still to compete with the cumulative loss of the best expert in hindsight. The observed losses can be adversarial or stochastic, giving rise to adversarial and stochastic multi-armed bandits.

10.1 Adversarial Multi-Armed Bandit

As in the learning with expert case, we need randomization in order to have a sublinear regret. Indeed, this is just a harder problem than LEA. However, we will assume that the adversary is **oblivious**, that is, he decides the losses of all the rounds before the game starts, but with the knowledge of the online algorithm. This makes the losses deterministic quantities and it avoids the inadequacy in our definition of regret when the adversary is adaptive (see Arora et al. [2012]).

This kind of problems where we do not receive the full-information, i.e., we do not observe the loss vector, are called **bandit problems**. The name comes from the problem of a gambler who plays a pool of slot machines, that can be called "one-armed bandits". On each round, the gambler places his bet on a slot machine and his goal is to win almost as much money as if he had known in advance which slot machine would return the maximal total reward.

In this problem, we clearly have an *exploration-exploitation trade-off*. In fact, on one hand we would like to play at the slot machine which, based on previous rounds, we believe will give us the biggest win. On the other hand, we have to explore the slot machines to find the best ones. On each round, we have to solve this trade-off.

Given that we do not observe completely observe the loss, we cannot use our two frameworks: Online Mirror Descent (OMD) and Follow-The-Regularized-Leader (FTRL) both needs the loss functions or at least lower bounds to them.

One way to solve this issue is to construct *stochastic estimates* of the unknown losses. This is a natural choice given that the prediction strategy has to be a randomized one even in the full-information setting, as we saw in Section 6.8. So, in each round t we construct a probability distribution over the arms $\boldsymbol{x}_t$ and we sample one action A_t according to this probability distribution. Then, we only observe the coordinate A_t of the loss vector $\boldsymbol{g}_t \in \mathbb{R}^d$. One possibility to have a stochastic estimate of the losses is to use an *importance-weighted estimator*: Construct the estimator $\tilde{\boldsymbol{g}}_t$ of the unknown vector $\boldsymbol{g}_t$ in the following way:

$$\tilde{\boldsymbol{g}}_t = \begin{cases} \frac{g_{t,i}}{x_{t,i}}, & i = A_t \\ 0, & \text{otherwise} \end{cases} .$$

Note that this estimator has all the coordinates equal to 0, except the coordinate corresponding the arm that was pulled.

This estimator is unbiased, that is $\mathbb{E}_{A_t}[\tilde{\boldsymbol{g}}_t] = \boldsymbol{g}_t$. To see why, note that $\tilde{g}_{t,i} = \mathbf{1}[A_t = i]\frac{g_{t,i}}{x_{t,i}}$ and $\mathbb{E}_{A_t}[\mathbf{1}[A_t = i]] = x_{t,i}$. Hence, for $i = 1, \dots, d$, we have

$$\mathbb{E}_{A_t}[\tilde{g}_{t,i}] = \mathbb{E}_{A_t}\left[\mathbf{1}[A_t = i]\frac{g_{t,i}}{x_{t,i}}\right] = \frac{g_{t,i}}{x_{t,i}}\mathbb{E}_{A_t}[\mathbf{1}[A_t = i]] = g_{t,i} .$$

Let's also calculate the (uncentered) variance of the coordinates of this estimator. We have

$$\mathbb{E}_{A_t}[\tilde{g}_{t,i}^2] = \mathbb{E}_{A_t}\left[\mathbf{1}[A_t = i]\frac{g_{t,i}^2}{x_{t,i}^2}\right] = \frac{g_{t,i}^2}{x_{t,i}} \,.$$

We can now think of using OMD with these estimated losses and an entropic regularizer. Hence, assume $\|\boldsymbol{g}_t\|_\infty \leq L_\infty$ and set $\psi : \mathbb{R}_+^d \to \mathbb{R}$ defined as $\psi(\boldsymbol{x}) = \sum_{i=1}^d x_i \ln x_i$, that is the unnormalized negative entropy. Also, set $\boldsymbol{x}_1 = [1/d, \dots, 1/d]$. Using the OMD analysis, we have

$$\sum_{t=1}^T \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{u} \rangle \leq \frac{\ln d}{\eta} + \frac{\eta}{2} \sum_{t=1}^T \|\tilde{\boldsymbol{g}}_t\|_\infty^2 \,.$$

We can now take the expectation at both sides and use the fact that

$$\mathbb{E}[\|\tilde{\boldsymbol{g}}_t\|_\infty^2] = \mathbb{E}\left[\frac{g_{t,A_t}^2}{x_{t,A_t}^2}\right] = \mathbb{E}\left[\mathbb{E}\left[\frac{g_{t,A_t}^2}{x_{t,A_t}^2}\middle| A_1, \dots, A_{t-1}\right]\right] = \mathbb{E}\left[\sum_{i=1}^d \frac{g_{t,i}^2}{x_{t,i}}\right],$$

to get

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^T g_{t,A_t}\right] - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle &= \mathbb{E}\left[\sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle\right] - \sum_{t=1}^T \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle = \mathbb{E}\left[\sum_{t=1}^T \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^T \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{u} \rangle\right] \\ &\leq \frac{\ln d}{\eta} + \frac{\eta}{2} \sum_{t=1}^T \mathbb{E}[\|\tilde{\boldsymbol{g}}_t\|_\infty^2] \leq \frac{\ln d}{\eta} + \frac{\eta}{2} \sum_{t=1}^T \sum_{i=1}^d \mathbb{E}\left[\frac{g_{t,i}^2}{x_{t,i}}\right] \,. \end{aligned} \tag{10.1}$$

We are now in troubles, because the terms in the sum scale as $\max_{i=1,\dots,d} \frac{1}{x_{t,i}}$. So, we need a way to control the smallest probability over the arms.

One way to do it, is to take a convex combination of $\boldsymbol{x}_t$ and a uniform probability. That is, we can predict with $\tilde{\boldsymbol{x}}_t = (1-\alpha)\boldsymbol{x}_t + \alpha[1/d, \dots, 1/d]^\top$, where α will be chosen in the following. So, α can be seen as the minimum amount of exploration we require to the algorithm. Its value will be chosen by the regret analysis to optimally trade-off exploration vs exploitation. The resulting algorithm is in Algorithm 10.1.

Algorithm 10.1 Exponential Weights with Explicit Exploration for Multi-Armed Bandit

Require: $\eta, \alpha > 0$
Set $\boldsymbol{x}_1 = [1/d, \dots, 1/d]$
for $t = 1$ **to** T **do**
  Set $\tilde{\boldsymbol{x}}_t = (1-\alpha)\boldsymbol{x}_t + \alpha[1/d, \dots, 1/d]^\top$
  Draw A_t according to $P(A_t = i) = \tilde{x}_{t,i}$
  Select expert A_t
  Observe *only* the loss of the selected arm $g_{t,A_t} \in [-L_\infty, L_\infty]$ and pay it
  Construct the estimate $\tilde{g}_{t,i} = \begin{cases} \frac{g_{t,i}}{\tilde{x}_{t,i}}, & i = A_t \\ 0, & \text{otherwise} \end{cases}$ for $i = 1, \dots, d$
  $x_{t+1,i} \propto x_{t,i} \exp(-\eta \tilde{g}_{t,i}),\ i = 1, \dots, d$
end for

The same probability distribution is used in the estimator:

$$\tilde{\boldsymbol{g}}_t = \begin{cases} \frac{g_{t,i}}{\tilde{x}_{t,i}}, & i = A_t \\ 0, & \text{otherwise} \end{cases} \,. \tag{10.2}$$

So, we now have that $\frac{1}{\tilde{x}_{t,i}} \leq \frac{d}{\alpha}$. However, we pay a price in the bias introduced:

$$\sum_{t=1}^{T} \langle \tilde{\boldsymbol{g}}_t, \tilde{\boldsymbol{x}}_t - \boldsymbol{u} \rangle = (1-\alpha) \sum_{t=1}^{T} \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle + \frac{\alpha}{d} \sum_{t=1}^{T} \sum_{i=1}^{d} \tilde{g}_{t,i} - \alpha \sum_{t=1}^{T} \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{u} \rangle \; .$$

Observing that $\mathbb{E}[\sum_{i=1}^{d} \tilde{g}_{t,i}] = \sum_{i=1}^{d} g_{t,i} \leq d L_\infty$ and $\mathbb{E}[-\langle \tilde{\boldsymbol{g}}_t, \boldsymbol{u} \rangle] = -\langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \leq L_\infty$, we have

$$\mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{\boldsymbol{g}}_t, \tilde{\boldsymbol{x}}_t - \boldsymbol{u} \rangle\right] \leq (1-\alpha) \mathbb{E}\left[\sum_{t=1}^{T} \langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle\right] + 2\alpha L_\infty T = (1-\alpha)\mathbb{E}[\mathrm{Regret}_T(\boldsymbol{u})] + 2\alpha L_\infty T \; .$$

Putting together the last inequality and the upper bound to the expected regret in (10.1), we have

$$\mathbb{E}\left[\sum_{t=1}^{T} g_{t,A_t}\right] - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \leq \frac{(1-\alpha)\ln d}{\eta} + \frac{(1-\alpha)\eta d^2 L_\infty^2 T}{2\alpha} + 2\alpha L_\infty T \leq \frac{\ln d}{\eta} + \frac{\eta d^2 L_\infty^2 T}{2\alpha} + 2\alpha L_\infty T \; .$$

Setting $\alpha \propto \sqrt{d^2 L_\infty \eta}$ and $\eta \propto \left(\frac{\ln d}{d L_\infty^{3/2} T}\right)^{2/3}$, we obtain a regret of $O(L_\infty (dT)^{2/3} \ln^{1/3} d)$ as $T \to \infty$.

We did obtain a sublinear regret, but it is way worse than the $O(\sqrt{T \ln d})$ of the full-information case. However, while it is expected that the bandit case must be more difficult than the full information one, it turns out that this is not the optimal strategy.

10.1.1 Exponential-weight algorithm for Exploration and Exploitation: Exp3

It turns out that the algorithm above actually works, even without the mixing with the uniform distribution! We were just too loose in our regret guarantee. So, we will analyse the following algorithm, that is called Exponential-weight algorithm for Exploration and Exploitation (Exp3), that is nothing else than OMD with entropic regularizer and stochastic estimates of the losses. Note that now we will assume that $g_{t,i} \in [0, L_\infty]$.

Algorithm 10.2 Exp3

Require: $\eta > 0$

$\boldsymbol{x}_1 = [1/d, \dots, 1/d]$

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

  Draw A_t according to $P(A_t = i) = x_{t,i}$

  Select expert A_t

  Observe *only* the loss of the selected arm $g_{t,A_t} \in [0, L_\infty]$ and pay it

  Construct the estimate $\tilde{g}_{t,i} = \begin{cases} \frac{g_{t,i}}{x_{t,i}}, & i = A_t \\ 0, & \text{otherwise} \end{cases}$ for $i = 1, \dots, d$

  $x_{t+1,i} \propto x_{t,i} \exp(-\eta \tilde{g}_{t,i}),\ i = 1, \dots, d$

end for

This time we will use the local norm regret bound for OMD. The reason is that, when we use the strong convexity, we are upper bounding the terms in the sum with the inverse of the smallest eigenvalue of the Hessian of the regularizer. However, we can do better if we consider the local norms. In fact, in the coordinates where $\boldsymbol{x}_t$ is small, we have a smaller growth of the divergence. This can be seen also graphically in Figure 10.1. Indeed, for the entropic regularizer, we have that the Hessian is a diagonal matrix:

$$(\nabla^2 \psi(\boldsymbol{x}))_{ii} = \frac{1}{x_i},\ i = 1, \dots, d \; .$$

Summing the inequality of Lemma 6.16 with $t = 1, \dots, T$, the above expression of the Hessian gives a regret of

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq \frac{B_\psi(\boldsymbol{u}; \boldsymbol{x}_1)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{i=1}^{d} z_{t,i} g_{t,i}^2,$$

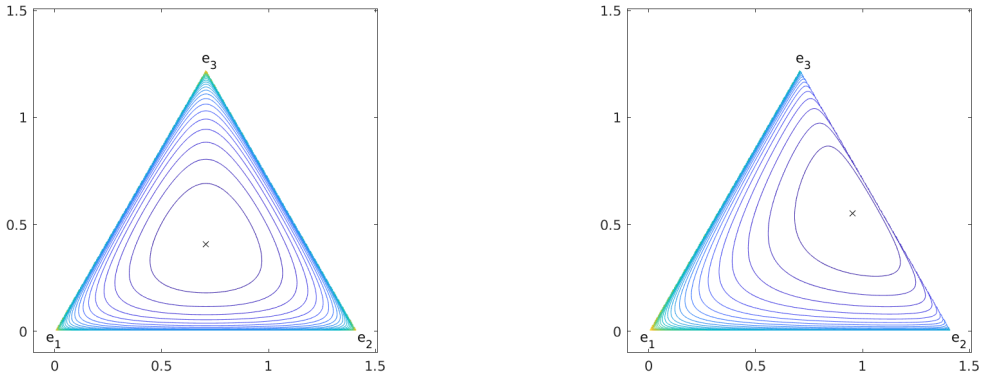

Figure 10.1: Contour plots of the KL divergence in 3-dimensions when $\boldsymbol{x}_t = [1/3, 1/3, 1/3]$ (left) and when $\boldsymbol{x}_t = [0.1, 0.45, 0.45]$ (right).

where $\boldsymbol{z}_t = \alpha_t \boldsymbol{x}_t + (1-\alpha_t)\boldsymbol{x}_{t+1}$ and $\alpha_t \in [0,1]$. Note that for any $\alpha_t \in [0,1]$ $\boldsymbol{z}_t$ is in the probability simplex, so this upper bound is always better than

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \le \frac{B_\psi(\boldsymbol{u}; \boldsymbol{x}_1)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\infty^2,$$

that we derived just using the strong convexity of the entropic regularizer.

However, we do not know the exact value of $\boldsymbol{z}_t$, but only that it is on the line segment between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t+1}$. Yet, if you could say that $z_{t,i} \propto x_{t,i}$, in the bandit case we would obtain an expected regret guarantee of $O(\sqrt{dT \ln d})$, greatly improving the bound we proved above! In other words, it might be possible to get the regret guarantee

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \le \frac{\ln d}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{i=1}^{d} x_{t,i} g_{t,i}^2, \ \forall \boldsymbol{u} \in V, \tag{10.3}$$

where $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_1 = 1, x_i \ge 0\}$. Fortunately, we have an alternative expression in Lemma 6.16 to help us.

Remark 10.1. *It is possible to prove* (10.3) *from first principles using the specific properties for the entropic regularizer rather than using Lemma 6.16. However, we prefer not to do it because such proof would not shed any light on what is actually going on.*

To use Lemma 6.16, we need a handle on $z'_{t,i}$ that lies on the line segment between $\boldsymbol{x}_t$ and $\tilde{\boldsymbol{x}}_{t+1} = \operatorname{argmin}_{\boldsymbol{x} \in \mathbb{R}^d_+} \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{1}{\eta_t} B_\psi(\boldsymbol{x}; \boldsymbol{x}_t)$. Given that we only need an upper bound, we can just take a look at $x_{t,i}$ and $\tilde{x}_{t+1,i}$ and see which one is bigger. This is easy to do: using the definition of $\tilde{\boldsymbol{x}}_t$, we have

$$\ln(\tilde{x}_{t+1,i}) + 1 = \ln(x_{t,i}) + 1 - \eta g_{t,i},$$

that is

$$\tilde{x}_{t+1,i} = x_{t,i} \exp(-\eta g_{t,i}) \ .$$

Assuming $g_{t,i} \ge 0$, we have $\tilde{x}_{t+1,i} \le x_{t,i}$ that implies $z_{t,i} \le x_{t,i}$.

Overall, we have the following improved regret guarantee for the Learning with Experts setting with positive losses.

Theorem 10.2. *Assume $g_{t,i} \ge 0$ for $t = 1, \dots, T$ and $i = 1, \dots, d$. Let $V = \{\boldsymbol{x} \in \mathbb{R}^d : \|\boldsymbol{x}\|_1 = 1, x_i \ge 0\}$ and $\eta > 0$. Using OMD with the entropic regularizer $\psi : \mathbb{R}^d_+ \to \mathbb{R}$ defined as $\psi(\boldsymbol{x}) = \sum_{i=1}^d x_i \ln x_i$, learning rate η, and $\boldsymbol{x}_1 = [1/d, \dots, 1/d]$ gives the following regret guarantee*

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t \rangle - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \le \frac{\ln d}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{i=1}^{d} x_{t,i} g_{t,i}^2, \ \forall \boldsymbol{u} \in V \ .$$

Armed with this new tool, we can now turn to the multi-armed bandit problem again.

Let's now consider the OMD with entropic regularizer, learning rate η, and set $\tilde{\boldsymbol{g}}_t$ equal to the stochastic estimate of $\boldsymbol{g}_t$, as in Algorithm 10.2. Applying Theorem 10.2 and taking the expectation, we have

$$\mathbb{E}\left[\sum_{t=1}^{T} g_{t,A_t}\right] - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}\rangle = \mathbb{E}\left[\sum_{t=1}^{T}\langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x}_t\rangle - \sum_{t=1}^{T}\langle \tilde{\boldsymbol{g}}_t, \boldsymbol{u}\rangle\right] \le \frac{\ln d}{\eta} + \frac{\eta}{2}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{i=1}^{d} x_{t,i}\tilde{g}_{t,i}^2\right] .$$

Now, focusing on the terms $\mathbb{E}[x_{t,i}\tilde{g}_{t,i}^2]$, we have

$$\mathbb{E}\left[\sum_{i=1}^{d} x_{t,i}\tilde{g}_{t,i}^2\right] = \mathbb{E}\left[\mathbb{E}\left[\sum_{i=1}^{d} x_{t,i}\tilde{g}_{t,i}^2 \middle| A_1, \dots, A_{t-1}\right]\right] = \mathbb{E}\left[\sum_{i=1}^{d} x_{t,i}\frac{g_{t,i}^2}{x_{t,i}}\right] \le dL_\infty^2 . \tag{10.4}$$

So, setting $\eta = \sqrt{\frac{2\ln d}{dL_\infty^2 T}}$, we have

$$\mathbb{E}\left[\sum_{t=1}^{T} g_{t,A_t}\right] - \sum_{t=1}^{T}\langle \boldsymbol{g}_t, \boldsymbol{u}\rangle \le L_\infty\sqrt{2dT\ln d} .$$

So, with a tighter analysis we showed that, even without an explicit exploration term, OMD with entropic regularizer solves the multi-armed bandit problem paying only a factor $\sqrt{d}$ more than the full information case. However, this is still not the optimal regret!

In the next section, we will see that changing the regularizer, *with the same analysis*, will remove the $\sqrt{\ln d}$ term in the regret.

10.1.2 Optimal Regret Using OMD with Tsallis Entropy

In this section, we present the Implicitly Normalized Forecaster (INF) also known as OMD with Tsallis entropy for multi-armed bandit.

Define $\psi_q : \mathbb{R}_+^d \to \mathbb{R}$ as $\psi_q(\boldsymbol{x}) = \frac{1}{1-q}\left(1 - \sum_{i=1}^{d} x_i^q\right)$, where $q \in [0,1]$ and in $q = 1$ we extend the function by continuity. This is the negative **Tsallis entropy** of the vector $\boldsymbol{x}$. This is a strict generalization of the Shannon entropy, because when q goes to 1, $\psi_q(\boldsymbol{x})$ converges to the negative (Shannon) entropy of $\boldsymbol{x}$.

We will instantiate OMD with this regularizer for the multi-armed problem, as in Algorithm 10.3.

Algorithm 10.3 INF Algorithm (OMD with Tsallis Entropy for Multi-Armed Bandit)

Require: $\eta > 0$

$\boldsymbol{x}_1 = [1/d, \dots, 1/d]$

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

  Draw A_t according to $P(A_t = i) = x_{t,i}$

  Select expert A_t

  Observe *only* the loss of the selected arm $g_{t,A_t} \in [0, L_\infty]$ and pay it

  Construct the estimate $\tilde{g}_{t,i} = \begin{cases} \frac{g_{t,i}}{x_{t,i}}, & i = A_t \\ 0, & \text{otherwise} \end{cases}$ for $i = 1, \dots, d$

  $\tilde{\boldsymbol{x}}_{t+1} = \operatorname{argmin}_{\boldsymbol{x}\in V}\ \eta\langle \tilde{\boldsymbol{g}}_t, \boldsymbol{x}\rangle - \frac{1}{1-q}\sum_{i=1}^{d} x_i^q + \frac{q}{1-q}\sum_{i=1}^{d} x_{t,i}^{q-1}x_i$

end for

Note that $\operatorname{argmin}_{\boldsymbol{x}}\ \psi_q(\boldsymbol{x}) = \frac{1}{d}$ and $\min_{\boldsymbol{x}} \psi_q(\boldsymbol{x}) = \frac{1-d^{1-q}}{1-q}$.

We will not use any interpretation of this regularizer from the information theory point of view. As we will see in the following, the only reason to choose it is its Hessian. In fact, the Hessian of this regularizer is still diagonal and it is equal to

$$(\nabla^2\psi_q(\boldsymbol{x}))_{ii} = \frac{q}{x_i^{2-q}} .$$

Now, we can use again Lemma 6.16. So, for any $\boldsymbol{u} \in V$, we obtain

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq \frac{B_{\psi_q}(\boldsymbol{u}; \boldsymbol{x}_1)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \boldsymbol{g}_t^\top (\nabla^2 \psi_q(\boldsymbol{z}'_t))^{-1} \boldsymbol{g}_t = \frac{d^{1-q} - \sum_{i=1}^{d} u_i^q}{\eta(1-q)} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{i=1}^{d} g_{t,i}^2 (z'_{t,i})^{2-q},$$

where $\boldsymbol{z}'_t = \alpha_t \boldsymbol{x}_t + (1-\alpha_t)\tilde{\boldsymbol{x}}_{t+1}$, $\alpha_t \in [0,1]$, and $\tilde{\boldsymbol{x}}_{t+1} = \operatorname{argmin}_{\boldsymbol{x} \in \mathbb{R}^d_+} \langle \boldsymbol{g}_t, \boldsymbol{x} \rangle + \frac{1}{\eta_t} B_{\psi_q}(\boldsymbol{x}; \boldsymbol{x}_t)$.

As we did for Exp3, now we need an upper bounds to the $z'_{t,i}$. From the definition of $\tilde{\boldsymbol{x}}_t$ and ψ, we have

$$-\frac{q}{1-q} \tilde{x}_{t+1,i}^{q-1} = -\frac{q}{1-q} x_{t,i}^{q-1} - \eta g_{t,i},$$

that is

$$\tilde{x}_{t+1,i} = \frac{x_{t,i}}{\left(1 + \frac{1-q}{q} \eta g_{t,i} x_{t,i}^{1-q}\right)^{\frac{1}{1-q}}} .$$

So, if $g_{t,i} \geq 0$, $\tilde{x}_{t+1,i} \leq x_{t,i}$, that implies that $z'_{t,i} \leq x_{t,i}$.

Hence, putting all together, we have

$$\sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle \leq \frac{d^{1-q} - \sum_{i=1}^{d} u_i^q}{\eta(1-q)} + \frac{\eta}{2q} \sum_{t=1}^{T} \sum_{i=1}^{d} g_{t,i}^2 x_{t,i}^{2-q} .$$

We can now specialize the above reasoning, considering $q = 1/2$ in the Tsallis entropy, to obtain the following theorem.

Theorem 10.3. *Assume $g_{t,i} \in [0, L_\infty]$. Set $q = 1/2$ and $\boldsymbol{x}_1 = [1/d, \dots, 1/d]$. Then, Algorithm 10.3 satisfies*

$$\mathbb{E}\left[\sum_{t=1}^{T} g_{t,A_t}\right] - \sum_{t=1}^{T} \langle \boldsymbol{g}_t, \boldsymbol{u} \rangle \leq \frac{2\sqrt{d}}{\eta} + \eta \sqrt{d} L_\infty^2 T .$$

Proof. We only need to calculate the terms

$$\mathbb{E}\left[\sum_{i=1}^{d} \tilde{g}_{t,i}^2 x_{t,i}^{3/2}\right] .$$

Proceeding as in (10.4), we obtain

$$\mathbb{E}\left[\sum_{i=1}^{d} \tilde{g}_{t,i}^2 x_{t,i}^{3/2}\right] = \mathbb{E}\left[\mathbb{E}\left[\sum_{i=1}^{d} x_{t,i}^{3/2} \tilde{g}_{t,i}^2 \middle| A_1, \dots, A_{t-1}\right]\right] = \mathbb{E}\left[\sum_{i=1}^{d} x_{t,i}^{3/2} \frac{g_{t,i}^2}{x_{t,i}^2} x_{t,i}\right] = \mathbb{E}\left[\sum_{i=1}^{d} g_{t,i}^2 \sqrt{x_{t,i}}\right]$$

$$\leq \mathbb{E}\left[\sqrt{\sum_{i=1}^{d} g_{t,i}^2} \sqrt{\sum_{i=1}^{d} g_{t,i}^2 x_{t,i}}\right] \leq L_\infty^2 \sqrt{d} . \qquad \square$$

Choosing $\eta \propto \frac{1}{L_\infty \sqrt{T}}$, we finally obtain an expected regret of $O(L_\infty \sqrt{dT})$ as $T \to \infty$, that can be proved to be the optimal one.

There is one last thing: How do we compute the predictions of this algorithm? In each step, we have to solve a constrained optimization problem. So, we can write the corresponding Lagrangian:

$$L(\boldsymbol{x}, \beta) = \sum_{i=1}^{d} \left(\eta \tilde{g}_{t,i} + \frac{q}{1-q} x_{t,i}^{q-1}\right) x_i - \frac{1}{1-q} \sum_{i=1}^{d} x_i^q + \beta \left(\sum_{i=1}^{d} x_i - 1\right) .$$

From the KKT conditions, we have

$$x_{t+1,i} = \left[\frac{1-q}{q}\left(\beta + \frac{q}{1-q} x_{t,i}^{q-1} + \eta \tilde{g}_{t,i}\right)\right]^{\frac{1}{q-1}} .$$

and we also know that $\sum_{i=1}^{d} x_{t+1,i} = 1$. So, we have a 1-dimensional problem in β that must be solved in each round.

10.2 Stochastic Bandits

We will now consider the *stochastic bandit* setting. Here, each arm is associated with an unknown probability distribution. At each time step, the algorithm selects one arm A_t and it receives a loss (or reward) g_{t,A_t} drawn i.i.d. from the distribution of the arm A_t. We focus on minimizing the *pseudo-regret*, that is the regret with respect to the optimal action in expectation, rather than the optimal action on the sequence of realized losses:

$$\text{Regret}_T := \mathbb{E}\left[\sum_{t=1}^{T} g_{t,A_t}\right] - \min_{i=1,\dots,d} \mathbb{E}\left[\sum_{t=1}^{T} g_{t,i}\right] = \mathbb{E}\left[\sum_{t=1}^{T} g_{t,A_t}\right] - \min_{i=1,\dots,d} \mu_i,$$

where we denoted by μ_i the expectation of the distribution associated with the arm i.

Remark 10.4. *The usual notation in the stochastic bandit literature is to consider rewards instead of losses. Instead, to keep our notation coherent with the OCO literature, we will consider losses. The two things are completely equivalent up to a multiplication by* -1.

Before presenting our first algorithm for stochastic bandits, we will introduce some basic notions on concentration inequalities that will be useful in our definitions and proofs.

10.2.1 Concentration Inequalities Bits

Suppose that $X_1, X_2, \dots, X_n$ is a sequence of independent and identically distributed random variables and with mean $\mu = \mathbb{E}[X_1]$ and variance $\sigma = \text{Var}[X_1]$. Having observed $X_1, X_2, \dots, X_n$ we would like to estimate the common mean μ. The most natural estimator is the *empirical mean*

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} X_i \,.$$

Linearity of expectation shows that $\mathbb{E}[\hat{\mu}] = \mu$, which means that $\hat{\mu}$ is an *unbiased estimator* of μ. Yet, $\hat{\mu}$ is a random variable itself. So, can we quantify how far $\hat{\mu}$ will be from μ?

We could use Chebyshev's inequality to upper bound the probability that $\hat{\mu}$ is far from μ:

$$\mathbb{P}\{|\hat{\mu} - \mu| \geq \epsilon\} \leq \frac{\text{Var}[\hat{\mu}]}{\epsilon^2} \,.$$

Using the fact that $\text{Var}[\hat{\mu}] = \frac{\sigma^2}{n}$, we have that

$$\mathbb{P}\{|\hat{\mu} - \mu| \geq \epsilon\} \leq \frac{\sigma^2}{n\epsilon^2} \,.$$

So, we can expect the probability of having a "bad" estimate to go to zero as one over the number of samples in our empirical mean. Is this the best we can get? To understand what we can hope for, let's take a look at the central limit theorem.

We know that, defining $S_n = \sum_{t=1}^{n}(X_i - \mu)$, $\frac{S_n}{\sqrt{n\sigma^2}} \to N(0,1)$, the standard Gaussian distribution, as n goes to infinity. This means that

$$\mathbb{P}\{\hat{\mu} - \mu \geq \epsilon\} = \mathbb{P}\{S_n \geq n\epsilon\} = \mathbb{P}\left\{\frac{S_n}{\sqrt{n\sigma^2}} \geq \sqrt{\frac{n}{\sigma^2}}\epsilon\right\} \approx \int_{\epsilon\sqrt{\frac{n}{\sigma^2}}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \mathrm{d}x,$$

where the approximation comes from the central limit theorem. The integral cannot be calculated with a closed form, but we can easily upper bound it. Indeed, for $a > 0$, we have

$$\int_a^{\infty} \exp\left(-\frac{x^2}{2}\right) \mathrm{d}x = \int_a^{\infty} \frac{x}{x} \exp\left(-\frac{x^2}{2}\right) \mathrm{d}x \leq \frac{1}{a}\int_a^{\infty} x \exp\left(-\frac{x^2}{2}\right) \mathrm{d}x = \frac{1}{a}\exp\left(-\frac{a^2}{2}\right) \,.$$

Hence, we have

$$\mathbb{P}\{\hat{\mu} - \mu \geq \epsilon\} \lessapprox \sqrt{\frac{\sigma^2}{2\pi\epsilon^2 n}} \exp\left(-\frac{n\epsilon^2}{2\sigma^2}\right) . \tag{10.5}$$

This is better than what we got with Chebyshev's inequality and we would like to obtain an exact bound with a similar asymptotic rate. To do that, we will focus our attention on *subgaussian* random variables.

Definition 10.5 (Subgaussian Random Variable)**.** *We say that a random variable is σ-**subgaussian** if for all $\lambda \in \mathbb{R}$ we have that $\mathbb{E}[\exp(\lambda X)] \leq \exp(\lambda^2\sigma^2/2)$.*

Example 10.6. *The following random variable are subgaussian:*

- *If X is Gaussian with mean zero and variance σ^2, then X is σ-subgaussian.*
- *If X has mean zero and $X \in [a, b]$ almost surely, then X is $(b-a)/2$-subgaussian.*

We have the following properties for subgaussian random variables.

Lemma 10.7 ([Lattimore and Szepesvári, 2020, Lemma 5.4])**.** *Assume that X_1 and X_2 are independent and σ_1-subgaussian and σ_2-subgaussian respectively. Then,*

(a) $\mathbb{E}[X_1] = 0$ and $\mathrm{Var}[X_1] \leq \sigma_1^2$.

(b) cX_1 is $|c|\sigma_1$-subgaussian.

(c) $X_1 + X_2$ is $\sqrt{\sigma_1^2 + \sigma_2^2}$-subgaussian.

Subgaussians random variables behaves like Gaussian random variables, in the sense that their tail probabilities are upper bounded by the ones of a Gaussian of variance σ^2. To prove it, let's first state the Markov's inequality.

Theorem 10.8 (Markov's inequality)**.** *For a non-negative random variable X and $\epsilon > 0$, we have that $\mathbb{P}\{X \geq \epsilon\} \leq \frac{\mathbb{E}[X]}{\epsilon}$.*

With Markov's inequality, we can now formalize the above statement on subgaussian random variables.

Theorem 10.9. *If a random variable is σ-subgaussian, then $\mathbb{P}\{X \geq \epsilon\} \leq \exp\left(-\frac{\epsilon^2}{2\sigma^2}\right)$.*

Proof. For any $\lambda > 0$, we have

$$\mathbb{P}\{X \geq \epsilon\} = \mathbb{P}\{\exp(\lambda X) \geq \exp(\lambda\epsilon)\} \leq \frac{\mathbb{E}[\exp(\lambda X)]}{\exp(\lambda\epsilon)} \leq \exp(\lambda^2\sigma^2/2 - \lambda\epsilon) .$$

Minimizing the right hand side of the inequality w.r.t. λ, we have the stated result. □

An easy consequence of the above theorem is that the empirical average of subgaussian random variables concentrates around its expectation, *with the same asymptotic rate in* (10.5).

Corollary 10.10. *Assume that $X_i - \mu$ are independent, σ-subgaussian random variables. Then, for any $\epsilon \geq 0$, we have*

$$\mathbb{P}\left\{\hat{\mu} \geq \mu + \epsilon\right\} \leq \exp\left(-\frac{n\epsilon^2}{2\sigma^2}\right) \qquad \textit{and} \qquad \mathbb{P}\left\{\hat{\mu} \leq \mu - \epsilon\right\} \leq \exp\left(-\frac{n\epsilon^2}{2\sigma^2}\right),$$

where $\hat{\mu} = \frac{1}{n}\sum_{i=1}^n X_i$.

Equating the upper bounds on the r.h.s. of the inequalities in the Corollary to δ, we have the equivalent statement that, with probability at least $1 - 2\delta$, we have

$$\mu \in \left[\hat{\mu} - \sqrt{\frac{2\sigma^2 \ln\frac{1}{\delta}}{n}}, \hat{\mu} + \sqrt{\frac{2\sigma^2 \ln\frac{1}{\delta}}{n}}\right] .$$

Algorithm 10.4 Explore-Then-Commit Algorithm

Require: $T, m \in \mathbb{N}, 1 \leq m \leq \frac{T}{d}$
1: $S_{0,i} = 0, \hat{\mu}_{0,i} = 0,\ i = 1, \ldots, d$
2: **for** $t = 1$ **to** T **do**
3: Choose $A_t = \begin{cases} (t \mod d) + 1, & t \leq dm \\ \operatorname{argmin}_i\ \hat{\mu}_{dm,i}, & t > dm \end{cases}$
4: Observe g_{t,A_t} and pay it
5: $S_{t,i} = S_{t-1,i} + \mathbf{1}[A_t = i]$
6: $\hat{\mu}_{t,i} = \frac{1}{S_{t,i}} \sum_{j=1}^{t} g_{j,A_j} \mathbf{1}[A_j = i],\ i = 1, \ldots, d$
7: **end for**

10.2.2 Explore-Then-Commit Algorithm

We are now ready to present the most natural algorithm for the stochastic bandit setting, called Explore-Then-Commit (ETC) algorithm. That is, we first identify the best arm over md exploration rounds and then we commit to it. This algorithm is summarized in Algorithm 10.4.

In the following, we will denote by $S_{t,i} = \sum_{j=1}^{t} \mathbf{1}[A_j = i]$, that is the number of times that the arm i was pulled in the first t rounds.

Define by $\mu^\star$ the expected loss of the arm with the smallest expectation, that is $\min_{i=1,\ldots,d}\ \mu_i$. Critical quantities in our analysis will be the *gaps*, $\Delta_i := \mu_i - \mu^\star$ for $i = 1, \ldots, d$, that measure the expected difference in losses between the arms and the optimal one. In particular, we can decompose the regret as a sum over the arms of the expected number of times we pull an arm multiplied by its gap.

Lemma 10.11. *For any policy of selection of the arms, the regret is upper bounded by*

$$\mathrm{Regret}_T = \sum_{i=1}^{d} \mathbb{E}[S_{T,i}]\Delta_i \ .$$

Proof. Observe that

$$\sum_{t=1}^{T} g_{t,A_t} = \sum_{t=1}^{T} \sum_{i=1}^{d} g_{t,i} \mathbf{1}[A_t = i] \ .$$

Hence,

$$\begin{aligned}
\mathrm{Regret}_T &= \mathbb{E}\left[\sum_{t=1}^{T} g_{t,A_t}\right] - T\mu^\star = \mathbb{E}\left[\sum_{t=1}^{T} (g_{t,A_t} - \mu^\star)\right] = \sum_{i=1}^{d} \sum_{t=1}^{T} \mathbb{E}[\mathbf{1}[A_t = i](g_{t,i} - \mu^\star)] \\
&= \sum_{i=1}^{d} \sum_{t=1}^{T} \mathbb{E}[\mathbb{E}[\mathbf{1}[A_t = i](g_{t,i} - \mu^\star)|A_t]] = \sum_{i=1}^{d} \sum_{t=1}^{T} \mathbb{E}[\mathbf{1}[A_t = i]\mathbb{E}[g_{t,i} - \mu^\star|A_t]] \\
&= \sum_{i=1}^{d} \sum_{t=1}^{T} \mathbb{E}[\mathbf{1}[A_t = i]](\mu_i - \mu^\star) \ .
\end{aligned}$$

□

The above Lemma quantifies the intuition that in order to have a small regret we have to select the suboptimal arms less often then the best one.

We are now ready to prove the regret guarantee of the ETC algorithm.

Theorem 10.12. *Assume that the losses of the arms minus their expectations are* 1*-subgaussian and* $1 \leq m \leq T/d$*. Then, ETC guarantees a regret of*

$$\mathrm{Regret}_T \leq m \sum_{i=1}^{d} \Delta_i + (T - md) \sum_{i=1}^{d} \Delta_i \exp\left(-\frac{m\Delta_i^2}{4}\right) \ .$$

Proof. Let's assume without loss of generality that the optimal arm is the first one.

So, for $i \neq 1$, we have

$$\begin{aligned}\sum_{t=1}^{T} \mathbb{E}[\mathbf{1}[A_t = i]] &= m + (T - md)\mathbb{P}\left\{\hat{\mu}_{md,i} \leq \min_{j \neq i} \hat{\mu}_{md,j}\right\} \\ &\leq m + (T - md)\mathbb{P}\left\{\hat{\mu}_{md,i} \leq \hat{\mu}_{md,1}\right\} \\ &= m + (T - md)\mathbb{P}\left\{\hat{\mu}_{md,1} - \mu_1 - (\hat{\mu}_{md,i} - \mu_i) \geq \Delta_i\right\} .\end{aligned}$$

From Lemma 10.7, we have that $\hat{\mu}_{md,i} - \mu_i - (\hat{\mu}_{md,1} - \mu_1)$ is $\sqrt{2/m}$-subgaussian. So, from Theorem 10.9, we have

$$\mathbb{P}\left\{\hat{\mu}_{md,1} - \mu_1 - (\hat{\mu}_{md,i} - \mu_i) \geq \Delta_i\right\} \leq \exp\left(-\frac{m\Delta_i^2}{4}\right) . \qquad \square$$

The bound shows the trade-off between exploration and exploitation: if m is too big, we pay too much during the exploration phase (first term in the bound). On the other hand, if m is small, the probability to select a suboptimal arm increases (second term in the bound). Knowing all the gaps Δ_i, it is possible to choose m that minimizes the bound.

For example, in that case that $d = 2$, the regret is upper bounded by

$$m\Delta + (T - 2m)\Delta \exp\left(-m\frac{\Delta^2}{4}\right) \leq m\Delta + T\Delta \exp\left(-m\frac{\Delta^2}{4}\right),$$

that is minimized by

$$m = \frac{4}{\Delta^2} \ln \frac{T\Delta^2}{4} .$$

Remembering that m must be a natural number we can choose

$$m = \max\left(\left\lceil \frac{4}{\Delta^2} \ln \frac{T\Delta^2}{4} \right\rceil, 1\right) .$$

When $\frac{T\Delta^2}{4} \leq 1$, we select $m = 1$. So, we have $\Delta + (T - 2)\Delta \leq T\Delta \leq \frac{4}{\Delta}$. Hence, the regret is upper bounded by

$$\min\left(\Delta T, \frac{4}{\Delta}\left(1 + \max\left(\ln \frac{T\Delta^2}{4}, 0\right)\right) + \Delta\right) = O\left(\frac{\ln T}{\Delta}\right) \text{ as } T \text{ goes to infinity.}$$

The main drawback of this algorithm is that its optimal tuning depends on the gaps Δ_i. Assuming the knowledge of the gaps account to make the stochastic bandit problem completely trivial. However, its tuned regret bound gives us a baseline to which compare other bandit algorithms. In particular, in the next section we will present an algorithm that achieves the same asymptotic regret without any knowledge of the gaps.

10.2.3 Upper Confidence Bound Algorithm

The ETC algorithm has the disadvantage of requiring the knowledge of the gaps to tune the exploration phase. Moreover, it solves the exploration vs. exploitation trade-off in a clunky way. It would be better to have an algorithm that smoothly mixes exploration and exploitation *in a data-dependent way*. So, we now describe an optimal and adaptive strategy called Upper Confidence Bound (UCB) algorithm. It employs the principle of *optimism in the face of uncertainty*, to select in each round the arm that has the *potential to be the best one*.

UCB works keeping an estimate of the expected loss of each arm and also a confidence interval at a certain probability. Roughly speaking, we have that with probability at least $1 - \delta$

$$\mu_i \in \left[\hat{\mu}_i - \sqrt{\frac{2 \ln \frac{1}{\delta}}{S_{t-1,i}}}, \hat{\mu}_i\right],$$

where the "roughly" comes from the fact that $S_{t-1,i}$ is a random variable itself. Then, UCB will query the arm with the smallest lower bound, that is the one that could potentially have the smallest expected loss.

Algorithm 10.5 Upper Confidence Bound Algorithm

Require: $\alpha > 2, T \in \mathbb{N}$

$S_{0,i} = 0, \hat{\mu}_{0,i} = 0,\ i = 1, \dots, d$

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

  Choose $A_t = \text{argmin}_{i=1,\dots,d} \begin{cases} \mu_{t-1,i} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,i}}}, & \text{if } S_{t-1,i} \neq 0 \\ -\infty, & \text{otherwise} \end{cases}$

  Observe g_{t,A_t} and pay it

  $S_{t,i} = S_{t-1,i} + \mathbf{1}[A_t = i]$

  $\hat{\mu}_{t,i} = \frac{1}{S_{t,i}} \sum_{j=1}^{t} g_{t,A_t} \mathbf{1}[A_t = i],\ i = 1, \dots, d$

end for

Remark 10.13. *The name Upper Confidence Bound comes from the fact that traditionally stochastic bandits are defined over rewards, rather than losses. So, in our case we actually use the lower confidence bound in the algorithm. However, to avoid confusion with the literature, we still call it Upper Confidence Bound algorithm.*

The proof works by proving that, once we have queried a suboptimal arm enough times, we will pull it again only if its confidence interval or the one around the best arm do not contain the true means. In turn, the probability that the confidence intervals are wrong is small because we use concentration inequalities to construct them. Moreover, a delicate point is to deal with the fact that the number of times we pull an arm is a random variable, making the application of the concentration inequality delicate. We go around it considering the worst number of pulls and taking a union bound on it.

The algorithm is summarized in Algorithm 10.5 and we can prove the following regret bound.

Theorem 10.14. *Assume that the losses of the arms are* 1*-subgaussian and let* $\alpha > 2$*. Then, UCB guarantees a regret of*

$$\text{Regret}_T \leq \frac{\alpha}{\alpha - 2} \sum_{i=1}^{d} \Delta_i + \sum_{i:\Delta_i > 0} \frac{8\alpha \ln T}{\Delta_i} .$$

Proof. We analyze one arm at the time. Also, without loss of generality, assume that the optimal arm is the first one. For arm i, we want to prove that $\mathbb{E}[S_{T,i}] \leq \frac{8\alpha \ln T}{\Delta_i^2} + \frac{\alpha}{\alpha - 2}$.

The proof is based on the fact that once I have sampled an arm enough times, the probability to take a suboptimal arm is small.

Let $t^\star$ the biggest time index such that $S_{t^\star - 1,i} \leq \frac{8\alpha \ln T}{\Delta_i^2}$. If $t^\star = T$, then the statement above is true. Hence, we can safely assume $t^\star < T$. Now, for $t > t^\star$, we have

$$S_{t-1,i} > \frac{8\alpha \ln T}{\Delta_i^2} . \tag{10.6}$$

Consider $t > t^\star$ and such that $A_t = i$, then we claim that at least one of the two following equations must be true:

$$\hat{\mu}_{t-1,1} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,1}}} \geq \mu_1, \tag{10.7}$$

$$\hat{\mu}_{t-1,i} + \sqrt{\frac{2\alpha \ln t}{S_{t-1,i}}} < \mu_i . \tag{10.8}$$

If the first one is true, the confidence interval around our estimate of the expectation of the optimal arm does not contain μ_1. On the other hand, if the second one is true the confidence interval around our estimate of the expectation μ_i does not contain μ_i. So, we claim that if $t > t^\star$ and we selected a suboptimal arm, then at least one of these two bad events happened.

Let's prove the claim by contradiction: *if both the inequalities above are false*, $t > t^\star$, and $A_t = i$, we have

$$\begin{aligned}
\hat{\mu}_{t-1,1} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,1}}} &< \mu_1 \qquad \text{((10.7) false)} \\
&= \mu_i - \Delta_i \\
&< \mu_i - 2\sqrt{\frac{2\alpha \ln T}{S_{t-1,i}}} \qquad \text{(for (10.6))} \\
&\leq \mu_i - 2\sqrt{\frac{2\alpha \ln t}{S_{t-1,i}}} \\
&\leq \hat{\mu}_{t-1,i} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,i}}} \qquad \text{((10.8) false)},
\end{aligned}$$

that, by the selection strategy of the algorithm, would imply $A_t \neq i$.

Note that $S_{t^\star,i} \leq \frac{8\alpha \ln T}{\Delta_i^2} + 1$. Hence, we have

$$\begin{aligned}
\mathbb{E}\left[S_{T,i}\right] &= \mathbb{E}[S_{t^\star,i}] + \sum_{t=t^\star+1}^{T} \mathbb{E}[\mathbf{1}[A_t = i, \text{(10.7) or (10.8) true}]] \\
&\leq \frac{8\alpha \ln T}{\Delta_i^2} + 1 + \sum_{t=t^\star+1}^{T} \mathbb{E}[\mathbf{1}[\text{(10.7) or (10.8) true}]] \\
&\leq \frac{8\alpha \ln T}{\Delta_i^2} + 1 + \sum_{t=t^\star+1}^{T} \left(\mathbb{P}\{\text{(10.7) true}\} + \mathbb{P}\{\text{(10.8) true}\}\right) .
\end{aligned}$$

Now, we upper bound the probabilities in the sum. Given that the losses on the arms are i.i.d. and using the union bound, we have

$$\begin{aligned}
\mathbb{P}\left\{\hat{\mu}_{t-1,1} - \sqrt{\frac{2\alpha \ln t}{S_{t-1,1}}} \geq \mu_1\right\} &\leq \mathbb{P}\left\{\max_{s=1,\ldots,t-1} \frac{1}{s}\sum_{j=1}^{s} g_{j,1} - \sqrt{\frac{2\alpha \ln t}{s}} \geq \mu_1\right\} \\
&= \mathbb{P}\left\{\bigcup_{s=1}^{t-1}\left\{\frac{1}{s}\sum_{j=1}^{s} g_{j,1} - \sqrt{\frac{2\alpha \ln t}{s}} \geq \mu_1\right\}\right\} .
\end{aligned}$$

Hence, using Corollary 10.10 we have

$$\begin{aligned}
\mathbb{P}\{\text{(10.7) true}\} &\leq \sum_{s=1}^{t-1} \mathbb{P}\left\{\frac{1}{s}\sum_{j=1}^{s} g_{j,1} - \sqrt{\frac{2\alpha \ln t}{s}} \geq \mu_1\right\} \qquad \text{(union bound)} \\
&\leq \sum_{s=1}^{t-1} t^{-\alpha} = (t-1)t^{-\alpha} .
\end{aligned}$$

Given that the same bound holds for $\mathbb{P}\{\text{(10.8) true}\}$, we have

$$\begin{aligned}
\mathbb{E}\left[S_{T,i}\right] &\leq \frac{8\alpha \ln T}{\Delta_i^2} + 1 + \sum_{t=1}^{\infty} 2(t-1)t^{-\alpha} \leq \frac{8\alpha \ln T}{\Delta_i^2} + 1 + \sum_{t=2}^{\infty} 2t^{1-\alpha} \leq \frac{8\alpha \ln T}{\Delta_i^2} + 1 + 2\int_1^\infty x^{1-\alpha}\,\mathrm{d}x \\
&= \frac{8\alpha \ln T}{\Delta_i^2} + \frac{\alpha}{\alpha - 2} .
\end{aligned}$$

Using the decomposition of the regret in Lemma 10.11, $\mathrm{Regret}_T = \sum_{i=1}^{d} \Delta_i \mathbb{E}[S_{T,i}]$, we have the stated bound. □

The bound above can become meaningless if the gaps are too small. So, here we prove another bound that does not depend on the inverse of the gaps.

Theorem 10.15. *Assume that the losses of the arms minus their expectations are* 1*-subgaussian and let* $\alpha > 2$*. Then, UCB guarantees a regret of*

$$\mathrm{Regret}_T \leq 4\sqrt{2\alpha d T \ln T} + \frac{\alpha}{\alpha - 2} \sum_{i=1}^{d} \Delta_i \ .$$

Proof. Let $\Delta > 0$ be some value to be tuned subsequently and recall from the proof of Theorem 10.14 that for each suboptimal arm i we can bound

$$\mathbb{E}[S_{T,i}] \leq \frac{\alpha}{\alpha - 2} + \frac{8\alpha \ln T}{\Delta_i^2} \ .$$

Hence, using the regret decomposition in Lemma 10.11, we have

$$\begin{aligned}
\mathrm{Regret}_T &= \sum_{i:\Delta_i < \Delta} \Delta_i \mathbb{E}[S_{T,i}] + \sum_{i:\Delta_i \geq \Delta} \Delta_i \mathbb{E}[S_{T,i}] \\
&\leq T\Delta + \sum_{i:\Delta_i \geq \Delta} \Delta_i \mathbb{E}[S_{T,i}] \\
&\leq T\Delta + \sum_{i:\Delta_i \geq \Delta} \left(\Delta_i \frac{\alpha}{\alpha - 2} + \frac{8\alpha \ln T}{\Delta_i} \right) \\
&\leq T\Delta + \frac{\alpha}{\alpha - 2} \sum_{i=1}^{d} \Delta_i + \frac{8\alpha d \ln T}{\Delta} \ .
\end{aligned}$$

Choosing $\Delta = \sqrt{\frac{8\alpha d \ln T}{T}}$, we have the stated bound. □

Remark 10.16. *Note that while the UCB algorithm is considered parameter-free, we still have to know the subgaussianity of the arms. While this can be easily upper bounded for stochastic arms with bounded support, it is unclear how to do it without any prior knowledge on the distribution of the arms.*

It is possible to prove that the UCB algorithm is asymptotically optimal, in the sense of the following Theorem.

Theorem 10.17 ([Bubeck and Cesa-Bianchi, 2012, Theorem 2.2])**.** *Consider a strategy that satisfies* $\mathbb{E}[S_{T,i}] = o(T^a)$ *as* $T \to \infty$ *for any set of Bernoulli loss distributions, any arm* i *with* $\Delta_i > 0$ *and any* $a > 0$*. Then, for any set of Bernoulli loss distributions, the following holds*

$$\liminf_{T \to +\infty} \frac{\mathrm{Regret}_T}{\ln T} \geq \sum_{i:\Delta_i} \frac{1}{2\Delta_i} \ .$$

10.3 History Bits

The algorithm in Algorithm 10.1 is from Cesa-Bianchi and Lugosi [2006, Theorem 6.9]. The Exp3 algorithm was proposed in Auer et al. [2002b] and it used a small exploration rate to achieve the same regret we proved. The observation that the exploration in Exp3 can be removed completely is from Stoltz [2005].

The INF algorithm was proposed by Audibert and Bubeck [2009] and re-casted as an OMD procedure in Audibert et al. [2011]. The connection with the Tsallis entropy was done in Abernethy et al. [2015]. The specific proof presented here is new and it builds on the proof by Abernethy et al. [2015]. Note that Abernethy et al. [2015] proved the same regret bound for a FTRL procedure over the stochastic estimates of the losses (that they call Gradient-Based Prediction Algorithm), while here we proved it using a OMD procedure.

The ETC algorithm goes back to Robbins [1952], even if Robbins proposed what is now called epoch-greedy [Langford and Zhang, 2008]. For more history on ETC, take a look at chapter 6 in Lattimore and Szepesvári [2020]. The proofs presented here are from Lattimore and Szepesvári [2020] as well.

The use of confidence bounds and the idea of optimism first appeared in the work by Lai and Robbins [1985]. The first version of UCB is by Lai [1987]. The version of UCB I presented is by Auer et al. [2002a] under the name UCB1. Note that, rather than considering 1-subgaussian environments, Auer et al. [2002a] considers bandits where the rewards are confined to the $[0, 1]$ interval. The proof of Theorem 10.14 is a minor variation of the one of Theorem 2.1 in Bubeck and Cesa-Bianchi [2012], which also popularized the subgaussian setup. Theorem 10.15 is from Bubeck and Cesa-Bianchi [2012].

10.4 Exercises

Problem 10.1. *Design and analyse an FTRL version of Exp3 and Poly-INF/OMD with Tsallis entropy with time-varying non-decreasing regularizers.*

Problem 10.2. *Prove a similar regret bound to the one in Theorem 10.15 for an optimally tuned Explore-Then-Commit algorithm.*

Chapter 11

Saddle-Point Optimization and OCO Algorithms

In this chapter, we talk about solving saddle point problems with online convex optimization (OCO) algorithms, and the connection with game theory.

11.1 Saddle-Point Problems

We want to solve the following saddle point problem

$$\inf_{\boldsymbol{x}\in X}\sup_{\boldsymbol{y}\in Y} f(\boldsymbol{x},\boldsymbol{y})\,. \tag{11.1}$$

Let's say from the beginning that we need $\inf$ and $\sup$ rather than $\min$ and $\max$ because the minimum or maximum might not exist. Everytime we know for sure the $\inf/\sup$ are attained, we can substitute them with $\min/\max$.

While for the minimization of functions is clear what it means to solve it, it might not be immediate to see what is the meaning of "solving" the saddle point problem in (11.1). It turns out that the proper notion we are looking for is the one of saddle point.

Definition 11.1 (Saddle Point)**.** *Let $X \subseteq \mathbb{R}^n$, $Y \subseteq \mathbb{R}^m$, and $f : X \times Y \to \mathbb{R}$. A point $(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \in X \times Y$ is a **saddle point** of f in $X \times Y$ if*

$$f(\boldsymbol{x}^\star,\boldsymbol{y}) \le f(\boldsymbol{x}^\star,\boldsymbol{y}^\star) \le f(\boldsymbol{x},\boldsymbol{y}^\star), \quad \forall \boldsymbol{x}\in X, \boldsymbol{y}\in Y\,.$$

We will now state conditions under which there *exists* a saddle point that solves (11.1). First, we need an easy lemma.

Lemma 11.2. *Let f is a function from a non-empty product set $X \times Y$ to $\mathbb{R}$. Then,*

$$\inf_{\boldsymbol{x}\in X}\sup_{\boldsymbol{y}\in Y} f(\boldsymbol{x},\boldsymbol{y}) \ge \sup_{\boldsymbol{y}\in Y}\inf_{\boldsymbol{x}\in X} f(\boldsymbol{x},\boldsymbol{y})\,.$$

Proof. For any $\boldsymbol{x}' \in X$ and $\boldsymbol{y} \in Y$ we have that $f(\boldsymbol{x}',\boldsymbol{y}) \ge \inf_{\boldsymbol{x}\in X} f(\boldsymbol{x},\boldsymbol{y})$. This implies that $\sup_{\boldsymbol{y}\in Y} f(\boldsymbol{x}',\boldsymbol{y}) \ge \sup_{\boldsymbol{y}\in Y}\inf_{\boldsymbol{x}\in X} f(\boldsymbol{x},\boldsymbol{y})$ for all $\boldsymbol{x}' \in X$ that gives the stated inequality. □

We can now state the following theorem.

Theorem 11.3. *Let f any function from a non-empty product set $X \times Y$ to $\mathbb{R}$. A point $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ is a saddle point of f if and only if the supremum in*

$$\sup_{\boldsymbol{y}\in Y}\inf_{\boldsymbol{x}\in X} f(\boldsymbol{x},\boldsymbol{y}) \tag{11.2}$$

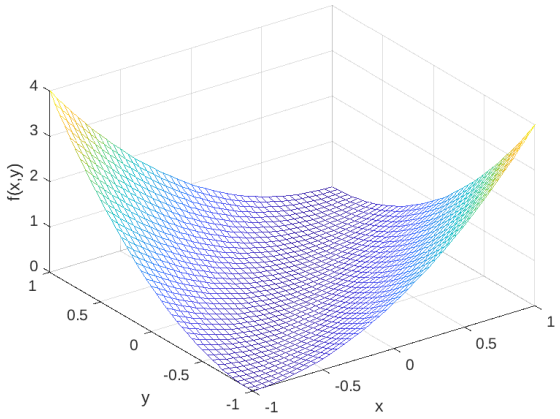

Figure 11.1: The function $f(x, y) = (x - y)^2$ does not have a saddle point.

is attained at $\boldsymbol{y}^\star$, the infimum in

$$\inf_{\boldsymbol{x}\in X} \sup_{\boldsymbol{y}\in Y} f(\boldsymbol{x}, \boldsymbol{y}) \tag{11.3}$$

is at $\boldsymbol{x}^\star$, and these two expressions are equal.

Proof. If $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ is a saddle point, then we have

$$\begin{aligned} f(\boldsymbol{x}^\star, \boldsymbol{y}^\star) &= \inf_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \boldsymbol{y}^\star) \leq \sup_{\boldsymbol{y}\in Y} \inf_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \boldsymbol{y}) \\ f(\boldsymbol{x}^\star, \boldsymbol{y}^\star) &= \sup_{\boldsymbol{y}\in Y} f(\boldsymbol{x}^\star, \boldsymbol{y}) \geq \inf_{\boldsymbol{x}\in X} \sup_{\boldsymbol{y}\in Y} f(\boldsymbol{x}, \boldsymbol{y}) \,. \end{aligned}$$

From Lemma 11.2, we have that these quantities must be equal, so that the three conditions in the theorem are satisfied.

For the other direction, if the conditions are satisfied, then we have

$$f(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \leq \sup_{\boldsymbol{y}\in Y} f(\boldsymbol{x}^\star, \boldsymbol{y}) = \inf_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \boldsymbol{y}^\star) \leq f(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \,.$$

Hence, $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ is a saddle point. □

Remark 11.4. *The above theorem implies that the set of saddle points, when nonempty, is the Cartesian product $X^\star \times Y^\star$, where $X^\star$ and $Y^\star$ are the sets of optimal solutions of the optimization problems* (11.2) *and* (11.3)*, respectively. In other words, $\boldsymbol{x}^\star$ and $\boldsymbol{y}^\star$ can be independently chosen from the sets $X^\star$ and $Y^\star$, respectively, to form a saddle point.*

Remark 11.5. *The above theorem tells a surprising thing: If a saddle point exists, then there might be multiple ones, and all of them must have the same minimax value. This might seem surprising, but it is due to the fact that the definition of saddle point is a global and not a local property. Moreover, if the* $\inf\sup$ *and* $\sup\inf$ *problem have different values, no saddle point exists.*

Remark 11.6. *Consider the case that the value of the* $\inf\sup$ *and* $\inf\sup$ *are different. If your favorite optimization procedure to find the solution of a saddle point problem does not distinguish between a* $\inf\sup$ *and* $\sup\inf$ *formulation, then it cannot be correct!*

Let's show a couple of examples that show that the conditions above are indeed necessary.

Example 11.7. *Let $f(x, y) = (x - y)^2$, $X = [-1, 1]$, and $Y = [-1, 1]$. Then, we have*

$$\inf_{x\in X} \sup_{y\in Y} (x - y)^2 = \inf_{x\in X} (1 + |x|)^2 = 1,$$

while

$$\sup_{y\in Y}\inf_{x\in X}\,(x-y)^2 = \sup_{y\in Y} 0 = 0\,.$$

Indeed, from Figure 11.1 we can see that there is no saddle point.

Example 11.8. *Let $f(x,y) = xy$, $X = (0,1]$, and $Y = (0,1]$. Then, we have*

$$\inf_{x\in X}\sup_{y\in Y}\,xy = \inf_{x\in X}\,x = 0$$

and

$$\sup_{y\in Y}\inf_{x\in X}\,xy = 0\,.$$

Here, even if inf sup is equal to sup inf, the saddle point does not exist because the inf in the first expression is not attained in a point of X.

Theorem 11.3 also tells us that in order to find a saddle point of a function f, we need to find the minimizer in $\boldsymbol{x}$ of $\sup_{\boldsymbol{y}\in Y}\,f(\boldsymbol{x},\boldsymbol{y})$ and the maximizer in y of $\inf_{\boldsymbol{x}\in X}\,f(\boldsymbol{x},\boldsymbol{y})$. Let's now use this knowledge to design a proper measure of progress towards the saddle point.

We might be tempted to use $f(\boldsymbol{x}',\boldsymbol{y}') - f(\boldsymbol{x}^\star,\boldsymbol{y}^\star)$ as a measure of suboptimality of $(\boldsymbol{x}',\boldsymbol{y}')$ with respect to the saddle point $(\boldsymbol{x}^\star,\boldsymbol{y}^\star)$. Unfortunately, this quantity can be negative or equal to zero for an infinite number of points $(\boldsymbol{x}',\boldsymbol{y}')$ that are not saddle points. We might then think to use some notion of distance to the saddle point, like $\|\boldsymbol{x}'-\boldsymbol{x}^\star\|_2^2+\|\boldsymbol{y}'-\boldsymbol{y}^\star\|_2^2$, but this quantity in general can go to zero at an arbitrarily slow rate. To see why consider the case that $f(\boldsymbol{x},\boldsymbol{y}) = h(\boldsymbol{x})$, so that the saddle point problem reduces to minimize a convex function. So, assuming only convexity, the rate of convergence to a minimizer of f can be arbitrarily slow. Hence, we need something different.

Observe that the Theorem 11.3 says one of the problems we should solve is

$$\inf_{\boldsymbol{x}\in X}\,h(\boldsymbol{x}),$$

where $h(\boldsymbol{x}) = \sup_{\boldsymbol{y}\in Y}\,f(\boldsymbol{x},\boldsymbol{y})$. In this view, the problem looks like a standard offline convex optimization problem, where the objective function has a particular structure. Moreover, in this view we only focus on the variables $\boldsymbol{x}$. The standard measure of convergence in this case for a point $\boldsymbol{x}'$, the *suboptimality gap*, can be written as

$$h(\boldsymbol{x}') - \inf_{\boldsymbol{x}\in X} h(\boldsymbol{x}) = \sup_{\boldsymbol{y}\in Y}\,f(\boldsymbol{x}',\boldsymbol{y}) - \inf_{\boldsymbol{x}\in X}\sup_{\boldsymbol{y}\in Y}\,f(\boldsymbol{x},\boldsymbol{y})\,.$$

We also have to find the maximizer with respect to $\boldsymbol{y}$ of the function $\inf_{\boldsymbol{x}\in X}\,f(\boldsymbol{x},\boldsymbol{y})$, hence we have

$$\sup_{\boldsymbol{y}\in Y}\,g(\boldsymbol{y}),$$

where $g(\boldsymbol{y}) = \inf_{\boldsymbol{x}\in X}\,f(\boldsymbol{x},\boldsymbol{y})$. This also implies another measure of convergence in which we focus only on the variable $\boldsymbol{y}$:

$$\sup_{\boldsymbol{y}\in Y}\,g(\boldsymbol{y}) - g(\boldsymbol{y}') = \sup_{\boldsymbol{y}\in Y}\inf_{\boldsymbol{x}\in X}\,f(\boldsymbol{x},\boldsymbol{y}) - \inf_{\boldsymbol{x}\in X}\,f(\boldsymbol{x},\boldsymbol{y}')\,.$$

Finally, in case we are interested in studying the quality of a joint solution $(\boldsymbol{x}',\boldsymbol{y}')$, a natural measure is a sum of the two measures above:

$$\sup_{\boldsymbol{y}\in Y}\,f(\boldsymbol{x}',\boldsymbol{y}) - \inf_{\boldsymbol{x}\in X}\sup_{\boldsymbol{y}\in Y}\,f(\boldsymbol{x},\boldsymbol{y}) + \sup_{\boldsymbol{y}\in Y}\inf_{\boldsymbol{x}\in X}\,f(\boldsymbol{x},\boldsymbol{y}) - \inf_{\boldsymbol{x}\in X}\,f(\boldsymbol{x},\boldsymbol{y}') = \sup_{\boldsymbol{y}\in Y}\,f(\boldsymbol{x}',\boldsymbol{y}) - \inf_{\boldsymbol{x}\in X}\,f(\boldsymbol{x},\boldsymbol{y}'),$$

where we assumed the existence of a saddle point to say that $\inf_{\boldsymbol{x}\in X}\sup_{\boldsymbol{y}\in Y}\,f(\boldsymbol{x},\boldsymbol{y}) = \sup_{\boldsymbol{y}\in Y}\inf_{\boldsymbol{x}\in X}\,f(\boldsymbol{x},\boldsymbol{y})$ from Theorem 11.3. This measure is called *duality gap*.

Definition 11.9 (Duality Gap)**.** *For a function $f : X\times Y\to\mathbb{R}$, define the **duality gap** on $(\boldsymbol{x}',\boldsymbol{y}')\in X\times Y$ as*

$$\sup_{\boldsymbol{y}\in Y}\,f(\boldsymbol{x}',\boldsymbol{y}) - \inf_{\boldsymbol{x}\in X}\,f(\boldsymbol{x},\boldsymbol{y}')$$

The duality gap is always non-negative even when the saddle point does not exist, since $\sup_{\boldsymbol{y}\in Y} \ f(\boldsymbol{x}', \boldsymbol{y}) \geq f(\boldsymbol{x}', \boldsymbol{y}') \geq \inf_{\boldsymbol{x}\in X} \ f(x, y')$, for all $\boldsymbol{x}' \in X$ and $\boldsymbol{y}' \in Y$.

Let's add even more intuition of the duality gap definition, using the one of ϵ-saddle point.

Definition 11.10 (ϵ-Saddle-Point)**.** *Let $X \subseteq \mathbb{R}^n$, $Y \subseteq \mathbb{R}^m$, and $f : X \times Y \to \mathbb{R}$. A point $(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \in X \times Y$ is an **ϵ-saddle point** of f in $X \times Y$ if*

$$f(\boldsymbol{x}^\star, \boldsymbol{y}) - \epsilon \leq f(\boldsymbol{x}^\star, \boldsymbol{y}^\star) \leq f(\boldsymbol{x}, \boldsymbol{y}^\star) + \epsilon, \quad \forall \boldsymbol{x} \in X, \boldsymbol{y} \in Y \ .$$

This definition is useful because we cannot expect to numerically calculate a saddle point with infinite precision, but we can be find something that satisfies the saddle point definition up to an ϵ. Obviously, any saddle point is also an ϵ-saddle point.

Now, the notion of ϵ-saddle point is equivalent up to a multiplicative constant to the ϵ duality gap, as detailed in the next lemma. The proof is left as an exercise (see Exercise 11.1).

Lemma 11.11. *If $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ is an ϵ-saddle point then its duality gap is upper bounded by 2ϵ. On the other hand, a duality gap of 2ϵ and the existence of a saddle point imply that the point is a 2ϵ-saddle.*

The above reasoning told us that finding the saddle point of the function f is equivalent to solving a maximization problem and a minimization problem. However, as we said above, the saddle point might not exist. So, let's now move to easily checkable sufficient conditions for the existence of a saddle point. For this, we can state the following theorem.

Theorem 11.12. *Let X, Y be compact convex subsets of $\mathbb{R}^n$ and $\mathbb{R}^m$ respectively. Let $f : X \times Y \to \mathbb{R}$ a continuous function, convex in its first argument, and concave in its second, and Lipschitz with respect to both. Then, we have that*

$$\min_{\boldsymbol{x}\in X} \max_{\boldsymbol{y}\in Y} \ f(\boldsymbol{x}, \boldsymbol{y}) = \max_{\boldsymbol{y}\in Y} \min_{\boldsymbol{x}\in X} \ f(\boldsymbol{x}, \boldsymbol{y}) \ .$$

This theorem gives us sufficient conditions to have the min-max problem equal to the max-min one. So, for example, thanks to the Weierstrass theorem (Theorem A.11), the assumptions in Theorem 11.12, in light of Theorem 11.3, are sufficient conditions for the existence of a saddle point.

We defer the proof of this theorem for a bit and we now turn ways to solve the saddle point problem in (11.1).

11.2 Solving Saddle-Point Problems with OCO

Let's show how to use Online Convex Optimization (OCO) algorithms to solve saddle point problems. We will state a procedure that is a direct generalization of the online-to-batch conversion we saw in Chapter 3.

Algorithm 11.1 Solving Saddle-Point Problems with OCO

Require: $\boldsymbol{x}_1 \in X, \boldsymbol{y}_1 \in Y$

1: **for** $t = 1, \dots, T$ **do**
2: X-Learner and Y-Learner simultaneously decide their outputs $\boldsymbol{x}_t \in X$ and $\boldsymbol{y}_t \in Y$
3: X-Learner receives $\ell_t(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{y}_t)$
4: Y-Learner receives $h_t(\boldsymbol{y}) = -f(\boldsymbol{x}_t, \boldsymbol{y})$
5: **end for**
6: **return** $\bar{\boldsymbol{x}}_T = \frac{1}{T}\sum_{t=1}^T \boldsymbol{x}_t$, $\bar{\boldsymbol{y}}_T = \frac{1}{T}\sum_{t=1}^T \boldsymbol{y}_t$

Suppose to use an OCO algorithm fed with losses $\ell_t(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{y}_t)$ that produces the iterates $\boldsymbol{x}_t$ and another OCO algorithm fed with losses $h_t(\boldsymbol{y}) = -f(\boldsymbol{x}_t, \boldsymbol{y})$ that produces the iterates $\boldsymbol{y}_t$. Then, we can state the following Theorem.

Theorem 11.13. *Let $f : X \times Y \to \mathbb{R}$. Then, with the notation in Algorithm 11.1, for any $\boldsymbol{x} \in X$, we have*

$$\frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x}_t, \boldsymbol{y}_t) - \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x}, \boldsymbol{y}_t) = \frac{\mathrm{Regret}_T^X(\boldsymbol{x})}{T},$$

where $\mathrm{Regret}_T^X(\boldsymbol{x}) = \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{x})$.

Moreover, for any $\boldsymbol{y} \in Y$, we have

$$\frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x}_t, \boldsymbol{y}) - \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x}_t, \boldsymbol{y}_t) = \frac{\mathrm{Regret}_T^Y(\boldsymbol{y})}{T},$$

where $\mathrm{Regret}_T^Y(\boldsymbol{y}) = \sum_{t=1}^T h_t(\boldsymbol{y}_t) - \sum_{t=1}^T h_t(\boldsymbol{y})$.

Also, if f is convex in the first argument and concave in the second and $\arg\max_{\boldsymbol{y}\in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y})$ *and* $\arg\min_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T)$ *are non-empty, then we have*

$$\max_{\boldsymbol{y}\in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y}) - \min_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T) \le \frac{\mathrm{Regret}_T^X(\boldsymbol{x}'_T) + \mathrm{Regret}_T^Y(\boldsymbol{y}'_T)}{T},$$

for any $\boldsymbol{x}'_T \in \arg\min_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T)$ and $\boldsymbol{y}'_T \in \arg\max_{\boldsymbol{y}\in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y})$.

Proof. The first two equalities are obtained simply observing that $\ell_t(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{y}_t)$ and $h_t(\boldsymbol{y}) = -f(\boldsymbol{x}_t, \boldsymbol{y})$.

For the inequality, using Jensen's inequality, we obtain

$$f(\bar{\boldsymbol{x}}_T, \boldsymbol{y}) - f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T) \le \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x}_t, \boldsymbol{y}) - \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x}, \boldsymbol{y}_t)\ .$$

Summing the first two equalities, using the above inequality, and taking $\boldsymbol{x} = \boldsymbol{x}'_T \in \arg\min_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T)$ and $\boldsymbol{y} = \boldsymbol{y}'_T \in \arg\max_{\boldsymbol{y}\in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y})$, we get the stated inequality. □

From this theorem and Lemma 11.11, we can immediately prove the following corollary.

Corollary 11.14. *Let $f : X \times Y \to \mathbb{R}$ continuous and assume that X and Y are compact. Consider Algorithm 11.1 and assume that the two online algorithms guarantee that their maximum regret over competitors in their feasible set is sublinear in T. Then, we have*

$$\lim_{T\to\infty} \max_{\boldsymbol{y}\in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y}) - \min_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T) = 0,$$

and $(\bar{\boldsymbol{x}}_T, \bar{\boldsymbol{y}}_T)$ converge to a saddle point.

Example 11.15. *Consider the saddle point problem*

$$\min_{|x|\le 2} \max_{|y|\le 2} (x-1)(y-1)\ .$$

The saddle point of this problem is $(x, y) = (1, 1)$. We can find it using, for example, Projected Online Gradient Descent with stepsizes $\eta_t = \frac{1}{\sqrt{t}}$. So, setting $\boldsymbol{x}_1 = \boldsymbol{y}_1 = 0$, we have the iterations

$$x_{t+1} = \max\left(\min\left(x_t - \frac{1}{\sqrt{t}}(y_t - 1), 2\right), -2\right)$$
$$y_{t+1} = \max\left(\min\left(y_t + \frac{1}{\sqrt{t}}(x_t - 1), 2\right), -2\right)\ .$$

According to Theorem 11.13, the duality gap in $(\frac{1}{T}\sum_{t=1}^T x_t, \frac{1}{T}\sum_{t=1}^T y_t)$ converges to 0.

Surprisingly, we can even prove a (simpler version of the) minimax theorem from the above result! In particular, we will use the additional assumption that there exist OCO algorithms that minimize ℓ_t and h_t have sublinear regret.

Proof of Theorem 11.12 with OCO assumption. From Lemma 11.2, we have one inequality. Hence, we now have to prove the other inequality.

We will use a constructive proof. Let's use Algorithm 11.1 and Theorem 11.13. For the first player, for any $\boldsymbol{x} \in X$ we have

$$\frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x}_t, \boldsymbol{y}_t) = \frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x}, \boldsymbol{y}_t) + \frac{\mathrm{Regret}_T^X(\boldsymbol{x})}{T} \leq f\left(\boldsymbol{x}, \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{y}_t\right) + \frac{\mathrm{Regret}_T^X(\boldsymbol{x})}{T} .$$

Observe that

$$\min_{\boldsymbol{x}\in X} f\left(\boldsymbol{x}, \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{y}_t\right) \leq \max_{\boldsymbol{y}\in Y}\min_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \boldsymbol{y}) .$$

Hence, using an OCO algorithm that has $o(T)$ regret for each $\boldsymbol{x} \in X$, we have

$$\frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x}_t, \boldsymbol{y}_t) \leq \max_{\boldsymbol{y}\in Y}\min_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \boldsymbol{y}) + o(1) .$$

In the same way, we have

$$-\frac{1}{T}\sum_{t=1}^{T} f(\boldsymbol{x}_t, \boldsymbol{y}_t) \leq -\min_{\boldsymbol{x}\in X}\max_{\boldsymbol{y}\in Y} f(\boldsymbol{x}, \boldsymbol{y}) + o(1) .$$

Summing the two inequalities, taking $T \to \infty$, and using the sublinear regret assumption, we have

$$\min_{\boldsymbol{x}\in X}\max_{\boldsymbol{y}\in Y} f(\boldsymbol{x}, \boldsymbol{y}) \leq \max_{\boldsymbol{y}\in Y}\min_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \boldsymbol{y}) . \qquad \square$$

11.2.1 Variations with Best Response and Alternation

In some cases, it is easy to compute the max with respect to $\boldsymbol{y} \in Y$ of $f(\boldsymbol{x}_t, \boldsymbol{y})$ for a given $\boldsymbol{x}_t$. For example, this is trivial for bilinear games over the probability simplex that we will see in Section 11.3. In these cases, we can remove the second learner and just use its *best response* in each round. Note that in this way we are making one of the two players "stronger" through the knowledge of its loss of the next round. However, this is perfectly fine: The proof in Theorem 11.13 is still perfectly valid.

Algorithm 11.2 Saddle-Point Optimization with OCO and Y-Best-Response

Require: $\boldsymbol{x}_1 \in X$
1: **for** $t = 1, \dots, T$ **do**
2: Set $\boldsymbol{y}_t \in \operatorname{argmax}_{\boldsymbol{y}\in Y} f(\boldsymbol{x}_t, \boldsymbol{y})$
3: X-Learner receives $\ell_t(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{y}_t)$ and produces $\boldsymbol{x}_{t+1} \in X$
4: **end for**
5: **return** $\bar{\boldsymbol{x}}_T = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{x}_t$, $\bar{\boldsymbol{y}}_T = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{y}_t$

In this case, the Y-player has an easy life: it knows the loss before making the prediction, hence it can just output the minimizer of the loss in Y. Hence, we also have that the regret of the Y-player will be non-positive and it will not show up in Theorem 11.13. Putting all together, we can state the following corollary.

Corollary 11.16. *Let $f : X \times Y \to \mathbb{R}$ be convex in the first argument, and concave in the second. With the notation in Algorithm 11.2, assume that X is compact, the argmax of the Y-player is never empty, and $\arg\min_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T)$ is not-empty. Then, we have*

$$\sup_{\boldsymbol{y}\in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y}) - \min_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T) \leq \frac{\mathrm{Regret}_T^X(\boldsymbol{x}'_T)}{T},$$

for any $\boldsymbol{x}'_T \in \arg\min_{\boldsymbol{x}\in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T)$ and where $\mathrm{Regret}_T^X(\boldsymbol{x}) = \sum_{t=1}^{T} \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^{T} \ell_t(\boldsymbol{x})$.

This alternative seems interesting from a theoretical point of view because it allows to avoid the complexity of learning in the Y space, for example removing the dependency on its dimension.

Of course, an analogous result can be stated using best-response for the X-player and an OCO algorithm for the Y-player, as in Algorithm 11.3.

Algorithm 11.3 Saddle-Point Optimization with OCO and X-Best-Response

Require: $\boldsymbol{y}_1 \in Y$
1: **for** $t = 1, \dots, T$ **do**
2: Set $\boldsymbol{x}_t = \operatorname{argmin}_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \boldsymbol{y}_t)$
3: Y-Learner receives $h_t(\boldsymbol{y}) = -f(\boldsymbol{x}_t, \boldsymbol{y})$ and produces $\boldsymbol{y}_{t+1} \in Y$
4: **end for**
5: **return** $\bar{\boldsymbol{x}}_T = \frac{1}{T} \sum_{t=1}^T \boldsymbol{x}_t$, $\bar{\boldsymbol{y}}_T = \frac{1}{T} \sum_{t=1}^T \boldsymbol{y}_t$

There is a third variant, very used in empirical implementations, especially of Counterfactual Regret Minimization (CFR) [Zinkevich et al., 2007]. It is called *alternation* and it breaks the simultaneous reveal of the actions of the two players. Instead, we use the updated prediction of the first player to construct the loss of the second player. Empirically, this variant seems to greatly speed-up the convergence of the duality gap.

Algorithm 11.4 Saddle-Point Optimization with OCO and Alternation

Require: $\boldsymbol{y}_1 \in Y$
1: **for** $t = 1, \dots, T$ **do**
2: X-Learner receives $\ell_t(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{y}_t)$ and produces $\boldsymbol{x}_{t+1} \in X$
3: Y-Learner receives $h_t(\boldsymbol{y}) = -f(\boldsymbol{x}_{t+1}, \boldsymbol{y})$ and produces $\boldsymbol{y}_{t+1} \in Y$
4: **end for**
5: **return** $\bar{\boldsymbol{x}}_T = \frac{1}{T} \sum_{t=1}^T \boldsymbol{x}_{t+1}$, $\bar{\boldsymbol{y}}_T = \frac{1}{T} \sum_{t=1}^T \boldsymbol{y}_t$

For this version, Theorem 11.13 does not hold anymore because the terms $f(\boldsymbol{x}_t, \boldsymbol{y}_t)$ and $f(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t)$ are now different, however we can prove a similar guarantee.

Theorem 11.17. *Let $f : X \times Y \to \mathbb{R}$ convex in the first argument and concave in the second. With the notation in Algorithm 11.4, assume that $\arg\min_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T)$ and $\arg\max_{\boldsymbol{y} \in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y})$ are non-empty. Then, for any $\boldsymbol{x}'_T \in \arg\min_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T)$ and any $\boldsymbol{y}'_T \in \arg\max_{\boldsymbol{y} \in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y})$, we have*

$$\max_{\boldsymbol{y} \in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y}) - \min_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T) \leq \frac{\operatorname{Regret}_T^X(\boldsymbol{x}'_T) + \operatorname{Regret}_T^Y(\boldsymbol{y}'_T) + \sum_{t=1}^T (f(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) - f(\boldsymbol{x}_t, \boldsymbol{y}_t))}{T},$$

where $\operatorname{Regret}_T^Y(\boldsymbol{y}) = \sum_{t=1}^T h_t(\boldsymbol{y}_t) - \sum_{t=1}^T h_t(\boldsymbol{y})$, $\operatorname{Regret}_T^X(\boldsymbol{x}) = \sum_{t=1}^T \ell_t(\boldsymbol{x}_t) - \sum_{t=1}^T \ell_t(\boldsymbol{x})$.

Proof. Note that $\ell_t(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{y}_t)$, and $h_t(\boldsymbol{y}) = -f(\boldsymbol{x}_{t+1}, \boldsymbol{y})$. By Jensen's inequality, we have

$$\begin{aligned}
T(f(\bar{\boldsymbol{x}}_T, \boldsymbol{y}) - f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T)) &\leq \sum_{t=1}^T f(\boldsymbol{x}_{t+1}, \boldsymbol{y}) - \sum_{t=1}^T f(\boldsymbol{x}, \boldsymbol{y}_t) \\
&= \sum_{t=1}^T f(\boldsymbol{x}_{t+1}, \boldsymbol{y}) - \sum_{t=1}^T f(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) + \sum_{t=1}^T f(\boldsymbol{x}_t, \boldsymbol{y}_t) - \sum_{t=1}^T f(\boldsymbol{x}, \boldsymbol{y}_t) \\
&\quad + \sum_{t=1}^T f(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) - \sum_{t=1}^T f(\boldsymbol{x}_t, \boldsymbol{y}_t) \\
&= \sum_{t=1}^T (h_t(\boldsymbol{y}_t) - h_t(\boldsymbol{y})) + \sum_{t=1}^T (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{x})) + \sum_{t=1}^T (f(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t) - f(\boldsymbol{x}_t, \boldsymbol{y}_t)) \,.
\end{aligned}$$

Taking $\boldsymbol{x} = \boldsymbol{x}'_T \in \arg\min_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T)$ and $\boldsymbol{y} = \boldsymbol{y}'_T \in \arg\max_{\boldsymbol{y} \in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y})$, we get the stated result. □

Remark 11.18. *Using OMD for the X-player we have that $f(\boldsymbol{x}_t, \boldsymbol{y}_t) = \ell_t(\boldsymbol{x}_t) \geq \ell_t(\boldsymbol{x}_{t+1}) = f(\boldsymbol{x}_{t+1}, \boldsymbol{y}_t)$. Hence, in this case the additional term in Theorem 11.17 is negative, showing a (marginal) improvement to the convergence rate.*

Next, we will show how to connect saddle point problems with Game Theory.

11.3 Game-Theory interpretation of Saddle-Point Problems

An instantiation of a saddle point problems also has an interpretation in Game Theory as a *Two-player Zero-Sum Game*. Note that Game Theory is a vast field and two-person zero-sum games are only a very small subset of the problems in this domain and what I describe here is an even smaller subset of this subset of problems.

Game theory studies what happens when self-interested agents interact. By self-interested, we mean that each agent has an ideal state of things he wants to reach, that can include a description of what should happen to other agents as well, and he works towards this goal. In two-person games the players act simultaneously and then they receive their losses. In particular, the X-player chooses the play $\boldsymbol{x}$ and the Y-player chooses the play $\boldsymbol{y}$, the X-player suffers the loss $f(\boldsymbol{x}, \boldsymbol{y})$ and the Y-player the loss $-f(\boldsymbol{x}, \boldsymbol{y})$. It is important to understand that this is only one round, that is it has only one play for each player. Note that the standard game-theoretic terminology uses payoffs instead of losses, but we will keep using losses for coherence with the online convex optimization notation we use in this book.

We consider the so-called *two-person normal-form games*, that is when the first player has n possible actions and the second player m. A player can use a *pure strategy*, that is a single fixed action, or randomize over a set of actions according to some probability distribution, a so-called *mixed strategy*. In this case, we consider $X = \Delta^{n-1}$ and $Y = \Delta^{m-1}$ and they are known as the *action spaces* for the two players. In this setting, for a pair of pure strategies $(\boldsymbol{e}_i, \boldsymbol{e}_j)$, the first player receives the loss $f(\boldsymbol{e}_i, \boldsymbol{e}_j)$ and the second player $-f(\boldsymbol{e}_i, \boldsymbol{e}_j)$, where $\boldsymbol{e}_i$ is the vector with all zeros but a '1' in position i. The goal of each player is to minimize the received loss. Given the discrete nature of this game, the function $f(\boldsymbol{x}, \boldsymbol{y})$ is the bilinear function $\boldsymbol{x}^\top M \boldsymbol{y}$, where M is a matrix with n rows and m columns. Hence, for a pair of mixed strategy $(\boldsymbol{x}, \boldsymbol{y})$, the *expected* loss of the first player is $\boldsymbol{x}^\top M \boldsymbol{y}$ and the one of the second player is $-\boldsymbol{x}^\top M \boldsymbol{y}$.

A fundamental concept in game theory is the one of *Nash Equilibrium*. We have a Nash equilibrium if all players are playing their best strategy to the other players' strategies. That is, none of the players has incentive to change their strategy if the other player does not change it. For the zero-sum two-person game, this can be formalized saying that $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ is a Nash equilibrium if

$$(\boldsymbol{x}^\star)^\top M \boldsymbol{y} \leq (\boldsymbol{x}^\star)^\top M \boldsymbol{y}^\star \leq \boldsymbol{x}^\top M \boldsymbol{y}^\star, \quad \forall \boldsymbol{x} \in \Delta^{n-1}, \boldsymbol{y} \in \Delta^{m-1} \, .$$

This is *exactly* the definition of saddle point for the function $f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top M \boldsymbol{y}$ that we gave in the previous section. Given that $f(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top M \boldsymbol{y}$ is continuous, convex in the first argument and concave in the second one, the sets $X = \Delta^{n-1}$ and $Y = \Delta^{m-1}$ are convex and compacts, we can deduce from Theorem 11.12 and Theorem 11.3 that a saddle point always exists. Hence, there is always at least one (possibly mixed) Nash equilibrium in two-person zero-sum games. The common value of the minimax and maxmin problem is called *value of the game* and we will denote it by $v^\star$.

For zero-sum two-person game the Nash equilibrium has an immediate interpretation: From the definition above, if the first player uses the strategy $\boldsymbol{x}^\star$ then his loss is less then the value of the game $v^\star$, regardless of the strategy of the second player. Analogously, if the second player uses the strategy $\boldsymbol{y}^\star$ then his loss is less then $-v^\star$, regardless of the strategy of the first player. Both players achieve the value of the game if they both play the Nash strategy. Moreover, even if one of the player would announce his strategy in advance to the other player, he would not increase his loss in expectation.

Example 11.19 (Cake cutting)**.** *Suppose to have a game between two people: The first player cuts the cake in two and the second one chooses a piece; the first player receives the piece that was not chosen. We can formalize it with the following matrix*

	larger piece	*smaller piece*
cut evenly	*0*	*0*
cut unevenly	*10*	*-10*

When the first player plays action i and the second player action j, the first player receives the loss $M(i,j)$ and the second player receives $-M(i,j)$. The losses represent how much less in percentage compared to half of the cake the first player is receiving. The second player receives the negative of the same number. It should be easy to convince oneself that the strategy pair is $(\text{cut evenly}, \text{larger piece})$ is an equilibrium with value of the game of 0. However, perhaps counter intuitively, this is not the only optimal strategy. In fact, for $0.5 \leq a \leq 1$, one can verify numerically that any pair $\boldsymbol{x}^\star = [1,0]^\top$, $\boldsymbol{y}^\star = [a, 1-a]$ is optimal, where the first player uses a pure strategy and the second player a mixed one.

Example 11.20 (Rock-paper-scissors)**.** *Let's consider the game of Rock-paper-scissors. We describe it with the following matrix*

	Rock	*Paper*	*Scissors*
Rock	*0*	*1*	*-1*
Paper	*-1*	*0*	*1*
Scissors	*1*	*-1*	*0*

It should be immediate to realize that there are no pure Nash equilibria for this game. However, there is a mixed Nash equilibrium when each player randomize the action with a uniform probability distribution over the three actions and value of the game equals to 0.

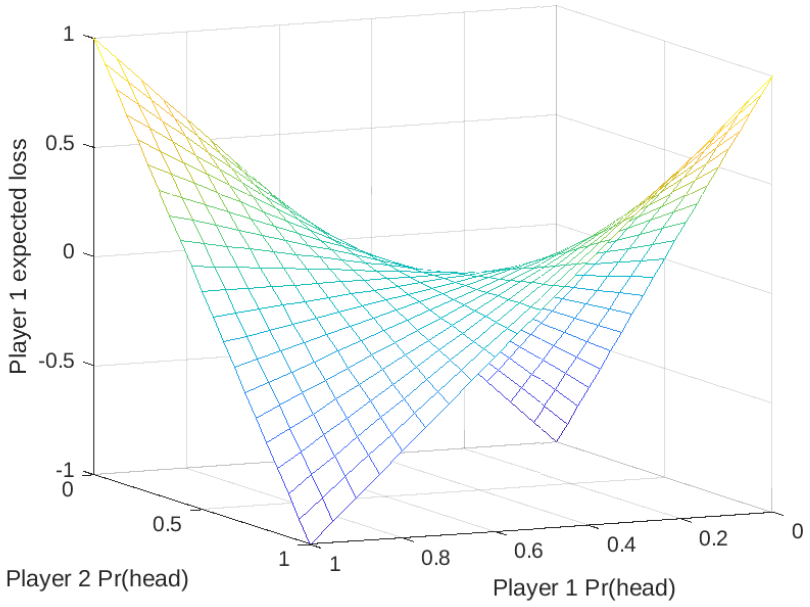

Figure 11.2: The saddle point in the Matching Pennies game.

Example 11.21 (Matching Pennies)**.** *In this game, both players show a face of a penny. If the two faces are the same, the first player wins both, otherwise the second player wins both. The associated matrix M is*

	head	*tail*
head	*-1*	*1*
tail	*1*	*-1*

It is easy to see that the Nash equilibrium is when both players randomize the face to show with equal probability.

In this simple case, we can visualize the saddle point associated to this problem in Figure 11.2

Unless the game is very small, we find Nash equilibria using numerical procedures that typically give us only approximate solutions. Hence, as for ϵ-saddle points, we also define an *ϵ-Nash equilibrium* for a zero-sum two-person game when $\boldsymbol{x}^\star$ and $\boldsymbol{y}^\star$ satisfy

$$(\boldsymbol{x}^\star)^\top M\boldsymbol{y} - \epsilon \leq (\boldsymbol{x}^\star)^\top M\boldsymbol{y}^\star \leq \boldsymbol{x}^\top M\boldsymbol{y}^\star + \epsilon, \quad \forall \boldsymbol{x} \in \Delta^{n-1}, \boldsymbol{y} \in \Delta^{m-1} .$$

Obviously, any Nash equilibrium is also an ϵ-Nash equilibrium.

From what we said in the previous section, it should be immediate to see how to numerically calculate the Nash equilibrium of a two-person zero-sum game. In fact, we know that we can use online convex optimization algorithms to find ϵ-saddle points, so we can do the same for ϵ-Nash equilibrium of two-person zero-sum games. Assuming that the average regret of both players is ϵ_T, Theorem 11.13 says that $\left(\frac{1}{T}\sum_{t=1}^T \boldsymbol{x}_t, \frac{1}{T}\sum_{t=1}^T \boldsymbol{y}_t\right)$ is a $2\epsilon_T$- Nash equilibrium.

11.4 Boosting as a Two-Person Game

We will now show that the Boosting problem can also be seen as the solution of a zero-sum two-person game.

Let $S = \{\boldsymbol{z}_t, y_t\}_{i=1}^m$ a training set of m couples features/labels, where $\boldsymbol{z}_i \in \mathbb{R}^d$ and $y_i = \{-1, 1\}$. Let $\mathcal{H} = \{h_i\}_{i=1}^n$ a set of n functions $h_i : \mathbb{R}^d \to \{-1, 1\}$.

The aim of boosting is to find a combination of the functions in $\mathcal{H}$ that has arbitrarily low misclassification error on S. Of course, this is not always possible. However, we will make a (strong) assumption: We assume that it is always possible to find a function $h_i \in \mathcal{H}$ such that its misclassification error over S weighted with any probability distribution is better than chance by a constant $\gamma > 0$. We now show that this assumption guarantees that the boosting problem is solvable!

First, we construct a matrix of the misclassifications for each function: $M \in \mathbb{R}^{n\times m}$ where

$$M(i,j) = \begin{cases} 1, & \text{if } h_i(\boldsymbol{z}_j) \neq y_j \\ 0, & \text{otherwise.} \end{cases}$$

Setting $X = \Delta^{n-1}$ and $Y = \Delta^{m-1}$, we write the saddle point problem/two-person zero-sum game

$$\min_{\boldsymbol{p}\in\Delta^{n-1}} \max_{\boldsymbol{q}\in\Delta^{m-1}} \boldsymbol{p}^\top M\boldsymbol{q} .$$

Given the definition of the matrix M, this is equivalent to

$$\min_{\boldsymbol{p}\in\Delta^{n-1}} \max_{\boldsymbol{q}\in\Delta^{m-1}} \sum_{i=1}^n \sum_{i=1}^m p_i q_j \mathbf{1}[h_i(\boldsymbol{z}_j) \neq y_j] .$$

Let's now formalize the assumption on the functions: We assume the existence of a *weak learning oracle* that, for any $\boldsymbol{q} \in \Delta^{m-1}$, returns $i^\star$ such that

$$\boldsymbol{e}_{i^\star}^\top M\boldsymbol{q} = \sum_{j=1}^m q_j \mathbf{1}[h_{i^\star}(\boldsymbol{z}_j) \neq y_j] \leq \frac{1}{2} - \gamma,$$

where $\gamma > 0$. In words, $i^\star$ is the index of the function in $\mathcal{H}$ that gives a $\boldsymbol{q}$-weighted error better than chance. Moreover, given that

$$\min_{\boldsymbol{p}\in\Delta^{n-1}} \boldsymbol{p}^\top M\boldsymbol{q} \leq \boldsymbol{e}_{i^\star}^\top M\boldsymbol{q} \leq \frac{1}{2} - \gamma$$

and

$$v^\star = \max_{\boldsymbol{q}\in\Delta^{m-1}} \min_{\boldsymbol{p}\in\Delta^{n-1}} \boldsymbol{p}^\top M\boldsymbol{q},$$

we have that the value of the game satisfies $v^\star \leq \frac{1}{2} - \gamma < \frac{1}{2}$. Using the inequality on the value of the game and the fact that the Nash equilibrium exists, we obtain that there exists $\boldsymbol{p}^\star \in \Delta^{n-1}$ such that for any $j = 1, \ldots, m$

$$\sum_{i=1}^n p_i^\star \mathbf{1}[h_i(\boldsymbol{z}_j) \neq y_j] = (\boldsymbol{p}^\star)^\top M\boldsymbol{e}_j \leq v^\star \leq \frac{1}{2} - \gamma < \frac{1}{2} . \tag{11.4}$$

In words, this means that every sample $\boldsymbol{z}_j$ is misclassified by less than half of the functions h_i when weighted by $\boldsymbol{p}^\star$. Hence, *we can correctly classify all the samples using a weighted majority vote rule where the weights over the function h_i are $\boldsymbol{p}^\star$*. This means that we can learn a perfect classifier rule using weak learners, through the solution of a minimax game. So, our job is to find a way to calculate this optimal distribution $\boldsymbol{p}^\star$ on the functions.

Given what we have said till now, a natural strategy is to use online convex optimization algorithms. In particular, we can use Algorithm 11.3, where in each round the X-player is the weak learning oracle, that knows the play $\boldsymbol{q}_t$ by the Y-Learner, while the Y-player is an OCO algorithm. Specialized to our setting we have the following algorithm. In words, the X-player looks for the function that has small enough weighted misclassification loss, where the weights are constructed by the Y-player.

Algorithm 11.5 Boosting through OCO

$q_{1,j} = 1/m, j = 1, \dots, m$
for $t = 1, \dots, T$ **do**
  Set i_t such that $\sum_{j=1}^m q_{t,j} \mathbf{1}[h_{i_t}(\boldsymbol{z}_j) \neq y_j] \leq 1/2 - \gamma$
  Y-Learner receives $\ell_t(\boldsymbol{q}) = -\sum_{j=1}^m q_j \mathbf{1}[h_{i_t}(\boldsymbol{z}_j) \neq y_j]$, pays $\ell_t(\boldsymbol{q}_t)$, and produces $\boldsymbol{q}_{t+1} \in \Delta^{m-1}$
end for
return $\bar{h}_T(\boldsymbol{z}) = \frac{1}{T}\sum_{t=1}^T h_{i_t}(\boldsymbol{z})$

Let's show a guarantee on the misclassification error of this algorithm. From the second equality of Theorem 11.13, for any $\boldsymbol{q} \in \Delta^{m-1}$, we have

$$\frac{1}{T}\sum_{t=1}^T \boldsymbol{e}_{i_t}^\top M \boldsymbol{q} = \frac{1}{T}\sum_{t=1}^T \boldsymbol{e}_{i_t}^\top M \boldsymbol{q}_t + \frac{\mathrm{Regret}_T^Y(\boldsymbol{q})}{T} \,.$$

From the assumption on the weak-learnability oracle, we have $\frac{1}{T}\sum_{t=1}^T \boldsymbol{e}_{i_t}^\top M \boldsymbol{q}_t \leq \frac{1}{2} - \gamma$. Moreover, choosing $\boldsymbol{q} = \boldsymbol{e}_j$ we have $\frac{1}{T}\sum_{t=1}^T \boldsymbol{e}_{i_t}^\top M \boldsymbol{q} = \frac{1}{T}\sum_{i=1}^T \mathbf{1}[h_{i_t}(\boldsymbol{z}_j) \neq y_j]$. Putting all together, for any $j = 1, \dots, m$, we have

$$\frac{1}{T}\sum_{i=1}^T \mathbf{1}[h_{i_t}(\boldsymbol{z}_j) \neq y_j] \leq \frac{1}{2} - \gamma + \frac{\mathrm{Regret}_T^Y(\boldsymbol{e}_j)}{T} \,.$$

If $\frac{\mathrm{Regret}_T^Y(\boldsymbol{e}_j)}{T} < \gamma$, less than half of the functions selected by the boosting procedure will make a mistake on $(\boldsymbol{z}_j, y_j)$. Given that the predictor is a majority rule, this means that the majority rule will make 0 mistakes on the training samples.

In this scheme, we construct $p_i^\star$ by approximating it with the frequency with which i_t is equal to i.

Let's now instantiate this framework with a specific OCO algorithm. For example, using EG as algorithm for the Y-player, we have that $\mathrm{Regret}_T^Y(\boldsymbol{e}_j) = O(\sqrt{T \ln n})$ as $T \to \infty$ for any $j = 1, \dots, m$, that implies that after $T = O(\frac{\ln n}{\gamma^2})$ rounds the training error is exactly 0. This is exactly the same guarantee achieved by AdaBoost [Freund and Schapire, 1995, 1997].

Boosting and Margins What happens if we keep boosting after the training error reaches 0? It turns out we maximize the *margin*, defined as $y_j \frac{1}{T}\sum_{t=1}^T h_{i_t}(\boldsymbol{z}_j)$. In fact, given that $\mathbf{1}[h_{i_t}(\boldsymbol{z}_j) \neq y_j] = \frac{1 - y_j h_{i_t}(\boldsymbol{z}_j)}{2}$, we have for any $j = 1, \dots, m$

$$2\gamma - \frac{2\,\mathrm{Regret}_T^Y(\boldsymbol{e}_j)}{T} \leq y_j \frac{1}{T}\sum_{i=1}^T h_{i_t}(\boldsymbol{z}_j) = y_j \bar{h}_T(\boldsymbol{z}_j) \,.$$

Hence, when the number of rounds goes to infinity the minimum margin on the training samples reaches 2γ. This property of boosting of maximizing the margin has been used as an explanation of the fact that in boosting additional rounds after the training error reaches 0 often keep improving the test error on test samples coming i.i.d. from the same distribution that generated the training samples.

The above reduction does not tell us how the training error precisely behaves. However, we can get this information changing the learning with expert algorithm. Indeed, we have seen in Chapter 9 that there are learning with

experts algorithm provably better than EG. We can use a learning with experts algorithm that guarantees a regret $O(\sqrt{T \cdot KL(\boldsymbol{q}; \boldsymbol{\pi})})$ as $T \to \infty$ for a given prior $\boldsymbol{\pi}$, where $\boldsymbol{\pi}$ is the uniform prior. This kind of algorithms allow us to upper bound the fraction of mistakes after any T iterations. Denote by k the number of misclassified samples after T iterations of boosting and set $\boldsymbol{q}_k$ as the vector whose coordinates are equal to $1/k$ if $\operatorname{sign}(\bar{h}_T(\boldsymbol{z}))$ misclassifies it. Hence, we have

$$2\gamma - \frac{2\operatorname{Regret}_T^Y(\boldsymbol{q}_k)}{T} \leq \frac{1}{k}\sum_{j=1}^{k} y_j \bar{h}_T(\boldsymbol{z}_j) \leq 0 \,. \tag{11.5}$$

Using the expression of the regret, we have that the fraction of misclassification errors $\frac{k}{m}$ is bounded by $O(\exp(-\gamma^2 T))$ as $T \to \infty$. That is, fraction of misclassification error goes to zero exponential fast in the number of boosting rounds. Again, a similar guarantee was proved for AdaBoost, but here we quickly derived it using a reduction from boosting to learning with experts, passing through two-person games.

11.5 Faster Rates Through Optimism

We saw that it is possible to solve convex/concave saddle point optimization problems using two online convex optimization algorithms playing against each other. We obtained a rate of convergence for the duality gap of $O(1/\sqrt{T})$. Now, we show that if the function is smooth we can achieve a faster rate using optimistic algorithms.

Assume that $f : X \times Y \to \mathbb{R}$ is smooth in an open interval containing its domain, in the sense that for any $\boldsymbol{x}, \boldsymbol{x}' \in X$ and $\boldsymbol{y}, \boldsymbol{y}' \in Y$, we have

$$\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y}) - \nabla_{\boldsymbol{x}} f(\boldsymbol{x}', \boldsymbol{y})\|_{X,\star} \leq L_{XX}\|\boldsymbol{x} - \boldsymbol{x}'\|_X \tag{11.6}$$
$$\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y}) - \nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y}')\|_{X,\star} \leq L_{XY}\|\boldsymbol{y} - \boldsymbol{y}'\|_Y \tag{11.7}$$
$$\|\nabla_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}) - \nabla_{\boldsymbol{y}} f(\boldsymbol{x}', \boldsymbol{y})\|_{Y,\star} \leq L_{XY}\|\boldsymbol{x} - \boldsymbol{x}'\|_X \tag{11.8}$$
$$\|\nabla_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}) - \nabla_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}')\|_{Y,\star} \leq L_{YY}\|\boldsymbol{y} - \boldsymbol{y}'\|_Y, \tag{11.9}$$

where $\nabla_{\boldsymbol{x}}$ and $\nabla_{\boldsymbol{y}}$ denote the gradients with respect to the first and second variable respectively, and we have denoted by $\|\cdot\|_X$ and $\|\cdot\|_Y$ the norms in X and Y respectively, while the norms with the $\star$ are their duals.

Remark 11.22. *At this point, one might be tempted to consider the maximum between the three quantities "to simplify the math", but the units are different!*

Let's use again two online algorithms to solve the saddle point problem $\min_{\boldsymbol{x}\in X}\max_{\boldsymbol{y}\in Y} f(\boldsymbol{x}, \boldsymbol{y})$. However, instead of using two standard no-regret algorithms, we will use two *optimistic ones*. Optimistic online algorithms use a hint on the next subgradient. We will use the same strategy and proof of the algorithm in Section 7.12.2, that is we will use the previous observed gradient as a prediction for the next one.

For example, use two Optimistic FTRL algorithms with fixed strongly convex regularizers and hint at time t constructed using the previous observed gradient: $\tilde{\ell}_t(\boldsymbol{x}) = \langle \boldsymbol{g}_{t-1}, \boldsymbol{x}\rangle$ where we set $\boldsymbol{g}_0 = 0$. We now show that these hints allow to cancel out terms when we consider the sum of the regrets and obtain a faster rate of $O(1/T)$ rather than just $O(1/\sqrt{T})$.

From the regret of Optimistic FTRL, for the X-player we have

$$\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell(\boldsymbol{u})) \leq \psi_X(\boldsymbol{u}) + \sum_{t=1}^{T}\left(\langle \boldsymbol{g}_{X,t} - \boldsymbol{g}_{X,t-1}, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle - \frac{\lambda_X}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|_X^2\right), \ \forall \boldsymbol{u} \in X \,.$$

From the Fenchel-Young inequality, we have $\langle \boldsymbol{g}_{X,t} - \boldsymbol{g}_{X,t-1}, \boldsymbol{x}_t - \boldsymbol{x}_{t+1}\rangle \leq \frac{\lambda_X}{4}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|_X^2 + \frac{1}{\lambda_X}\|\boldsymbol{g}_{X,t} - \boldsymbol{g}_{X,t-1}\|_{X,\star}^2$. Putting all together, we have

$$\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell(\boldsymbol{u})) \leq \psi_X(\boldsymbol{u}) + \sum_{t=1}^{T}\left(\frac{1}{\lambda_X}\|\boldsymbol{g}_{X,t} - \boldsymbol{g}_{X,t-1}\|_{X,\star}^2 - \frac{\lambda_X}{4}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|_X^2\right), \ \forall \boldsymbol{u} \in X \,.$$

Algorithm 11.6 Solving Saddle-Point Problems with Optimistic FTRL

Require: $\lambda_X > 0, \lambda_Y > 0$

$\boldsymbol{g}_{X,0} = \mathbf{0}, \boldsymbol{g}_{Y,0} = \mathbf{0}$

for $t = 1, \dots, T$ **do**

  $\boldsymbol{x}_t = \operatorname{argmin}_{\boldsymbol{x} \in X} \ \psi_X(\boldsymbol{x}) + \langle \boldsymbol{g}_{X,t-1}, \boldsymbol{x} \rangle + \sum_{i=1}^{t-1} \langle \boldsymbol{x}, \boldsymbol{g}_{X,i} \rangle$

  $\boldsymbol{y}_t = \operatorname{argmin}_{\boldsymbol{y} \in Y} \ \psi_Y(\boldsymbol{y}) + \langle \boldsymbol{g}_{Y,t-1}, \boldsymbol{y} \rangle + \sum_{i=1}^{t-1} \langle \boldsymbol{y}, \boldsymbol{g}_{Y,i} \rangle$

  Set $\boldsymbol{g}_{X,t} = \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_t, \boldsymbol{y}_t)$

  Set $\boldsymbol{g}_{Y,t} = -\nabla_{\boldsymbol{y}} f(\boldsymbol{x}_t, \boldsymbol{y}_t)$

end for

return $\bar{\boldsymbol{x}}_T = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{x}_t$, $\bar{\boldsymbol{y}}_T = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{y}_t$

Note that there are multiple choices of the coefficient in the Fenchel-Young inequality, but without additional information all choices are equally good.

Now, using the smoothness assumption, for $t \geq 2$ we have

$$\begin{aligned}
\|\boldsymbol{g}_{X,t} - \boldsymbol{g}_{X,t-1}\|_{X,\star}^2 &= \|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_t, \boldsymbol{y}_t) - \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1})\|_{X,\star}^2 \\
&\leq \left(\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_t, \boldsymbol{y}_t) - \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_{t-1}, \boldsymbol{y}_t)\|_{X,\star} + \|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_{t-1}, \boldsymbol{y}_t) - \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_{t-1}, \boldsymbol{y}_{t-1})\|_{X,\star} \right)^2 \\
&\leq 2L_{XX}^2 \|\boldsymbol{x}_{t-1} - \boldsymbol{x}_t\|_X^2 + 2L_{XY}^2 \|\boldsymbol{y}_{t-1} - \boldsymbol{y}_t\|_Y^2 \ .
\end{aligned}$$

We can proceed in the exact same way for the Y-player too.

Summing the regret of the two algorithms, we have

$$\begin{aligned}
\sum_{t=1}^{T} f(\boldsymbol{x}_t, \boldsymbol{y}) - \sum_{t=1}^{T} f(\boldsymbol{x}, \boldsymbol{y}_t) &\leq \psi_X(\boldsymbol{x}) + \psi_Y(\boldsymbol{y}) + \frac{\|\boldsymbol{g}_{X,1}\|_{X,\star}^2}{\lambda_X} + \frac{\|\boldsymbol{g}_{Y,1}\|_{Y,\star}^2}{\lambda_Y} \\
&\quad + \sum_{t=2}^{T} \left(\left(\frac{2L_{XX}^2}{\lambda_X} + \frac{2L_{XY}^2}{\lambda_Y} - \frac{\lambda_X}{4} \right) \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_X^2 + \left(\frac{2L_{YY}^2}{\lambda_Y} + \frac{2L_{XY}^2}{\lambda_X} - \frac{\lambda_Y}{4} \right) \|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|_Y^2 \right) \ .
\end{aligned}$$

Choosing $\lambda_X \geq 2\sqrt{2}(L_{XX} + L_{XY}\alpha)$ and $\lambda_Y \geq 2\sqrt{2}(L_{YY} + L_{XY}/\alpha)$ for any $\alpha > 0$ kills all the terms in the sum. In fact, we have

$$\frac{2L_{XX}^2}{\lambda_X} + \frac{2L_{XY}^2}{\lambda_Y} \leq \frac{2L_{XX}^2}{2\sqrt{2}L_{XX}} + \frac{2L_{XY}^2\alpha}{2\sqrt{2}L_{XY}} \leq \frac{\lambda_X}{4},$$

and similarly for the other term. One might wonder why we need to introduce α and if it can be just set to 1. However, α has units and it allows the sum of the smoothness coefficients, so it is better to keep it around to remember it.

Assuming that the regularizers are bounded over X and Y and using the usual online-to-batch conversion, we have that the duality gap evaluated at the pair $\left(\frac{1}{T} \sum_{t=1}^{T} \boldsymbol{x}_t, \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{y}_t \right)$ goes to zero as $O(1/T)$ when $T \to \infty$.

Overall, we can state the following theorem.

Theorem 11.23. *With the notation in Algorithm 11.6, let $f : X \times Y \to \mathbb{R}$ convex in the first argument and concave in the second, satisfying assumptions* (11.6)-(11.9)*. For a fixed $\alpha > 0$, let $\lambda_X \geq 2\sqrt{2}(L_{XX} + L_{XY}\alpha)$ and $\lambda_Y \geq 2\sqrt{2}(L_{YY} + L_{XY}/\alpha)$. Let $\psi_X : X \to \mathbb{R}$ be λ_X-strongly convex w.r.t. $\|\cdot\|_X$ and $\psi_Y : Y \to \mathbb{R}$ be λ_Y-strongly convex w.r.t. $\|\cdot\|_Y$. Assume* $\arg\max_{\boldsymbol{y} \in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y})$ *and* $\arg\min_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T)$ *non-empty. Then, we have*

$$\max_{\boldsymbol{y} \in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y}) - \min_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T) \leq \frac{\psi_X(\boldsymbol{x}'_T) - \psi_X(\boldsymbol{x}_1) + \psi_Y(\boldsymbol{y}'_T) - \psi_Y(\boldsymbol{y}_1) + \frac{\|\boldsymbol{g}_{X,1}\|_{X,\star}^2}{\lambda_X} + \frac{\|\boldsymbol{g}_{Y,1}\|_{Y,\star}^2}{\lambda_Y}}{T},$$

for any $\boldsymbol{x}'_T \in \arg\min_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T)$ and $\boldsymbol{y}'_T \in \arg\max_{\boldsymbol{y} \in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y})$.

Looking back at the proof of the algorithm, we have a faster convergence because regret of one player depends on the "stability" of the other player, measured by the terms $\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_X^2$ and $\|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|_Y^2$. Hence, we have a sort

of stabilization loop in which the stability of one algorithm makes the other more stable, that in turn stabilizes the first one even more. Indeed, we can also show that the regret of the two algorithms is not growing over time. Note that such result cannot be obtained just looking at the fact that the sum of the regret does not grow over time.

In fact, setting for example $\lambda_X \geq 4\sqrt{2}(L_{XX} + L_{XY}\alpha)$ and $\lambda_Y \geq 4\sqrt{2}(L_{YY} + L_{XY}/\alpha)$, we have that

$$\frac{2L_{YY}^2}{\lambda_Y} + \frac{2L_{XY}^2}{\lambda_X} - \frac{\lambda_Y}{4} \leq -\frac{\lambda_Y}{8}$$

and

$$\frac{2L_{XX}^2}{\lambda_X} + \frac{2L_{XY}^2}{\lambda_Y} - \frac{\lambda_X}{4} \leq -\frac{\lambda_X}{8} \, .$$

Hence, using the fact that the existence of a saddle point $(\boldsymbol{x}^\star, \boldsymbol{y}^\star)$ guarantee that $f(\boldsymbol{x}_t, \boldsymbol{y}^\star) - f(\boldsymbol{x}^\star, \boldsymbol{y}_t) \geq 0$, we have

$$\sum_{t=2}^{T} \left(\frac{\lambda_X}{8} \|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_X^2 + \frac{\lambda_Y}{8} \|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|_Y^2 \right) \leq \psi_X(\boldsymbol{x}^\star) - \psi_X(\boldsymbol{x}_1) + \psi_Y(\boldsymbol{y}^\star) - \psi_Y(\boldsymbol{y}_1) + \frac{\|\boldsymbol{g}_{X,1}\|_{X,\star}^2}{\lambda_X} + \frac{\|\boldsymbol{g}_{Y,1}\|_{Y,\star}^2}{\lambda_Y} \, . \tag{11.10}$$

Plugging this guarantee back in the regret of each algorithm, we have that their regret is bounded and independent of T. From (11.10), we also have that $\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_X^2$ and $\|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|_Y^2$ converge 0. Hence, the algorithms are getting more and more stable over time, even if they use constant regularizers.

Version with Optimistic OMD The exact same reasoning holds for Optimistic OMD, because the key terms of its regret bound are exactly the same of the one of Optimistic FTRL. To better show this fact, we also instantiate the Optimistic OMD with stepsizes equal to $\frac{1}{\lambda_X}$ and $\frac{1}{\lambda_Y}$ for X-player and Y-player respectively. Following the same reasoning above and the regret bound of Optimistic OMD, we obtain the following theorem.

Theorem 11.24. *With the notation in Algorithm 11.7, let $f : X \times Y \to \mathbb{R}$ convex in the first argument and concave in the second, satisfying assumptions* (11.6)-(11.9)*. For a fixed $\alpha > 0$, let $\lambda_X \geq 2\sqrt{2}(L_{XX} + L_{XY}\alpha)$ and $\lambda_Y \geq 2\sqrt{2}(L_{YY} + L_{XY}/\alpha)$. Let $\psi_X : X \to \mathbb{R}$ be 1-strongly convex w.r.t. $\|\cdot\|_X$ and $\psi_Y : Y \to \mathbb{R}$ be 1-strongly convex w.r.t. $\|\cdot\|_Y$. Assume $\arg\max_{\boldsymbol{y} \in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y})$ and $\arg\min_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T)$ non-empty. Then, we have*

$$\max_{\boldsymbol{y} \in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y}) - \min_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T) \leq \frac{B_{\psi_X}(\boldsymbol{x}'_T; \boldsymbol{x}_1) + B_{\psi_Y}(\boldsymbol{y}'_T; \boldsymbol{y}_1) + \frac{\|\boldsymbol{g}_{X,1}\|_{X,\star}^2}{\lambda_X} + \frac{\|\boldsymbol{g}_{Y,1}\|_{Y,\star}^2}{\lambda_Y}}{T},$$

for any $\boldsymbol{x}'_T \in \arg\min_{\boldsymbol{x} \in X} f(\boldsymbol{x}, \bar{\boldsymbol{y}}_T)$ and $\boldsymbol{y}'_T \in \arg\max_{\boldsymbol{y} \in Y} f(\bar{\boldsymbol{x}}_T, \boldsymbol{y})$.

Algorithm 11.7 Solving Saddle-Point Problems with Optimistic OMD

Require: $\lambda_X > 0, \lambda_Y > 0, \boldsymbol{x}_1 \in X, \boldsymbol{y}_1 \in Y$
1: $\boldsymbol{g}_{X,0} = \boldsymbol{0}, \boldsymbol{g}_{Y,0} = \boldsymbol{0}$
2: **for** $t = 1, \ldots, T$ **do**
3: Set $\boldsymbol{g}_{X,t} = \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_t, \boldsymbol{y}_t)$
4: Set $\boldsymbol{g}_{Y,t} = -\nabla_{\boldsymbol{y}} f(\boldsymbol{x}_t, \boldsymbol{y}_t)$
5: $\boldsymbol{x}_{t+1} = \operatorname{argmin}_{\boldsymbol{x} \in X} \langle 2\boldsymbol{g}_{X,t} - \boldsymbol{g}_{X,t-1}, \boldsymbol{x} \rangle + \lambda_X B_{\psi_X}(\boldsymbol{x}; \boldsymbol{x}_t)$
6: $\boldsymbol{y}_{t+1} = \operatorname{argmin}_{\boldsymbol{y} \in Y} \langle 2\boldsymbol{g}_{Y,t} - \boldsymbol{g}_{Y,t-1}, \boldsymbol{y} \rangle + \lambda_Y B_{\psi_Y}(\boldsymbol{y}; \boldsymbol{y}_t)$
7: **end for**
8: **return** $\bar{\boldsymbol{x}}_T = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{x}_t, \bar{\boldsymbol{y}}_T = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{y}_t$

Example 11.25. *Consider the bilinear saddle point problem*

$$\min_{\boldsymbol{x} \in X} \max_{\boldsymbol{y} \in Y} \boldsymbol{x}^\top A \boldsymbol{y} \, .$$

In this case, we have that $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{y}) = A\boldsymbol{y}$, $\nabla_{\boldsymbol{y}} f(\boldsymbol{x}, \boldsymbol{y}) = A^\top \boldsymbol{x}$, $L_{XX} = 0$, $L_{YY} = 0$, and $L_{XY} = \|A\|_{op}$ where $\|\cdot\|_{op}$ is the operator norm of the matrix A. The specific shape of the operator norm depends on the norms we use on X and Y. For example, we choose the Euclidean norm on both X and Y, the operator norm of A is the largest singular value of A. On the other hand, if $X = \Delta^{n-1}$ and Δ^{m-1} as in the two-person zero-sum games, then the operator norm of a matrix A is the maximum absolute value of the entries of A.

11.6 Prescient Online Mirror Descent and Be-The-Regularized-Leader

The above result is interesting from a game-theoretic point of view, because it shows that two player can converge to an equilibrium without any "communication", if instead we only care about converging to the saddle point, we can easily do better. In fact, we can use the fact that it is fine if one of the two players "cheats" by looking at the loss at the beginning of each round, making its regret non-positive.

For example, we saw the use of Best Response. However, Best Response only guarantees non-positive regret, while for the optimistic proof above we need some specific negative terms. It turns out we can achieve them with another algorithm: Prescient Online Mirror Descent, that predicts in each round with $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x} \in V} \ \ell_t(\boldsymbol{x}) + \frac{1}{\eta_t} B_\psi(\boldsymbol{x}; \boldsymbol{x}_{t-1})$.

Algorithm 11.8 Prescient Online Mirror Descent

Require: Non-empty closed convex $V \subseteq X \subseteq \mathbb{R}^d$, $\psi : X \to \mathbb{R}$ strictly convex and differentiable on $\operatorname{int} X$, $\boldsymbol{x}_0 \in \operatorname{int} X$, $\eta_1, \dots, \eta_T > 0$

1: **for** $t = 1$ **to** T **do**
2: Receive $\ell_t : V \to \mathbb{R}$ subdifferentiable in V
3: $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x} \in V} \ \ell_t(\boldsymbol{x}) + \frac{1}{\eta_t} B_\psi(\boldsymbol{x}; \boldsymbol{x}_{t-1})$
4: Pay $\ell_t(\boldsymbol{x}_t)$
5: **end for**

Theorem 11.26. *Let $\psi : X \to \mathbb{R}$ differentiable in $\operatorname{int} X$, closed, and strictly convex. Let $V \subseteq X$ a non-empty closed convex set. Assume $\boldsymbol{x}_t \in \operatorname{int} X$, ℓ_t subdifferentiable in V, and $\eta_{t+1} \leq \eta_t$, for $t = 1, \dots, T$. Then, $\forall \boldsymbol{u} \in V$, the following inequality holds*

$$\sum_{t=1}^{T} \ell_t(\boldsymbol{x}_{t+1}) - \sum_{t=1}^{T} \ell_t(\boldsymbol{u}) \leq \max_{1 \leq t \leq T-1} \frac{B_\psi(\boldsymbol{u}; \boldsymbol{x}_t)}{\eta_T} - \sum_{t=1}^{T} \frac{1}{\eta_t} B_\psi(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) \ .$$

Moreover, if η_t is constant, i.e., $\eta_t = \eta \ \forall t = 1, \cdots, T$, we have

$$\sum_{t=1}^{T} (\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) \leq \frac{B_\psi(\boldsymbol{u}; \boldsymbol{x}_0)}{\eta} - \frac{1}{\eta} \sum_{t=1}^{T} B_\psi(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}) \ .$$

Proof. From the first-order optimality condition on the update, we have that there exists $\boldsymbol{g}_t \in \partial \ell_t(\boldsymbol{x}_t)$ such

$$\langle \eta_t \boldsymbol{g}_t + \nabla \psi(\boldsymbol{x}_t) - \nabla \psi(\boldsymbol{x}_{t-1}), \boldsymbol{u} - \boldsymbol{x}_t \rangle \geq 0, \quad \forall \boldsymbol{u} \in V \ .$$

Hence, we have

$$\begin{aligned} \eta_t(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\leq \langle \eta_t \boldsymbol{g}_t, \boldsymbol{x}_t - \boldsymbol{u} \rangle = \langle \nabla \psi(\boldsymbol{x}_{t-1}) - \nabla \psi(\boldsymbol{x}_t), \boldsymbol{x}_t - \boldsymbol{u} \rangle + \langle \eta_t \boldsymbol{g}_t + \nabla \psi(\boldsymbol{x}_t) - \nabla \psi(\boldsymbol{x}_{t-1}), \boldsymbol{x}_t - \boldsymbol{u} \rangle \\ &\leq \langle \nabla \psi(\boldsymbol{x}_{t-1}) - \nabla \psi(\boldsymbol{x}_t), \boldsymbol{x}_t - \boldsymbol{u} \rangle \\ &= B_\psi(\boldsymbol{u}, \boldsymbol{x}_{t-1}) - B_\psi(\boldsymbol{u}, \boldsymbol{x}_t) - B_\psi(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}), \end{aligned}$$

where in the last equality we used Lemma 6.6. Dividing by η_t and summing over $t = 1, \dots, T$, we have

$$\begin{aligned}
\sum_{t=1}^{T}(\ell_t(\boldsymbol{x}_t) - \ell_t(\boldsymbol{u})) &\leq \sum_{t=1}^{T}\left(\frac{1}{\eta_t}B_\psi(\boldsymbol{u};\boldsymbol{x}_{t-1}) - \frac{1}{\eta_t}B_\psi(\boldsymbol{u};\boldsymbol{x}_t)\right) - \sum_{t=1}^{T}\frac{1}{\eta_t}B_\psi(\boldsymbol{x}_t,\boldsymbol{x}_{t-1}) \\
&= \frac{1}{\eta_1}B_\psi(\boldsymbol{u};\boldsymbol{x}_0) - \frac{1}{\eta_T}B_\psi(\boldsymbol{u};\boldsymbol{x}_T) + \sum_{t=1}^{T-1}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right)B_\psi(\boldsymbol{u};\boldsymbol{x}_t) - \sum_{t=1}^{T}\frac{1}{\eta_t}B_\psi(\boldsymbol{x}_t,\boldsymbol{x}_{t-1}) \\
&\leq \frac{1}{\eta_1}D^2 + D^2\sum_{t=1}^{T-1}\left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right) - \sum_{t=1}^{T}\frac{1}{\eta_t}B_\psi(\boldsymbol{x}_t,\boldsymbol{x}_{t-1}) \\
&= \frac{1}{\eta_1}D^2 + D^2\left(\frac{1}{\eta_T} - \frac{1}{\eta_1}\right) - \sum_{t=1}^{T}\frac{1}{\eta_t}B_\psi(\boldsymbol{x}_t,\boldsymbol{x}_{t-1}) \\
&= \frac{D^2}{\eta_T} - \sum_{t=1}^{T}\frac{1}{\eta_t}B_\psi(\boldsymbol{x}_t,\boldsymbol{x}_{t-1}),
\end{aligned}$$

where we denoted by $D^2 = \max_{1\leq t\leq T-1} B_\psi(\boldsymbol{u};\boldsymbol{x}_t)$.

The second statement is left as exercise. □

The regret of Prescient Online Mirror Descent contains the negative terms we needed from the optimistic algorithms.

Analogously, we can obtain a version of FTRL that uses the knowledge of the current loss: Be-The-Regularized-Leader (BTRL), that predicts in each time step with $\boldsymbol{x}_t \in \operatorname{argmin}_{\boldsymbol{x}\in V}\ \psi_t(\boldsymbol{x}) + \sum_{i=1}^{t}\ell_i(\boldsymbol{x})$. In the case that $\psi_t \equiv 0$, then Be-The-Regularized-Leader becomes the Be-The-Leader algorithm. BTRL can be thought as Optimistic FTRL where $\tilde{\ell}_t = \ell_t$. Hence, from the regret of Optimistic FTRL, we immediately have the following theorem.

Theorem 11.27. *Let $V \subseteq \mathbb{R}^d$ be convex, closed, and non-empty. Assume for $t = 1, \dots, T$ that $\psi_t + \sum_{i=1}^{t}\ell_i$ is proper, closed, and λ_t-strongly convex w.r.t. $\|\cdot\|$. Then, for all $\boldsymbol{u} \in V$ we have*

$$\sum_{t=1}^{T}\ell(\boldsymbol{x}_t) + \sum_{t=1}^{T}\ell_t(\boldsymbol{u}) \leq \psi_{T+1}(\boldsymbol{u}) - \psi_1(\boldsymbol{x}_1) - \sum_{t=1}^{T}\left(-\frac{\lambda_t}{2}\|\boldsymbol{x}_t - \boldsymbol{x}_{t+1}\|^2 + \psi_t(\boldsymbol{x}_{t+1}) - \psi_{t+1}(\boldsymbol{x}_{t+1})\right).$$

Remark 11.28. *In the Be-The-Leader algorithm, if all the $\lambda_t = 0$, then the theorem states that the regret is non-positive.*

Notably, the non-negative gradients terms are missing in the bound of BTRL, but we still have the negative ones associated to the change in $\boldsymbol{x}_t$.

Using, for example, BTRL for the X-player and Optimistic FTRL for the Y-player, we have

$$\begin{aligned}
\sum_{t=1}^{T}f(\boldsymbol{x}_t,\boldsymbol{y}) - \sum_{t=1}^{T}f(\boldsymbol{x},\boldsymbol{y}_t) &\leq \psi_X(\boldsymbol{x}) + \psi_Y(\boldsymbol{y}) + \frac{\|\boldsymbol{g}_1^Y\|_{Y,\star}^2}{\lambda_Y} \\
&\quad + \sum_{t=2}^{T}\left(\left(\frac{2L_{XY}^2}{\lambda_Y} - \frac{\lambda_X}{4}\right)\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_X^2 + \left(\frac{2L_{YY}^2}{\lambda_Y} - \frac{\lambda_Y}{4}\right)\|\boldsymbol{y}_t - \boldsymbol{y}_{t-1}\|_Y^2\right).
\end{aligned}$$

11.7 History Bits

Theorem 11.3 is Rockafellar [1970, Lemma 36.2].

The proof of Theorem 11.13 is from Liu and Orabona [2021] and in turn is based on the one in Abernethy and Wang [2017]: Liu and Orabona [2021] stressed the dependency of the regret on a competitor that can be useful for refined bounds. Different variant of this theorem are known in the game theory community as "Folk Theorems", because such result was widely known among game theorists in the 1950s, even though no one had published it.

The celebrated minimax theorem for zero-sum two-person games was first discovered by John von Neumann in the 1920s [Neumann, 1928, Neumann and Morgenstern, 1944]. The version is state here is a simplification of the generalization due to Sion [1958]. The proof here is from Abernethy and Wang [2017]. A similar proof is in Cesa-Bianchi and Lugosi [2006] based on a discretization of the space that in turn is based on the one in Freund and Schapire [1996, 1999b].

Algorithms 11.2 and 11.3 are a generalization of the algorithm for boosting in Freund and Schapire [1996, 1999b]. Algorithm 11.2 was also used in Abernethy and Wang [2017] to recover variants of the Frank-Wolfe algorithm [Frank and Wolfe, 1956].

It is not clear who invented alternation: It was a known trick used in implementations of CFR for the computer poker competition from 2010 or so[1]. Note that in CFR the method of choice is Regret Matching [Hart and Mas-Colell, 2000]. However, Kroer [2020] empirically shows that alternation improves a lot even OGD for solving bilinear games. Tammelin et al. [2015] explicitly include this trick in their implementation of an improved version of CFR called CFR+, claiming that it would still guarantee convergence. However, Farina et al. [2019] pointed out that averaging of the iterates in alternation might not produce a solution to the min-max problem, providing a counterexample. Theorem 11.17 is from Burch et al. [2019].

There is also a complementary view on alternation: Zhang et al. [2021] link alternating updates to Gauss-Seidel methods in numerical linear algebra, in contrast to the simultaneous updates of the Jacobi method. Also, they provide a good review of the optimization literature on the advantages of alternation, but this paper and the papers they cite do not seem to be aware of the use of alternation in CFR.

The reduction from boosting to learning with expert is from Freund and Schapire [1996]. It seems that the question if a weak learner can be boosted into a strong learner was originally posed by Kearns and Valiant [1988] (see also Kearns [1988]) but I could not verify this claim because I could not find the paper anywhere. It was answered in the positive by Schapire [1990]. The AdaBoost algorithm is from Freund and Schapire [1995, 1997]. The idea of using algorithms that guarantee a KL regret bound in (11.5) is from Luo and Schapire [2014].

Daskalakis et al. [2011] proposed the first no-regret algorithm that achieved a rate of $O(\frac{\ln T}{T})$ for the duality gap when used by the two players of a zero-sum game without any communication between the players. However, the algorithm was rather complex and they posed the problem of obtaining the same or faster rate with a simpler algorithm. Rakhlin and Sridharan [2013a] solved this problem showing that two Optimistic OMD algorithms can solve the problem in a simpler way. Theorems 11.23 and 11.24 derive directly from Rakhlin and Sridharan [2013a, Corollary 5]. For some reason Rakhlin and Sridharan [2013a, Corollary 5] was missed in recent years, so the $O(1/T)$ convergence for smooth saddle-point problems using optimistic gradient descent/ascent has been rediscovered a number of times [e.g., Hsieh et al., 2019, Mokhtari et al., 2020] However, the optimistic gradient descent/ascent for saddle-point problems is much older: It was proposed for the first time by Popov [1980], as a modification of the Arrow-Hurwicz method. Roughly 30 years later, the optimistic algorithms were rediscovered, first as pure online learning algorithms [Chiang et al., 2012, Rakhlin and Sridharan, 2013b] and then used to solve saddle-point problems [Rakhlin and Sridharan, 2013a].

The possibility to achieve constant regret for each player observed after Theorem 11.23 is from Luo [2022].

The use of Prescient Online Mirror Descent in saddle point optimization is from Wang et al. [2021], but when renaming $\boldsymbol{x}_t$ to $\boldsymbol{x}_{t+1}$ it is also equivalent to implicit online mirror descent [Kivinen and Warmuth, 1997, Kulis and Bartlett, 2010, Campolongo and Orabona, 2020]. Theorem 11.26 is from the guarantee of implicit online mirror descent in Campolongo and Orabona [2020].

There is also a tight connection between optimistic updates using the previous gradients and classic approaches to solve saddle-point optimization. In fact, Gidel et al. [2019] showed that using two optimistic gradient descent algorithms to solve a saddle point problems can be seen as a variant of the Extra-gradient updates [Korpelevich, 1976], while Mokhtari et al. [2020] show that they can be interpreted as an approximated proximal point algorithm.

For generic saddle-point problems, Popov [1980] proved the asymptotic convergence of the iterates when using two optimistic OGD algorithms. This old result was unknown to the majority of the community until recently and it implies some later weaker results [e.g., Daskalakis et al., 2018]. The extension to the Mirror Descent case was done by Semenov [2017] to solve the more general problem of variational inequalities, but only for distance generating

[1] Christian Kroer, 2021, personal communication.

functions that are differentiable on the entire feasible set.[2] This means that this proof does not cover the optimistic exponentiated gradient. In turn, this little known result was also recently rediscovered [e.g., Lee et al., 2021, Theorem 4]. Hsieh et al. [2021, Theorem 7] proved the asymptotic convergence of the last iterate for Optimistic FTRL with linear losses and an adaptive regularization weight, assuming either strict convexity/concavity or that the regularizer to be differentiable on the entire domain. Note that the proof in Hsieh et al. [2021, Theorem 7] can be easily adapted to prove the same result for optimistic OMD with a fixed and small enough learning rate. Lei et al. [2021] prove the asymptotic convergence of optimistic exponentiated gradient for saddle-point problems, under the assumption that some of the optimality conditions are satisfied in a strict way.

For the specific case of bilinear games, stronger results can be proven on the last-iterate convergence. Liang and Stokes [2019] proved that if the matrix A is square, full-rank, and the problem unconstrained, then the iterates of two Optimistic OGD will converge exponentially fast to the saddle point in the origin. Daskalakis and Panageas [2019] proved the asymptotic convergence of optimistic OMD/FTRL Exponentiated Gradient with fixed stepsize for bilinear games over probability simplexes, assuming a unique saddle point. Wei et al. [2021] proved an exponential rate for the same algorithm under the same assumptions. Hsieh et al. [2021, Theorem 8] removed the unique saddle point assumption, proving asymptotic convergence for Optimistic FTRL with entropic regularization and an adaptive regularization weight. Once again, this proof can be easily modified to prove the same result for optimistic OMD with a fixed and small enough learning rate.

11.8 Exercises

Problem 11.1. *Prove Lemma 11.11.*

Problem 11.2. *Let $f(\boldsymbol{x}, \boldsymbol{y})$ be a convex-concave function, Lipschitz with respect to both variables. Use two FTRL algorithms to find the saddle-point, using as regularizers $\psi_t(\boldsymbol{x}) \propto \sqrt{t}\psi(\boldsymbol{x})$, where ψ is differentiable and 1-strongly convex with respect to a norm $\|\cdot\|$. Then, using the inequality in Exercise 7.2, show that the trajectory of the iterates of the above strategy are bounded on any convex-concave saddle-point problem with at least one saddle-point, even in unbounded domains and any number of dimensions.*

[2] Equation 17 in Semenov [2017] needs the fact that the distance generating function is continuously differentiable everywhere for a limit operation. This assumption is not stated, making the proof essentially wrong, for example, for the entropy function. As far as I know, no one had realized this issue in that paper before.

Chapter 12

Sequential Investment and Universal Portfolio Algorithms

We now describe another online learning problem: Sequential investment in a market with $d \geq 2$ stocks. The behavior of the market is specified by arbitrarily chosen non-negative market gains vectors $\boldsymbol{w}_1, \dots, \boldsymbol{w}_T$, each of them in $\mathbb{R}^d_{\geq 0}$. The coordinates of the market gains vectors represent the ratio between closing and opening price for the stocks. In this game, each day we allocate our wealth buying a number of stocks and at the end of the day we sell all the stocks. An investment strategy is specified by a vector $\boldsymbol{x}_t \in \mathbb{R}^d$, where $0 \leq x_{t,i} \leq 1$ and $\|\boldsymbol{x}_t\|_1 = 1$, and its elements specifies the fraction of the wealth invested on each stock at time t. We will also assume that the wealth of the investor is infinitely divisible. Hence, assuming an initial wealth of \$1, our wealth at the end of round T is

$$\mathrm{Wealth}_T = \sum_{i=1}^{d} \mathrm{Wealth}_{T-1}\, w_{T,i} x_{T,i} = \mathrm{Wealth}_{T-1} \langle \boldsymbol{w}_T, \boldsymbol{x}_T \rangle = \prod_{t=1}^{T} \langle \boldsymbol{w}_t, \boldsymbol{x}_t \rangle \ .$$

To define a regret, we have to decide what is the class of comparators. Here, we compare with the *best constant rebalanced portfolio*. A constant rebalanced portfolio follows the same strategy we use, but its allocation of the stocks is the same on each time step. Denoting by $\boldsymbol{u}$ the vector in the simplex of the constant rebalanced portfolio and initial wealth of \$1, its wealth is $\mathrm{Wealth}_T(\boldsymbol{u}) = \prod_{t=1}^{T} \langle \boldsymbol{w}_t, \boldsymbol{u} \rangle$.

Given the multiplicative nature of this game, difference of wealths do not make much sense. Hence, we consider the ratio of the wealth of the best constant rebalanced portfolio and the one of the algorithm or equivalently the difference of the logarithms:

$$\mathrm{Regret}_T(\boldsymbol{u}) = \ln \mathrm{Wealth}_T(\boldsymbol{u}) - \ln \mathrm{Wealth}_T = \sum_{t=1}^{T} \ln \langle \boldsymbol{w}_t, \boldsymbol{u} \rangle - \sum_{t=1}^{T} \ln \langle \boldsymbol{w}_t, \boldsymbol{x}_t \rangle \ . \tag{12.1}$$

Rewritten in this way, it should be clear that this game is just an online convex optimization game where $V = \Delta^{d-1}$ and the convex losses are $\ell_t(\boldsymbol{x}) = -\ln \langle \boldsymbol{w}_t, \boldsymbol{x} \rangle$.

We will say that a portfolio algorithm is **universal** if the regret against any constant rebalanced portfolio and with any sequence of market gain vectors is sublinear in time.

Why Constant Rebalanced Portfolios? The class of strategy has advantages that are not immediate to see. First of all, despite of the word "constant" this strategy is a very active one and very different from the buy-and-hold one that consists in buying a set of stocks in the first round and selling them at the end of the game. Let's consider an example that clearly shows its advantage. Consider only 2 stocks with a sequence of market vectors equal to $(1, \frac{1}{2}), (1, 2), (1, \frac{1}{2}), (1, 2), \dots$. On the long run none of these two stocks yield any value and so the buy-and-hold strategy does not bring any growth. Instead, the best constant rebalanced portfolio here is $\boldsymbol{u} = (\frac{1}{2}, \frac{1}{2})$ and it gives an exponentially increasing wealth of $(\frac{9}{8})^{t/2} \approx 1.06^t$. If it seems slow, we need to realize that this means that you would double your wealth in 12 days.

Another motivation to compare with constant rebalanced portfolios is that it can be shown that in the case the market vectors are i.i.d. from some fixed (unknown) distribution, then the constant rebalanced portfolio is asymptotically optimal.

12.1 Portfolio Selection with Exponentiated Gradient

Given that the problem in (12.1) is an online convex optimization game, we could think of using any of the algorithms we saw till now. In particular, given that the feasible set is the probability simplex and that the loss are convex, we can try to use Exponentiated Gradient. However, this problem turns out to be particularly challenging. In fact, the loss functions are not Lipschitz, so their gradients are unbounded.

In fact, using the upper bound to the regret of exponentiated gradient, we have

$$\sum_{t=1}^{T} \ln\langle \boldsymbol{w}_t, \boldsymbol{u}\rangle - \sum_{t=1}^{T} \ln\langle \boldsymbol{w}_t, \boldsymbol{x}_t\rangle \leq \frac{\ln d}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T} \|\boldsymbol{g}_t\|_\infty^2 = \frac{\ln d}{\eta} + \frac{\eta}{2}\sum_{t=1}^{T} \frac{\|\boldsymbol{w}_t\|_\infty^2}{(\langle \boldsymbol{x}_t, \boldsymbol{w}_t\rangle)^2}\,.$$

The terms in the sum are potentially unbounded, so we will assume that the market gains are in the range $[c, C]$. In this way, we have $\frac{\|\boldsymbol{w}_t\|_\infty^2}{(\langle \boldsymbol{x}_t, \boldsymbol{w}_t\rangle)^2} \leq \frac{C^2}{c^2}$. Setting the learning rate to $\eta = \frac{\sqrt{\ln d}}{\frac{C}{c}\sqrt{T}}$, we obtain a regret upper bound of

$$\frac{C}{c}\sqrt{T \ln d}\,. \tag{12.2}$$

While the average regret is vanishing, we had to use the additional assumption on the market gains so the resulting algorithm is not universal, according to our definition above. In the next section, we will see that we can remove the constraint on the market gains and also achieve a much better dependency in T.

12.2 Universal Portfolio Selection with F-Weighted Portfolio Algorithms

Algorithm 12.1 F-Weighted Portfolio Selection

Require: $F : \Delta^{d-1} \to \mathbb{R}$ probability density function
1: $\text{Wealth}_0 = 1$
2: **for** $t = 1$ **to** T **do**
3: Set $\boldsymbol{x}_t = \frac{\int_{\Delta^{d-1}} \boldsymbol{x}\, \text{Wealth}_{t-1}(\boldsymbol{x})\, \mathrm{d}F(\boldsymbol{x})}{\int_{\Delta^{d-1}} \text{Wealth}_{t-1}(\boldsymbol{x})\, \mathrm{d}F(\boldsymbol{x})}$
4: Receive $\boldsymbol{w}_t \in \mathbb{R}^d_{\geq 0}$
5: $\text{Wealth}_t = \text{Wealth}_{t-1}\langle \boldsymbol{w}_t, \boldsymbol{x}_t\rangle$
6: **end for**

To solve the problem of the unbounded gradients, we will consider a different strategy. In particular, we will use the **F-weighted portfolio algorithms** [Cover, 1991] (Algorithm 12.1 that predict at each step with

$$\boldsymbol{x}_t = \frac{\int_{\Delta^{d-1}} \boldsymbol{x}\, \text{Wealth}_{t-1}(\boldsymbol{x})\, \mathrm{d}F(\boldsymbol{x})}{\int_{\Delta^{d-1}} \text{Wealth}_{t-1}(\boldsymbol{x})\, \mathrm{d}F(\boldsymbol{x})},$$

where $F : \Delta^{d-1} \to \mathbb{R}$ is a probability density function. To understand this formula, consider a generic constant rebalanced portfolio allocation $\boldsymbol{x} \in \Delta^{d-1}$ whose wealth after $t-1$ rounds is $\text{Wealth}_{t-1}(\boldsymbol{x})$. Now, consider a probability distribution of all the possible constant rebalanced portfolios proportional to $\text{Wealth}_{t-1}(\boldsymbol{x})F(\boldsymbol{x})$, where $F(\boldsymbol{x})$ is a probability distribution. We have that $\boldsymbol{x}_t$ defined above is nothing else than the average portfolio according to this distribution. Hence, we give more importance in the average to successful portfolios that are also weighted highly by the distribution F.

The above strategy also gives us a closed form expression for the wealth of the algorithm:

$$\mathrm{Wealth}_t = \int_{\Delta^{d-1}} \mathrm{Wealth}_t(\boldsymbol{x})\,\mathrm{d}F(\boldsymbol{x}),$$

that is, the wealth of the F-weighted portfolio is the average wealth of constant rebalanced portfolios according to the distribution F. Let's prove this expression by induction. At $t = 1$ the claim is true, so let's assume it is true at $t-1$ and let's prove it at t. We have

$$\begin{aligned}\mathrm{Wealth}_t = \mathrm{Wealth}_{t-1}\langle \boldsymbol{w}_t, \boldsymbol{x}_t\rangle &= \int_{\Delta^{d-1}} \mathrm{Wealth}_{t-1}(\boldsymbol{x})\,\mathrm{d}F(\boldsymbol{x})\frac{\int_{\Delta^{d-1}}\langle \boldsymbol{w}_t, \boldsymbol{x}\rangle\,\mathrm{Wealth}_{t-1}(\boldsymbol{x})\,\mathrm{d}F(\boldsymbol{x})}{\int_{\Delta^{d-1}}\mathrm{Wealth}_{t-1}(\boldsymbol{x})\,\mathrm{d}F(\boldsymbol{x})}\\ &= \int_{\Delta^{d-1}} \mathrm{Wealth}_t(\boldsymbol{x})\,\mathrm{d}F(\boldsymbol{x})\,. \end{aligned} \tag{12.3}$$

Cover and Ordentlich [1996] proved the universal property using two possible settings of F: the Dirichlet$(1/2,\dots,1/2)$ distribution and the uniform distribution. For these two distribution they proved the following regret upper bounds.

Theorem 12.1. *Consider the F-weighted portfolio strategy and an arbitrary sequence of market gains $\boldsymbol{w}_t \in \mathbb{R}^d_{\geq 0}$ for $t = 1,\dots,T$.*

- *If F is equal to the Dirichlet$(1/2,\dots,1/2)$ distribution, then we have*

$$\mathrm{Regret}_T = \ln \mathrm{Wealth}_T(\boldsymbol{u}) - \ln \mathrm{Wealth}_T \leq \ln \frac{\sqrt{\pi}\Gamma\left(T+\frac{d}{2}\right)}{\Gamma\left(\frac{d}{2}\right)\Gamma\left(T+\frac{1}{2}\right)} \leq \frac{d-1}{2}\ln(2(T+1)),$$

  where $\Gamma(x) = \int_0^{+\infty} t^{x-1}\exp(-t)\,\mathrm{d}t$ is the gamma function.

- *If F is equal to the uniform distribution, then we have*

$$\mathrm{Regret}_T = \ln \mathrm{Wealth}_T(\boldsymbol{u}) - \ln \mathrm{Wealth}_T \leq \ln \binom{T+d-1}{d-1} \leq (d-1)\left(\ln\left(\frac{T}{d-1}+1\right)+1\right)\,.$$

Observe that both prediction strategies assure that we never produce a vector where the loss is infinite and it allows us to show a vanishing average regret. Moreover, it can be shown that this first regret upper bound is optimal up to constant additive terms. Also, the proof shows that regret upper bounds are essentially tight in the case that in each round the market gains are vectors with all zeros but one coordinate with a '1'.

To prove this theorem, we will need some tools from information theory.

12.3 Information Theory Bits

We will need some basic result from information theory and in particular upper bounds used in the *method of types*.

Definition 12.2 (Type of a sequence of symbols)**.** *Consider a sequence of T symbols $\boldsymbol{x} = (x_1,\dots,x_T)$, where each $x_t \in \{1,\dots,d\}$. The **type** of the sequence is the vector $\boldsymbol{n}$ of the fractions of times each symbol appears, i.e., $\boldsymbol{n} = [N(1;\boldsymbol{x})/T,\dots,N(d;\boldsymbol{x})/T]$, where $N(j;\boldsymbol{x}) := \sum_{t=1}^T \mathbf{1}[x_t = j]$.*

First, we prove the following lemma.

Lemma 12.3. *Let $T \geq 1$, $\boldsymbol{x} \in \mathbb{R}^d_{\geq 0}$ such that $\sum_{i=1}^d x_i = T$, $\boldsymbol{u} \in \mathbb{R}^d_{\geq 0}$, and $\sum_{i=1}^d u_i = 1$. Then, we have*

$$\prod_{i=1}^d u_i^{x_i} \leq \prod_{i=1}^d \left(\frac{x_i}{T}\right)^{x_i}\,.$$

Proof. If there exists i such that $u_i^{x_i} = 0$, the inequality is verified. So, in the following we assume that $u_i^{x_i} \neq 0$ for $i = 1, \dots, d$. So, if for a given i we have $u_i = 0$ we must have $x_i = 0$ as well. Hence, we have

$$\prod_{i=1}^{d} u_i^{x_i} = \prod_{i:u_i \neq 0} u_i^{x_i} = \prod_{i:u_i \neq 0} e^{x_i \ln u_i} = \prod_{i:u_i \neq 0} e^{T\left(\frac{x_i}{T} \ln \frac{x_i}{T} + \frac{x_i}{T} \ln \frac{u_i T}{x_i}\right)} = \prod_{i:u_i \neq 0} e^{T(-H(\boldsymbol{x}/T) - D(\boldsymbol{x}/T; \boldsymbol{u}))}$$
$$\leq \prod_{i:u_i \neq 0} e^{-TH(\boldsymbol{x}/T)} = \prod_{i=1}^{d} \left(\frac{x_i}{T}\right)^{x_i},$$

where $0 \ln 0 := 0$, $H(\boldsymbol{x}) := -\sum_{i=1}^{d} x_i \ln x_i$ is the entropy, and $D(\boldsymbol{x}; \boldsymbol{y}) := \sum_{i=1}^{d} x_i \ln \frac{x_i}{y_i}$ is the KL divergence, and the inequality is due to the fact that the KL divergence is non-negative. □

Theorem 12.4. *Consider an alphabet of d symbols and consider the set of all the sequences of length T symbols. Let the set of all possible types of these sequence by $Q \subset \Delta^{d-1}$. Denote by $\mathrm{Type}_T(\boldsymbol{n})$ the set of all the sequences of length T symbols with a type $\boldsymbol{n}$, that is $\mathrm{Type}_T(\boldsymbol{n}) := \{\boldsymbol{x} \in \{1, \dots, d\}^T : \boldsymbol{x} \text{ has type } \boldsymbol{n}\}$. Then, for any $\boldsymbol{u} \in Q$, we have $|\mathrm{Type}_T(\boldsymbol{u})| \leq \prod_{t=1}^{d} u_i^{-u_i T}$.*

Proof. We prove the inequality with a probabilistic argument. Consider all the possible sequences $\boldsymbol{y}$ of length T generated i.i.d. from a distribution over d symbols equal to $\boldsymbol{u} = [u_1, \dots, u_d]^\top \in \Delta^{d-1}$. The sum of the probabilities of all these sequences is 1. Among all the sequences, there are the ones with type $\boldsymbol{u}$. Hence, for any $\boldsymbol{u} \in Q$, we have

$$1 \geq \sum_{\boldsymbol{y} \text{ has type } \boldsymbol{u}} \mathbb{P}\{N(1; \boldsymbol{y}) = u_1, \dots, N(d; \boldsymbol{y}) = u_d\} = \sum_{\boldsymbol{y} \text{ has type } \boldsymbol{u}} \prod_{i=1}^{d} u_i^{u_i T} = |\mathrm{Type}_T(\boldsymbol{u})| \prod_{i=1}^{d} u_i^{u_i T},$$

where the first equality is due the i.i.d. assumption and the second one is due to the fact that all the sequences of type $\boldsymbol{u}$ have the same probability. □

12.4 Proof of Theorem 12.1

First of all, we need a couple of technical lemmas.

Lemma 12.5. *Let $a_1, \dots, a_T > 0$ and $b_1, \dots, b_T > 0$. Then, we have*

$$\frac{\sum_{t=1}^{T} a_t}{\sum_{t=1}^{T} b_t} \leq \max_{t=1,\dots,T} \frac{a_t}{b_t}.$$

Proof. Let $j^\star = \mathrm{argmax}_{t=1,\dots,T} \frac{a_t}{b_t}$. Then, we have

$$\frac{\sum_{t=1}^{T} a_t}{\sum_{t=1}^{T} b_t} = \frac{a_{j^\star}\left(1 + \sum_{t \neq j^\star} \frac{a_t}{a_{j^\star}}\right)}{b_{j^\star}\left(1 + \sum_{t \neq j^\star} \frac{b_t}{b_{j^\star}}\right)} \leq \frac{a_{j^\star}}{b_{j^\star}},$$

because $\frac{a_{j^\star}}{b_{j^\star}} \geq \frac{a_t}{b_t}$ for all t, that implies $\frac{a_t}{a_{j^\star}} \leq \frac{b_t}{b_{j^\star}}$. □

Lemma 12.6. *Let $T \geq 1$ and $a_1, \dots, a_T \in \{0, 1, \dots, T\}$ such that $\sum_{t=1}^{T} a_t = T$. Then, the function $f(a_1, \dots, a_T) = \prod_{t=1}^{T} \frac{a_t^{a_t}}{\Gamma\left(a_t + \frac{1}{2}\right)}$ is maximized when $a_1 = T$ and $a_2 = a_3 = \dots = a_T = 0$.*

Proof. First, let's show that for $a_r \geq a_s$, we have $f(a_1, \dots, a_r, \dots, a_s, \dots, a_T)/f(a_1, \dots, a_r+1, \dots, a_s-1, \dots, a_T) \leq 1$. This ratio is equal to $\phi(a_r)/\phi(a_s - 1)$, where

$$\phi(x) := \frac{x^x}{\Gamma\left(x + \frac{1}{2}\right)} \frac{\Gamma\left(x + \frac{3}{2}\right)}{(x+1)^{x+1}} = \frac{x^x}{(x+1)^{x+1}} \left(x + \frac{1}{2}\right),$$

where we used $\Gamma(x+1) = x\Gamma(x)$. Given that for $a_r = a_s - 1$ we have that $\phi(a_r)/\phi(a_s - 1) = 1$, we have to show that the ϕ is decreasing. The derivative of $\ln \phi(x)$ is

$$(\ln \phi)'(x) = \ln(x) - \ln(x+1) + \frac{1}{x+\frac{1}{2}} = -\int_x^{x+1} \frac{1}{y}\,\mathrm{d}y + \frac{1}{x+\frac{1}{2}} \leq -\frac{1}{\int_x^{x+1} y\,\mathrm{d}y} + \frac{1}{x+\frac{1}{2}}$$
$$= -\frac{1}{x+\frac{1}{2}} + \frac{1}{x+\frac{1}{2}} = 0\,.$$

Repeatedly applying the fact that $f(a_1, \ldots, a_r, \ldots, a_s, \ldots, a_T)/f(a_1, \ldots, a_r + 1, \ldots, a_s - 1, \ldots, a_T) \leq 1$ to couple of variables, gives the stated result. □

The key lemma is the following one.

Lemma 12.7. *For any fixed rebalanced portfolio $\boldsymbol{u}$ and any sequence of market gains $\boldsymbol{w}_t \in \mathbb{R}^d_{\geq 0}$ for $t = 1, \ldots, T$, a F-weighted portfolio guarantees*

$$\frac{\mathrm{Wealth}_T(\boldsymbol{u})}{\mathrm{Wealth}_T} \leq \max_{\boldsymbol{j} \in \{1,\ldots,d\}^T} \frac{\prod_{t=1}^T u_{j_t}}{\int \prod_{t=1}^T x_{j_t} \mathrm{d}F(\boldsymbol{x})}\,.$$

Proof. Observe that

$$\mathrm{Wealth}_T(\boldsymbol{u}) = \prod_{t=1}^T \langle \boldsymbol{w}_t^i \boldsymbol{u} \rangle = \prod_{t=1}^T \sum_{i=1}^d w_{t,i} u_i = \sum_{\boldsymbol{j} \in \{1,\ldots,d\}^T} \prod_{t=1}^T u_{j_t} w_{t,j_t},$$

where in the last equality we have expressed the products as sum over all the possible combinations of the terms in the sum. In the same way, using (12.3), we have

$$\mathrm{Wealth}_T = \int_{\Delta^{d-1}} \mathrm{Wealth}_T(\boldsymbol{x})\,\mathrm{d}F(\boldsymbol{x}) = \int_{\Delta^{d-1}} \prod_{t=1}^T \langle \boldsymbol{w}_t, \boldsymbol{x} \rangle\,\mathrm{d}F(\boldsymbol{x}) = \sum_{\boldsymbol{j} \in \{1,\ldots,d\}^T} \int_{\Delta^{d-1}} \prod_{t=1}^T x_{j_t} w_{t,j_t}\,\mathrm{d}F(\boldsymbol{x})\,.$$

We now use Lemma 12.5, to obtain

$$\frac{\mathrm{Wealth}_T(\boldsymbol{u})}{\mathrm{Wealth}_T} = \frac{\sum_{\boldsymbol{j} \in \{1,\ldots,d\}^T} \prod_{t=1}^T u_{j_t} w_{t,j_t}}{\sum_{\boldsymbol{j} \in \{1,\ldots,d\}^T} \int_{\Delta^{d-1}} \prod_{t=1}^T x_{j_t} w_{t,j_t}\,\mathrm{d}F(\boldsymbol{x})} = \frac{\sum_{\boldsymbol{j} \in \{1,\ldots,d\}^T, \prod_{t=1}^T w_{t,j_t} > 0} \prod_{t=1}^T u_{j_t} w_{t,j_t}}{\sum_{\boldsymbol{j} \in \{1,\ldots,d\}^T, \prod_{t=1}^T w_{t,j_t} > 0} \int_{\Delta^{d-1}} \prod_{t=1}^T x_{j_t} w_{t,j_t}\,\mathrm{d}F(\boldsymbol{x})}$$
$$\leq \max_{\boldsymbol{j} \in \{1,\ldots,d\}^T, \prod_{t=1}^T w_{t,j_t} > 0} \frac{\prod_{t=1}^T u_{j_t} w_{t,j_t}}{\int_{\Delta^{d-1}} \prod_{t=1}^T x_{j_t} w_{t,j_t}\,\mathrm{d}F(\boldsymbol{x})}$$
$$= \max_{\boldsymbol{j} \in \{1,\ldots,d\}^T, \prod_{t=1}^T w_{t,j_t} > 0} \frac{\prod_{t=1}^T u_{j_t}}{\int_{\Delta^{d-1}} \prod_{t=1}^T x_{j_t}\,\mathrm{d}F(\boldsymbol{x})} \leq \max_{\boldsymbol{j} \in \{1,\ldots,d\}^T} \frac{\prod_{t=1}^T u_{j_t}}{\int_{\Delta^{d-1}} \prod_{t=1}^T x_{j_t}\,\mathrm{d}F(\boldsymbol{x})}\,.$$ □

This Lemma has a very easy and powerful interpretation: for a given $\boldsymbol{j} \in \{1, \ldots, d\}^T$, we have that $\prod_{t=1}^T u_{j_t}$ is nothing else that the wealth of the constant rebalanced portfolio $\boldsymbol{u}$ when the market gains for $t = 1, \ldots, T$ are

$$\boldsymbol{w}_t = [0, \ldots, 0, \underbrace{1}_{\text{position } j_t}, 0, \ldots, 0] \in \mathbb{R}^d\,.$$

Similarly for the wealth of the algorithm. This means that the Lemma shows that the worst case regret ratio is achieved for market gains where only one coordinate is 1 and all the others are zero. So, this lemma allows us to simplify the analysis by completely removing the market gains from the upper bound. As such, the upper bound is worst-case with respect to the sequence of market gains. Yet, given that with our choices of F distributions the final regret is logarithmic, it also shows that the market gains can only have a limited influence on the regret of the algorithm.

We can now prove the regret guarantees for universal portfolio with two different distributions F.

Proof of Theorem 12.1. Let's first consider the Dirichlet distribution. First, observe that it can be shown that $\int \prod_{t=1}^{T} x_{j_t} \mathrm{d}F(\boldsymbol{x})$ has a closed form expression for the specific distribution we have selected:

$$\int \prod_{t=1}^{T} x_{j_t} \mathrm{d}F(\boldsymbol{x}) = \frac{\Gamma\left(\frac{d}{2}\right)}{\Gamma\left(T + \frac{d}{2}\right)} \prod_{t=1}^{T} \frac{\Gamma\left(N(t; \boldsymbol{j}) + \frac{1}{2}\right)}{\sqrt{\pi}},$$

where $N(t; \boldsymbol{j})$ is $\sum_{i=1}^{T} \mathbf{1}[j_i = t]$. Moreover, using Lemma 12.3, we have

$$\prod_{t=1}^{T} u_{j_t} = \prod_{i=1}^{d} u_i^{N(i;\boldsymbol{j})} \leq \prod_{i=1}^{d} \left(\frac{N(i; \boldsymbol{j})}{T}\right)^{N(i;\boldsymbol{j})}.$$

Putting everything together and using Lemmas 12.7 and 12.6, we have

$$\begin{aligned}
\max_{\boldsymbol{j} \in \{1,\dots,d\}^T} \frac{\prod_{t=1}^{T} u_{j_t}}{\int_{\Delta^{d-1}} \prod_{t=1}^{T} x_{j_t} \, \mathrm{d}F(\boldsymbol{x})} &\leq \max_{a_1,\dots,a_T \geq 0 : \sum_{t=1}^{T} a_t = T} \pi^{T/2} \frac{\Gamma\left(T + \frac{d}{2}\right)}{\Gamma\left(\frac{d}{2}\right) T^T} \prod_{t=1}^{T} \frac{a_t^{a_t}}{\Gamma\left(a_t + \frac{1}{2}\right)} \\
&\leq \pi^{T/2} \frac{\Gamma\left(T + \frac{d}{2}\right)}{\Gamma\left(\frac{d}{2}\right) T^T} \frac{T^T}{\Gamma\left(T + \frac{1}{2}\right)} \frac{1}{\pi^{\frac{T-1}{2}}} = \frac{\sqrt{\pi} \Gamma\left(T + \frac{d}{2}\right)}{\Gamma\left(\frac{d}{2}\right) \Gamma\left(T + \frac{1}{2}\right)}.
\end{aligned}$$

Analogously, when F is the uniform distribution, it can be shown that

$$\int_{\Delta^{d-1}} \prod_{t=1}^{T} x_{j_t} \, \mathrm{d}F(\boldsymbol{x}) = \frac{1}{\binom{T+d-1}{d-1} |\operatorname{Type}_T(N(1; \boldsymbol{j})/T, \dots, N(d; \boldsymbol{j})/T)|},$$

where $|\operatorname{Type}_T(a_1, \dots, a_d)|$ is the number of sequences of type $(a_1, \dots, a_d)$. From Theorem 12.4, we have that $|\operatorname{Type}_T(N(1; \boldsymbol{j})/T, \dots, N(d; \boldsymbol{j})/T)| \leq \prod_{i=1}^{d} \left(\frac{N(i;\boldsymbol{j})}{T}\right)^{-N(i;\boldsymbol{j})}$. Hence, when F is the uniform distribution, we have

$$\max_{\boldsymbol{j} \in \{1,\dots,m\}^T} \frac{\prod_{t=1}^{T} u_{j_t}}{\int_{\Delta^{d-1}} \prod_{t=1}^{T} x_{j_t} \, \mathrm{d}F(\boldsymbol{x})} \leq \binom{T + d - 1}{d - 1}.$$

The second inequality is obtained using the inequality $\binom{n}{k} \leq \left(\frac{en}{k}\right)^k$. □

12.5 Portfolio Selection through Online-Newton-Step

While the F-weighted portfolio algorithm can have optimal regret, its computational complexity is very high, both in T and d. In fact, we have to calculate an integral over d dimensions in each step. While it is possible to have a closed form update with complexity $O(t)$ [Cover and Ordentlich, 1996], it is also natural to look for alternatives with a lower computational complexity.

One might be tempted to use any other OCO algorithm to solve the portfolio selection problem. However, as we have seen in Section 12.1, the gradients in general can be unbounded. Hence, we again consider an easier setting: We assume that the market gains are both lower bounded by $c > 0$ and upper bounded by $C < \infty$. We stress that Theorem 12.1 does not require this assumption, but we use it to derive a simpler algorithm, still with a logarithmic regret.

Now, we might be tempted to think that the losses are strongly convex, because Theorem 12.1 shows a logarithmic regret. This is clearly false: The Hessian of each function has rank 1 and unfortunately in the worst case the sum of T losses can also have a singular Hessian. Even assuming the Hessian of the sum of the losses is not singular, we would need its minimal eigenvalue to grow over time. Again, this does not happen in a worst-case scenario. Hence, we need another way.

Observe that the losses in portfolio selection are 1-exp-concave. In fact, $\exp(-\ell_t(\boldsymbol{x})) = \langle \boldsymbol{w}_t, \boldsymbol{x} \rangle$ that is concave. Hence, if the gradients and the domain are bounded, we can apply the Online-Newton-Step algorithm. In particular,

following Example 7.31, we need to find $\beta \leq \frac{1}{2}$ such that $|\beta\langle\nabla\ell_t(\boldsymbol{x}), \boldsymbol{x}-\boldsymbol{u}\rangle| \leq \frac{1}{2}$ for any $\boldsymbol{x}, \boldsymbol{u} \in \Delta^{d-1}$. For the gradient, we have $\nabla\ell_t(\boldsymbol{x}) = \frac{\boldsymbol{w}_t}{\langle \boldsymbol{w}_t, \boldsymbol{x}\rangle}$, hence $\|\nabla\ell_t(\boldsymbol{x})\|_\infty \leq \frac{C}{c}$. Moreover, $\|\boldsymbol{x}-\boldsymbol{u}\|_1 \leq 2$ for any $\boldsymbol{x}, \boldsymbol{u} \in \Delta^{d-1}$. Hence, we can set $\beta = \frac{c}{4C}$.

The resulting algorithm is the following one.

Algorithm 12.2 Online Newton Step for Portfolio Selection

Require: $\lambda > 0, C < \infty, c > 0$
1: Set $\beta = \frac{c}{4C}$
2: **for** $t = 1$ **to** T **do**
3: Set $\boldsymbol{x}_t = \operatorname{argmin}_{\boldsymbol{x}\in\Delta^{d-1}} \sum_{i=1}^{t-1}\langle \boldsymbol{g}_i, \boldsymbol{x}\rangle + \frac{\lambda}{2}\|\boldsymbol{x}\|_2^2 + \frac{\beta}{2}\sum_{i=1}^{t-1}(\langle \boldsymbol{g}_i, \boldsymbol{x}-\boldsymbol{x}_i\rangle)^2$
4: Receive $\ell_t(\boldsymbol{x}) = -\ln\langle \boldsymbol{w}_t, \boldsymbol{x}\rangle$ and pay $\ell_t(\boldsymbol{x}_t)$
5: Set $\boldsymbol{g}_t = -\frac{\boldsymbol{w}_t}{\langle \boldsymbol{w}_t, \boldsymbol{x}_t\rangle}$
6: **end for**

As shown in Exercise 7.11, the update can also be divided in two steps: first minimizing over the entire space and then a Bregman projection.

As we have seen in Section 7.10, the regret upper bound of the Online-Newton-Step for any $\boldsymbol{u} \in \Delta^{d-1}$ is

$$\text{Regret}_T = \ln \text{Wealth}_T(\boldsymbol{u}) - \ln \text{Wealth}_T = \sum_{t=1}^T \ln\langle \boldsymbol{w}_t, \boldsymbol{u}\rangle - \sum_{t=1}^T \ln\langle \boldsymbol{w}_t, \boldsymbol{x}_t\rangle \leq \frac{\lambda}{2}\|\boldsymbol{u}\|_2^2 + \frac{d}{2\beta}\ln\left(1 + \frac{\beta T L^2}{d\lambda}\right),$$

where $\|\nabla\ell_t(\boldsymbol{x}_t)\|_2 \leq L$. In our setting, this becomes

$$\text{Regret}_T = \ln \text{Wealth}_T(\boldsymbol{u}) - \ln \text{Wealth}_T = \sum_{t=1}^T \ln\langle \boldsymbol{w}_t, \boldsymbol{u}\rangle - \sum_{t=1}^T \ln\langle \boldsymbol{w}_t, \boldsymbol{x}_t\rangle \leq \frac{\lambda}{2} + \frac{2dC}{c}\ln\left(1 + \frac{CT}{4c\lambda}\right),$$

where we used the fact that $\|\nabla\ell_t(\boldsymbol{x}_t)\|_2 \leq \frac{\sqrt{d}C}{c}$.

12.6 Application: Portfolio Selection and Continuous Coin-Betting

Consider the coin-betting online problem. We said that it corresponds to $V = [-1, 1]$ and $\ell_t(x) = \ln(1 + c_t x)$, where the coin outcome $c_t \in \{-1, 1\}$. We also considered an extension of this problem, the *continuous coin* betting problem, where $c_t \in [-1, 1]$. Now, we show how to reduce the continuous coin-betting problem to portfolio selection.

The reduction is straightforward: We consider portfolio selection with 2 stocks, and we set the market gains to $w_{t,1} = 1 + c_t$ and $w_{t,2} = 1 - c_t$. Note that $w_{t,1}, w_{t,2} \geq 0$, so they are legal market gains. Define $[x_t, 1 - x_t]$ the play of a 2-stocks portfolio algorithm, where $0 \leq x_t \leq 1$. Then, taking

$$\beta_t = 2x_t - 1$$

as the signed betting fraction a continuous-coin-betting algorithm on c_t ensures that the gain in the coin betting problem coincides with the gain in the portfolio selection problem.

The proof is immediate. We have

$$w_{t,1}x_t + w_{t,2}(1 - x_t) = x_t + x_t c_t + (1 - c_t)(1 - x_t) = x_t + x_t c_t + 1 - x_t - c_t + c_t x_t = 1 + c_t\beta_t,$$

where β_t is the signed betting fraction equal to $2x_t - 1$. Given that $x_t \in [0, 1]$, the range of the betting fractions is in $[-1, 1]$ as we wanted.

This implies that we can use the above reduction to transform any portfolio algorithm with 2 stocks into an algorithm for continuous coin betting. In turn, this means that portfolio algorithms with 2 stocks can be used for online convex optimization, see Chapter 9. This does not make the continuous coin-betting algorithm superfluous because the update rule of universal portfolio algorithm even with 2 stocks is still linear in the iteration number.

12.7 Application: From Portfolio Regret to Time-Uniform Concentration Inequalities

In this section, we will show another application of portfolio algorithms: The regret guarantee of portfolio algorithms can be used to derive time-uniform concentration inequalities.

Consider a sequence of random variables $Z_1, Z_2, \dots$ and assume that $E[Z_t|Z_1, \dots, Z_{t-1}] = \mu$ for all t. Also, assume that Z_t has values on $[0, 1]$. We want to estimate μ, giving confidence intervals $[l_t, u_t]$ that are valid for any t with probability at least $1 - \delta$, that is, $\mathbb{P}\{\forall t, \mu \notin [l_t, u_t]\} \le \delta$. The classic approach is to estimate μ with $\hat{\mu}_t = \frac{1}{t}\sum_{i=1}^{t} Z_i$, then invoke a concentration inequality that holds uniformly over time to obtain the confidence intervals. However, for most of the well-known concentration inequalities, we would have *vacuous* confidence intervals when t is small, in the sense that $u_t - l_t > 1$ for all t smaller than some constant.

Here, we will show how a portfolio algorithm immediately gives rise to non-vacuous time-uniform confidence intervals, even with a single sample!

The idea is the following: We will construct a continuous coin-betting game on a fair coin from the problem of estimating the unknown mean. Then, we will use the following theorem that says that the probability to make a large amount of money at any moment in time is small.

Theorem 12.8 (Ville's inequality)**.** *Let $Y_0, Y_1, \dots$, a non-negative supermartingale and $\delta \in (0, 1]$. Then, we have*

$$\mathbb{P}\left\{\max_t Y_t \ge \frac{1}{\delta}\right\} \le \mathbb{E}[Y_0]\delta \ .$$

Warm-up: From KT to a concentration inequality As a warm-up example, consider the KT algorithm that bets $\beta_t \operatorname{Wealth}_{t-1}$ in round t on the outcome of the continuous coin $c_t = Z_t - \mu$. We have that the wealth of the algorithm is a martingale and hence a supermartingale. Indeed, we have

$$\begin{aligned}\mathbb{E}[\operatorname{Wealth}_t \,|\, \operatorname{Wealth}_{t-1}] &= \operatorname{Wealth}_{t-1}\, \mathbb{E}[1 + \beta_t c_t |\, \operatorname{Wealth}_{t-1}] \\ &= \operatorname{Wealth}_{t-1}\, \mathbb{E}[1 + \beta_t (Z_t - \mu) |\, \operatorname{Wealth}_{t-1}] = \operatorname{Wealth}_{t-1} \ .\end{aligned}$$

Moreover, the wealth is non-negative because KT guarantees a non-negative wealth on any sequence of coins.

So, starting with \$1, using Ville's inequality and Theorem 9.5, we obtain

$$\delta \ge \mathbb{P}\left\{\max_t \operatorname{Wealth}_t \ge \frac{1}{\delta}\right\} = \mathbb{P}\left\{\max_t \ln \operatorname{Wealth}_t \ge \ln \frac{1}{\delta}\right\} \ge \mathbb{P}\left\{\max_t \frac{\left(\sum_{i=1}^{t} c_t\right)^2}{4t} - \frac{1}{2}\ln(Kt) \ge \ln \frac{1}{\delta}\right\},$$

where K is a universal constant. An equivalent statement is to say that with probability at least $1-\delta$ we have uniformly over t that

$$\left|\frac{1}{t}\sum_{i=1}^{t} Z_i - \mu\right| = \frac{1}{t}\left|\sum_{i=1}^{t} c_i\right| \le \frac{2\sqrt{\ln \frac{\sqrt{Kt}}{\delta}}}{t} \ .$$

Observe that $Y_t = \sum_{i=1}^{t}(Z_i - \mu)$ is a martingale, so let's compare this concentration to Hoeffding-Azuma inequality in Theorem 3.13. We almost get the same thing, but here we have an additional $\ln \sqrt{T}$ term. However, this concentration is uniform over time, that justifies the additional term in the logarithm. To summarize, the regret/wealth guarantee of KT implies a concentration inequality.

From Universal Portfolio to a concentration inequality Let's now improve this reasoning using a universal portfolio algorithm to bet on the same outcomes. Similarly to Section 12.6, we will transform the random variables Z_t to a sequence of two market gains. Set $w_{t,1} = 1 + \frac{Z_t - \mu}{\mu}$ and $w_{t,2} = 1 - \frac{Z_t - \mu}{1-\mu}$. Given that $Z_t - \mu \in [-\mu, 1 - \mu]$, we have that $w_{t,1}$ and $w_{t,2}$ are non-negative. Moreover, as before we still have that the wealth will be a martingale. So, we have

$$\max_{\boldsymbol{u} \in \Delta} \sum_{i=1}^{t} \ln \langle \boldsymbol{w}_i, \boldsymbol{u}\rangle = \max_{\beta \in [-\frac{1}{1-\mu}, \frac{1}{\mu}]} \sum_{i=1}^{t} \ln\left(1 + (Z_i - \mu)\beta\right) \ .$$

Reasoning as above, we have

$$\mathbb{P}\left\{\max_t \max_{\beta\in[-\frac{1}{1-\mu},\frac{1}{\mu}]} \sum_{i=1}^{t} \ln\left(1+(Z_i-\mu)\beta\right) - \mathrm{Regret}_t \geq \ln\frac{1}{\delta}\right\} \leq \delta,$$

where Regret_t is the regret of the portfolio algorithm with 2 stocks. If we use a portfolio algorithm with logarithmic regret, given that $[-1,1] \subset [-\frac{1}{1-\mu}, \frac{1}{\mu}]$, this expression gives always a bigger wealth than the KT strategy we have just seen. In turn, a bigger wealth corresponds to a tighter concentration.

Let's be more precise instantiating the above idea with a F-weighted portfolio algorithm, with F equal to the Beta(1/2,1/2) distribution.

Theorem 12.9. *Let $\delta \in (0,1)$. Assume $Z_1, Z_2, \ldots$ a sequence of random variables such that for each i we have $0 \leq Z_i \leq 1$ and $\mathbb{E}[Z_i|Z_1,\ldots,Z_{i-1}] = \mu$ almost surely. Let $G_t(\beta,\mu) := \sum_{i=1}^{t} \ln\left(1+\beta(Z_i-\mu)\right)$, $R_t := \ln\frac{\sqrt{\pi}\Gamma(t+1)}{\Gamma(t+\frac{1}{2})}$, and*

$$S_t := \left\{m \in [0,1] : \max_{\beta\in[-\frac{1}{1-m},\frac{1}{m}]} G_t(\beta,m) - R_t \leq \ln\frac{1}{\delta}\right\}.$$

Then, with probability at least $1-\delta$ and uniformly over t, after observing t random variables we have $\mu \in \cap_{i=1}^{t} S_i$.

Moreover, we have that S_t is an interval $[l_t, u_t] \subseteq [0,1]$ for all $t = 1,\ldots,T$.

Proof. Following the reasoning above and the fact that the regret of the F-weighted portfolio algorithm with 2 stocks and $F = Beta(1/2,1/2)$ is upper bounded by R_t using Theorem 12.1, we get that $\mu \in S_t$. From the fact that this holds with probability $1-\delta$ uniformly over time, we get that at time t μ must be in the intersection of the sets $S_1,\ldots,S_t$.

For the second claim, first denote by $\beta^\star(m) := \mathrm{argmax}_{\beta\in[-\frac{1}{1-m},\frac{1}{m}]}\, G_t(\beta,m)$ and $\hat{G}_t(m) = \max_{\beta\in[-\frac{1}{1-\mu},\frac{1}{\mu}]} G_t(\beta,m)$.

Observe that the derivative of G_t with respect to its first argument is

$$G_t'(\beta,m) = \sum_{i=1}^{t} \frac{Z_i - m}{1+\beta(Z_i-m)}.$$

So, we have that $G_t'(0,\hat{\mu}_t) = 0$. Given that $G_t(\beta,m)$ is concave in β, then $G_t(\beta,\hat{\mu}_t)$ has a maximum with respect to the first argument in $\beta = 0$ and the value of the function is 0.

For $m' > \hat{\mu}_t$, $G_t'(0,m') < 0$. Since $G_t(\beta,m')$ is concave in β, we have $\beta^\star(m') < 0$. In the same way, for $m' < \hat{\mu}_t$ we have $\beta^\star(m') > 0$.

Now, let us start with $m' > \hat{\mu}_t$ and prove that $\hat{G}_t(m)$ is nondecreasing, the other side is analogous. Consider $m_1 > m_2 > \hat{\mu}_t$. Given that $\beta^\star(m_2) < 0$, we have

$$\hat{G}_t(m_2) = G_t(\beta^\star(m_2), m_2) \leq G_t(\beta^\star(m_2), m_1) \leq G_t(\beta^\star(m_1), m_1) = \hat{G}_t(m_1),$$

where the first inequality is due to the fact that $G_t(\beta,m)$ is nondecreasing in m when $\beta < 0$ and the second inequality is due to the fact that the negative part of the interval $[-\frac{1}{1-m_1}, \frac{1}{m_1}]$ contains the negative part of the interval $[-\frac{1}{1-m_2}, \frac{1}{m_2}]$ and we know the maximum β is negative. Hence, $\hat{G}_t(m)$ is a quasi-convex function of m and hence S_t is an interval. □

Remark 12.10. *Given that S_t is an interval, we can find $S_t = [l_t, u_t]$ efficiently using the bisection algorithm.*

Now, we gather some more intuition on the inequality of Theorem 12.9.

It may not seem obvious if the maximum log wealth is a better candidate for constructing a confidence sequence than the standard ones like Bernoulli KL-divergence based bound [e.g., Garivier and Cappé, 2011, Theorem 10], which works for random variables supported in $[0,1]$:

$$\mathbb{P}\left\{\max_t\ t \cdot KL(\hat{\mu}_t, \mu) - \ln f(t) \geq \ln\frac{1}{\delta}\right\} \leq \delta,$$

where $\hat{\mu}_t = \frac{1}{t}\sum_{i=1}^t Z_i$, $KL(p,q) := p\ln\frac{p}{q} + (1-p)\ln\frac{1-p}{1-q}$, and $f(t)$ grows polynomially in t or slower. So, in the following proposition we show that the maximum log wealth is never worse than the KL divergence, which supports a viewpoint that the KL divergence is a special case of the maximum log wealth and that confidence bounds constructed with the maximum wealth are never worse than those with KL divergence, ignoring the minor difference in $\ln f(t)$.

Proposition 12.11. *Let $Z_1, \dots, Z_t \in [0,1]$, $\hat{\mu}_t = \frac{1}{t}\sum_{i=1}^t Z_i$, and $\mu \in [0,1]$. Then,*

$$\max_{\beta \in [-\frac{1}{1-\mu}, \frac{1}{\mu}]} \sum_{i=1}^t \ln\left(1 + \beta(Z_i - \mu)\right) \geq t \cdot KL\left(\hat{\mu}_t, \mu\right),$$

where we achieve the equality if $X_1, \dots, X_t \in \{0,1\}$ almost surely.

Proof. Using Jensen's inequality, we have for any $Z \in [0,1]$ that

$$\begin{aligned}\ln(1+\beta(Z-\mu)) &= \ln[Z(1+\beta(1-\mu)) + (1-Z)(1+\beta(0-\mu))] \\ &\geq Z\ln(1+\beta(1-\mu)) + (1-Z)\ln(1+\beta(0-\mu)),\end{aligned}$$

Note that we achieve equality when $Z = 1$ or $Z = 0$. Then, we have

$$\begin{aligned}\max_{\beta \in [-\frac{1}{1-\mu}, \frac{1}{\mu}]} \sum_{i=1}^t \ln(1+\beta(Z_i-\mu)) &\geq \max_{\beta \in [-\frac{1}{1-\mu}, \frac{1}{\mu}]} \sum_i Z_i \ln(1+\beta(1-\mu)) + (1-Z_i)\ln(1-\beta\mu) \\ &= \max_{\beta \in [-\frac{1}{1-\mu}, \frac{1}{\mu}]} t[\hat{\mu}_t \ln(1+\beta(1-\mu)) + (1-\hat{\mu}_t)\ln(1-\beta\mu)].\end{aligned}$$

As the right hand side is concave in β, it remains to maximize the right hand side over β. The solution is $\beta = \frac{\hat{\mu}_t - \mu}{\mu(1-\mu)}$ with which the maximum becomes $t \cdot KL(\hat{\mu}_t, \mu)$. □

Second, we show that the confidence intervals obtained by the numerical inversion of Theorem 12.9 are *never vacuous*.

Theorem 12.12. *Under the assumptions of Theorem 12.9, for any t we have that $u_t - \ell_t \leq u_1 - \ell_1 = 1 - \frac{\delta}{2}$.*

Proof. First of all, it should be clear that the width of the confidence intervals $u_t - \ell_t$ are always smaller than the one calculated with 1 sample, $u_1 - \ell_1$. With only one sample, the upper and lower bound have a closed formula. Indeed, the argmax of $G_t(\beta, m)$ with respect to β over $[-\frac{1}{1-m}, \frac{1}{m}]$ is achieved in $\beta = \frac{1}{m}$ if $Z_1 - m > 0$ and in $\beta = -\frac{1}{1-m}$ for $Z_1 - m < 0$. This implies that

$$\ell_1 = \frac{Z_1}{\exp(R_1 + \ln\frac{1}{\delta})} \qquad \text{and} \qquad u_1 = 1 - \frac{1-Z_1}{\exp(R_1 + \ln\frac{1}{\delta})}.$$

Given that $R_1 = \ln 2$, subtracting the upper bound from the lower bound we get the stated bound for any Z_1. □

Finally, besides implying the KL bound above, the implicit concentration in Theorem 12.9 also implies an empirical Bernstein time-uniform concentration. It is worth stressing that the numerically evaluated interval $[l_t, u_t]$ is strictly smaller than the upper bound in the following theorem, given that Theorem 12.12 tells us that the width of numerically calculated intervals are always strictly smaller than 1.

Theorem 12.13. *Under the assumptions of Theorem 12.9, denote by $\hat{\mu}_i = \frac{1}{i}\sum_{j=1}^i Z_j$, $V_i = \sum_{j=1}^i (Z_j - \hat{\mu}_i)^2$, $\hat{R}_i = \ln\frac{\sqrt{\pi}\Gamma(i+1)}{\delta\Gamma(i+\frac{1}{2})}$, and*

$$\epsilon_i = \frac{(4/3)i\hat{R}_i + \sqrt{16/9 i^2\hat{R}_i^2 + 8V_i\hat{R}_i(i^2 - 2i\hat{R}_i)}}{2i^2 - 4i\hat{R}_i}.$$

Then, with probability at least $1-\delta$ uniformly for all t such that $t > 2\hat{R}_t$, we have

$$\max_{i=1,\dots,t} \hat{\mu}_i - \epsilon_i \leq \mu \leq \min_{i=1,\dots,t} \hat{\mu}_i + \epsilon_i.$$

For t big enough the deviation is roughly $\frac{(4/3)\ln(\sqrt{t}/\delta)}{t} + \frac{\sqrt{2V_t \ln(\sqrt{t}/\delta)}}{t}$, similarly to the inequalities in Audibert et al. [2009], Maurer and Pontil [2009].

Remark 12.14. *One can easily see that, in terms of the scaling with $\ln(1/\delta)$, the factor $\sqrt{2(\frac{1}{t}V_t)/t}$ is the optimal one due to the central limit theorem. For the scaling with t, by changing the weight distribution F it is also possible to obtain $\sqrt{\frac{2(\frac{1}{t}V_t)\ln(\ln(V_t))+o(\ln\ln t)}{t}}$ as $t \to \infty$, which matches the law of the iterated logarithm (thus asymptotically optimal), see Orabona and Jun [2021].*

To prove this theorem, we first need a technical lemma.

Lemma 12.15. *Let $f(x) = ax + b(\ln(1-|x|)+|x|)$, where $a \in \mathbb{R}$ and $b \geq 0$. Then, $\operatorname{argmax}_{x\in[-1,1]} f(x) = \frac{a}{|a|+b}$ and $\max_{x\in[-1,1]} f(x) = b\psi(\frac{a}{b}) \geq \frac{a^2}{(4/3)|a|+2b}$, where $\psi(x) = |x| - \ln(|x|+1)$.*

Proof. If $b = 0$, we have that argmax is $\operatorname{sign}(a)$. If $a = 0$, the argmax is 0. Hence, in the following, we can assume a and b to be different than 0.

We can rewrite the maximization problem as

$$\operatorname*{argmax}_x f(x) = b \operatorname*{argmax}_x \frac{a}{b}x + \ln(1-|x|) + |x| \,.$$

From the optimality condition, we have that $\frac{a}{b} - \frac{\operatorname{sign}(x^\star)}{1-|x^\star|} + \operatorname{sign}(x^\star) = 0$, that implies $x^\star = \frac{a}{|a|+b}$. Substituting this expression in f, we obtain the stated expression. The inequality is obtained by the elementary inequality $\ln(1+x) \leq x \cdot \frac{6+x}{6+4x}$ for $x \geq 0$. □

We can now prove Theorem 12.13.

Proof of Theorem 12.13. For a given t, set ϵ_t equal to $\mu - \hat{\mu}_t$, so that $\epsilon + \hat{\mu}_t \in [0,1]$.

Consider the function $f(a) = \frac{\ln(1+a)-a}{a^2/2}$ for $a > -1$. Set $|x| \leq 1$ and $|\beta| < 1$, so we have $\beta x \geq -|\beta| > -1$. From the sign of the first derivative, we have that $f(a)$ is increasing. Hence, we have

$$\ln(1+\beta x) = \beta x + \frac{1}{2}(\beta x)^2 f(\beta x) \geq \beta x + \frac{1}{2}(\beta x)^2 f(-|\beta|) = \beta x + x^2(|\beta| + \ln(1-|\beta|)) \,.$$

Hence, for any $\beta \in (-1,1)$, we have

$$\begin{aligned}
\sum_{i=1}^t \ln(1+\beta(Z_i-\mu)) &= \sum_{i=1}^t \ln(1+\beta(Z_i-\hat{\mu}_t-\epsilon_t)) \\
&\geq \beta\sum_{i=1}^t (Z_i-\hat{\mu}_t-\epsilon_t) + (\ln(1-|\beta|)+|\beta|)\left(\sum_{i=1}^t (Z_i-\hat{\mu}_t)^2 + \epsilon_t^2 t - 2\epsilon_t\sum_{i=1}^t (Z_i-\hat{\mu}_t)\right) \\
&= -\epsilon_t\beta t + (\ln(1-|\beta|)+|\beta|)\left(\sum_{i=1}^t (Z_i-\hat{\mu}_t)^2 + \epsilon_t^2 t\right) \,.
\end{aligned}$$

Hence, we have

$$\max_{\beta\in[-1,1]} \sum_{i=1}^t \ln(1+\beta(Z_i-\hat{\mu}_t-\epsilon_t)) = \left(\sum_{i=1}^t (Z_i-\hat{\mu}_t)^2 + \epsilon_t^2 t\right)\psi\left(\frac{|\epsilon_t| t}{\sum_{i=1}^t (Z_i-\hat{\mu}_t)^2 + \epsilon_t^2 t}\right),$$

where $\psi(x) = |x| - \ln(|x|+1)$ and the equality is due to Lemma 12.15. From the inequality in Lemma 12.15 we also obtain

$$\max_{\beta\in[-1,1]} \sum_{i=1}^t \ln(1+\beta(Z_i-\hat{\mu}_t-\epsilon_t)) \geq \frac{\epsilon_t^2 t^2}{(4/3)|\epsilon_t| t + 2\sum_{i=1}^t (Z_i-\hat{\mu}_t)^2 + 2\epsilon_t^2 t} \,.$$

Now, note that for any $\mu \in [0,1]$ the interval $[-\frac{1}{1-\mu}, \frac{1}{\mu}]$ is contained in $[-1,1]$. Hence, from Theorem 12.9, uniformly on all t with probability at least $1-\delta$, we have

$$\frac{\epsilon_t^2 t^2}{(4/3)|\epsilon_t|t + 2\sum_{i=1}^t (Z_i - \hat{\mu}_t)^2 + 2\epsilon_t^2 t} \leq R_t + \ln \frac{1}{\delta} = \hat{R}_t \,.$$

Assuming ϵ_t positive and solving for it, we have the stated upper bound. By the symmetry of the formula, the expression for negative ϵ_t has the opposite sign. □

12.8 History Bits

The distributional approach to betting and gambling was pioneered by Kelly [1956]. This approach assumes that the market gains are i.i.d. from a (known) distribution. Cover and Ordentlich [1996] proposed the first portfolio algorithm with a minimax regret, without assumptions over the market gains, improving over the result in Cover [1991]. The proof presented here follows the one in Cover and Ordentlich [1996], while Lemma 12.3 and Theorem 12.4 are adapted from the proofs of Cover and Thomas [2006, Theorem 11.1.2 and Theorem 11.1.3], respectively. Vovk and Watkins [1998] proved that F-weighted portfolio algorithms are instantiations of the Aggregating Algorithm [Vovk, 1990].

The use of EG for portfolio selection was proposed by Helmbold et al. [1998]. Stochastic approximations to the F-weighted portfolio update were proposed by Blum and Kalai [1997, 1999], Kalai and Vempala [2002]. Kalai and Vempala [2002] also presents a simpler proof for the uniform distribution case. The use of the Online Netwon Step for portfolio selection was proposed by Hazan et al. [2006, 2007]. There is also a line of research that aims at designing algorithms on the efficiency-regret Pareto frontier [see, e.g., Tsai et al., 2023, and references therein].

The reduction in Section 12.6 is folklore, and it appears in Orabona and Jun [2021].

The results and proofs from Section 12.6 are taken from Orabona and Jun [2021]. Note that the computational complexity to calculate S_t in Theorem 12.9 is $O(t^2)$. Looser confidence intervals from portfolio algorithms but with complexity $O(t)$ have been proposed in Orabona and Jun [2021] and Ryu and Bhatt [2022].

Ville's inequality is proved in Ville [1939, Page 84], where its meaning is exactly associated to betting. In fact, Ville [1939] also introduced the concept of martingale as the wealth of a betting strategy on a fair coin. The ideas of Ville were later used to design ideal tests for randomness of infinite sequences [Schnorr, 1971, Levin, 1976, Gács, 2005], "ideal" because none of these tests is computable.

There are two papers that at the same time and independently explicitly link hypothesis testing on a finite sequence of outcomes to betting. One is Cover [1974, Example 3], where an optimal betting strategy is defined in terms of the null hypothesis. The other one is Robbins and Siegmund [1974] that constructs confidence sequences from novel betting schemes and explicitly recognizes the connection between the sequential probability ratio test [Wald, 1945]. Very surprisingly, Robbins and Siegmund [1974, Section 9] seem to have proposed and analyzed the famous strategy of Krichevsky and Trofimov [1981] 7 years before them. However, while in the information theory literature these ideas flourished and gave birth to results on coding, compression, minimum description length, and gambling, they seem to disappear from the statistics community for 30 years. In this view, it is remarkable that Cover [1974] was submitted to Annals of Statistics and probably rejected, as it can be inferred from the footnote on its first page.

In fact, in the statistics literature gambling strategies reappear again only in the '90-'00s thanks to the book and papers by Shafer and Vovk. In particular, Vovk [1993], Shafer and Vovk [2001] aimed to found probabilities on a game-theoretic ground through betting schemes. However, the foundational approach in Shafer and Vovk [2001] also means that all the betting strategies they propose do not have closed form expressions and they cannot be easily implemented. Moreover, Shafer and Vovk [2001] does not contain any explicit concentration inequality, while the first concentration for game-theoretic probability derived by a betting scheme is in Vovk [2007], that derives a game-theoretic Hoeffding's inequality.

The first paper to consider an implementable strategy for testing through betting is by Hendriks [2018]. Directly building on Shafer and Vovk [2001] and Shafer et al. [2011], Hendriks [2018] proposes to construct testing martingales and confidence sequences for bounded random variables as uniform mixtures of constant betting strategies, effectively a F-wighted portfolio with the uniform distribution. Hendriks [2018] also showed empirically the good performance of the proposed approach in a simple statistical test. Waudby-Smith and Ramdas [2021] seem to follow the same approach as Hendriks [2018] but propose a number of heuristic betting algorithms to maximize the wealth as well as

a discrete version of the uniform mixture that appeared in Hendriks [2018]. The idea of using the regret of gambling algorithms to prove concentration inequalities appear for the first time in Jun and Orabona [2019]. In particular, Jun and Orabona [2019] show how to easily derive a Law of Iterated Logarithm for sub-Gaussian random vectors in Banach spaces from the regret of a one-dimensional betting algorithm and the direction/magnitude reduction (see Section 9.4). In turn, their work was based on the seminal work in Rakhlin and Sridharan [2017] that showed an *equivalence* between the regret guarantee of online learning algorithms with linear losses and concentration inequalities. However, the proof technique in Jun and Orabona [2019] is different from the one in Rakhlin and Sridharan [2017] and it is specific to online algorithms that guarantee a non-negative exponential wealth for biased inputs. In particular, it allows to derive time-uniform concentrations, like the law of the iterated logarithm, that are not possible with the method in Rakhlin and Sridharan [2017]. More recently, Jang et al. [2023] showed how to derive tighter PAC-Bayes bounds from the regret of portfolio algorithms.

12.9 Exercises

Problem 12.1. *Assume that the market gains $w_{t,i}$ are bounded in $[c, C]$. Find a variant of EG can does not need to know c and C and it achieves up to constants the same guarantee in* (12.2)*.*

Appendix A

Appendix

A.1 The Lambert Function and Its Applications

The Lambert function $W(x) : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is defined by the equality

$$x = W(x) \exp\left(W(x)\right) \qquad \text{for } x \geq 0\ . \tag{A.1}$$

Hence, from the definition we have that $\exp(W(x)/2) = \sqrt{\frac{x}{W(x)}}$.

Theorem A.1 ([Hoorfar and Hassani, 2008, Theorem 2.3])**.**

$$W(x) \leq \ln \frac{x + C}{1 + \ln(C)}, \quad \forall x > -\frac{1}{e},\ C > \frac{1}{e}.$$

The following lemma provides upper and lower bounds on $W(x)$.

Theorem A.2 ([Orabona and Pál, 2016, Lemma 17])**.** *The Lambert function $W(x)$ satisfies*

$$0.6321 \ln(x + 1) \leq W(x) \leq \ln(x + 1),\ \forall x \geq 0\ .$$

Proof. The inequalities are satisfied for $x = 0$, hence we in the following we assume $x > 0$. We first prove the lower bound. From (A.1) we have

$$W(x) = \ln\left(\frac{x}{W(x)}\right)\ . \tag{A.2}$$

From this equality, using the elementary inequality $\ln(x) \leq \frac{a}{e} x^{\frac{1}{a}}$ for any $a > 0$, we get

$$W(x) \leq \frac{1}{a\, e}\left(\frac{x}{W(x)}\right)^{a} \quad \forall a > 0,$$

that is

$$W(x) \leq \left(\frac{1}{a\, e}\right)^{\frac{1}{1+a}} x^{\frac{a}{1+a}} \quad \forall a > 0. \tag{A.3}$$

Using (A.3) in (A.2), we have

$$W(x) \geq \ln\left(\frac{x}{\left(\frac{1}{a\, e}\right)^{\frac{1}{1+a}} x^{\frac{a}{1+a}}}\right) = \frac{1}{1+a} \ln\left(a\, e\, x\right) \quad \forall a > 0\ .$$

Consider now the function $g(x) = \frac{x}{x+1} - \frac{b}{\ln(1+b)(b+1)} \ln(x+1)$ defined in $[0, b]$ where b is a positive number that will be decided in the following. This function has a maximum in $x^\star = (1 + \frac{1}{b}) \ln(1 + b) - 1$, the derivative is positive in

$[0, x^\star]$ and negative in $[x^\star, b]$. Hence the minimum is in $x = 0$ and in $x = b$, where it is equal to 0. Using the property just proved on g, setting $a = \frac{1}{x}$, we have

$$W(x) \geq \frac{x}{x+1} \geq \frac{b}{\ln(1+b)(b+1)} \ln(x+1) \;\; \forall x \leq b \, .$$

For $x > b$, setting $a = \frac{x+1}{ex}$, we have

$$W(x) \geq \frac{e\, x}{(e+1)x+1} \ln(x+1) \geq \frac{e\, b}{(e+1)b+1} \ln(x+1) \tag{A.4}$$

Hence, we set b such that

$$\frac{e\, b}{(e+1)b+1} = \frac{b}{\ln(1+b)(b+1)}$$

Numerically, $b = 1.71825...$, so

$$W(x) \geq 0.6321 \ln(x+1) \, .$$

For the upper bound, we use Theorem A.1 and set $C = 1$. □

Theorem A.3. *Let $a, b > 0$. Then, the Fenchel conjugate of $f(x) = b \exp(x^2/(2a))$ is*

$$f^\star(\theta) = \sqrt{a}|\theta|\sqrt{W(a\theta^2/b^2)} - b \exp\left(\frac{W(a\theta^2/b^2)}{2}\right) = \sqrt{a}|\theta| \left(\sqrt{W(a\theta^2/b^2)} - \frac{1}{\sqrt{W(a\theta^2/b^2)}}\right) \, .$$

Moreover,

$$f^\star(\theta) \leq \sqrt{a}|\theta|\sqrt{\ln(a\theta^2/b^2+1)} - b \, .$$

Proof. First, observe that

$$\max_x \; \theta x - b \exp(x^2/(2a)) = \max_y \; b \left(\frac{\sqrt{a}\theta}{b} y - \exp(y^2/2)\right) \, .$$

Also, by the definition of Lambert function, we have

$$\operatorname*{argmax}_y \; uy - \exp(y^2/2) = \operatorname{sign}(u)\sqrt{W(y^2)},$$

where $\operatorname{sign}(u) = 0$ for $u = 0$. Hence, $\operatorname{argmax}_x \; \theta x - b \exp(x^2/(2a)) = \operatorname{sign}(\theta)\sqrt{a}\sqrt{W(a\theta^2/b^2)}$. So,

$$\begin{aligned}
\max_x \; \theta x - b \exp(x^2/(2a)) &= \max_y \; b \left(\frac{\sqrt{a}\theta}{b} y - \exp(y^2/2)\right) \\
&= \sqrt{a}|\theta|\sqrt{W(a\theta^2/b^2)} - b \exp\left(\frac{W(a\theta^2/b^2)}{2}\right) \\
&= \sqrt{a}|\theta|\sqrt{W(a\theta^2/b^2)} - b\sqrt{\frac{a\theta^2/b^2}{W(a\theta^2/b^2)}} \\
&= \sqrt{a}|\theta| \left(\sqrt{W(a\theta^2/b^2)} - \frac{1}{\sqrt{W(a\theta^2/b^2)}}\right) \, .
\end{aligned}$$

□

The upper bound is obtained through Theorem A.2 and upper bounding $-b \exp\left(\frac{W(a\theta^2/b^2)}{2}\right)$ with $-b$.
For the lower bound, observe that

$$-b \exp\left(\frac{W(a\theta^2/b^2)}{2}\right) \geq -b \exp\left(\frac{1}{2} \ln\left(a\theta^2/b^2+1\right)\right) = -b\sqrt{a\theta^2/b^2+1} = -\sqrt{a\theta^2+b^2} \geq -\sqrt{a}|\theta| - b$$

A.2 Topology Bits

Definition A.4 (Bounded Set)**.** *A non-empty subset M in a metric space X with metric $d(\cdot,\cdot)$ is **bounded** if $\sup_{\boldsymbol{x},\boldsymbol{y}\in M}\ d(x,y) < \infty$.*

Definition A.5 (Open and Closed Sets)**.** *A subset M of a metric space X is said to be **open** if it contains a ball centered on each of its points. A subset M of X is **closed** if its complement in X is open.*

Remark A.6. *A set can be open, closed, both, or neither. In particular, the empty set and the entire metric space are both closed and open. An example of a set that is neither is the set $(0,1]$ on $\mathbb{R}$.*

Definition A.7 (Neighborhood)**.** *In a metric space X, a set V is a **neighbourhood** of a point $\boldsymbol{x}$ if there exists an open ball with centre $\boldsymbol{x}$ and radius $r > 0$ contained in V.*

Definition A.8 (Interior point and Interior of a Set)**.** *A point $\boldsymbol{x}$ is an **interior point** of a set V if V is a neighborhood of $\boldsymbol{x}$. The **interior** of a set V, denoted by $\operatorname{int} V$, is the set of all interior points of V.*

Note that $\operatorname{int} V$ is the largest open set contained in V.

Definition A.9 (Boundary points and Boundary of a Set)**.** *A point $\boldsymbol{x}$ is a **boundary point** of V in a metric space X with metric $d(\cdot,\cdot)$ if every neighborhood of $\boldsymbol{x}$ contains points of V as well as points not in V. The **boundary of** V, denoted by $\operatorname{bdry} V$, is the set of all boundary points of V.*

Theorem A.10. *For any subset A of a Euclidean space, A is compact if and only if it is closed and bounded.*

Theorem A.11 (Weierstrass for extended functions [Bauschke et al., 2003, Theorem 1.28])**.** *Let $\mathcal{X}$ a Hausdorff space, let $f : \mathcal{X} \to [-\infty, +\infty]$ be lower semicontinuous, and let V be a compact subset of $\mathcal{X}$. Suppose $V \cap \operatorname{dom} f \neq \{\}$. Then, f achieves its infimum over V.*

Bibliography

J. Abernethy, C. Lee, A. Sinha, and A. Tewari. Online linear optimization via smoothing. In *Conference on Learning Theory (COLT)*, pages 807–823, 2014. URL http://proceedings.mlr.press/v35/abernethy14. 88

J. D. Abernethy and J.-K. Wang. On Frank-Wolfe and equilibrium computation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/7371364b3d72ac9a3ed8638e6f0be2c9-Paper.pdf. 141, 142

J. D. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In Rocco A. Servedio and Tong Zhang, editors, *Proc. of Conference on Learning Theory (COLT)*, pages 263–274. Omnipress, 2008. URL https://repository.upenn.edu/cgi/viewcontent.cgi?article=1492&context=statistics_papers. 61, 87

J. D. Abernethy, C. Lee, and A. Tewari. Fighting bandits with a new kind of smoothness. In *Advances in Neural Information Processing Systems 28*, pages 2197–2205. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/6030-fighting-bandits-with-a-new-kind-of-smoothness.pdf. 124

A. Agarwal and E. Hazan. New algorithms for repeated play and universal portfolio management. Technical Report TR-740-05, Princeton University Technical Report, 2005. URL ftp://ftp.cs.princeton.edu/techreports/2005/740.pdf/. 87

A. Agarwal, A. Rakhlin, and P. Bartlett. Matrix regularization techniques for online multitask learning. *EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2008-138*, 2008. URL https://www2.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-138.html. 88

N. Agarwal, R. Anil, E. Hazan, T. Koren, and C. Zhang. Disentangling adaptive gradient methods from learning rates. *arXiv preprint arXiv:2002.11803*, 2020. URL https://arxiv.org/abs/2002.11803. 34

M. A. Aizerman, E. M. Braverrnan, and L. I. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, 25(6):917–936, 1964. URL https://cs.uwaterloo.ca/~y328yu/classics/kernel.pdf. 95

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2006/file/0afa92fc0f8a9cf051bf2961b06ac56b-Paper.pdf. 88

R. Arora, O. Dekel, and A. Tewari. Online bandit learning against an adaptive adversary: from regret to policy regret. In *Proc. of the International Conference on Machine Learning (ICML)*, January 2012. URL https://www.microsoft.com/en-us/research/publication/online-bandit-learning-adaptive-adversary-regret-policy-regret/. 112

J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proc. of the Conference on Learning Theory (COLT)*, 2009. URL https://www.di.ens.fr/willow/pdfscurrent/COLT09a.pdf. 124

J.-Y. Audibert, R. Munos, and C. Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009. URL https://www.sciencedirect.com/science/article/pii/S030439750900067X. 154

J.-Y. Audibert, S. Bubeck, and G. Lugosi. Minimax policies for combinatorial prediction games. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 107–132, 2011. URL http://proceedings.mlr.press/v19/audibert11a.html. 124

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002a. URL https://homes.di.unimi.it/cesa-bianchi/Pubblicazioni/ml-02.pdf. 125

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002b. URL http://rob.schapire.net/papers/AuerCeFrSc01.pdf. 124

P. Auer, N. Cesa-Bianchi, and C. Gentile. Adaptive and self-confident on-line learning algorithms. *J. Comput. Syst. Sci.*, 64(1):48–75, 2002c. URL http://homes.dsi.unimi.it/~cesabian/Pubblicazioni/jcss-02.pdf. 34

K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001. URL https://link.springer.com/article/10.1023/A:1010896012157. 61, 89

S. Bakin. *Adaptive regression and model selection in data mining problems*. PhD thesis, The Australian National University, 1999. URL https://openresearch-repository.anu.edu.au/bitstream/1885/9449/6/Bakin_S_1999.pdf. 88

H. H. Bauschke and J. M. Borwein. Legendre functions and the method of random Bregman projections. *Journal of convex analysis*, 4(1):27–67, 1997. URL https://www.heldermann-verlag.de/jca/jca04/jca04002.pdf. 61

H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*, volume 408. Springer, 2011. URL https://link.springer.com/book/10.1007/978-1-4419-9467-7. 13, 14, 37, 38, 39, 46, 53

H. H. Bauschke, J. M. Borwein, and P. L. Combettes. Bregman monotone optimization algorithms. *SIAM Journal on Control and Optimization*, 42(2):596–636, 2003. URL https://epubs.siam.org/doi/pdf/10.1137/S0363012902407120. 60, 159

H. H. Bauschke, J. Bolte, and M. Teboulle. A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017. URL https://people.ok.ubc.ca/bauschke/Research/103.pdf. 88

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003. URL https://www.sciencedirect.com/science/article/abs/pii/S0167637702002316. 55, 60, 61

A. Ben-Tal, T. Margalit, and A. Nemirovski. The ordered subsets mirror descent optimization method with applications to tomography. *SIAM Journal on Optimization*, 12(1):79–108, 2001. URL https://epubs.siam.org/doi/pdf/10.1137/S1052623499354564. 61

A. Beygelzimer, F. Orabona, and C. Zhang. Efficient online bandit multiclass learning with $\tilde{O}(\sqrt{T})$ regret. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 488–497. JMLR. org, 2017. URL https://arxiv.org/abs/1702.07958. 95

B. Birnbaum, N. R. Devanur, and L. Xiao. New convex programs and distributed algorithms for Fisher markets with linear and spending constraint utilities. Technical Report MSR-TR-2010-112, Microsoft Research, August 2010. URL https://www.microsoft.com/en-us/research/publication/new-convex-programs-and-distributed-algorithms-for-fisher-markets-with-linear-and-spendi 88

B. Birnbaum, N. R. Devanur, and L. Xiao. Distributed algorithms via gradient descent for Fisher markets. In *Proceedings of the 12th ACM conference on Electronic commerce*, pages 127–136, 2011. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/prop_response.pdf. 88

D. Blackwell and D. Freedman. On the amount of variance needed to escape from a strip. *The Annals of Probability*, 1(5):772–787, 1973. URL https://projecteuclid.org/journals/annals-of-probability/volume-1/issue-5/On-the-Amount-of-Variance-Needed-to-Escape-from-a/10.1214/aop/1176996845.full. 34

H.-D. Block. The Perceptron: A model for brain functioning. I. *Reviews of Modern Physics*, 34(1):123–135, 1962. URL https://journals.aps.org/rmp/abstract/10.1103/RevModPhys.34.123. 95

A. Blum and A. Kalai. Universal portfolios with and without transaction costs. In *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, pages 309–313, 1997. URL https://dl.acm.org/doi/pdf/10.1145/267460.267518. 155

A. Blum and A. Kalai. Universal portfolios with and without transaction costs. *Machine Learning*, 35:193–205, 1999. URL https://link.springer.com/article/10.1023/A:1007530728748. 155

A. Blum, A. Kalai, and J. Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *COLT*, volume 99, pages 203–208, 1999. URL https://www.ri.cmu.edu/pub_files/pub1/blum_a_1999_1/blum_a_1999_1.pdf. 23

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787. URL https://web.stanford.edu/~boyd/cvxbook/. 8

L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967. URL https://www.sciencedirect.com/science/article/pii/0041555367900407. 60

S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012. URL http://sbubeck.com/SurveyBCB12.pdf. 124, 125

N. Burch, M. Moravcik, and M. Schmid. Revisiting CFR+ and alternating updates. *Journal of Artificial Intelligence Research*, 64:429–443, 2019. URL https://arxiv.org/abs/1810.11542. 142

N. Campolongo and F. Orabona. Temporal variability in implicit online learning. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/9239be5f9dc4058ec647f14fd04b1290-Abstract.html. 142

N. Cesa-Bianchi. Analysis of two gradient-based algorithms for on-line regression. *Journal of Computer and System Sciences*, 59(3):392–411, 1999. URL https://www.sciencedirect.com/science/article/pii/S0022000099916355. 16

N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006. URL https://www.cambridge.org/us/academic/subjects/computer-science/pattern-recognition-and-machine-learning/prediction-learning-and-games?format=HB&isbn=9780521841085. 16, 40, 61, 81, 83, 98, 124, 142

N. Cesa-Bianchi, Y. Freund, D. P. Helmbold, D. Haussler, R. E. Schapire, and M. K Warmuth. How to use expert advice. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of Computing*, pages 382–391, 1993. URL https://dl.acm.org/doi/pdf/10.1145/167088.167198. 61

N. Cesa-Bianchi, P. M. Long, and M. K. Warmuth. Worst-case quadratic loss bounds for prediction using linear functions and gradient descent. *IEEE Transactions on Neural Networks*, 7(3):604–619, 1996. URL https://ieeexplore.ieee.org/document/501719. 34

N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth. How to use expert advice. *J. ACM*, 44(3):427–485, 1997. URL https://cseweb.ucsd.edu/~yfreund/papers/expertAdvice.pdf. 61

N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Trans. Inf. Theory*, 50(9):2050–2057, 2004. URL https://homes.di.unimi.it/~cesabian/Pubblicazioni/J20.pdf. 23

N. Cesa-Bianchi, Y. Mansour, and G. Stoltz. Improved second-order bounds for prediction with expert advice. In *International Conference on Computational Learning Theory*, pages 217–232. Springer, 2005. URL http://www.cs.tau.ac.il/~mansour/papers/05colt.pdf. 34

N. Cesa-Bianchi, Y. Mansour, and G. Stoltz. Improved second-order bounds for prediction with expert advice. *Machine Learning*, 66(2):321–352, 2007. URL https://link.springer.com/content/pdf/10.1007/s10994-006-5001-7.pdf. 34

K. Chaudhuri, Y. Freund, and D. J. Hsu. A parameter-free hedging algorithm. In *Advances in neural information processing systems*, pages 297–305, 2009. URL https://arxiv.org/abs/0903.2851. 109, 110

G. Chen and M. Teboulle. Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *SIAM Journal on Optimization*, 3(3):538–543, 1993. URL https://epubs.siam.org/doi/pdf/10.1137/0803026. 46

A. Chernov and V. Vovk. Prediction with advice of unknown number of experts. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010. URL https://arxiv.org/abs/1408.2040. 110

C.-K. Chiang, T. Yang, C.-J. Lee, M. Mahdavi, C.-J. Lu, R. Jin, and S. Zhu. Online optimization with gradual variations. In *Proc. of the Conference on Learning Theory (COLT)*, volume 23, pages 6.1–6.20, 2012. URL http://proceedings.mlr.press/v23/chiang12.html. 61, 89, 142

T. Cover. Behavior of sequential predictors of binary sequences. In *Proc. of the 4th Prague Conference on Information Theory, Statistical Decision Functions and Random Processes*, pages 263–272. Publishing House of the Czechoslovak Academy of Sciences, 1965. URL https://isl.stanford.edu/~cover/papers/paper3.pdf. 61

T. M Cover. Universal gambling schemes and the complexity measures of Kolmogorov and Chaitin. Technical Report 12, Department of Statistics, Stanford University, 1974. URL https://purl.stanford.edu/js411qm9805. 155

T. M. Cover. Universal portfolios. *Mathematical Finance*, pages 1–29, 1991. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9965.1991.tb00002.x. 145, 155

T. M. Cover and E. Ordentlich. Universal portfolios with side information. *IEEE Transactions on Information Theory*, 42(2):348–363, 1996. URL https://ieeexplore.ieee.org/document/485708. 146, 149, 155

T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, 2006. 155

F. Cucker and D. X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, New York, NY, USA, 2007. 94

A. Cutkosky. *Algorithms and Lower Bounds for Parameter-free Online Learning*. PhD thesis, Stanford University, 2018. URL https://www-cs.stanford.edu/people/ashokc/papers/thesis.pdf. 41

A. Cutkosky. Anytime online-to-batch, optimism and acceleration. In K. Chaudhuri and R. Salakhutdinov, editors, *Proc. of the 36th International Conference on Machine Learning*, volume 97 of *Proc. of Machine Learning Research*, pages 1446–1454, Long Beach, California, USA, 09–15 Jun 2019a. PMLR. URL http://proceedings.mlr.press/v97/cutkosky19a/cutkosky19a.pdf. 23

A. Cutkosky. Combining online learning guarantees. In *Proc. of the Conference on Learning Theory (COLT)*, 2019b. URL https://arxiv.org/abs/1902.09003. 110

A. Cutkosky. Better full-matrix regret via parameter-free online learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8836–8846. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/6495cf7ca745a9443508b86951b8e33a-Paper.pdf. 34

A. Cutkosky and F. Orabona. Black-box reductions for parameter-free online learning in Banach spaces. In *Proc. of the Conference on Learning Theory (COLT)*, 2018. URL https://arxiv.org/abs/1802.06293. 110

C. Daskalakis and I. Panageas. Last-iterate convergence: Zero-sum games and constrained min-max optimization. In *10th Innovations in Theoretical Computer Science (ITCS) conference*, 2019. URL https://arxiv.org/pdf/1807.04252.pdf. 143

C. Daskalakis, A. Deckelbaum, and A. Kim. Near-optimal no-regret algorithms for zero-sum games. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms*, pages 235–254. SIAM, 2011. URL https://epubs.siam.org/doi/pdf/10.1137/1.9781611973082.21. 142

C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng. Training GANs with optimism. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJJySbbAZ. 142

S. de Rooij, T. van Erven, P. D. Grünwald, and W. M. Koolen. Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research*, 15(37):1281–1316, 2014. URL http://jmlr.org/papers/v15/rooij14a.html. 88

G. Denevi, M. Pontil, and D. Stamos. Online parameter-free learning of multiple low variance tasks. In *Conference on Uncertainty in Artificial Intelligence*, pages 889–898. PMLR, 2020. URL https://proceedings.mlr.press/v124/denevi20a.html. 111

C. Ding, D. Zhou, X. He, and H. Zha. r_1-PCA: rotational invariant l_1-norm principal component analysis for robust subspace factorization. In *Proceedings of the 23rd international conference on Machine learning*, pages 281–288, 2006. URL https://ranger.uta.edu/~chqding/papers/R1PCA.pdf. 88

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *COLT*, 2010. URL https://stanford.edu/~jduchi/projects/DuchiHaSi10_colt.pdf. 32, 34

G. Farina, C. Kroer, and T. Sandholm. Online convex optimization for sequential decision processes and extensive-form games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1917–1925, 2019. URL https://arxiv.org/pdf/1809.03075.pdf. 142

G. E. Flaspohler, F. Orabona, J. Cohen, S. Mouatadid, M. Oprescu, P. Orenstein, and L. Mackey. Online learning with optimism and delay. In M. Meila and T. Zhang, editors, *Proc. of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3363–3373. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/flaspohler21a.html. 89

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800030109. 142

Y. Freund and R. E. Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Second European Conference, EuroCOLT'95 Barcelona, Spain, March 13–15*, pages 23–37. Springer, 1995. 61, 136, 142

Y. Freund and R. E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, COLT '96, pages 325—-332, New York, NY, USA, 1996. Association for Computing Machinery. URL http://www.cs.cmu.edu/~ninamf/LGO10/wm-minimax.pdf. 142

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997. URL https://www.sciencedirect.com/science/article/pii/S002200009791504X. 61, 136, 142

Y. Freund and R. E. Schapire. Large margin classification using the Perceptron algorithm. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 209—-217, New York, NY, USA, 1998. Association for Computing Machinery. URL https://dl.acm.org/doi/10.1145/279943.279985. 95

Y. Freund and R. E. Schapire. Large margin classification using the Perceptron algorithm. *Machine Learning*, pages 277–296, 1999a. URL https://cseweb.ucsd.edu/~yfreund/papers/LargeMarginsUsingPerceptron.pdf. 95

Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999b. URL https://cseweb.ucsd.edu/~yfreund/papers/games_long.pdf. 142

D. Fudenberg and D. K. Levine. Consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, 19:1065—-1089, 1995. URL https://dash.harvard.edu/bitstream/handle/1/3198694/fudenberg_consistency.pdf. 61

P. Gács. Uniform test of algorithmic randomness over a general space. *Theoretical Computer Science*, 341(1-3):91–137, 2005. URL https://www.sciencedirect.com/science/article/pii/S030439750500188X. 155

P. Gaillard, G. Stoltz, and T. Van Erven. A second-order bound with excess losses. In *Conference on Learning Theory*, pages 176–196. PMLR, 2014. URL https://proceedings.mlr.press/v35/gaillard14.html. 89

A. Garivier and O. Cappé. The KL-UCB algorithm for bounded stochastic bandits and beyond. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 359–376, 2011. URL https://proceedings.mlr.press/v19/garivier11a.html. 152

C. Gentile. The robustness of the p-norm algorithms. *Machine Learning*, 53(3):265–299, 2003. URL https://link.springer.com/article/10.1023/A:1026319107706. 61, 95

C. Gentile and N. Littlestone. The robustness of the p-norm algorithms. In *Proc. of the Twelfth Annual Conference on Computational Learning Theory*, COLT '99, pages 1–11, New York, NY, USA, 1999. ACM. URL http://doi.acm.org/10.1145/307400.307405. 61, 95

G. Gidel, H. Berard, G. Vignoud, P. Vincent, and S. Lacoste-Julien. A variational inequality perspective on generative adversarial networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=r1laEnA5Ym. 142

G. J. Gordon. *Approximate solutions to Markov decision processes*. PhD thesis, Carnegie Mellon University, 1999a. URL https://www.proquest.com/docview/304499958. 16, 87

G. J. Gordon. Regret bounds for prediction problems. In *Proc. of the twelfth annual conference on Computational learning theory (COLT)*, pages 29–40, 1999b. URL http://www.cs.cmu.edu/~ggordon/colt99.ps.gz. 16, 87

A. J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. In *Proc. of the 10th Annual Conference on Computational Learning Theory*, pages 171–183, 1997. URL https://dl.acm.org/doi/pdf/10.1145/267460.267493?casa_token=1D4SKEw0fjUAAAAA:Kzh3RPTbgUukpJCf4a-WHmrxUnhPAw-xYZK4pkzcTCdFfRNTE1cnF4c78DJr3krWJ8Uu30RxoCvf. 61

A. J. Grove, N. Littlestone, and D. Schuurmans. General convergence results for linear discriminant updates. *Machine Learning*, 43(3):173–210, 2001. URL https://link.springer.com/content/pdf/10.1023/A:1010844028087.pdf. 61

J. Hannan. Approximation to Bayes risk in repeated play. In *Contributions to the Theory of Games, Volume III*, pages 97–140. Princeton University Press, 1957. URL http://www-stat.wharton.upenn.edu/~steele/Resources/Projects/SequenceProject/Hannan.pdf. 5, 87

S. Hart and A. Mas-Colell. A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, 68(5):1127–1150, 2000. URL https://www.ma.imperial.ac.uk/~dturaev/Hart0.pdf. 142

E. Hazan and S. Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *Proc. of the 21st Conference on Learning Theory*, 2008. URL http://colt2008.cs.helsinki.fi/papers/46-Hazan.pdf. 87

E. Hazan, A. Kalai, S. Kale, and A. Agarwal. Logarithmic regret algorithms for online convex optimization. In *International Conference on Computational Learning Theory*, pages 499–513. Springer, 2006. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.3483&rep=rep1&type=pdf. 34, 89, 155

E. Hazan, A. Agarwal, and S. Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007. URL https://link.springer.com/content/pdf/10.1007/s10994-007-5016-8.pdf. 155

E. Hazan, A. Rakhlin, and P. L. Bartlett. Adaptive online gradient descent. In *Advances in Neural Information Processing Systems*, pages 65–72, 2008. URL https://papers.nips.cc/paper/3319-adaptive-online-gradient-descent.pdf. 34

D. P. Helmbold, R. E. Schapire, Y. Singer, and M. K. Warmuth. On-line portfolio selection using multiplicative updates. *Mathematical Finance*, 8(4):325–347, 1998. URL http://rob.schapire.net/papers/HelmboldScSiWa98.pdf. 155

H. Hendriks. Test martingales for bounded random variables. *arXiv preprint arXiv:2109.08923*, 2018. URL https://arxiv.org/abs/1801.09418. 155, 156

A. Hoorfar and M. Hassani. Inequalities on the Lambert W function and hyperpower function. *J. Inequal. Pure and Appl. Math*, 9(2):5–9, 2008. URL http://emis.ams.org/journals/JIPAM/images/107_07_JIPAM/107_07_www.pdf. 157

Y.-G. Hsieh, F. Iutzeler, J. Malick, and P. Mertikopoulos. On the convergence of single-call stochastic extra-gradient methods. *Advances in Neural Information Processing Systems*, 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/4625d8e31dad7d1c4c83399a6eb62f0c-Abstract.html. 142

Y.-G. Hsieh, K. Antonakopoulos, and P. Mertikopoulos. Adaptive learning in continuous games: Optimal regret bounds and convergence to Nash equilibrium. In *Conference on Learning Theory*, pages 2388–2422. PMLR, 2021. URL https://proceedings.mlr.press/v134/hsieh21a.html. 143

K. Jang, K.-S. Jun, I. Kuzborskij, and F. Orabona. Tighter PAC-Bayes bounds through coin-betting. *arXiv preprint arXiv:2302.05829*, 2023. URL https://arxiv.org/abs/2302.05829. 156

L. Jie, F. Orabona, M. Fornoni, B. Caputo, and N. Cesa-Bianchi. OM-2: An online multi-class multi-kernel learning algorithm. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 43–50. IEEE, 2010. URL https://homes.di.unimi.it/~cesabian/Pubblicazioni/OM-2.pdf. 88

P. Joulani, A. György, and C. Szepesvári. A modular analysis of adaptive (non-)convex optimization: Optimism, composite objectives, and variational bounds. In *Proc. of the International Conference on Algorithmic Learning Theory (ALT)*, volume 76, pages 681–720, 2017. URL http://proceedings.mlr.press/v76/joulani17a.html. 61, 89

K.-S. Jun and F. Orabona. Parameter-free online convex optimization with sub-exponential noise. In *Proc. of the Conference on Learning Theory (COLT)*, 2019. URL http://proceedings.mlr.press/v99/jun19a/jun19a.pdf. 156

S. Kakade, S. Shalev-Shwartz, and A. Tewari. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. Technical report, TTIC, 2009. URL http://www.cs.huji.ac.il/~shais/papers/KakadeShalevTewari09.pdf. 61, 75, 88

A. T. Kalai and S. Vempala. Efficient algorithms for universal portfolios. *Journal of Machine Learning Research*, pages 423–440, 2002. URL https://jmlr.org/papers/v3/kalai02a.html. 155

M. Kearns. Thoughts on hypothesis boosting. Machine Learning class project, December 1988. URL https://www.cis.upenn.edu/~mkearns/papers/boostnote.pdf. 142

M. Kearns and L. G. Valiant. Learning boolean formulae or finite automata is as hard as factoring. Technical Report TR-14-88, Harvard University Aikem Computation Laboratory, 1988. 142

J. L. Kelly, jr. A new interpretation of information rate. *IRE Transactions on Information Theory*, 2(3):185–189, 1956. URL https://ieeexplore.ieee.org/document/1056803. 155

J. Kivinen and M. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132(1):1–63, January 1997. URL https://users.soe.ucsc.edu/~manfred/pubs/J36.pdf. 34, 61, 142

J. Kivinen and M. K. Warmuth. Averaging expert predictions. In *European Conference on Computational Learning Theory*, pages 153–167. Springer, 1999. URL https://users.soe.ucsc.edu/~manfred/pubs/C50.pdf. 89

W. M. Koolen and T. van Erven. Second-order quantile methods for experts and combinatorial games. In *Proc. of the Conference On Learning Theory (COLT)*, pages 1155–1175, 2015. URL https://arxiv.org/abs/1502.08009. 110

G. M. Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976. 142

R. Krichevsky and V. Trofimov. The performance of universal encoding. *IEEE Trans. on Information Theory*, 27(2):199–207, 1981. URL https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1056331. 110, 155

C. Kroer. IEOR8100: Economics, AI, and optimization. lecture note 5: Computing Nash equilibrium via regret minimization. Technical report, Columbia University, 2020. URL http://www.columbia.edu/~ck2945/files/s20_8100/lecture_note_5_nash_from_rm.pdf. 142

B. Kulis and P. L. Bartlett. Implicit online learning. In *International Conference on Machine Learning*, pages 575–582, 2010. URL https://icml.cc/Conferences/2010/papers/429.pdf. 142

S. Lacoste-Julien, M. Schmidt, and F. Bach. A simpler approach to obtaining an O(1/t) convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002*, 2012. URL https://arxiv.org/abs/1212.2002. 34

T. L. Lai. Adaptive treatment allocation and the multi-armed bandit problem. *The Annals of Statistics*, pages 1091–1114, 1987. URL https://projecteuclid.org/euclid.aos/1176350495. 125

T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1): 4–22, 1985. URL https://core.ac.uk/download/pdf/82425825.pdf. 125

T. L. Lai and C. Z. Wei. Least squares estimates in stochastic regression models with applications to identification and control of dynamic systems. *The Annals of Statistics*, 10(1):154–166, 1982. URL https://projecteuclid.org/journals/annals-of-statistics/volume-10/issue-1/Least-Squares-Estimates-in-Stochastic-Regression-Models-with-Applications-to/10.1214/aos/1176345697.full. 89

J. Langford and T. Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824, 2008. URL https://papers.nips.cc/paper/3178-the-epoch-greedy-algorithm-for-multi-armed-bandits-with-side-information. 124

T. Lattimore and C. Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020. URL https://tor-lattimore.com/downloads/book/book.pdf. 119, 124

C.-W. Lee, C. Kroer, and H. Luo. Last-iterate convergence in extensive-form games. In *Advances in Neural Information Processing Systems*, volume 34, pages 14293–14305, 2021. URL https://proceedings.neurips.cc/paper/2021/file/77bb14f6132ea06dea456584b7d5581e-Paper.pdf. 143

Q. Lei, S. G. Nagarajan, I. Panageas, and X. Wang. Last iterate convergence in no-regret learning: constrained min-max optimization for convex-concave landscapes. In *International Conference on Artificial Intelligence and Statistics*, pages 1441–1449. PMLR, 2021. URL https://proceedings.mlr.press/v130/lei21a.html. 143

L. A. Levin. Uniform tests of randomness. *Doklady Akademii Nauk*, 227(1):33–35, 1976. English version available at https://www.cs.bu.edu/fac/lnd/dvi/rnd76.pdf. 155

T. Liang and J. Stokes. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In K. Chaudhuri and M. Sugiyama, editors, *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 907–915. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/liang19b.html. 143

N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Information and computation*, 108(2):212–261, 1994. URL https://www.sciencedirect.com/science/article/pii/S0890540184710091. 61

M. Liu and F. Orabona. A parameter-free algorithm for convex-concave min-max problems. *arXiv preprint arXiv:2103.00284*, 2021. URL https://arxiv.org/abs/2103.00284. 141

H. Lu, R. M. Freund, and Y. Nesterov. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28(1):333–354, 2018. URL https://arxiv.org/abs/1610.05708. 88

H. Luo. CSCI 659: Introduction to online optimization/learning, lecture 4, 2022. URL https://haipeng-luo.net/courses/CSCI659/2022_fall/lectures/lecture4.pdf. 142

H. Luo and R. E. Schapire. A drifting-games analysis for online learning and applications to boosting. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/ccb0989662211f61edae2e26d58ea92f-Paper.pdf. 142

A. Maurer and M. Pontil. Empirical Bernstein bounds and sample variance penalization. In *Proc. of the Conference on Learning Theory*, 2009. URL https://arxiv.org/abs/0907.3740. 154

B. McMahan and J. Abernethy. Minimax optimal algorithms for unconstrained linear optimization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2724–2732. Curran Associates, Inc., 2013. URL http://papers.nips.cc/paper/5148-minimax-optimal-algorithms-for-unconstrained-linear-optimization.pdf. 41

H. B. McMahan. Follow-the-regularized-leader and mirror descent: Equivalence theorems and L1 regularization. In *Proc. of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS*, pages 525–533, 2011. URL http://proceedings.mlr.press/v15/mcmahan11b/mcmahan11b.pdf. 88

H B. McMahan. A survey of algorithms and analysis for adaptive online learning. *The Journal of Machine Learning Research*, 18(1):3117–3166, 2017. URL http://www.jmlr.org/papers/volume18/14-428/14-428.pdf. 88

H. B. McMahan and F. Orabona. Unconstrained online linear learning in Hilbert spaces: Minimax algorithms and normal approximations. In *Proc of the Annual Conference on Learning Theory, COLT*, 2014. URL https://arxiv.org/abs/1403.0628. 110

H. B. McMahan and M. J. Streeter. Adaptive bound optimization for online convex optimization. In *COLT*, 2010. URL https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/36483.pdf. 32, 34

H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hrafnkelsson, T. Boulos, and J. Kubica. Ad click prediction: a view from the trenches. In *Proc. of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2013. URL https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41159.pdf. 88

Z. Mhammedi and W. M Koolen. Lipschitz and comparator-norm adaptivity in online learning. In *Conference on Learning Theory*, pages 2858–2887. PMLR, 2020. URL http://proceedings.mlr.press/v125/mhammedi20a/mhammedi20a.pdf. 110

J. Milnor. Games against nature. Technical Report RM-679, RAND PROJECT AIR FORCE, 1951. URL https://www.rand.org/pubs/research_memoranda/RM0679.html. 5

A. Mokhtari, A. Ozdaglar, and S. Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *International Conference on Artificial Intelligence and Statistics*, pages 1497–1507. PMLR, 2020. URL https://arxiv.org/pdf/1901.08511.pdf. 142

J. Negrea, B. Bilodeau, N. Campolongo, F. Orabona, and D. Roy. Minimax optimal quantile and semi-adversarial regret via root-logarithmic regularizers. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/file/dcd2f3f312b6705fb06f4f9f1b55b55c-Paper.pdf. 110

A. S. Nemirovskij and D. Yudin. *Problem complexity and method efficiency in optimization*. Wiley, New York, NY, USA, 1983. URL https://books.google.com/books/about/Problem_Complexity_and_Method_Efficiency.html?id=6ULvAAAAMAAJ. 60

Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259, 2009. URL https://link.springer.com/content/pdf/10.1007/s10107-007-0149-x.pdf. 88

Y. Nesterov and V. Shikhman. Quasi-monotone subgradient methods for nonsmooth convex minimization. *Journal of Optimization Theory and Applications*, 165(3):917–940, 2015. URL https://link.springer.com/article/10.1007/s10957-014-0677-5. 23

J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928. 142

J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1944. 142

A. B Novikoff. On convergence proofs for perceptrons. In *Proc. of the Symposium of the Mathematical Theory of Automata*, volume XII, pages 615–622. Wiley, New York, 1963. URL http://classes.engr.oregonstate.edu/eecs/spring2020/cs519-400/extra/novikoff-1962.pdf. 95

F. Orabona. Dimension-free exponentiated gradient. In *Advances in Neural Information Processing Systems 26*, pages 1806–1814. Curran Associates, Inc., 2013. URL https://papers.nips.cc/paper/4920-dimension-free-exponentiated-gradient.pdf. 42, 110

F. Orabona and K.-S. Jun. Tight concentrations and confidence sequences from the regret of universal portfolio. *arXiv preprint arXiv:2110.14099*, 2021. URL https://arxiv.org/abs/2110.14099. 154, 155

F. Orabona and D. Pál. Scale-free algorithms for online linear optimization. In *International Conference on Algorithmic Learning Theory*, pages 287–301. Springer, 2015. URL https://arxiv.org/abs/1502.05744. 34, 88

F. Orabona and D. Pál. Coin betting and parameter-free online learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 577–585. Curran Associates, Inc., 2016. URL https://arxiv.org/pdf/1602.04128.pdf. 41, 98, 110, 157

F. Orabona and D. Pál. Scale-free online learning. *Theoretical Computer Science*, 716:50–69, 2018. URL https://arxiv.org/pdf/1601.01974.pdf. Special Issue on ALT 2015. 34, 41, 88

F. Orabona and D. Pál. Parameter-free stochastic optimization of variationally coherent functions. *arXiv preprint arXiv:2102.00236*, 2021. URL https://arxiv.org/abs/2102.00236. 110

F. Orabona and T. Tommasi. Training deep networks without learning rates through coin betting. In *Advances in Neural Information Processing Systems*, pages 2160–2170, 2017. URL https://arxiv.org/abs/1705.07795. 110

F. Orabona, L. Jie, and B. Caputo. Online-batch strongly convex multi kernel learning. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 787–794. IEEE, 2010. URL https://ieeexplore.ieee.org/document/5540137. 88

F. Orabona, N. Cesa-Bianchi, and C. Gentile. Beyond logarithmic bounds in online learning. In N. D. Lawrence and M. Girolami, editors, *Proc. of the 15th International Conference on Artificial Intelligence and Statistics, AISTATS, La Palma, Canary Islands, April 21-23, 2012*, volume 22 of *JMLR Proceedings*, pages 823–831. JMLR.org, 2012a. URL http://proceedings.mlr.press/v22/orabona12/orabona12.pdf. 89

F. Orabona, L. Jie, and B. Caputo. Multi kernel learning with online-batch optimization. *Journal of Machine Learning Research*, 13(2), 2012b. URL https://jmlr.csail.mit.edu/papers/volume13/orabona12a/orabona12a.pdf. 88

F. Orabona, K. Crammer, and N. Cesa-Bianchi. A generalized online mirror descent with applications to classification and regression. *Machine Learning*, 99:411–435, 2015. URL https://arxiv.org/abs/1304.2994. 88, 89

B. Polyak. Existence theorems and convergence of minimizing sequences in extremum problems with restrictions. *Dokl. Akad. Nauk SSSR*, 166(2):287—290, 1966. URL https://www.researchgate.net/publication/265564392_Existence_theorems_and_convergence_of_minimizing_sequences_in_extremum_problems_with_restrictions. 34

L. D. Popov. A modification of the Arrow-Hurwicz method for search of saddle points. *Mathematical notes of the Academy of Sciences of the USSR*, 28(5):845–848, 1980. URL https://link.springer.com/article/10.1007/BF01141092?error=cookies_not_supported&code=8d34e3d9-bc6b-4c0c-9c69-0656f1caf391. 61, 142

A. Rakhlin and K. Sridharan. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, pages 3066–3074, 2013a. URL https://arxiv.org/abs/1311.1869. 142

A. Rakhlin and K. Sridharan. Online learning with predictable sequences. In *Proc. of the Conference on Learning Theory (COLT)*, volume 30, pages 993–1019, 2013b. URL http://proceedings.mlr.press/v30/Rakhlin13.html. 61, 89, 142

A. Rakhlin and K. Sridharan. On equivalence of martingale tail bounds and deterministic regret inequalities. In *Proc. of the Conference On Learning Theory (COLT)*, pages 1704–1722, 2017. URL https://proceedings.mlr.press/v65/rakhlin17a.html. 156

S. J. Reddi, S. Kale, and S. Kumar. On the convergence of Adam and beyond. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/pdf?id=ryQu7f-RZ. 8

H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952. URL https://projecteuclid.org/euclid.bams/1183517370. 124

H. Robbins and D. Siegmund. The expected sample size of some tests of power one. *The Annals of Statistics*, 2(3):415–436, 1974. URL https://projecteuclid.org/journals/annals-of-statistics/volume-2/issue-3/The-Expected-Sample-Size-of-Some-Tests-of-Power-One/10.1214/aos/1176342704.pdf. 155

R. T. Rockafellar. *Convex functions and dual extremum problems*. PhD thesis, Harvard University, 1963. 16

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970. URL https://press.princeton.edu/titles/1815.html. 9, 13, 38, 61, 141

F. Rosenblatt. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.335.3398&rep=rep1&type=pdf. 95

J. J. Ryu and A. Bhatt. On confidence sequences for bounded random processes via universal gambling strategies. *arXiv preprint arXiv:2207.12382*, 2022. URL https://arxiv.org/abs/2207.12382. 155

S. Sachs, H. Hadiji, T. van Erven, and C. Guzmán. Between stochastic and adversarial online convex optimization: Improved regret bounds via smoothness. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2202.07554. 89

L. J. Savage. The theory of statistical decision. *Journal of the American Statistical Association*, 46(253):55–67, 1951. URL https://www.jstor.org/stable/2280094. 5

L. J. Savage. *The foundations of statistics*. Wiley, 1954. URL https://archive.org/details/foundationsofsta0000sava/. 5

R. E. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990. URL https://www.cs.princeton.edu/~schapire/papers/strengthofweak.pdf. 142

C.-P. Schnorr. A unified approach to the definition of random sequences. *Mathematical systems theory*, 5(3):246–258, 1971. URL https://link.springer.com/content/pdf/10.1007/BF01694181.pdf. 155

V. V. Semenov. A version of the mirror descent method to solve variational inequalities. *Cybernetics and Systems Analysis*, 53(2):234–243, 2017. URL https://link.springer.com/article/10.1007/s10559-017-9923-9. 142, 143

G. Shafer and V. Vovk. *Probability and finance: it's only a game!* John Wiley & Sons, 2001. 155

G. Shafer, A. Shen, N. Vereshchagin, and V. Vovk. Test martingales, Bayes factors and p-values. *Statistical Science*, 26(1):84–101, 2011. URL https://projecteuclid.org/journals/statistical-science/volume-26/issue-1/Test-Martingales-Bayes-Factors-and-p-Values/10.1214/10-STS347.full. 155

S. Shalev-Shwartz. *Online Learning: Theory, Algorithms, and Applications*. PhD thesis, The Hebrew University, 2007. URL https://www.cs.huji.ac.il/~shais/papers/ShalevThesis07.pdf. 57, 87

S. Shalev-Shwartz and Y. Singer. Online learning meets optimization in the dual. In *International Conference on Computational Learning Theory*, pages 423–437. Springer, 2006. URL https://storage.googleapis.com/pub-tools-public-publication-data/pdf/25.pdf. 87

S. Shalev-Shwartz and Y. Singer. Logarithmic regret algorithms for strongly convex repeated games. Technical report, The Hebrew University, 2007a. URL https://www.cse.huji.ac.il/~shais/papers/ShalevSi07report.pdf. 88

S. Shalev-Shwartz and Y. Singer. Convex repeated games and Fenchel duality. In *Advances in neural information processing systems*, pages 1265–1272, 2007b. URL https://ttic.uchicago.edu/~shai/papers/ShalevSi06_fench.pdf. 87

S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proc. of the International Conference on Machine Learning*, pages 807–814, 2007. URL https://ttic.uchicago.edu/~nati/Publications/Pegasos.pdf. 34

L. Sharrock and C. Nemeth. Coin sampling: Gradient-based bayesian inference without learning rates. In *International Conference on Machine Learning*, 2023. URL https://arxiv.org/abs/2301.11294. 111

L. Sharrock, D. Dodd, and C. Nemeth. CoinEM: Tuning-free particle-based variational inference for latent variable models. *arXiv preprint arXiv:2305.14916*, 2023a. URL https://arxiv.org/abs/2305.14916. 111

L. Sharrock, L. Mackey, and C. Nemeth. Learning rate free bayesian inference in constrained domains. *arXiv preprint arXiv:2305.14943*, 2023b. URL https://arxiv.org/abs/2305.14943. 111

M. Sion. On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176, 1958. URL https://projecteuclid.org/journals/pacific-journal-of-mathematics/volume-8/issue-1/On-general-minimax-theorems/pjm/1103040253.full. 142

J. Steinhardt and P. Liang. Adaptivity and optimism: An improved exponentiated gradient algorithm. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 1593–1601, 2014. URL https://cs.stanford.edu/~pliang/papers/eg-icml2014.pdf. 89

G. Stoltz. *Incomplete information and internal regret in prediction of individual sequences*. PhD thesis, Université Paris Sud-Paris XI, 2005. URL https://tel.archives-ouvertes.fr/tel-00009759/document. 124

M. Streeter and B. McMahan. No-regret algorithms for unconstrained online convex optimization. In *Advances in Neural Information Processing Systems 25*, pages 2402–2410. Curran Associates, Inc., 2012. URL https://arxiv.org/pdf/1211.2260.pdf. 41, 110

M. Streeter and H. B. McMahan. Less regret via online conditioning. *arXiv preprintarXiv:1002.4862*, 2010. URL https://arxiv.org/abs/1002.4862. 34

O. Tammelin, N. Burch, M. Johanson, and M. Bowling. Solving heads-up limit Texas hold'em. In *Twenty-fourth international joint conference on artificial intelligence*, 2015. URL https://poker.cs.ualberta.ca/publications/2015-ijcai-cfrplus.pdf. 142

C.-E. Tsai, H.-C. Cheng, and Y.-H. Li. Online self-concordant and relatively smooth minimization, with applications to online portfolio selection and learning quantum states. In *International Conference on Algorithmic Learning Theory*, pages 1481–1483. PMLR, 2023. URL https://proceedings.mlr.press/v201/tsai23a.html. 155

D. van der Hoeven, A. Cutkosky, and H. Luo. Comparator-adaptive convex bandits. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19795–19804. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/e4f37b9ed429c1fe5ce61860d9902521-Paper.pdf. 110

T. van Erven, W. M. Koolen, S. D. Rooij, and P. Grünwald. Adaptive Hedge. In *Advances in Neural Information Processing Systems*, pages 1656–1664, 2011. URL https://papers.nips.cc/paper/4191-adaptive-hedge.pdf. 88

J. Ville. *Étude critique de la notion de collectif*. Gauthier-Villars, Paris, 1939. URL http://archive.numdam.org/item/THESE_1939__218__1_0/. 155

V. Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1751-5823.2001.tb00457.x. 89

V. Vovk. On-line regression competitive with Reproducing Kernel Hilbert Spaces. In J.-Y. Cai, S. B. Cooper, and A. Li, editors, *Theory and Applications of Models of Computation*, volume 3959 of *Lecture Notes in Computer Science*, pages 452–463. Springer Berlin Heidelberg, 2006. URL https://arxiv.org/pdf/cs/0511058.pdf. 110

V. Vovk. Hoeffding's inequality in game-theoretic probability. *arXiv preprint arXiv:0708.2502*, 2007. URL https://arxiv.org/abs/0708.2502. 155

V. Vovk and C. Watkins. Universal portfolio selection. In *Proc. of the Conference on Computational Learning Theory*, pages 12–23, 1998. URL https://dl.acm.org/doi/10.1145/279943.279947. 155

V. G. Vovk. Aggregating strategies. *Proc. of Computational Learning Theory, 1990*, pages 371–386, 1990. URL https://www.sciencedirect.com/science/article/pii/B9781558601468500321. 61, 155

V. G. Vovk. A logic of probability, with application to the foundations of statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55(2):317–351, 1993. URL https://www.jstor.org/stable/2346196. 155

A. Wald. Sequential tests of statistical hypotheses. *The annals of mathematical statistics*, 16(2):117–186, 1945. URL https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-16/issue-2/Sequential-Tests-of-Statistical-Hypotheses/10.1214/aoms/1177731118.full. 155

A. Wald. *Statistical decision functions*. Wiley, 1950. 5

J.-K. Wang, J. Abernethy, and K. Y. Levy. No-regret dynamics in the fenchel game: A unified framework for algorithmic convex optimization. *arXiv preprint arXiv:2111.11309*, 2021. URL https://arxiv.org/abs/2111.11309. 142

M. K. Warmuth and A. K. Jagota. Continuous and discrete-time nonlinear gradient descent: Relative loss bounds and convergence. In *Electronic proceedings of the 5th International Symposium on Artificial Intelligence and Mathematics*, volume 326, 1997. URL https://users.soe.ucsc.edu/~manfred/pubs/C45.pdf. 61

I. Waudby-Smith and A. Ramdas. Estimating means of bounded random variables by betting. *arXiv preprint arXiv:2010.09686*, 2021. URL https://arxiv.org/abs/2010.09686. 155

C.-Y. Wei, C.-W. Lee, M. Zhang, and H. Luo. Linear last-iterate convergence in constrained saddle-point optimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=dx11_7vm5_r. 143

L. Xiao. Dual averaging method for regularized stochastic learning and online optimization. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22, pages 2116–2124. Curran Associates, Inc., 2009. URL https://proceedings.neurips.cc/paper/2009/file/7cce53cf90577442771720a370c3c723-Paper.pdf. 88

L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/xiao10JMLR.pdf. 88

G. Zhang, Y. Wang, L. Lessard, and R. Grosse. Near-optimal local convergence of alternating gradient descent-ascent for minimax optimization. *arXiv preprint arXiv:2102.09468*, 2021. URL https://arxiv.org/abs/2102.09468. 142

T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proc. of International Conference on Machine Learning*, pages 919–926, New York, NY, USA, 2004. ACM. URL http://tongzhang-ml.org/papers/icml04-stograd.pdf. 23, 34

Z. Zhang, A. Cutkosky, and I. Paschalidis. PDE-based optimal strategy for unconstrained online learning. In *International Conference on Machine Learning*, pages 26085–26115. PMLR, 2022. URL https://proceedings.mlr.press/v162/zhang22d.html. 110

J. Zimmert and Y. Seldin. Tsallis-INF: An optimal algorithm for stochastic and adversarial bandits. *J. Mach. Learn. Res.*, 22:1–49, 2021. URL https://jmlr.csail.mit.edu/papers/v22/19-753.html. 88

M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proc. of the International Conference on Machine Learning*, pages 928–936, 2003. URL http://ra.adm.cs.cmu.edu/anon/usr0/ftp/usr/anon/2003/CMU-CS-03-110.pdf. 16

M. Zinkevich. *Theoretical guarantees for algorithms in multi-agent settings*. PhD thesis, School of Computer Science, Carnegie Mellon University, 2004. URL http://martin.zinkevich.org/publications/thesis.ps. 88

M. Zinkevich, M. Johanson, M. Bowling, and C. Piccione. Regret minimization in games with incomplete information. In *Advances in neural information processing systems*, volume 20, pages 1729–1736, 2007. URL https://proceedings.neurips.cc/paper/2007/file/08d98638c6fcd194a4b1e6992063e944-Paper.pdf. 132