Online Convex Optimization:
Algorithms, Learning, and Duality

Victor Sanches Portella

Dissertação apresentada
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
obtenção do título
de
Mestre em Ciências

Programa: Ciência da Computação
Orientador: Prof. Dr. Marcel Kenji de Carli Silva

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da CNPq

São Paulo, Junho de 2019

Online Convex Optimization:
Algorithms, Learning, and Duality

Esta versão da dissertação contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 03/05/2019. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof. Dr. Carlos Eduardo Ferreira - IME - USP
- Prof. Dr. Aritanan Borges Garcia Gruber - UFABC
- Prof. Dr. Carlos Henrique Cardonha - IBM Research

Abstract

Online Convex Optimization (OCO) is a field in the intersection of game theory, optimization, and machine learning which has been receiving increasing attention due to its recent applications to a wide range of topics such as complexity theory and graph sparsification. Besides the usually simple description and implementation of OCO algorithms, a lot of this recent success is due to a deepening of our understanding of the OCO setting and their algorithms by using cornerstone ideas from convex analysis and optimization such as the powerful results from convex duality theory.

In this text we present a mostly self-contained introduction to the field of online convex optimization. We first describe the online learning and online convex optimization settings, proposing an alternative way to formalize both of them so we can make formal claims in a clear and unambiguous fashion while not cluttering the readers understanding. We then present an overview of the main concepts of convex analysis we use, with a focus on building intuition. With respect to algorithms for OCO, we first present and analyze the Adaptive Follow the Regularized Leader (AdaFTRL) together with an analysis which relies mainly on the duality between strongly convex and strongly smooth functions. We then describe the Adaptive Online Mirror Descent (AdaOMD) and the Adaptive Dual Averaging (AdaDA) algorithms and analyze both by writing them as special cases of the AdaFTRL algorithm. Additionally, we show simple sufficient conditions for Eager and Lazy Online Mirror Descent (the non-adaptive counter-parts of AdaOMD and AdaDA) to be equivalent. We also present the well-known AdaGrad and Online Newton Step algorithms as special cases of the AdaReg algorithm, proposed by Gupta, Koren, and Singer, which is itself a special case of the AdaOMD algorithm. We conclude by taking a bird's-eyes view of the connections shown throughout the text, forming a "genealogy" of OCO algorithms, and discuss some possible path for future research.

Resumo

Otimização Convexa Online (OCO) é uma área na intersecção de teoria dos jogos, otimização e aprendizado de máquina que tem recebido maior atenção recentemente devido a suas recentes aplicações em uma grande gama de áreas como complexidade computacional e esparsificação de grafos. Além dos algoritmos de OCO usualmente terem descrições diretas e poderem ser implementados de forma relativamente simples, muito do recente sucesso da área foi possível graças a um melhor entendimento do cenário e dos algoritmos de OCO que se deu com uso de conhecidas ideias de análise e otimização convexa como a poderosa teoria de dualidade convexa.

Nesse texto nós apresentamos uma introdução (em sua maioria auto-contida) à área de otimização convexa online. Primeiro, descrevemos os cenários de aprendizado online e de otimização convexa online, propondo uma forma alternativa de formalizar ambos os modelos de forma que conseguimos enunciar afirmações claras e formais de forma que não atrapalha o entendimento do leitor. Nós então apresentamos um resumo dos principais conceitos e resultados de análise convexa que usamos no texto com um foco em criar intuição sobre os mesmos. Com relação a algoritmos para OCO, nós começamos apresentando o algoritmo *Adaptive Follow the Regularized Leader (AdaFTRL)* e analisamos sua eficácia com um resultado sobre a dualidade de funções *strongly convex* e *strongly smooth*. Na sequência, descrevemos os algoritmos *Adaptive Online Mirror Descent (AdaOMD)* e *Adaptive Dual Averaging (AdaDA)*, analisando a eficácia de cada um escrevendo eles como instâncias do algoritmo AdaFTRL. Além disso, nós mostramos condições simples para que as versões *Eager* e *Lazy* do *Online Mirror Descent* (que são as versões não adaptativas do AdaOMD e do AdaDA, respectivamente) sejam equivalentes. Também apresentamos os algoritmos *AdaGrad* e *Online Newton Step*, bem conhecidos na literatura sobre OCO, como casos especiais do algoritmo *AdaReg*, esse último um algoritmo proposto por Gupta, Koren, and Singer, que, por sua vez, é um caso especial do algoritmo AdaOMD. Nós concluímos o texto com uma visão global das conexões entre os algoritmos que mostramos durante o texto, formando uma “genealogia” de algoritmos para OCO, além de discutirmos possíveis direções futuras de pesquisa.

Acknowledgements

Although it is impossible to fully express how grateful I am for the people that helped me during my master's, I cannot say this text is complete without thanking them.

First and foremost, thanks to my family for their unconditional support even in the darkest and most difficult moments: my mother Solange Sanches, my father José Alexandre Portella, my brother Daniel Sanches Portella, and my sister Bianca Sanches Portella. I hold you all dearly in my heart. I would like to thank my mom specially for hearing me unreservedly and lovingly every time that I needed.

I cannot fully express in anyway how grateful I am for having as my advisor Professor Marcel Kenji de Carli Silva. I was able to fully undestand the meaning of the expression "drinking from a firehose" under his supervision. Looking back I can appreciate how much I have learned about computer science and mathematics due to him. However, and maybe more importantly, I am thankful for the many things he did that transcended his duties as an advisor. If I ever become an advisor to any student, I hope I can do for them at least a fraction of what Professor Marcel has done for me.

Another person which I owe a lot is Professor Aritanan Gruber. I will be eternally grateful for his willingness to help me, in academic matters and otherwise. His passion for mathematics and computer science is intoxicating and he certainly helped me make many of the decisions I have made up to this point.

I was also fortunate to have met Professors Marco Molinaro and Carlos Cardonha, who helped me a lot in my PhD applications, even though they had met me only on my qualification exam. Without their help I would not have been able to enjoy the many opportunities that were presented to me and, for that and much more, I shall be eternally grateful.

My friends offered me incredible support and affection, filling my life with moments of joy and playfulness. Thank Mateus Barros for being one of the best friends I could wish for, for helping me have many moments of joy, and for cutting by half the number of typos in this text. Thanks Renato Cordeiro for your truthfully gold friendship and for all our deep conversations about life, the universe and everything else. Thanks Karina Awoki for being a short friend with an enormous heart who helps me with many of my personal dilemmas. Thanks Ruan Costa and Gervásio Santos for being with me and for being supportive even after I almost killed us back in Australia. Thanks Vinicius Vendramini for being extremely supportive and for always being there for your friends. Thanks Ludmila Ferreira for always being a freehearted, spontaneous, and extremely joyful friendship. Thanks Leonardo Contador for being with me since my childhood and for, even after so many years, being such a close and good friend who I see as a brother of mine.

Finally, I would like to thank Ana Caroline de França. Writing your name made me smile in the same way I do when I see you smile. Thanks for this and much more.

Contents

Chapter 1

Introduction and Preliminaries

Consider the following scenario: a spam filter receives emails one at a time and needs to classify each of them as spam or not as soon as they arrive. After that, a user checks whether the filter's classification is right or wrong, penalizing it in the latter case. This problem can be modeled into the *online learning* setting [67, 68].

An online learning problem is roughly a game made of multiple rounds in which three entities participate: nature, player, and adversary/enemy. At the beginning of each round, nature reveals a query to both the player and the adversary. Then, the player tries to predict the correct answer to nature's query, while the adversary simultaneously picks the correct answer. At the end of the round, the enemy's and player's choices are revealed to each other. The player then suffers a loss based on the quality of her prediction compared to the enemy's answer. In the spam filtering problem described earlier, for example, nature simply reveals, at each round, one email. The player (the spam filter) classifies the email. The adversary (the user assessing the email's classification) then picks the correct classification for the email of the current round, and the filter is penalized at the end of the round if it misclassified the email.

In online learning problems we are usually interested in devising strategies for the player which minimize, in some sense, the player's cumulative loss throughout the game. Different from the field of statistical learning [50, 71], in which we assume that the pairing of nature's queries and enemy's answers are governed by a fixed probability distribution which is unknown to the player, in online learning problems the enemy may be adversarial to the player. That is, the adversary may have the clear intention of maximizing the player's cumulative loss. As expected, it is impossible for the player to minimize her cumulative loss against adversarial enemies. This holds since an enemy who knows the player's strategy can, at each round, assign a loss of 1 to the player's prediction while associating a loss of 0 to all other possible choices the player could have picked. Instead, we measure the quality of the player's strategy through the notion of *regret* borrowed from game theory. Given a set $\mathcal{H}$ of functions from queries to possible player predictions, the regret measures how sorry the player is for not using the best function in $\mathcal{H}$ in hindsight. When looking at the regret, we are usually interested in strategies for the player which have worst-case regret which grows sub-linearly in the number of rounds of the game. That is, the "average regret" of the player goes to zero as the number of rounds goes to infinity.

Even though the notion of regret seems easier to handle and minimize, it is impossible for the player to attain sub-linear regret already in simple online learning problems [26]. There is, however, a special case of online learning problems in which it is usually possible to attain sub-linear regret: online convex optimization [19, 36, 67]. This framework models multi-round games with two entities: a player and an enemy. At each round, the player has to pick a point $x \in X$ from a convex set X in an euclidean space while the enemy simultaneously picks a convex function $f\colon X \to \mathbb{R}$ and, at the end of the round, the player suffers a loss of $f(x)$. It turns out that the convexity assumption is not very restrictive, making the framework still able to model a wide range of problems. Maybe more importantly, the design and analysis of strategies for the player benefits greatly from the classical field of convex analysis and optimization, while still using ideas from game theory and statistical learning. Recently, the powerful duality theory from convex analysis has driven rapid and sustained progress in the field of online convex optimization. Looking at the algorithms from online convex optimization through the lens of convex duality theory is helping to better understand their inner workings, unifying analyses of previously unrelated algorithms, and aiding in the design of brand new algorithms [33, 48, 69].

As already mentioned, we may still model many problems from the online learning setting as online convex optimization problems, even if sometimes we may need to allow the player to randomize her choices (see, for example, [67, Section 1.2.2]). Online convex optimization has been drawing the attention of the theoretical computer science community recently since its algorithms have been finding applications in areas unrelated to learning such as graph sparsification [3, 20], almost linear-time algorithms for the approximate maximum flow problem [25], a efficient method for some semidefinite programs [7], and computational complexity theory [12, 40]. One of the driving forces behind many of these application is the recent *Big Data* phenomenon. It refers to the growing need of devising algorithms that handle huge amounts of data that make even quadratic algorithms impractical. In such applications, it is usually acceptable to compute only approximate solutions if they can be obtained very efficiently. A deep understanding of the inner workings of algorithms for online convex optimization may reveal more interesting applications or even allow us to improve existing results.

In this text we present a mostly self-contained introduction to the field of online convex optimization, mainly based on the works of Shalev-Shwartz [67], Hazan [36], McMahan [48], Bubeck [18, 19], and Gupta, Koren, and Singer [33], with a focus on looking at and analyzing algorithms through the lens of convex duality. To do so, after introducing and discussing the online learning and online convex optimization frameworks, we give a brief overview of the convex analysis concepts we use, with a focus on building intuition instead of proving the stated results. We then start to present algorithms for online convex optimization, with a focus on the perspective of convex analysis duality. This allows us to present unified analyses of many algorithms and to show interesting connections among them. Moreover, in our presentation we propose an alternative way of formalizing the online learning and online convex optimization frameworks, which allows us to make formal claims in a clear, unambiguous, and hopefully transparent way. At the same time, we have tried to make our presentation familiar to anyone acquainted with the basics of online convex optimization so that the formalization does not clutter one's understanding of the content of the text. In particular, our formalization aids us in stating and proving equivalence between many algorithms, which culminates in a kind of "genealogy" among online convex optimization algorithms, which we present and discuss on Chapter 7.

1.1 Notation and Preliminaries

In this section we collect basic (and usually standard) notation and results (without proofs) which will be used throughout the text. One may skip most of this section, using it only as reference when needed.

Throughout the text, we use **bold** words for formal definitions, and *italic* words for loose and usually non-formal definitions, or sometimes simply for emphasis. For example, let us define the

Iverson bracket: if P is a predicate, we set

$$[P] := \begin{cases} 1 & \text{if } P \text{ is true,} \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, if P is false, then $[P]$ is **strongly zero**, that is, any expression multiplied by $[P]$ is 0, even in the case where the expression is invalid or undefined. For example, if we set $f(\alpha) := [\alpha \neq 0]\alpha^{-1}$ for every $\alpha \in \mathbb{R}$, we have $f(0) = 0$ even though x^{-1} is not defined for $x = 0$.

Most of our notation is collected on Tables 1.1 to 1.3. Still, similarly to the case of the Iverson bracket, some of the notation on these tables will be more carefully defined on the remainder of this section.

Table 1.1: Basic Notation

$[P]$	$:=$	1 if the predicate P is true, and (strongly) zero otherwise
X^Y	$:=$	the set of functions from the set Y to the set X
X^n	$:=$	$X^{[n]}$ for every $n \in \mathbb{N}$; note that $X^0 = \{\varnothing\}$
$X + Y$	$:=$	$\{\, x + y : x \in X, y \in Y \}$ for any subsets X, Y of an Euclidean space
αX	$:=$	$\{\, \alpha x : x \in X \}$ for every $\alpha \in \mathbb{R}$ and every subset X of an Euclidean space
$\oplus$	$:=$	the direct sum of two vectors or two sets of vectors
$\mathbb{1}$	$:=$	properly sized real vector with all entries equal to 1
e_i	$:=$	characteristic vector of i on the implicit from the context set $\mathbb{R}^E$ (i.e., $e_i(j) := [j = i]$ for each $j \in E$)
$\mathrm{Diag}(x)$	$:=$	diagonal matrix with $\mathrm{Diag}(x)_{i,i} := x_i$ for each possible index i
$\mathrm{diag}(X)$	$:=$	vector with $\mathrm{diag}(X)_i = X_{i,i}$ for each possible index i

Table 1.2: Frequently Used Sets

$[n]$	$:=$	$\{1, \dots, n\}$ for each $n \in \mathbb{N}$
$\mathbb{E}$	$:=$	fixed euclidean space with inner product $\langle \cdot, \cdot \rangle$
$\mathbb{R}_+$	$:=$	$\{\, \alpha \in \mathbb{R} : \alpha \geq 0\}$
$\mathbb{R}_{++}$	$:=$	$\{\, \alpha \in \mathbb{R} \setminus \{0\} : \alpha \geq 0\}$
Δ_E	$:=$	$\{\, x \in [0,1]^E : \mathbb{1}^\mathsf{T} x = 1\}$, the simplex in $\mathbb{R}^E$, where E is a finite set
$\mathbb{R}^{n \times m}$	$:=$	$\mathbb{R}^{[n] \times [m]}$, that is, the set of real $n \times m$ matrices
$\mathbb{S}^d$	$:=$	set of symmetric $d \times d$ matrices
$\mathbb{S}^d_+$	$:=$	set of positive semidefinite $d \times d$ matrices
$\mathbb{S}^d_{++}$	$:=$	set of positive definite $d \times d$ matrices
$\mathcal{S}_d$	$:=$	$\{\, X \in \mathbb{S}^d_+ : \mathrm{Tr}(X) = 1\}$, the spectraplex on $\mathbb{S}^d$

1.1.1 Sequences

For any nonempty set X, define $\mathrm{Seq}(X) := \bigcup_{n=0}^{\infty} X^n$ and define by **sequences** the elements of $\mathrm{Seq}(X)$. Let $T \in \mathbb{N}$ and let X be a nonempty set. Throughout the text, we will stick to the

Table 1.3: Sequence Notation

$\mathrm{Seq}(X)$	$\coloneqq$	$\bigcup_{n=0}^{\infty} X^n$
$\langle\rangle$	$\coloneqq$	the empty sequence
$\langle x_1, \dots, x_t\rangle$	$\coloneqq$	sequence of length t whose k-th element is x_k for each $k \in [t]$
$\langle x_i, \dots, x_j\rangle$	$\coloneqq$	$\langle\rangle$ if $i > j$, the sequence whose k-th element is x_{i+k-1} for each $k \in [j-i+1]$ otherwise
$\boldsymbol{x}_{i:j}$	$\coloneqq$	$\langle x_i, \dots, x_j\rangle$

convention of using bold letters for sequences, even though we use the un-bolded letter with an index to denote the sequence elements. Formally, for any $\boldsymbol{x} \in X^T \subseteq \mathrm{Seq}(X)$ (that is, a sequence of elements from X of size T), we set $x_i \coloneqq \boldsymbol{x}_i$ for each $i \in [T]$. Moreover, we use angle brackets to write sequences in full, that is, for every $\boldsymbol{x} \in X^T$ and $i, j \in [T]$ with $i \leq j$, we have that $\langle x_i, \dots, x_j\rangle$ denotes the sequence whose $j - i + 1$ elements are, in order[1], $x_i, x_{i+1}, \dots, x_j$. Finally, $\langle\rangle$ denotes the empty sequence, and for any $\boldsymbol{x} \in X^T$ and $i, j \in \mathbb{Z}$ with $j < i$ or such that $j < 0$, define $\langle x_i, \dots, x_j\rangle \coloneqq \langle\rangle$. We also denote by $\boldsymbol{x}_{i:j}$ the sequence $\langle x_i, \dots, x_j\rangle$ for any $\boldsymbol{x} \in \mathrm{Seq}(X)$ and $i, j \in \mathbb{Z}$. Most of the notation for sequences is listed on Table 1.3.

1.1.2 Probability

In some parts of the text, we will use a bit of probability theory. Thus, let us define the basic concepts and objects of probability theory to ensure the reader follows our notation. We assume the reader is acquainted with basic probability theory.

Let Ω be a set, and let $\Sigma \subseteq 2^{\Omega}$, where 2^{Ω} denotes the power set of Ω. The set Σ is a **σ-algebra** on Ω if

(i) $\Omega \in \Sigma$,

(ii) for every $E \in \Sigma$ we have $\Omega \setminus E \in \Sigma$,

(iii) If $\{E_i\}_{i=0}^{\infty}$ is countable and such that $E_i \in \Sigma$ for every $i \in \mathbb{N}$, then $\bigcup_{i=0}^{\infty} E_i \in \Sigma$.

Moreover, if Σ is a σ-algebra, we say that (Ω, Σ) is a **measurable space**. We may refer simply to Ω as a measurable space when Σ is made clear by the context. Finally, for any set $\mathcal{O} \subseteq 2^{\Omega}$, define the σ-algebra **generated** by $\mathcal{O}$ (on Ω) by

$$\sigma(\mathcal{O}) \coloneqq \bigcap \{\, \mathcal{M} \subseteq 2^{\Omega} : \mathcal{O} \subseteq \mathcal{M} \text{ and } \mathcal{M} \text{ is a } \sigma \text{ algebra on } \Omega\}.$$

For any set $\mathcal{O} \subseteq 2^{\Omega}$, the above set is indeed a σ-algebra with $Ocal \subseteq \sigma(\mathcal{O})$ (see [65, Theorem 1.10]).

Many of the spaces which we have to handle are *topological spaces*[2], which have natural σ-algebras which are usually associated to them, the *Borel σ-algebras*. Formally, let $(X, \mathcal{O})$ be a topological space. Then the **Borel σ-algebra** on $(X, \mathcal{O})$ is the σ-algebra $\sigma(\mathcal{O})$. The Borel σ-algebra on $(X, \mathcal{O})$ may be denoted by $\mathcal{B}(X)$ is the topology $\mathcal{O}$ is clear from the context. For example, the

[1] Note that, by the definition of sequence, the indexes of the elements of $\langle x_i, \dots, x_j\rangle$ range from 1 to $i - j + 1$. Therefore, for any $k \in \{1, \dots, j - i + 1\}$, the k-th element of $\langle x_i, \dots, x_j\rangle$ is x_{i+k-1}.

[2] that is, are sets X equipped with a *topology* $\mathcal{O} \subseteq 2^X$ on X (also know as the open sets of the topological space $(X, \mathcal{O})$). To say that $\mathcal{O}$ is a topology on X means that $\mathcal{O}$ contains both $\varnothing$ and X, is closed under arbitrary unions, and closed under finite intersections. We do not expand on this subject since it is not the focus of this section and neither of the text, but the interested reader may see [65, Definitions 2.3] and the discussion and results that follow it.

Borel σ-algebra on $\mathbb{R}$ (equipped with the usual topology) is the σ-algebra generated by the set $\mathcal{O}_{\mathbb{R}} \coloneqq \{\, (a,b) \subseteq \mathbb{R} : a,b \in \mathbb{R}, a \leq b\}$ of open intervals on $\mathbb{R}$, where $(a,b) \coloneqq \{\, x \in \mathbb{R} : a < x < b\}$ for any $a,b \in \mathbb{R}$. More generally, the Borel σ-algebra on $\mathbb{R}^d$ (equipped with the usual topology) is the σ-algebra generated by the set of open rectangles on $\mathbb{R}^d$, that is,

$$\{\, l_1 \times l_2 \times \cdots \times l_d \subseteq \mathbb{R}^d : l_i \in O_{\mathbb{R}} \text{ for each } i \in [d]\}.$$

Throughout the text we assume that $\mathbb{R}$ and $\mathbb{R}^d$ are each equipped with their respective Borel σ-algebras.

Before jumping to probability spaces, it is worth talking about unions and Cartesian products of measurable spaces since we shall face some of these cases when talking about probability on sequences as defined on the previous section. Let $(B_1, \mathcal{B}_1), \ldots, (B_k, \mathcal{B}_k)$ be measurable spaces. Then, unless stated otherwise, we equip the set $B_1 \times \cdots \times B_k$ with the σ-algebra generated $\sigma(\mathcal{B}_1 \times \cdots \times \mathcal{B}_k)$. Measurable spaces of the latter kind are called **product spaces** (see [35, Section 37] and [8, Section 2.6] for details and discussion on product spaces). Thus, if $(B, \mathcal{B})$ is a measurable space, then the product space $(B^n, \sigma(\mathcal{B}^n))$ is also a measurable space. Finally, if $(B, \mathcal{B})$ is a measurable space, then

$$\sigma\Big(\bigcup_{n=0}^{\infty} \sigma(\mathcal{B}^n)\Big) \tag{1.1}$$

is a σ-algebra on $\operatorname{Seq}(B)$. Thus, unless stated otherwise, we equip the above σ-algebra to $\operatorname{Seq}(B)$.

Let us now define probability spaces. A **probability space** is a triple $(\Omega, \Sigma, \mathbb{P})$, where

(i) Ω is a set, called **sample space**,

(ii) $\Sigma \subseteq 2^{\Omega}$ is a σ-algebra on Ω whose elements are called **events**,

(iii) $\mathbb{P} \colon \Sigma \to [0,1]$ is a function, called **probability measure** or **probability distribution**, such that $\mathbb{P}(\Omega) = 1$, $\mathbb{P}(\varnothing) = 0$, and such that it is countably additive, that is, if $\{E_i\}_{i=0}^{\infty}$ is a countable family of pairwise disjoint events from Σ, then

$$\mathbb{P}\Big(\bigcup_{i=0}^{\infty} E_i\Big) = \sum_{i=1}^{\infty} \mathbb{P}(E_i).$$

If (Ω, Σ) is a measurable space, then a **probability distribution** on (Ω, Σ) is a function $\mathbb{P} \colon \Omega \to \mathbb{R}_+$ such that $(\Omega, \Sigma, \mathbb{P})$ is a probability space. Let $(\Omega, \Sigma, \mathbb{P})$ be a probability space and let $(B, \mathcal{B})$ be a measurable space. We say that a function $X \colon \Omega \to B$ is a **random variable** if X is a **measurable function** (w.r.t. the measurable spaces (Ω, Σ) and $(B, \mathcal{B})$), that is, if for every $F \in \mathcal{B}$ we have $X^{-1}(F) \coloneqq \{\, \omega \in \Omega : X(\omega) \in F\} \in \Sigma$. In particular, for a function from a measurable space (Ω, Σ) to $\mathbb{R}$ to be a random variable it needs to be measurable w.r.t. (Ω, Σ) and $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$. If $(B, \mathcal{B})$ is a measurable space and $X \colon \Omega \to B$ is a random variable, for every $b \in B$ and $F \in \mathcal{B}$ we set

$$\mathbb{P}(X = b) \coloneqq \mathbb{P}(X^{-1}(\{b\})) \qquad \text{and} \qquad \mathbb{P}(X \in F) \coloneqq \mathbb{P}(X^{-1}(F)).$$

One fact that is tirelessly used in probability theory is that, if $\mathcal{D}$ is a probability distribution on a measurable space $(B, \mathcal{B})$, then there are[3] a probability space $(\Omega', \Sigma', \mathbb{P}')$ and a random variable $X \colon \Omega' \to B$ such that $\mathbb{P}'(X \in F) = \mathcal{D}(F)$ for every $F \in \mathcal{F}$. In this case, we write $X \sim \mathcal{D}$ or we say that X **follows** the probability distribution $\mathcal{D}$. We say that a function $L \colon \Omega \to \mathbb{R}$ is **Borel-measurable** if it is measurable w.r.t. the Borel σ-algebra on $\mathbb{R}$. We say that events $A, B \in \Sigma$ are

[3] Namely, the probability space $(B, \mathcal{B}, \mathcal{D})$ and the random variable given by $X(b) \coloneqq b$ for every $b \in B$.

independent events if $\mathbb{P}(A \cap B) = \mathbb{P}(A)\mathbb{P}(B)$. Moreover, two random variables $X \colon \Omega \to A$ and $Y \colon \Omega \to B$, where $(A, \mathcal{A})$ and $(B, \mathcal{B})$ are measurable spaces, are **independent** if, for every $E \in \mathcal{A}$ and $F \in \mathcal{B}$ we have

$$\mathbb{P}(X \in E, Y \in F) \coloneqq \mathbb{P}(X^{-1}(E) \cap Y^{-1}(F)) = \mathbb{P}(X \in E)\mathbb{P}(Y \in F).$$

Let $(\Omega, \Sigma, \mathbb{P})$ be a probability space, let $(B_1, \mathcal{B}_1), \dots, (B_k, \mathcal{B}_k)$ be measurable spaces, and let $X_1, \dots, X_k$ be random variables on $(\Omega, \Sigma, \mathbb{P})$ with $X_i \colon \Omega \to B_i$ for each $i \in [k]$. If $(U, \mathcal{U})$ is a measurable space and $f \colon B_1 \times \ldots \times B_k \to U$ is measurable w.r.t. $\sigma(\mathcal{B}_1 \times \ldots \times \mathcal{B}_k)$ and $\mathcal{U}$, then we abuse notation and define the random variable

$$[f(X_1, \dots, X_k)](\omega) \coloneqq f(X_1(\omega), \dots, X_k(\omega)), \qquad \forall \omega \in \Omega.$$

We will avoid this notation when it may cause confusion.

Moreover, the **expectation** of a random variable $X \colon \Omega \to \mathbb{R}$ is the Lebesgue integral[4]

$$\mathbb{E}[X] \coloneqq \int X(\omega) d\mathbb{P}(\omega)$$

when it exists. If the above integral is well-defined for the random variable X, then we say that X is **Lesbegue integrable** (w.r.t. $\mathcal{P}$). Interestingly (and more easily understandable), if $X \colon \Omega \to \mathbb{R}$ is a random variable and there is a finite set $\mathcal{I} \subseteq \mathbb{R}$ such that $\sum_{x \in \mathcal{I}} \mathbb{P}(X = x) = 1$, then

$$\mathbb{E}[X] = \sum_{x \in \mathcal{I}} x\mathbb{P}(X = x).$$

1.1.3 Linear Algebra Results

Let us now define some of the notation and state some basic results we use about concepts from linear algebra. We still suppose the reader is reasonably acquainted with linear algebra. Thus, we do not aim to be thorough in our exposition, and we only define the most used concepts and we state (and sometimes prove) only some results we use on the text.

For any $a, b \in \mathbb{R}^d$ and $\circ \in \{\leq, \geq, =, <, >\}$, we shall write $a \circ b$ when $a_i \circ b_i$ for each $i \in [d]$. Let $A \in \mathbb{R}^{m \times d}$. If $m = d$, then A is **square**, and if $A = A^{\mathsf{T}}$, then A is **symmetric**. We denote the set of all symmetric $d \times d$ matrices by $\mathbb{S}^d$. If A is square and $A_{i,j}$ is zero for every distinct $i, j \in [d]$, then A is **diagonal**. Finally, the following notation will be useful:

- for any vector $x \in \mathbb{R}^d$, $\operatorname{Diag}(x) \in \mathbb{S}^d$ is a diagonal matrix with diagonal entries given by $\operatorname{Diag}(x)_{i,i} \coloneqq x_i$ for each $i \in [d]$, and
- for any matrix $X \in \mathbb{R}^{d \times d}$, $\operatorname{diag}(X) \in \mathbb{R}^d$ is defined by $\operatorname{diag}(X)_i \coloneqq X_{i,i}$ for each $i \in [d]$ and it is called the **diagonal** of X.

Throughout the text, we denote by I the **identity matrix**, a properly sized diagonal matrix with $I_{i,j} = [i = j]$ for any i, j in the set of possible indices, that is, the identity matrix has one in each diagonal entry and zeroes everywhere else. The **trace** of $A \in \mathbb{R}^{d \times d}$ is $\operatorname{Tr}(A) \coloneqq \sum_{i=1}^{d} A_{i,i}$, and one may verify that

- $\operatorname{Tr}(AB) = \operatorname{Tr}(BA)$ for real matrices A, B such that the products AB and BA are defined,

[4] We skip the definition of Lesbegue integral for the sake of brevity, but one can find the definition and in-depth discussions in

- $(A, B) \in \mathbb{S}^d \times \mathbb{S}^d \mapsto \operatorname{Tr}(AB)$ is an inner product on $\mathbb{S}^d$, and
- $\mathbb{S}^d$ equipped with the above inner product is an euclidean space.

We denote by $\det(A)$ the **determinant** of A (see [39, Section 0.3] for a proper definition and discussion). We will use, without giving a proof, the following properties of the determinant:

- $\det(AB) = \det(A)\det(B)$ for matrices $A, B \in \mathbb{R}^{d \times d}$,
- $\det(A) = \det(A^\mathsf{T})$ for $A \in \mathbb{R}^{d \times d}$, and
- $\det(\operatorname{Diag}(x)) = \prod_{i=1}^d x_i$ for any $x \in \mathbb{R}^d$.

Let $A \in \mathbb{S}^d$. If $x^\mathsf{T} A x \geq 0$ for every $x \in \mathbb{R}^d$, denoted by $A \succeq 0$, then A is **positive semidefinite**, and if $x^\mathsf{T} A x > 0$ for every $x \in \mathbb{R}^d \setminus \{0\}$, denoted by $A \succ 0$, then A is **positive definite**. Moreover, for any $B \in \mathbb{S}^d$ we write $A \succeq B$ if $A - B \succ 0$ and we write $A \succ B$ if $A - B \succ 0$. Define

$$\mathbb{S}^d_+ \coloneqq \{\, X \in \mathbb{S}^d : X \succeq 0 \} \qquad \text{and} \qquad \mathbb{S}^d_{++} \coloneqq \{\, X \in \mathbb{S}^d : X \succ 0 \}.$$

The **eigenvalues** of A are the d roots of the polynomial $\lambda \in \mathbb{R} \mapsto \det(\lambda I - A)$ (which are all real since A is symmetric and real, see [39, Theorem 4.1.3]). If $\lambda \in \mathbb{R}$ is an eigenvalue of A, then any $v \in \mathbb{R}^d \setminus \{0\}$ such that $Av = \lambda v$ is said to be an **eigenvector** of A (associated with λ). The function $\lambda^\uparrow \colon \mathbb{S}^d \to \mathbb{R}^d$ extracts the eigenvalues of a matrix in non-decreasing order. We say that a matrix $Q \in \mathbb{R}^{d \times d}$ is **orthogonal** if $Q^{-1} = Q^\mathsf{T}$. Note that, if $Q \in \mathbb{R}^{d \times d}$ is an orthogonal matrix, then

$$1 = \det(I) = \det(QQ^\mathsf{T}) = \det(Q)\det(Q^\mathsf{T}) = \det(Q)^2.$$

That is, the determinant of any orthogonal matrix is either 1 or -1. To conclude, let us state some results we use in future chapters.

Theorem 1.1.1 (Spectral Decomposition Theorem, see [39, Theorem 4.1.5])**.** *If $A \in \mathbb{S}^d$, then there exists an orthogonal matrix $Q \in \mathbb{R}^{d \times d}$ such that $A = Q \operatorname{Diag}(\lambda^\uparrow(A)) Q^\mathsf{T}$.*

Corollary 1.1.2. *If $A \in \mathbb{S}^d$, then $\operatorname{Tr}(A) = \mathbb{1}^T \lambda^\uparrow(A)$ and $\det(A) = \prod_{i=1}^d \lambda_i^\uparrow(A)$.*

Proof. Define $\Lambda \coloneqq \operatorname{Diag}(\lambda^\uparrow(A))$. By Theorem 1.1.1, there is an orthogonal matrix $Q \in \mathbb{R}^{d \times d}$ such that $A = Q \Lambda Q^\mathsf{T}$. Then

$$\operatorname{Tr}(A) = \operatorname{Tr}(Q \Lambda Q^\mathsf{T}) = \operatorname{Tr}(Q^\mathsf{T} Q \Lambda) = \operatorname{Tr}(\Lambda) = \mathbb{1}^\mathsf{T} \lambda^\uparrow(A)$$

and

$$\det(Q \Lambda Q^\mathsf{T}) = \det(Q)\det(\Lambda)\det(Q^\mathsf{T}) = \det(\Lambda) = \prod_{i=1}^d \lambda_i^\uparrow(A). \qquad \square$$

Theorem 1.1.3 ([39, Theorem 4.1.10])**.** *Let $A \in \mathbb{S}^d$. Then $A \succeq 0$ if and only if $\lambda^\uparrow(A) \geq 0$ and $A \succ 0$ if and only if $\lambda^\uparrow(A) > 0$.*

Let $A \in \mathbb{S}^d_+$. A matrix $A^{1/2} \in \mathbb{S}^d_+$ is a **square root** of A if $(A^{1/2})^2 = A$. The next proposition shows that such a matrix is unique, and it shows how to construct it from the spectral decomposition of the matrix.

Proposition 1.1.4 ([39, Theorem 7.2.6])**.** *Let $A \in \mathbb{S}^d_+$. Then A has a unique square root matrix $A^{1/2} \in \mathbb{S}^d_+$. Moreover, if $A = Q \operatorname{Diag}(\lambda^\uparrow(A)) Q^T$, where $Q \in \mathbb{R}^{d \times d}$ is an orthogonal matrix, then $A^{1/2} = Q \operatorname{Diag}(\mu) Q^T$, where $\mu \in \mathbb{R}^d$ is defined by $\mu_i \coloneqq \lambda_i^\uparrow(A)^{1/2}$ for each $i \in [d]$.*

Lemma 1.1.5 ([39, Corollary 7.7.4])**.** *Let $A, B \in \mathbb{S}^d_+$ be such that $A \succeq B$. Then $A^{1/2} \succeq B^{1/2}$.*

Chapter 2

Online Learning and Online Convex Optimization

Roughly, online learning is a setting where three entities, which we call nature, player, and enemy, play a multi-round game. At each round, nature begins by revealing a query. Then, simultaneously, the player picks her guess of the answer and the enemy picks the "true answer" of the query. At the end of the round, the player suffers a loss based on how bad her prediction was when compared to the true answer given by the enemy. This seemingly simple setting has many applications in fields such as machine learning [68], optimization [19], and game theory [24]. One of the main aspects of this setting which makes it so useful and which distinguishes it from the classical statistical learning setting [71] is the lack of any statistical assumptions on the entities playing the game. In particular, the enemy in the online learning setting can be adversarial to the player, trying to maximize her cumulative loss. As expected, this setting in its full generality is too hard for the player. For that reason we look at the special case of *online convex optimization*, where two entities, player and enemy, play a multi-round game. At each round of this game the player picks a point x from a convex set X in an euclidean space $\mathbb{E}$, and the enemy simultaneously picks a convex function $f\colon \mathbb{E} \to (-\infty, +\infty]$ from a fixed set $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$. At the end of the round, the player suffers a loss given by $f(x)$. In later chapters we will see that there are plenty of interesting algorithms for the player which, under mild assumptions, guarantee low *regret*, where the latter is a performance measure borrowed from game theory which we define later. In this chapter we will see that many interesting problems from the online learning setting can be solved either by directly modeling it as an online convex optimization instance, or by using algorithms from the latter as subroutines.

It is worth noting that our presentation of the online learning and online optimization settings is slightly unusual if compared to the current literature, such as of the surveys [19, 36, 67]. Our goal with our presentation is to leave no room for doubt about the information which each entity has access to at any moment, and to be able to make precise and formal statements about these settings. At the same time, we try to make the presentation familiar enough for the existing community and for those familiar with optimization so that the formalism does not clutter one's understanding.

On Section 2.1, we define the online learning setting, and look at the simpler realizable case with a finite hypothesis set. On Section 2.2 we define and motivate many classic problems which fit the online learning setting. On Section 2.3 we argue that it is uninformative to look at the raw loss of the player on the online learning setting. We introduce the notion of *regret* as an alternative measure of player quality, and we show that it is impossible to attain low regret already in simple online learning problems. On Section 2.4 we compare the online learning to the statistical learning settings, and show how to use the former to solve tasks of the latter. On Section 2.5 we define the online (convex) optimization setting, we show that it is a special case of online learning, and we look at some problems from the online learning setting which can be modeled as online convex optimization problems. On Section 2.6 we look at techniques to devise player strategies for the online learning setting given that we have good algorithms for some online convex optimization problems. Finally, on Section 2.7 we further discuss the notion of regret, its intuition, and we present the idea of *policy regret*.

2.1 The Online Learning Setting

Online Learning is a setting which models problems where a forecaster/player has to sequentially predict a competitor's answers to a sequence of queries by an entity called nature (not yet mentioned here). We will usually call such a competitor as an enemy since it will often be adversarial to the player. Immediately after each of her choices, the player suffers some kind of loss which depends on the accuracy of her prediction. As expected, the goal of the player is to attain the lowest possible, in some sense, cumulative loss throughout the game. A classic example is the spam filtering problem, where the a spam filter receives, sequentially, emails which need to be classified as spam or non-spam. Immediately after each email classification, a penalty is charged on the filter in case of misclassification. This example is illustrative because it helps to intuitively justify one of the main peculiarities of this setting: there are no statistical assumptions over the sequence of answers picked by the adversary. Actually, we will usually be interested in the performance of the player against adversarial enemies, that is, competitors which pick answers/points[1] with the clear goal of maximizing the players cumulative loss. This makes sense in the spam filtering example, since spam creators evolve with time, actively trying to bypass detection by the spam filter. Let us formally define this setting before continuing our discussion. The following definitions use the notation of *sequences* extensively, which are formally defined in Section 1.1.

Definition 2.1.1 (Online learning instance)**.** An **online learning** (OL) **instance** is a quadruple (X, D, Y, L), where X, D, and Y are arbitrary sets which we call the **query**, **decision**, and **label sets** of the instance, respectively, and $L\colon D \times Y \to \mathbb{R}$ is a function, which we call the **loss function** of the instance.

Let $\mathcal{P} := (X, D, Y, L)$ be an online learning instance. We associate with $\mathcal{P}$ the function $\mathrm{OL}_{\mathcal{P}}$, which receives the following parameters:

- $\mathrm{NATURE}\colon \mathbb{N} \to X$, which we call **nature oracle**;
- $\mathrm{PLAYER}\colon \mathrm{Seq}(X) \times \mathrm{Seq}(Y) \to D$, which we call **player oracle**;
- $\mathrm{ENEMY}\colon \mathrm{Seq}(X) \times \mathrm{Seq}(D) \to Y$, which we call **enemy oracle**;
- $T \in \mathbb{N}$, which we call the number of **rounds** or **iterations**.

It is worth warning that player and enemy oracles will usually not be defined for pairs of sequences of arbitrary size. Instead, they will be defined only for pairs of sequences with size which may appear on the online learning setting that we define later on. Still, we say that a player (or enemy) oracle is a function from $\mathrm{Seq}(X) \times \mathrm{Seq}(Y)$ (or $\mathrm{Seq}(X) \times \mathrm{Seq}(D)$) to D (or Y) for convenience.

Define $\mathrm{OL}_{\mathcal{P}}$ in an iterative way as in Algorithm 2.1. It is worth noting that, for $t \in \mathbb{N} \setminus \{0\}$, we consider to be the t-th *round* the iteration of Algorithm 2.1 in which are defined the t-th elements of

[1] The choices of nature, player, and enemy in our setting will be represented by points from certain sets. Thus, we may refer to their choices by "points" throughout the remainder of the text.

Algorithm 2.1 Definition of $\mathrm{OL}_{\mathcal{P}}(\mathrm{NATURE}, \mathrm{PLAYER}, \mathrm{ENEMY}, T)$

```
Input: NATURE, PLAYER, and ENEMY which are a nature, player, and enemy oracles for $\mathcal{P}$,
    respectively, and $T \in \mathbb{N}$.
Output: $(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}) \in X^T \times D^T \times Y^T$.
  for $t = 1$ to $T$ do
      $x_t \leftarrow \mathrm{NATURE}(t)$
      $d_t \leftarrow \mathrm{PLAYER}\big(\langle x_1, \dots, x_t \rangle, \langle y_1, \dots, y_{t-1} \rangle\big)$
      $y_t \leftarrow \mathrm{ENEMY}\big(\langle x_1, \dots, x_t \rangle, \langle d_1, \dots, d_{t-1} \rangle\big)$
  return $(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y})$
```

the sequence of points picked by the oracles. Even though this is intuitive in Algorithm 2.1, one may get confused later in the text when we define more complex algorithms and start talking about its actions on round t.

Let us look at what is happening on Algorithm 2.1 for an online learning $\mathcal{P} \coloneqq (X, D, Y L)$. In this setting, a player is participating in a game made of a sequence of rounds with a competitor, who we call enemy since it can, and will in the analysis of the algorithms we describe in later chapters, be adversarial to the player. At round t, nature presents a query $x_t \in X$. The player then picks a prediction $d_t \in D$, and the enemy simultaneously picks the "true label" $y_t \in Y$. Although this is not formally defined in the algorithm, one may imagine that at the end of the round the player suffers a loss of $L(d_t, y_t)$, where L is the loss function from $\mathcal{P}$, which is fixed throughout the game. One important aspect of this game is that, in round t, the player and the enemy know all the queries from nature until round t, besides knowing the points played by each other until the round $t - 1$. That is, at round t both the player and the enemy know $x_1, \dots, x_t$, the player knows $y_1, \dots, y_{t-1}$, and the enemy knows $d_1, \dots, d_{t-1}$, so they may adapt to each others' previous choices.

At this point, it is worth mentioning that the terminology on the literature is not uniform. Online learning is a loosely used term for many related settings and problems. For example, online learning in [19] is a setting quite similar to the one given by Algorithm 2.1, but without the nature oracle, a difference which does not affect the capabilities of the framework by much. Moreover, one may find works with "online learning" in its title which mainly talk about online (convex) optimization, a special case of online learning which we define in Section 2.5. Our definition of online learning is based mainly on [67] and on [50], and it fits the definition of classic papers of the field such as the ones introducing the Perceptron [62, 63] and Winnow algorithms [47]. Nonetheless, our presentation of online learning draws from other sources as well [19, 36, 38].

Before continuing, one may be puzzled by these functions which we call oracles used in the definition of online learning. We realize that this is a unusual way of defining the OL setting, and may not be ideal in all situations, but there are some reasons to define the setting in this way. One aspect of online learning (and of other settings we look at later) that we want to make crystal clear is which kind of information each of the "participants" has access to at each moment. For example, the idea that at round $t \in \mathbb{N}$ both player and enemy make their choices "simultaneously" is formally described in Algorithm 2.1 by the fact that d_t is computed without knowledge of y_t, and vice-versa. Still, both participants make these choices with knowledge of the queries from nature up to round t, that is, $\langle x_1, \dots, x_t \rangle$ as in Algorithm 2.1, as well as each other's choices on rounds 1 to $t - 1$. Knowing which information each oracle has access to may start to get complicated when we look at worst-case scenario or probabilistic players and enemies. As we will see, oracles give us the ability to formalize claims in sometimes insightful ways, leaving no room for doubt about the information each participant has access to at each round of the game.

Although this is not our main focus in the text, it is interesting to first look at strategies for the player in the *realizable case*. Let $\mathcal{P} \coloneqq (X, D, Y, L)$ be an online learning instance and let ENEMY be an enemy oracle for $\mathcal{P}$. Then ENEMY is **represented** by $h^* \colon X \to D$ if, for any $T \in \mathbb{N}$, any player oracle PLAYER for $\mathcal{P}$, and any nature oracle NATURE for $\mathcal{P}$, we have that if

$$(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}) = \mathrm{OL}_{\mathcal{P}}(\mathrm{NATURE}, \mathrm{PLAYER}, \mathrm{ENEMY}, T),$$

then $L(h^*(x_t), y_t) = 0$ for every $t \in [T]$. That is, there is some function $h^* \colon X \to D$ governing the pairing of nature's queries and the "correct" (zero-loss) prediction to it. Taking inspiration from the statistical learning setting, we name each function from $\mathcal{H}$ a **hypothesis** . In the **realizable case**, the goal is to build a player oracle $\mathrm{PLAYER}_{\mathcal{H}}$ for $\mathcal{P}$, which can depend on the hypothesis set $\mathcal{H}$, that attains small loss against enemy oracles represented by a hypothesis in $\mathcal{H}$.

As an example, consider the online learning instance

$$\mathcal{P} \coloneqq (X, \{0,1\}, \{0,1\}, L), \text{ where } L(d, y) \coloneqq [d \neq y] \text{ for every } d, y \in \{0,1\}. \tag{2.1}$$

In this instance, at each round nature picks a point $x \in X$. The player then has to give a label 0 or 1 to it, trying to match the enemy's classification, suffering a loss of 1 in the case of a mistake. Instances of this form are known as *online binary classification* problems, and we will define this class of instances more carefully in the next section. For an example of an online binary classification instance, consider X as a set of pictures, and suppose we want to devise a player oracle which can predict if there is a visible dog in a given picture of X or not. The ideal scenario is the one in which there is a set $\mathcal{H} \subseteq \{0,1\}^X$ and a function $h^* \in \mathcal{H}$ such that, for every $x \in X$, we have $h^*(x) = 1$ if and only if there is a visible dog in x. That is, h^* is a function which assigns the correct answer to each one of the images. In this case, we want to build a player oracle which, with knowledge of $\mathcal{H}$, can make a small number of mistakes against an enemy represented by h^*. More generally, in the realizable case of an OL instance $\mathcal{P} \coloneqq (X, D, Y, L)$ with a hypothesis set we want to devise a player oracle $\mathrm{PLAYER}_{\mathcal{H}}$ which performs well against any enemy represented by some hypothesis in $\mathcal{H}$. We index the player by the hypothesis set $\mathcal{H}$ since its behavior may (and usually will) depend on $\mathcal{H}$.

For the sake of simplicity, let us look at the case of the realizable online binary classification with a finite hypothesis set. That is, let $\mathcal{P}$ be as in (2.1) and let $\mathcal{H} \subseteq \{0,1\}^X$ be a nonempty and finite hypothesis set. Intuitively, since in the realizable case we want strategies for a player which has knowledge of $\mathcal{H}$, we can guess that the smaller the size of $\mathcal{H}$, the smaller the upper bounds we devise on the number of player mistakes will be. Indeed, suppose we have a player oracle which, at round $t \in \mathbb{N}$, has a set $\mathcal{H}_t \subseteq \mathcal{H}$ of candidates for the "true hypotheses" which represents the enemy oracle. Moreover, at round $t \in \mathbb{N}$ this player picks a hypothesis $h_t \in \mathcal{H}$ and makes its prediction according to h_t. At the end of the round, this player builds $\mathcal{H}_{t+1}$, the set of candidate hypotheses for the next round, by discarding the current hypothesis h_t in case of a mistake. It is easy to see that such a player does not make more than $|\mathcal{H}| - 1$ mistakes in the realizable case, independently of the number of rounds of the game. An easy and quite natural improvement of this idea can be made: at each round, predict in the same way as the majority of the hypotheses, and after each mistake discard all of the hypotheses which made a mistake. This strategy is known as *halving* (see [50, Section 7.2.1] and [67, Section 1.2.1]), and an oracle which formally implements it is defined on Algorithm 2.2.

Fortunately, such a simple[2] strategy already yields an upper bound in the number of mistakes of the player which grows logarithmically with the size of the hypothesis set.

Theorem 2.1.2 ([67, Theorem 2.1])**.** Let $\mathcal{P} \coloneqq (X, \{0,1\}, \{0,1\}, L)$ be an online learning instance as in (2.1). Moreover, let $\mathcal{H} \subseteq \{0,1\}^X$ be nonempty and let ENEMY be an enemy oracle for $\mathcal{P}$

[2] It is worth noting that such a simple algorithm has its subtleties. If the hypotheses are not given explicitly, it may be computationally expensive to explicitly compute the value of every hypothesis at the query picked by the nature oracle for each round.

Algorithm 2.2 Definition of $\text{HALVING}_{\mathcal{H}}(\langle x_1, \dots, x_T\rangle, \langle y_1, \dots, y_{T-1}\rangle)$

```
Input:
    (i) A nonempty set $\mathcal{H} \subseteq \{0,1\}^X$ for some set $X$ and
    (ii) sequences $\boldsymbol{x} \in X^T$ and $\boldsymbol{y} \in \{0,1\}^{T-1}$ for some $T \in \mathbb{N} \setminus \{0\}$.
Output: $d_T \in \{0,1\}$.
  $\mathcal{H}_1 \leftarrow \mathcal{H}$
  for $t = 1$ to $T$ do
      $\mathcal{H}_t^{(0)} \leftarrow \{\, h \in \mathcal{H}_t : h(x_t) = 0\}$
      $\mathcal{H}_t^{(1)} \leftarrow \{\, h \in \mathcal{H}_t : h(x_t) = 1\}$
      $d_t \leftarrow \big[|\mathcal{H}_t^{(1)}| \geq |\mathcal{H}_t^{(0)}|\big]$
      if $t < T$ then                              ▷ Compute candidates for next round if needed
          if $d_t \neq y_t$ then
              $\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t^{(1-d_t)}$
          else
              $\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t$
  return $d_T$
```

which is represented by a function in $\mathcal{H}$. Finally, let $T \in \mathbb{N}$ and NATURE be a nature oracle for $\mathcal{P}$, and set $(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}) \coloneqq \text{OL}_{\mathcal{P}}(\text{NATURE}, \text{HALVING}_{\mathcal{H}}, \text{ENEMY}, T)$. Then

$$\sum_{t=1}^{T} L(d_t, y_t) \leq \lg|\mathcal{H}|.$$

Proof. For each $t \in [T]$, let $\mathcal{H}_t$ be defined as in $\text{HALVING}_{\mathcal{H}}(\boldsymbol{x}, \boldsymbol{y}_{1:T-1})$ and define

$$\mathcal{H}_{T+1} \coloneqq \begin{cases} \mathcal{H}_T & \text{if } d_T = y_T, \\ \{\, h \in \mathcal{H}_T : h(x_T) = y_T\} & \text{otherwise,} \end{cases}$$

that is, $\mathcal{H}_{T+1}$ is defined as it would have been had the halving algorithm played $T+1$ rounds. Moreover, let $M = \{i(1), i(2), \dots, i(m)\} \subseteq [T]$ be the set of indices $i(k)$ such that $\mathcal{H}_{i(k)} \neq \mathcal{H}_{i(k)+1}$. By the definition of the halving algorithm, one can see that M is the set of rounds in which the player has made a mistake, that is, the rounds $t \in [T]$ such that $d_t \neq y_t$. By assumption, there is $h^* \in \mathcal{H}$ such thatENEMY is represented by h^*, that is, $h^* \in \mathcal{H}$ is such that $h^*(x_t) = y_t$ for each $t \in [T]$. Moreover, by the definition of HALVING, for each $h \in \mathcal{H}$ we have that $h \notin \mathcal{H}_{T+1}$ if and only if there is $t \in M$ such that $h(x_t) = d_t \neq y_t$, that is, if and only if h has made a mistake on one of the rounds in M. Since $h^*(x_t) = y_t$ for every $t \in \mathbb{N}$, we have $h^* \in \mathcal{H}_{T+1}$. Thus, $|\mathcal{H}_{T+1}| \geq 1$.

Moreover, let $t \in M$ and let both $\mathcal{H}_t^{(0)}$ and $\mathcal{H}_t^{(1)}$ be as in the definition of $\text{HALVING}_{\mathcal{H}}(\boldsymbol{x}, \boldsymbol{y}_{1:T-1})$. By the definition of the HALVING oracle, we have $|\mathcal{H}_t^{(d_t)}| \geq |\mathcal{H}_t^{1-d_t}|$. Since $\{\mathcal{H}_t^{(d_t)}, \mathcal{H}_t^{(1-d_t)}\}$ is a partition of $\mathcal{H}_t$ and $\mathcal{H}_{t+1} = \mathcal{H}_t^{(1-d_t)}$, we conclude that

$$|\mathcal{H}_{t+1}| = |\mathcal{H}_t^{1-d_t}| \leq \frac{|\mathcal{H}_t|}{2}, \qquad \forall t \in M.$$

Thus, by a simple induction,

$$1 \leq |\mathcal{H}_{T+1}| \leq \frac{|\mathcal{H}_{i(m)}|}{2} \leq \frac{|\mathcal{H}_{i(m-1)}|}{2^2} \leq \dots \leq \frac{|\mathcal{H}_1|}{2^m} = \frac{|\mathcal{H}|}{2^{|M|}}.$$

That is,

$$\sum_{t=1}^{T} L(d_t, y_t) = \sum_{t=1}^{T} [d_t \neq y_t] = |M| \leq \lg|\mathcal{H}|. \qquad \square$$

The above result holds for the case with a finite hypothesis set. Thus, one may be wondering if it is possible to say something about hypothesis sets of infinite size. We can, but in this case we need some kind of measure of the complexity of such hypothesis classes. A discussion from this point of view is not the focus of this text, but a great discussion on online learnability of general hypothesis classes on the classification case can be found on [67, Section 3].

2.2 Examples of Online Learning Problems

In this section we describe a series of classic and interesting classes of *online learning problems*, that is, instances with some specific form. Looking at these known problems will make the concepts shown in the last section more concrete and will help familiarize the reader with most of the online learning notation introduced so far. Additionally, some of these problems already encompass many other interesting and more specific online learning tasks. In some cases, as for the prediction with expert advice problem, the capability of the problem of modeling other online learning problems can single-handedly motivate this problem. Throughout the remainder of the text, when an OL instance fits into one of the problems we shall see, we may refer to it simply as a *problem* or as an *instance* of a problem. We shall use the latter when simply saying that it is a "problem" does not make it clear if we are talking about a specific instance or if we are talking about all instances which fit the description of the problem.

2.2.1 Online Classification

Let us look now at the online classification problem. In this kind of instance the player oracle receives, at each round, a point x from an arbitrary set X. It then has to assign a class (or label) from a finite set Y, and then the enemy reveals the correct class where x belongs. The goal of the player oracle is to assign classes to the points of X presented during the game in a way to match as accurately as possible the enemy oracle's classification. Formally, an online learning instance is said to be an **online classification problem** when it has the form (X, Y, Y, L), where X is an arbitrary set, Y is some finite set, and $L\colon Y \times Y \to \mathbb{R}$ is a function, although we usually have $L(d, y) \coloneqq [d \neq y]$ for every $d, y \in Y$.

A concrete example is the *spam filtering problem*, where at each round the player receives an email, represented by a point x in some set X. The player has to classify it as spam or not, and the user then reveals the true classification of the email. Note that the user is not adversarial to the player in this case. Rather, the user is the one that wants the most for the predictions to be accurate. Instead the spammers, which in this case play the role of nature, are the ones that are adaptive and adversarial[3] to the player, the spam detector. For the sake of concreteness, let us suppose that each email is represented in a format known as *bag-of-words*: we have a fixed set of words Σ, our dictionary, and each email is represented by a vector $x \in \mathbb{N}^{\Sigma}$, where x_w is the number of times the word $w \in \Sigma$ appears in the email. Thus, the spam filtering problem can be modeled,

[3] In our framework nature cannot be adaptive to the players' choices since it is not aware of such decisions. However, spammers usually tend to modify the emails they send in a way that it looks less like spam. Thus, as the OL game goes on, spammers will progressively send emails which do not look like spam for the filter, while the user is able to identify such emails as spam. That is, from the point of view of the spam filter, the user is the one who is being adversarial by marking seemingly inoffensive emails as spam, not the spammers/nature.

given a dictionary Σ, as the online learning instance $(\mathbb{N}^{\Sigma}, \{0,1\}, \{0,1\}, L)$, where $L(d,y) \coloneqq [d \neq y]$ for every $d, y \in Y$. This problem fits a type of online classification instances that deserves special attention, which is when there are only two distinct labels, a case which we call **online binary classification**.

2.2.2 Prediction with Expert Advice

An online learning instance is a **prediction with expert advice problem** or simply an **experts' problem** if it is of the form (A^E, A, Y, L), where A is some arbitrary set of possible "actions", E is a finite set of "experts", Y is nonempty, and $|L(a,y)| \leq 1$ for every $a \in A$ and $y \in Y$ (i.e., losses are bounded). The intuitive idea of this problem is that there is a finite set E of experts and at each round each expert suggests an action from a set A. With this information, the player has to pick an action from A and the enemy oracle then reveals the costs of the actions[4]. In this problem we are usually interested in devising player oracles that perform as well, in some sense, as the cost of the actions of the best expert in hindsight, even for adversarial enemy oracles. This is certainly one of the most studied and classic online learning problems, probably because many other problems can be modeled as a prediction with expert advice instance (for a plethora of examples, see [6, 24]). To motivate this problem, let us look at some examples of problems which can be modeled as prediction with expert advice problems.

A good example is the *online routing* problem: every day a driver has to pick one of many routes to go to work, wishing to minimize her total travel time[5]. A good measure of effectiveness of the driver's strategy is to compare the time she has spent driving in T days with the time she would have spent had she chosen the best fixed route in hindsight. Even though it may not be clear at first, we can fit this problem into the prediction with expert advice problem. If the set of routes is P, the key idea is to think of one expert for each route $p \in P$ whose advice is p at every round. Moreover, the enemy plays the role of assigning costs to the routes every round/day. That is, an instance of online routing can be modeled as the instance $(P^P, P, [-1,1]^P, L)$ of the experts' problem, where P is the set of possible routes, and $L(p,y) \coloneqq y_p$ for every $p \in P$ and $y \in [-1,1]^P$. It is worth noting that the crux of modeling problems as an experts' problem is usually the choice of the set of experts and which are their advice at each round. Thus, we can fit online routing in the prediction with expert advice problem: each route is an expert's advice, the player suffers the loss of the expert she has chosen to follow, and the cost of every route is revealed after the player makes her choice.

One interesting example of a problem which can be modeled as an experts' problem is the task from the machine learning field of *meta-learning*, which we only describe loosely for the sake of building intuition. Suppose we have a task where we have to predict, given a point in a set X, the "correct" label from a set Y. However, instead of learning from scratch, the player has a set $E \subseteq Y^X$ of predictors. The player's task is, given a sequence of points from X, to make not much more prediction mistakes than the best predictor from E. This problem fits seamlessly into the experts framework. At each round t, we have a point $x_t \in X$ for which we have to assign a label from Y. For that we receive the advice $e(x_t)$ of each predictor $e \in E$. With this information in hands, the player makes its prediction $d_t \in Y$, and the true label $y_t \in Y$ is then revealed, incurring a loss of $[d_t \neq y_t]$ to the player (or a loss given by some other loss function). This is useful when, for example, one has different predictors which work well in different scenarios, and one wants to select which predictor to use dynamically.

As we are going to see later on Section 2.6, the true power of the prediction with expert advice

[4] Note that, for any $y \in Y$, the function $L(\cdot, y)$ associates to each action a cost.

[5] We suppose that, at the end of the day the driver gets to see the travel time of that day for each route, even though it is more realistic to suppose the player has no information about the routes except for the one she picked.

problem shines when the player is randomized: at round t the player maintains weights $p_t \in \Delta_E$ over the set E of experts which represents a probability distribution over the experts. Then, instead of deterministically picking an action at each round, the player samples at round t an expert[6] to follow according to the distribution represented by p_t, that is, each expert $e \in E$ is sampled with probability $p_t(e)$. Later we show that the expected cost at round t of such strategy is given by $p_t^{\mathsf{T}} c_t$, where $c \in \mathbb{R}^E$ is given by $c_t(e) \coloneqq L(e, y_t)$ for each $e \in E$ where y_t is the point picked by the enemy at round t. We save the details of this randomization for Section 2.6.2.

One interesting application of randomized strategies for the experts' problem is boosting [66] which, again, we describe loosely since our goal is only to motivate the experts' problem. Besides, for the sake of simplicity we focus on the task of boosting an algorithm for *(statistical) classification*, a problem from statistical learning theory which we define slightly more formally on Section 2.4. In the statistical classification problem, the player has to build a predictor which assigns to points from a set X the correct labels from a set Y. The correct pairing of points from X to labels Y is assumed to be governed by a function $c\colon X \to Y$ that is unknown to the player, which we call a *concept*. In order to build such a predictor, we have access to a *training set* $\mathcal{T} \coloneqq \{(x_1, c(x_1)), (x_2, c(x_2)), \dots, (x_m, c(x_m))\} \subseteq X \times Y$. In the task of *boosting for statistical classification*, the player still has to build a good predictor from X to Y based on a training set $\mathcal{T} \subseteq X \times Y$ for the concept $c\colon X \to Y$, but in this case the player also has access to a function $\mathcal{A}\colon \Delta_{\mathcal{T}} \to Y^X$, a *weak learner*, which can be thought as an "almost worthless" predictor. Loosely speaking, a weak learner receives a probability distribution over the training set, and outputs a function from X to Y which predicts only *slightly better* than an uniformly random assignment from X to Y. This probability distribution given to the weak learner is a way to tweak the predictor which $\mathcal{A}$ generates so that it focuses more on certain points of the training set. The idea in a problem of boosting is to iteratively build predictors from $\mathcal{A}$ and update the probability distribution over the training set at each iteration so that, at a given iteration, the weak learner focuses on the points from $\mathcal{T}$ which all the previously generated predictors *together*[7] miss-classify. As it is shown in [66], the final predictor built from a combination of all the predictors generated by $\mathcal{A}$ during this process is a good one (given that the training set and the number of rounds are big enough, besides other conditions).

The boosting task is modeled as an experts problem as follows. Each entry from the training set $\mathcal{T}$ is modeled as an expert. At round t, the player picks weights (or probabilities) $p_t \in \Delta_{\mathcal{T}}$ for the points in $\mathcal{T}$, which yield a predictor $r_t \coloneqq \mathcal{A}(p_t)$. Then, the enemy simply outputs $\mathcal{T}$, and the loss of the player is given by the function $L(r, \mathcal{T}) \coloneqq \sum_{(x,y) \in \mathcal{T}} [r(x) \neq y]$ for each $r \in Y^X$. Interestingly, even though each one of the functions $r_t \coloneqq \mathcal{A}(p_t)$ induced by the weights p_t picked by the player are bad predictors, if the player in this experts' problem suffers a low amount of cumulative loss (in some sense), then the average predictions given by all the functions $r_1, \dots, r_T$ happens to be a good predictor (given that some conditions on the training set and the number of rounds are satisfied).

[6] One may find it odd that we sample an expert, and not an action. This is a choice of strategy for the player, that is, following this strategy the player always picks an action suggested by some expert. This guarantees a good performance in expectation if compared to the loss of the best expert in hindsight. Still, one may come up with cases where the player would benefit from choosing actions none of the experts suggested.

[7] Maybe by taking the average prediction, the majority prediction or simply the prediction given by a predictor sampled uniformly at random.

2.2.3 Online Regression

Formally, the **online regression problem** encompasses the online learning instances of the form[8] $(\mathbb{R}^d, \mathbb{R}, \mathbb{R}, L)$, where $L\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is some arbitrary function. Some common loss functions used for this type of problems are $(d, y) \in \mathbb{R} \times \mathbb{R} \mapsto |d - y|$ and $(d, y) \in \mathbb{R} \times \mathbb{R} \mapsto (d - y)^2$. The format of online regression problems may seem too broad, and indeed it is. The structure of these problems is usually contained in the hypothesis set $\mathcal{H} \subseteq (\mathbb{R}^d \to \mathbb{R})$ of functions the player has to compete with. For example, if we were in the realizable case, there would be, for each enemy, a "true function" which maps the entries given by nature to the correct labels of the enemy. Moreover, if $\mathcal{H}$ had no structure whatsoever (or if the player were not aware of any structure), e.g. if $\mathcal{H}$ is the set of all functions from $\mathbb{R}^d$ to $\mathbb{R}$, it would be impossible for a player to do much. More frequently, one wants to compete against a more restricted class of functions. Some examples of common hypothesis sets are $\{\, x \in X \mapsto w^{\mathsf{T}} x : w \in \mathbb{R}^d \}$ and $\{\, x \in X \mapsto x^{\mathsf{T}} W x + b^{\mathsf{T}} x : W \in \mathbb{R}^{d \times d}, b \in \mathbb{R}^d \}$, cases which we call **online linear regression** and **online quadratic regression**, respectively.

As an example, let us look at the *dynamic pricing* problem (inspired from [19, Section 1.2]). In this problem, a vendor sequentially receives customers in a way such that the characteristics of the customer from round t is represented by a vector $x_t \in \mathbb{R}^d$. After receiving the customer, the vendor sets a price $d_t \in \mathbb{R}$ to charge, while the customer picks a maximum price $y_t \in \mathbb{R}$ he is willing to pay. Finally, an intuitive-looking loss function for this problem is $L(d, y) \coloneqq -d[d \leq y]$ for every $d, y \in \mathbb{R}$. In this case, the vendor may try do model the relation between customer characteristics and their respective maximal prices with a linear function $x \in \mathbb{R}^d \mapsto w^{\mathsf{T}} x$, for example. In this case, it makes sense to compare the performance of the vendor only against the performance of other linear functions. One may notice that the type of functions the vendor chooses to model this customer-price relationship depends on her prior knowledge about such a relation. Additionally, this choice affects the complexity of the strategy of the player: strategies for online linear regression are usually more efficiently computable than ones for online quadratic regression or for even more complex regression models.

2.2.4 Sequential Investment

Let us look at the problem of *sequential investment* or *constantly rebalanced portfolio.* In this problem an investor has a set A of assets over which she has to distribute her wealth at the beginning of every day. After choosing a distribution, which may be represented by a point $d \in \Delta_A$, the market reveals the ratio of returns of each one of the assets for the day. Namely, it reveals a non-zero vector $r \in \mathbb{R}^A_+$ such that, if the investor's wealth at the beginning of the day was $\rho_0 \in \mathbb{R}_+$, then her wealth at the end of the day is $\sum_{a \in A} \rho_0 r_a d_a = \rho_0 r^{\mathsf{T}} d$.

Let us look at the case where the investor has $\rho_0 \in \mathbb{R}_+$ of initial wealth, and makes investments for T days, where at day $t \in [T]$ her wealth distribution is given by $d_t \in \Delta_A$ and the market return ratios for day t is given by $r_t \in \mathbb{R}^A_+ \setminus \{0\}$. Then, the investor's total wealth at the end of day T is

$$\rho_T \coloneqq \rho_0 \prod_{t=1}^{T} d_t^{\mathsf{T}} r_t.$$

Thus, to maximize her total wealth at the end of day T the investor has to, in an online fashion,

[8] Even though we define this problem using $\mathbb{R}^d$ as the query set, one can consider the more general case where the query space for the regression is a more general Euclidean space such as the space of matrices with the trace inner product.

maximize her final wealth ratio

$$\frac{\rho_T}{\rho_0} = \prod_{t=1}^{T} d_t^{\mathsf{T}} r_t.$$

This problem still does not quite fit the online learning setting. However, note that maximizing the above quantity is equivalent to maximizing its logarithm or, instead, minimizing the negative logarithm of the wealth ratio. That is, the investor wants to minimize

$$-\sum_{t=1}^{T} \ln d_t^{\mathsf{T}} r_t,$$

and the above better fits the online learning setting.

Formally, an online learning instance is a **sequential investment** instance if it is of the form $\mathcal{P}(\{0\}, \Delta_A, \mathbb{R}_+^A, L)$, where $L(a,r) \coloneqq -\ln a^{\mathsf{T}} r$ for every $a \in \Delta_A$ and every $r \in \mathbb{R}_+^A$. This setting is more naturally cast in the online optimization setting, which we are going to see later. Still, this problem has interesting properties, and modeling it as an online learning instance leaves some room for tweaking the model (for example, one could make nature give "hints" about which assets will be more valuable that day). This problem in the worst-case scenario was first investigated by Cover [27], who called *universal portfolio selection* strategies which could perform well in this setting even against an adversarial market. One of the special features of this problem is its loss function L, which is convex and *exp-concave* with respect to its first argument. Roughly, the latter means that L is curved in the direction of its gradients. We will see in Chapter 6 that exp-concavity can be exploited by player oracles, obtaining in this way very good bounds on cumulative loss.

2.3 Loss Minimization Impossibility and Regret

On Section 2.1, we have quickly looked at online learning problems $\mathcal{P} \coloneqq (X, D, Y, L)$ in the realizable case. Recall that, in this case, we want to perform well against enemies whose behavior can be predicted with 0 loss by a function/hypothesis from a known set $\mathcal{H} \subseteq D^X$. As expected, this assumption is too strong for most of the problems in online learning. Take as an example the spam filtering problem. It is unrealistic to suppose that there is a function which correctly classifies any possible email for a user as spam or not-spam, since spammers change their attacks over time for example. In this way, emails which were considered spam at some point in time may be considered not spam later on, and vice-versa. The following simple proposition shows what is almost obvious: in the general setting against adaptive and adversarial enemies, there is no hope in trying to minimize cumulative loss. Indeed, if the cost of all possible predictions of the player is 1, for example, there is nothing to do. Even in slightly more interesting cases, looking at the raw cumulative loss seems uninformative.

Proposition 2.3.1. Define the prediction with expert advice instance $\mathcal{P} \coloneqq (\{1,2\}^2, \{1,2\}, [-1,1]^2, L)$, where $L(d,y) \coloneqq y_d$ for every $d \in \{1,2\}$ and $y \in [-1,1]^2$. Moreover, let NATURE and PLAYER be nature and player oracles for $\mathcal{P}$, respectively. Finally, define the enemy oracle $\text{ENEMY}^*_{\text{PLAYER}}$ for every $t \in \mathbb{N}$, every $\boldsymbol{x} \in (\{1,2\}^2)^{t+1}$, and every $\boldsymbol{d} \in (\{1,2\}^2)^t$ by

$$\text{ENEMY}^*_{\text{PLAYER}}(\boldsymbol{x}, \boldsymbol{d}) \coloneqq e_{d_{t+1}} \qquad \begin{aligned} &\text{with } d_{t+1} \coloneqq \text{PLAYER}(\boldsymbol{x}, \boldsymbol{y}'), \text{ where} \\ &(\boldsymbol{x}_{1:t}, \boldsymbol{d}', \boldsymbol{y}') \coloneqq \text{OL}_{\mathcal{P}}(\text{NATURE}, \text{PLAYER}, \text{ENEMY}^*_{\text{PLAYER}}, t). \end{aligned}$$

Then, by setting $(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}) \coloneqq \text{OL}_{\mathcal{P}}(\text{NATURE}, \text{PLAYER}, \text{ENEMY}^*_{\text{PLAYER}}, T)$ we have

$$\sum_{t=1}^{T} L(d_t, y_t) = T.$$

Proof. Let NATURE be a nature oracle for $\mathcal{P}$, let $T \in \mathbb{N}$, and define

$$(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}) \coloneqq \mathrm{OL}_{\mathcal{P}}(\mathrm{NATURE}, \mathrm{PLAYER}, \mathrm{ENEMY}^{*}_{\mathrm{PLAYER}}, T).$$

By the definition of $\mathrm{ENEMY}^{*}_{\mathrm{PLAYER}}$, we have that $y_t = e_{d_t}$ for every $t \in [T]$, that is, $L(d_t, y_t) = 1$ for every $t \in [T]$. □

Note that the enemy oracle used in the above simple proposition is special: it knows the strategy of the player. Thus, even though the enemy does not have access to the t-th choice of the player during round t in the usual rules of the game, this special enemy knows exactly which point the player picks on the current round. This is the most extreme case of adversarial enemy one can think off: one that makes the player suffer the maximal amount of loss possible no matter what the player does.

Thus, the worst-case cumulative loss of a player may not give us much information about her performance: both sophisticated and simple-minded player oracles are usually indistinguishable if we look only at the cumulative loss. Intuitively, the problem is that we are posing, in some sense, an unrealistic goal: to obtain low cumulative loss when even a player with hindsight could perform no better. A more informative metric of performance originated from game theory [24] is the notion of *regret*, which measures how better or worse the player oracle performs when compared to some function of queries to predictions. The name “regret” comes from the idea that the regret with respect to some function measures how “sorry” the player oracle is for not using this function as its strategy throughout the whole game. Let us define regret formally.

Definition 2.3.2 (Regret for Online Learning)**.** Let $\mathcal{P} \coloneqq (X, D, Y, L)$ be an OL instance, let $h\colon X \to D$, and let $T \in \mathbb{N}$. Moreover, let $\boldsymbol{x} \in X^T$, $\boldsymbol{d} \in D^T$, and $\boldsymbol{y} \in Y^T$. The **regret** of $\boldsymbol{d}$ with respect to the sequence $\boldsymbol{y}$ and the function h (and w.r.t. the sequence $\boldsymbol{x}$ and the loss function L) is

$$\mathrm{Regret}(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}, h, L) \coloneqq \sum_{t=1}^{T} \big(L(d_t, y_t) - L(h(x_t), y_t) \big).$$

If PLAYER is a player oracle for $\mathcal{P}$, define

$$\mathrm{Regret}(\boldsymbol{x}, \mathrm{PLAYER}, \boldsymbol{y}, h, L) \coloneqq \mathrm{Regret}(\boldsymbol{x}, \boldsymbol{d}', \boldsymbol{y}, h, L),$$

where

$$d'_t = \mathrm{PLAYER}(\langle x_1, \dots, x_t \rangle, \langle y_1, \dots, y_{t-1} \rangle), \qquad \forall t \in [T]. \tag{2.2}$$

Additionally, for every $\mathcal{H} \subseteq D^X$, the **regret** of $\boldsymbol{d}$ with respect to the sequence $\boldsymbol{y}$ and the set $\mathcal{H}$ (and w.r.t. the sequence $\boldsymbol{x}$ and the loss function L) is

$$\mathrm{Regret}(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}, \mathcal{H}, L) \coloneqq \sup_{h \in \mathcal{H}} \mathrm{Regret}(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}, h, L) = \sum_{t=1}^{T} L(d_t, y_t) - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} L\big(h(x_t), y_t\big).$$

Moreover, If PLAYER is a player oracle for $\mathcal{P}$, define

$$\mathrm{Regret}(\boldsymbol{x}, \mathrm{PLAYER}, \boldsymbol{y}, \mathcal{H}, L) \coloneqq \mathrm{Regret}(\boldsymbol{x}, \boldsymbol{d}', \boldsymbol{y}, \mathcal{H}, L),$$

where $\boldsymbol{d}'$ is defined as in (2.2). Finally, for every nature, player, and enemy oracles PLAYER, ENEMY, and NATURE for $\mathcal{P}$, respectively, for every function $h\colon X \to D$ and every set $\mathcal{H} \subseteq D^X$ define

$$\mathrm{Regret}_T^{\mathcal{P}}(\mathrm{NATURE}, \mathrm{PLAYER}, \mathrm{ENEMY}, h) \coloneqq \mathrm{Regret}(\boldsymbol{x}', \boldsymbol{d}', \boldsymbol{y}', h, L)$$

and

$$\mathrm{Regret}_T^{\mathcal{P}}(\mathrm{NATURE}, \mathrm{PLAYER}, \mathrm{ENEMY}, \mathcal{H}) \coloneqq \mathrm{Regret}(\boldsymbol{x}', \boldsymbol{d}', \boldsymbol{y}', \mathcal{H}, L),$$

where $(\boldsymbol{x}', \boldsymbol{d}', \boldsymbol{y}') \coloneqq \mathrm{OL}_{\mathcal{P}}(\mathrm{NATURE}, \mathrm{PLAYER}, \mathrm{ENEMY}, T)$, omitting $\mathcal{P}$ from the notation when it is clear from the context.

As in the realizable case, the function h and the set $\mathcal{H}$ in the above definitions are usually called **hypothesis** and **hypothesis set**, respectively.

Let us take a step back and look at the intuitive meaning of these regret functions. The regret of a player oracle with respect to a function h measures how much better the player would have performed had she used as her strategy throughout the game the function h. Similarly, the regret with respect to a hypothesis set $\mathcal{H}$ measures how much better the player would have performed had she used as her strategy throughout the game the hypothesis from $\mathcal{H}$ best suited for this game, that is, the one which performs best.

Let us look at some hypothesis classes for some of the problems we have seen on Section 2.2. Let $\mathcal{P} \coloneqq (A^E, A, Y, L)$ be an instance of prediction with expert advice. Intuitively, the player should not be expected to magically perform better than any of the experts. Thus, if no expert in E performs well in the game (with certain nature and enemy oracles), then it is reasonable to expect that the player will perform poorly as well. On the other hand, if there is some expert in E which is good, that is, whose advice yields losses much lower than that of the other experts throughout the game, a good player oracle should be expected to notice that and learn to follow this good expert. So, a natural and traditionally used hypothesis class for the experts problem is $\mathcal{H} \coloneqq \{\, x \in A^E \mapsto x_e : e \in E \}$, that is, the class of predictors which follow the same expert at every round. Thus, the regret w.r.t. $\mathcal{H}$ measures how sorry the player is for not following in every round the best expert in hindsight.

Let us look at the case of online regression. As briefly commented on Section 2.2, the hypothesis class used for online regression is what defines most of the structure of the problem. If we are in the online linear regression case, and no linear function performs well in the game with certain nature and enemy oracles at hand, the player may still be able to get low regret, since she only needs to perform as well as the best linear function, none of which perform well in the current game. Finally, in online classification, a natural hypothesis set is a set of naive predictors, usually easy to come up with (such as a fixed random assignment of labels, or predictors based in very simple rules). Thus, the task in online classification is to classify as well as any of these benchmark predictors.

Given a learning problem and a hypothesis set, we are interested in finding player oracles that attain low regret w.r.t. this hypothesis set for any online learning instance from this class. However, we have yet to say what is *low regret*. First in words and not so formally, we consider the regret of the player w.r.t. a hypothesis set $\mathcal{H}$ to be low if it is sublinear in the number of rounds T, which happens only if, for any $h \in \mathcal{H}$, the difference of the loss of the player at round t and the loss of the function h in the same round (i.e. $L(h(x_t), y_t)$) goes to zero as the number t goes to infinity. Intuitively, if $L(d_t, y_t) - L(h(x_t), y_t) \to 0$ as $t \to +\infty$ for any $h \in \mathcal{H}$, it means that the player is being able to "learn" how to perform as well as the best hypothesis $\mathcal{H}$ for the game. Formally, let PLAYER be a player oracle for the OL instance $\mathcal{P} \coloneqq (X, D, Y, L)$. We say that PLAYER attains **low regret** w.r.t. a hypothesis set $\mathcal{H}$ and to nature and enemy oracles NATURE and ENEMY for $\mathcal{P}$, respectively, if

$$\lim_{T \to \infty} \frac{1}{T} \mathrm{Regret}_T(\mathrm{NATURE}, \mathrm{PLAYER}, \mathrm{ENEMY}, \mathcal{H}) = 0.$$

That is, the player attains low regret with respect to $\mathcal{H}$ and to the oracles NATURE and ENEMY if the regret grows sublinearly w.r.t. the number of rounds T. We may also say that PLAYER attains low-regret w.r.t. a single hypothesis instead of a hypothesis set with a definition similar to the one above. We will usually be interested in player oracles that attain low regret for *any* pair of enemy and nature oracles for the online learning instance at hand. We say that a player oracle attains **low regret** w.r.t. a hypothesis set $\mathcal{H}$ if it attains low regret w.r.t. $\mathcal{H}$ and to any pair of nature and enemy oracles for $\mathcal{P}$. Although we are not going to use this terminology in this text, it is worth noting that player oracles whose regret against any nature and enemy oracles grows sublinearly w.r.t. the number of rounds of the game are said to be **Hannan consistent** (see [24, Section 4.2]). Additionally, we will be interested in the speed with which the above limit goes to 0. The faster it converges to zero, the fewer rounds the player needs to "learn", in some sense.

We have seen that regret is, at least intuitively, a performance measure which seems more sensible than looking at the raw cumulative loss. Additionally, note that regret is exactly the loss of the player if we are in the realizable case, which contributes to the idea that this performance measure is a good generalization unifying the realizable and non-realizable cases. Still, pursuing regret sublinear in the number of rounds happens to be an impossible mission in quite simple cases, as the next proposition due to Cover [26] shows. The proof idea is simple: consider an instance of the prediction with experts advice with two experts, each with a distinct constant advice at every round. Then, a player who is put against its worst possible enemy (which attributes loss of 1 only to the player's current choice) in a game of $T \in \mathbb{N}$ rounds will suffer a loss of T, while one of the experts will have cumulative loss smaller than $T/2$.

Proposition 2.3.3 (Cover's impossibility result [26])**.** Define the prediction with expert advice instance $\mathcal{P} \coloneqq (\{1,2\}^2, \{1,2\}, [-1,1]^2, L)$, where $L(d,y) \coloneqq y_d$ for every $d \in \{1,2\}$ and $y \in [-1,1]^2$. Set $h_1(x) \coloneqq x_1$ and $h_2(x) \coloneqq x_2$ for every $x \in \{1,2\}$, and define $\mathcal{H} \coloneqq \{h_1, h_2\}$. Moreover, define the nature oracle NATURE for $\mathcal{P}$ by $\text{NATURE}(t) \coloneqq (1,2)^\mathsf{T}$ for every $t \in \mathbb{N} \setminus \{0\}$ and let PLAYER be a player oracle for $\mathcal{P}$. Finally, define the enemy oracle $\text{ENEMY}^*_{\text{PLAYER}}$ for every $t \in \mathbb{N}$, every $\boldsymbol{x} \in (\{1,2\}^2)^{t+1}$, and every $\boldsymbol{d} \in (\{1,2\}^2)^t$ by

$$\text{ENEMY}^*_{\text{PLAYER}}(\boldsymbol{x}, \boldsymbol{d}) \coloneqq e_{d_{t+1}} \qquad \begin{array}{l} \text{with } d_{t+1} \coloneqq \text{PLAYER}(\boldsymbol{x}, \boldsymbol{y}'), \text{ where} \\ (\boldsymbol{x}_{1:t}, \boldsymbol{d}', \boldsymbol{y}') \coloneqq \text{OL}_{\mathcal{P}}(\text{NATURE}, \text{PLAYER}, \text{ENEMY}^*_{\text{PLAYER}}, t). \end{array}$$

Then,

$$\text{Regret}_T(\text{NATURE}, \text{PLAYER}, \text{ENEMY}^*_{\text{PLAYER}}, \mathcal{H}) \geq \lfloor \tfrac{T}{2} \rfloor, \qquad \forall T \in \mathbb{N}.$$

Proof. Let $T \in \mathbb{N}$ and set $(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}) \coloneqq \text{OL}_{\mathcal{P}}(\text{NATURE}, \text{PLAYER}, \text{ENEMY}^*_{\text{PLAYER}}, T)$. By Proposition 2.3.1, we have that the player suffers a loss of 1 at every round, that is, $\sum_{t=1}^T L(d_t, y_t) = T$. Moreover, note that for every $t \in [T]$ we have $L(h_i(x_t), y_t) = L(x_t(i), y_t) \in \{0,1\}$ for any $i \in \{1,2\}$ and

$$L(h_1(x_t), y_t) + L(h_2(x_t), y_t) = L(x_t(1), y_t) + L(x_t(2), y_t) = 1.$$

Therefore, there is $i \in \{1,2\}$ such that

$$\sum_{t=1}^T L(h_i(x_t), y_t) = \sum_{t=1}^T L(x_t(i), y_t) = \sum_{t=1}^T L(x_t(i), y_t) \leq \lfloor \tfrac{T}{2} \rfloor. \qquad \square$$

2.4 Relation of Online Learning with Statistical Learning

In this text, our focus is the online learning setting, where a player in a sequential game has to compete against an adversarial enemy picking the "correct" answers to queries. Still, it is enlightening to compare the OL setting with the statistical learning setting [50, 71]. More specifically, in this section we look at the *Probably and Approximately Correct (PAC)* framework for *supervised learning*, i.e. learning from a previously sampled set of pairs of queries and labels, called *training sequence/set* (these concepts will be formally defined soon). We then compare these ideas with the ones developed so far for online learning, and we give a brief overview of techniques to convert algorithms for a player from the online learning setting to algorithms for the statistical (or batch) setting.

A **(supervised) statistical learning instance** is a quadruple $\mathcal{L} \coloneqq (X, D, Y, L)$, where X, D, and Y called **query**, **decision**, and **label** sets, respectively, each being a measurable space (i.e., each is equipped with a σ-algebra), and $L\colon D \times Y \to \mathbb{R}$ is the **loss function**, a Borel-measurable function. That is, a statistical learning instance has the same form of an OL instance, with the addition of the measurability conditions for us to work with probability distributions over these sets (or to create random variables with the loss function). The goal of the player[9] is to "learn" a certain unknown probability distribution $\mathcal{D}$ over $X \times Y$ equipped with the product σ-algebra of the σ-algebras for X and Y. In order to do so, the idea is that the player has access to a sequence of $\langle (x_1, y_1), \dots, (x_T, y_T) \rangle \in (X \times Y)^T$ of independent and identically distributed samples from the probability distribution $\mathcal{D}$ over $X \times Y$, a realization of a *training sequence*[10] (for $\mathcal{D}$). Formally, if $T \in \mathbb{N}$ and $\mathcal{D}$ is a probability distribution over $X \times Y$, then, a **training sequence** (of size T and for a probability distribution $\mathcal{D}$ on $X \times Y$) is a sequence $\boldsymbol{Z} \coloneqq \langle Z_1, \dots, Z_T \rangle \in ((X \times Y)^{\Omega})^T$ of $T \in \mathbb{N}$ independent random variables on a probability space $(\Omega, \Sigma, \mathbb{P})$ such that $Z_i \sim \mathcal{D}$ in $(\Omega, \Sigma, \mathbb{P})$ for each $i \in [T]$. We will abuse notation so that, if $\boldsymbol{Z}$ is a training sequence of size $T \in \mathbb{N}$ on a probability space $(\Omega, \Sigma, \mathbb{P})$, then we set $\boldsymbol{Z}(\omega) \coloneqq \langle Z_1(\omega), \dots, Z_T(\omega) \rangle$ for each $\omega \in \Omega$, that is, we treat $\boldsymbol{Z}$ as a random variable taking values on $(X \times Y)^T$.

Given a supervised statistical learning instance $\mathcal{L} \coloneqq (X, D, Y, L)$, our goal is to devise a function $\mathcal{A}$ from $\mathrm{Seq}(X \times Y)$ to D^X such that, given as input to $\mathcal{A}$ a sequence in $\mathrm{Seq}(X \times Y)$ (which will be a realization/evaluation of the training sequence we have access to), it outputs a hypothesis $h\colon X \to D$. Recall from Section 1.1.2 that, since X and Y are measurable spaces, $X \times Y$ is a measurable space (equipped with the σ-algebra from the product space) and, as discussed in Section 1.1.2, we equip on $\mathrm{Seq}(X \times Y)$ a properly crafted σ-algebra as described in (1.1). A **player oracle** for $\mathcal{L}$ is a measurable function[11] $\mathcal{A}\colon \mathrm{Seq}(X \times Y) \to D^X$ such that each function $h\colon X \to D$ in the image of $\mathcal{A}$ is measurable. The intuition is that, for each sequence of points from $\mathrm{Seq}(X \times Y)$ the player outputs a hypothesis which tries to predict, for each query in X, the probably best answer in D to the "enemy's answer" from Y. Formally, if $\boldsymbol{Z}$ is a training sequence on a probability space $(\Omega, \Sigma, \mathbb{P})$ and $\mathcal{A}$ is a player oracle for $\mathcal{L}$, then we define[12] the random variable

$$[\mathrm{TRAIN}(\mathcal{A}, \boldsymbol{Z})](\omega) \coloneqq \mathcal{A}(\boldsymbol{Z}(\omega)), \qquad \forall \omega \in \Omega.$$

Although we have said that the goal of the player is to learn the unknown distribution $\mathcal{D}$, we have not formally defined what it means for the player to *learn*. The idea is that the player wants to pick a function from X to D which tries to predict the pairing of points from $X \times Y$ sampled

[9] The entity "player" is not commonly used in the statistical learning literature, since it simply poses problems without needing any notion of "player" and "enemy". Still, attributing to an entity called player the goal that the learning problem poses is useful to compare this setting with the online one.

[10] Again, in traditional statistical learning this is a set, known as the *training set*, not a sequence. But we do not lose generality by dealing with sequences and it will ease the definition of some concepts later on.

[11] We use the notation $\mathcal{A}$ for a player for statistical learning instances to avoid confusion with online learning notation and due to tradition in the literature.

[12] One may argue that the following definition is innocuous and that it is equivalent to the random variable $\mathcal{A} \circ \boldsymbol{Z}$, which we would denote simply by $\mathcal{A}(\boldsymbol{Z})$ based on our conventions from Section 1.1.2. Still, in this case it is good to avoid any kind of confusion since we want the oracle $\mathcal{A}$ to receive only points from $X \times Y$, and not the entire random variable, which could give unfair information for the player about the unknown probability distribution that she wants to learn.

from the distribution $\mathcal{D}$ and, thus, to choose for each point in X a point in D to minimize her losses. More specifically, fixed an unknown to the player probability distribution $\mathcal{D}$ on $X \times Y$, the player wants to pick a measurable function $h \colon X \to D$ with small **generalization error** or **risk** w.r.t. $\mathcal{D}$, which is given by

$$\operatorname{risk}_{\mathcal{D}}(h) = \mathbb{E}_{(\bar{X},\bar{Y})\sim\mathcal{D}}[L(h(\bar{X}), \bar{Y})],$$

where $\mathbb{E}_{(\bar{X},\bar{Y})\sim\mathcal{D}}$ is the expectation taken with respect to a probability space $(\Omega, \Sigma, \mathbb{P})$ in which $(\bar{X}, \bar{Y})$ is a random variable with probability distribution given by $\mathcal{D}$. Usually, the player picks the function h from a known **hypothesis set** $\mathcal{H} \subseteq D^X$, that is, a set of measurable functions from X to D.

One example of a statistical learning instance is a statistical version of the prediction with expert advice problem. Let (A^E, A, Y, L) be a prediction with expert advice problem, and suppose that A and Y are measurable spaces, and that the function L is measurable (the set E is naturally a measurable space when equipped with its power set). In this version of the problem, the advice of the experts (and their respective costs) are bound to a probability distribution, that is, there is a distribution $\mathcal{D}$ on $A^E \times [-1,1]^E$ on which the training sequence is based. We are often interested in finding the expert whose advice, in expectation, minimizes the loss given by the loss function $L \colon A \times Y \to [-1,1]$. In the context of statistical learning, this is the same as picking a hypothesis in $\{\, x \in A^E \mapsto x_e : e \in E\}$ that minimizes the risk with respect to the probability distribution $\mathcal{D}$.

A natural question that arises after the above definitions is: how small should the error of the hypothesis be so that we can consider that the player was able to learn the unknown probability distribution from where the training sequence was sampled? In online learning, we have seen that a reasonable goal is to seek for player oracles with regret sublinear in the number of rounds for any pair of enemy and nature oracles. Since the statistical setting is a probabilistic setting by nature, we should expect the learnability condition to be probabilistic as well. Let us look first at the simpler **deterministic case**. In this case, we suppose that there is a (measurable) function $c \colon X \to Y$, a **target concept**, which determines the relation between X and Y that we want to learn. In this case, we need training sequences with a special form, which we call *deterministic training sequences*. A **deterministic training sequence** (for the concept $c \colon X \to Y$) based on a probability distribution $\mathcal{D}$ on X is a sequence $\boldsymbol{Z} = \langle (X_1, c \circ X_1), \dots, (X_T, c \circ X_T) \rangle$, where $X_1, \dots, X_T$ are independent random variables on a probability space $(\Omega, \Sigma, \mathbb{P})$, each following the probability distribution $\mathcal{D}$.

Moreover, overloading the notation of risk to capture the deterministic case, we define the **deterministic generalization error** or the **deterministic risk** of a measurable function $h \colon X \to D$ w.r.t. a probability distribution $\mathcal{D}$ on X and to a concept $c \colon X \to Y$ by

$$\operatorname{risk}_{\mathcal{D}}(h, c) = \mathbb{E}_{\bar{X}\sim\mathcal{D}}[L(h(\bar{X}), c(\bar{X}))],$$

Hence, given a concept set $C \subseteq Y^X$, we want to decide if it is possible to devise an algorithm for the player which, for any target concept $c \in C$ given together with a training sequence based on a fixed probability distribution, generates a hypothesis (i.e., a function from X to D) with small deterministic risk. This idea is formalized by *PAC-learnability*, concept first proposed in [70]. Here we closely follow the presentation from [50].

Definition 2.4.1 (PAC-learnability)**.** Let $\mathcal{L} \coloneqq (X, D, Y, L)$ be a statistical learning instance, and let $C \subseteq Y^X$ be a set of measurable functions. Then C is **PAC-learnable** for $\mathcal{L}$ if there is a player oracle $\mathcal{A}_C \colon \operatorname{Seq}(X \times Y) \to D^X$ for $\mathcal{L}$ and a real polynomial function $p \colon \mathbb{R}^3 \to \mathbb{R}$ such that, for any $c \in C$, any probability distribution $\mathcal{D}$ over X, any $\varepsilon > 0$, any $\delta > 0$, any $T \in \mathbb{N}$ with $T > p(1/\varepsilon, 1/\delta, T)$, and any deterministic training sequence $\boldsymbol{Z}$ of size T for c sampled from a probability distribution $\mathcal{D}$ on X on the probability space $(\Omega, \Sigma, \mathbb{P})$, we have

$$\mathbb{P}(\operatorname{risk}_{\mathcal{D}}(\mathrm{TRAIN}(\mathcal{A}_C, \boldsymbol{Z}), c) \le \varepsilon) \ge 1 - \delta.$$

Moreover, if there is an algorithm which implements $\mathcal{A}$ that runs in time $O(p(1/\varepsilon, 1/\delta, T))$, then C is **efficiently PAC-learnable** for $\mathcal{L}$. In this definition, we omit the dependence of the minimum size of T and the space required to represent the concept c for the sake of simplicity.

With the definition of PAC-learnability, we can already spot major differences between the statistical and the online learning settings. In the statistical case, we have strong statistical assumptions over the points sampled from $X \times Y$, while in the online case the points can be picked in a way adversarial to the player. Another major difference is that in statistical learning the player has access to all the points sampled from $X \times Y$ before picking a hypothesis, while in the online learning setting the player has to make a prediction (and thus suffer some kind of loss) for each round. Arguably, the most interesting difference is the way that each setting measures algorithm/oracle quality (regret and risk). In the statistical case, the player tries to pick a *single* function that generalizes well in expectation in *unseen* data sampled from a fixed distribution (the same distribution on which training sequence is based on), that is, that performs well in the "real world". Indeed, note that in the definition of PAC-learnability, we require the algorithm to be able to have low risk with respect to any fixed probability distribution, only requiring the training sequence to be sampled from this same probability distribution. On the other hand, in the online case the player is affected only by the points she sees during the game. Not only that, the player can, in some sense, change her strategy in the middle of the current game based on the points she has already seen, not committing to a single hypothesis as in the statistical scenario. While strategies used in these frameworks can differ in approach, with some effort, algorithms for online learning that have low regret can be used for statistical learning in some cases to obtain low risk [21, 46]. Later we will briefly look at the idea of some of these transformations, called *Online to Batch* conversions.

Before talking about online to batch conversions, let us look at the "deterministic case" assumption made on the PAC-learnability definition. Let (X, D, Y, L) be a statistical learning instance. Note that assuming there exists a (measurable) concept $c \in Y^X$ which correctly maps each query to its label in the training sequence, or even in the points from the "real world", which is done in the definition of risk with respect to a concept c, is restrictive. For example, if X is the set of pairs of weight and height of a certain population, and Y is the set of possible genders, there probably are two people with the same height and weight with different genders. Thus, assuming that there is a function which maps each distinct pair of height and weight in X to a gender in Y is not valid in this case. A better assumption is to suppose that, given the weight and the height of a person, there is a probability distribution over Y describing the probability of a person with these characteristics being of each gender in Y. Not only that, recall that a concept class C is PAC-learnable if, for any given concept $c \in C$, there is an algorithm which generates with high probability a hypothesis with arbitrarily small risk. However, it is often the case that there is no such hypothesis. That is, the PAC learning model pratically supposes that we are in the *realizable case*, that is, the case where there is a hypothesis with no risk. A model which captures the notion of learnability without relying on the deterministic or realizable case assumptions is the idea of *Agnostic PAC-learnability*.

Definition 2.4.2 (Agnostic PAC-learnability)**.** Let $\mathcal{L} \coloneqq (X, D, Y, L)$ be a statistical learning instance and let $\mathcal{H} \subseteq D^X$ be a set of measurable functions. Then $\mathcal{H}$ is **agnostic PAC-learnable** for $\mathcal{L}$ if there is a player oracle $\mathcal{A}\colon \operatorname{Seq}(X \times Y) \to D^X$ for $\mathcal{L}$ and a real polynomial function $p\colon \mathbb{R}^3 \to \mathbb{R}$ such that, for any probability distribution $\mathcal{D}$ over $X \times Y$, any $\varepsilon > 0$, any $\delta > 0$, any $T \in \mathbb{N}$ with $T > p(1/\varepsilon, 1/\delta, T)$, and any training sequence $\boldsymbol{Z}$ of length T sampled from $\mathcal{D}$ on the probability space $(\Omega, \Sigma, \mathbb{P})$, we have

$$\mathbb{P}(\operatorname{risk}_{\mathcal{D}}(\operatorname{TRAIN}(\mathcal{A}, \boldsymbol{Z})) - \inf_{h \in \mathcal{H}} \operatorname{risk}_{\mathcal{D}}(h)) \leq \varepsilon) \geq 1 - \delta.$$

Moreover, if there is an algorithm which implements $\mathcal{A}$ that runs in time $O(p(1/\varepsilon, 1/\delta, T))$, then $\mathcal{H}$ is **efficiently agnostic PAC-learnable** for $\mathcal{L}$. In this definition, we omit the dependence of the minimum size of T and the space required to represent the concept c for the sake of simplicity.

Note that the agnostic PAC-learnability model has more similarities to the idea of regret used in online learning if compared to the original PAC-learning model. In words, a hypothesis set $\mathcal{H}$ is agnostic PAC-learnable if there is an algorithm which, with a big enough training sequence, outputs with high probability a hypothesis in $\mathcal{H}$ which is as accurate as the best hypothesis in $\mathcal{H}$. Thus, a natural question is if we can translate, in some way, player oracles with low regret guarantees for a hypothesis set $\mathcal{H}$ into algorithms for agnostically PAC-learning the set $\mathcal{H}$. It is indeed possible to make such a conversion, with many different techniques proposed for different cases [21, 23, 29, 30, 46]. The details of the conversion are out of the scope of this text, but a brief overview is interesting nonetheless.

Let $\mathcal{L} \coloneqq (X, D, Y, L)$ be an statistical learning instance, let $\mathcal{H} \subseteq D^X$ be a hypothesis set for $\mathcal{L}$, and define the online learning instance $\mathcal{P} \coloneqq \mathcal{L}$. Even though both are the same, it is useful to have different notations for the cases when we look at (X, D, Y, L) as an online learning instance ($\mathcal{P}$) and when we look at it as an statistical learning instance ($\mathcal{L}$). Suppose there is a player oracle PLAYER for the online learning instance $\mathcal{P}$ which suffers low regret against any pair of nature and enemy oracles for $\mathcal{P}$. Finally, let $T \in \mathbb{N}$, let $\mathcal{D}$ be a probability distribution on $X \times Y$, and let

$$\boldsymbol{Z} \coloneqq \langle (X_1, Y_1), \dots, (X_T, Y_T) \rangle$$

be a training sequence for $\mathcal{L}$ of size T based on $\mathcal{D}$ on the probability space $(\Omega, \Sigma, \mathbb{P})$. The idea now is that we want to use, in some way, the oracle PLAYER in sequences of the form $\langle X_1, \dots, X_T \rangle$ and $\langle Y_1, \dots, Y_{t-1} \rangle$ for each $t \in [T]$. However, the latter sequences are random variables and, thus, what we would like to do is actually to define random variables of the form

$$\omega \in \Omega \mapsto \text{PLAYER}(\langle X_1(\omega), \dots, X_t(\omega) \rangle, \langle Y_1(\omega), \dots, Y_{t-1}(\omega) \rangle), \qquad \forall t \in [T]. \tag{2.3}$$

Some subtleties appear when we try to build random variables of this form, which we call *randomized player oracles*. For the sake of simplicity we defer the discussion of some of these subtleties to Section 2.6.2. For now, just assume that PLAYER is such that the functions in (2.3) are random variables, and let us discuss how to obtain, given $\omega \in \Omega$, a hypothesis with low risk.

Thus, let $\omega \in \Omega$ and set

$$x_t \coloneqq X_t(\omega) \qquad \text{and} \qquad y_t \coloneqq Y_t(\omega), \qquad \text{for each } t \in [T].$$

The idea to use PLAYER to obtain a low-risk hypothesis for $\mathcal{L}$ is to look at the regret of PLAYER against a pair of nature and enemy oracles for $\mathcal{P}$ such that, at round $t \in [T]$, nature outputs $x_t \in X$ and the enemy picks $y_t \in Y$. One annoying difference between online learning and statistical learning is that, in OL, the player makes predictions and, in the statistical learning setting, the player outputs an entire hypothesis. Still, note that during the game the player oracle for $\mathcal{P}$ at any round works as a hypothesis: given a query in X, the player oracle computes a prediction in D. Namely, for each $t \in [T]$ we define the hypothesis $h_t \colon X \to D$ given by

$$h_t(x) \coloneqq \text{PLAYER}(\langle x_1, \dots, x_{t-1}, x \rangle, \langle y_1, \dots, y_{t-1} \rangle), \qquad \forall x \in X.$$

In this way, we have that the regret of PLAYER w.r.t. a hypothesis $h \in \mathcal{H}$ and to the sequences $\langle x_1, \dots, x_T \rangle \in X^T$ and $\langle y_1, \dots, y_T \rangle \in Y^T$ (divided by T) is

$$\frac{1}{T} \sum_{t=1}^{T} L(h_t(x_t), y_t) - \frac{1}{T} \sum_{t=1}^{T} L(h(x_t), y_t). \tag{2.4}$$

This way of viewing regret is enlightening. The second sum is an estimator[13] of the risk of h w.r.t. the probability distribution $\mathcal{D}$, usually named *empirical risk*. Not only that, the above difference vanishes as T grows in the case where PLAYER is guaranteed to have sublinear regret in T. Thus, the crux of most of the techniques to convert online player oracles to statistical learning algorithms is how to build a final hypothesis from the set $h_1, h_2, \dots, h_T$ of hypotheses generated by the player oracle in a way that some guarantee on the risk can be derived from bonds on (2.4). Interestingly, the hypothesis h_T, the last one picked by the player in the game, does not necessarily has (with high probability) low risk since it is harder to relate it to the expression from (2.4). Some techniques used to build a final hypothesis are averaging the hypotheses (if possible) and picking a hypothesis from $h_1, \dots, h_T$ uniformly at random.

2.5 Online Convex Optimization

In this section we describe the *online convex optimization* setting, which may be seen as a special case of the online learning setting. Let us first describe the setting in an intuitive way, leaving the formalization for later. Recall from Section 1.1 that, throughout the text, $\mathbb{E}$ denotes an arbitrary euclidean space (finite-dimensional real vector space equipped with an inner product), and we denote its inner product by $\langle \cdot, \cdot \rangle$.

Similarly to the online learning setting, the OCO framework is a game played in rounds by a player and its enemy. At round t, the player picks a point x_t from a convex set $X \subseteq \mathbb{E}$, and the enemy picks, simultaneously, a convex function[14] $f_t \colon \mathbb{E} \to (-\infty, +\infty]$ from some set $\mathcal{F}$. At the end of the round, the player suffers the loss $f_t(x_t)$. Similarly to the online learning setting, at round t the player knows the previous functions $f_1, \dots, f_{t-1} \in \mathcal{F}$ played by the enemy, and the enemy knows the previous points $x_1, \dots, x_{t-1} \in X$ picked by the player. The goal of the player is to minimize, in some sense, the cumulative loss suffered along a sequence of T rounds. As one may already guess based on the results and discussions from Section 2.3, minimizing the raw cumulative loss is impossible in the case of adversarial enemy oracles. Thus, we shall define regret for OCO in a way analogous to the regret of the online learning setting. Let us now formalize this setting.

Definition 2.5.1 (Online (convex) optimization instance)**.** An **online optimization instance** is a pair $(X, \mathcal{F})$ where $X \subseteq \mathbb{E}$ is nonempty and $\mathcal{F} \subseteq (-\infty, +\infty]^X$ is a set of functions such that[15] $X \subseteq \operatorname{dom} f$ for every $f \in \mathcal{F}$, and it is an **online convex optimizaton (OCO) instance** if X and each $f \in \mathcal{F}$ are convex.

Let $\mathcal{C} := (X, \mathcal{F})$ be an online optimization instance. We associate with $\mathcal{C}$ the function[16] $\mathrm{OCO}_{\mathcal{C}}$, which takes the following parameters:

- PLAYER: $\mathrm{Seq}(\mathcal{F}) \to X$, which we call a **player oracle**;
- ENEMY: $\mathrm{Seq}(X) \to \mathcal{F}$, which we call an **enemy oracle**;
- $T \in \mathbb{N}$, which we call the number of **rounds** or **iterations**,

and outputs a point in $\mathrm{Seq}(X) \times \mathrm{Seq}(\mathcal{F})$. As in the case of online learning, we define the function $\mathrm{OCO}_{\mathcal{C}}$ in an iterative way in Algorithm 2.3. For $t \in \mathbb{N} \setminus \{0\}$ we consider to be the t-th *round* the

[13] Recall that the latter sum is a random variable by taking (x_t, y_t) as $\omega \in \Omega \mapsto (X_t(\omega), Y_t(\omega))$ for each $t \in [T]$, and we only fixed ω here to ease the discussion.

[14] We will use extended-real-valued functions, a convention justified in Chapter 3.

[15] We impose this condition on the effective domain of the functions since it would be mildly unfair to the player to make her suffer infinite loss in a single round.

[16] Although this function can be used for non-convex online optimization instances, we stick with the name OCO since the convex case is our main focus, with sporadic mentions to online optimization in its general form.

```
Algorithm 2.3 Definition of $\mathrm{OCO}_{\mathcal{C}}(\mathrm{PLAYER}, \mathrm{ENEMY}, T)$
Input:
    (i) An OCO instance $\mathcal{C}$,
    (ii) player and enemy oracles for $\mathcal{C}$ denoted by PLAYER and ENEMY, respectively, and
    (iii) a number $T \in \mathbb{N}$ of rounds.
Output: $(\boldsymbol{x}, \boldsymbol{f}) \in X^T \times \mathcal{F}^T$.
  for $t = 1$ to $T$ do
      $x_t \leftarrow \mathrm{PLAYER}\big(\langle f_1, \dots, f_{t-1} \rangle\big)$
      $f_t \leftarrow \mathrm{ENEMY}\big(\langle x_1, \dots, x_{t-1} \rangle\big)$
  return $(\boldsymbol{x}, \boldsymbol{f})$
```

iteration of Algorithm 2.3 in which are defined the t-th elements of the sequence of points picked by the oracles. Even though this is intuitive in Algorithm 2.3, one may get confused later in the text when we define more complex algorithms and start talking about its actions on round t.

Definition 2.5.2 (Regret for online convex optimization)**.** Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an online optimization instance and let $T \in \mathbb{N}$. The **regret** of $\boldsymbol{x} \in X^T$ with respect to $\boldsymbol{f} \in \mathcal{F}^T$ and to a point $u \in \mathbb{E}$ is

$$\mathrm{Regret}(\boldsymbol{x}, \boldsymbol{f}, u) \coloneqq \sum_{t=1}^{T} \big(f_t(x_t) - f_t(u)\big),$$

and the **regret** of $\boldsymbol{x} \in X^T$ w.r.t. $\boldsymbol{f} \in \mathcal{F}^T$ and to a set $U \subseteq \mathbb{E}$ is

$$\mathrm{Regret}(\boldsymbol{x}, \boldsymbol{f}, U) \coloneqq \sup_{u \in U} \mathrm{Regret}(\boldsymbol{x}, \boldsymbol{f}, u).$$

Moreover, let PLAYER be a player oracle for $\mathcal{C}$ and define $x'_t \coloneqq \mathrm{PLAYER}(\langle f_1, \dots, f_{t-1} \rangle)$ for each $t \in [T]$. Then, the **regret** of PLAYER with respect to $\boldsymbol{f} \in \mathcal{F}^T$ and to $u \in \mathbb{E}$ is

$$\mathrm{Regret}(\mathrm{PLAYER}, \boldsymbol{f}, u) \coloneqq \mathrm{Regret}(\boldsymbol{x}', \boldsymbol{f}, u),$$

and the regret of PLAYER w.r.t. $\boldsymbol{f} \in \mathcal{F}^T$ and to a set $U \subseteq \mathbb{E}$ is

$$\mathrm{Regret}(\mathrm{PLAYER}, \boldsymbol{f}, U) \coloneqq \mathrm{Regret}(\boldsymbol{x}', \boldsymbol{f}, U).$$

Finally, let ENEMY be player and enemy oracle for $\mathcal{C}$ and define the pair of sequences $(\boldsymbol{x}'', \boldsymbol{f}') \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{PLAYER}, \mathrm{ENEMY}, T)$. Then, the **regret** of PLAYER in T rounds w.r.t. ENEMY and to $u \in \mathbb{E}$ is

$$\mathrm{Regret}_T^{\mathcal{C}}(\mathrm{PLAYER}, \mathrm{ENEMY}, u) \coloneqq \mathrm{Regret}(\boldsymbol{x}'', \boldsymbol{f}', u),$$

and the **regret** of PLAYER in T rounds w.r.t. ENEMY and to $U \subseteq \mathbb{E}$ is

$$\mathrm{Regret}_T^{\mathcal{C}}(\mathrm{PLAYER}, \mathrm{ENEMY}, U) \coloneqq \mathrm{Regret}(\boldsymbol{x}'', \boldsymbol{f}', U),$$

where we omit $\mathcal{C}$ from the notation of regret when it is clear from context.

It is interesting to note that the regret for online optimization is computed comparing the loss of the player with that of fixed points, whereas in the case of online learning, regret is computed with respect to the loss of other functions (or hypotheses). Although this may seem arbitrary at first, the next theorem shows that the online optimization framework is a special case of the online learning setting with a nature oracle which is just a constant function. On account of this nature oracle, each hypothesis in the regret from online learning will evaluate to only one point. Thus, the regret for these online learning problems is exactly the regret defined here for online optimization instances.

Theorem 2.5.3. let $\mathcal{C} := (X, \mathcal{F})$ be an online optimization instance and define the online learning instance $\mathcal{P} := (\{0\}, X, \mathcal{F}, L)$, where[17] $L(x, f) := f(x)$ for every $(x, f) \in X \times \mathcal{F}$. Moreover, let $\mathrm{PLAYER}_{\mathrm{OCO}}$ and $\mathrm{ENEMY}_{\mathrm{OCO}}$ be player and enemy oracles for $\mathcal{C}$, respectively, and let $T \in \mathbb{N}$. Then, there are nature, player, and enemy oracles NATURE, $\mathrm{PLAYER}_{\mathrm{OL}}$, and $\mathrm{ENEMY}_{\mathrm{OL}}$ for $\mathcal{P}$, respectively, such that

$$(\mathbf{0}, \boldsymbol{x}, \boldsymbol{f}) = \mathrm{OL}_{\mathcal{P}}(\mathrm{NATURE}, \mathrm{PLAYER}_{\mathrm{OL}}, \mathrm{ENEMY}_{\mathrm{OL}}, T), \tag{2.5}$$

where $(\boldsymbol{x}, \boldsymbol{f}) := \mathrm{OCO}_{\mathcal{C}}(\mathrm{PLAYER}_{\mathrm{OCO}}, \mathrm{ENEMY}_{\mathrm{OCO}}, T)$ and $\mathbf{0}$ is a properly-sized sequence with all entries equal to 0. Additionally, for every $u \in \mathbb{E}$ we have $\mathrm{Regret}(\mathrm{PLAYER}_{\mathrm{OCO}}, \boldsymbol{f}, u) = \mathrm{Regret}(\mathbf{0}, \mathrm{PLAYER}_{\mathrm{OL}}, \boldsymbol{f}, h_u, L)$, where $h_u(0) := u$.

Proof. Define the nature, player, and enemy oracles NATURE, $\mathrm{PLAYER}_{\mathrm{OL}}$, and $\mathrm{ENEMY}_{\mathrm{OL}}$ for $\mathcal{P}$ by

$$\begin{aligned}
\mathrm{NATURE}(t) &:= 0 && \text{for every } t \in \mathbb{N}, \\
\mathrm{PLAYER}_{\mathrm{OL}}(\mathbf{0}, \boldsymbol{f}) &:= \mathrm{PLAYER}_{\mathrm{OCO}}(\boldsymbol{f}) && \text{for every } t \in \mathbb{N} \setminus \{0\} \text{ and } \boldsymbol{f} \in \mathcal{F}^{t-1}, \text{ and} \\
\mathrm{ENEMY}_{\mathrm{OL}}(\mathbf{0}, \boldsymbol{x}) &:= \mathrm{ENEMY}_{\mathrm{OCO}}(\boldsymbol{x}) && \text{for every } t \in \mathbb{N} \setminus \{0\} \text{ and } \boldsymbol{x} \in X^{t-1}.
\end{aligned}$$

By the definition of these oracles and of the functions $\mathrm{OL}_{\mathcal{P}}$ and $\mathrm{OCO}_{\mathcal{C}}$, it is clear that (2.5) holds. Moreover, if $\boldsymbol{x}$ and $\boldsymbol{y}$ are as in (2.5), if $u \in \mathbb{E}$, and if $h_u(0) := u$, then

$$\begin{aligned}
\mathrm{Regret}(\mathrm{PLAYER}_{\mathrm{OCO}}, \boldsymbol{f}, u) &= \sum_{t=1}^{T} (f_t(x_t) - f_t(u)) = \sum_{t=1}^{T} (L(x_t, f_t) - L(u, f_t)) \\
&= \sum_{t=1}^{T} (L(x_t, f_t) - L(h_u(0), f_t)) \\
&= \mathrm{Regret}(\mathbf{0}, \mathrm{PLAYER}_{\mathrm{OL}}, \boldsymbol{f}, h_u, L).
\end{aligned}$$

□

The above result formally proves what we had commented earlier: online (convex) optimization is a special case of online learning. Still, what we want to do is to model problems from online learning into the online *convex* optimization framework. The reason is that, as we are going to see in later chapters, there are player oracles for online convex optimization instances which, under some mild assumptions, have regret upper bounds which grow sublinearly with the number of rounds. Some problems from the online learning setting fit almost seamlessly into the online optimization setting. For example, the next proposition shows how to model online linear regression as an online optimization instance. Not only that, one may note that if the loss function L in the proposition is convex w.r.t. its first argument (that is, $L(\cdot, \alpha)$ is convex for any $\alpha \in \mathbb{R}$), the online optimization instance given is actually an OCO instance. Later, we will see one reduction of a problem from OL to OCO where convexity is essential.

Proposition 2.5.4. Let $\mathcal{P} := (\mathbb{R}^d, \mathbb{R}, \mathbb{R}, L)$ be an instance of online linear regression and let $\mathcal{C} := (\mathbb{R}^d, \mathcal{F})$ be a online optimization instance where

$$\mathcal{F} := \{\, w \in \mathbb{R}^d \mapsto L(w^{\mathsf{T}} x, y) : x \in \mathbb{R}^d, y \in \mathbb{R}\}.$$

[17] One may note that there are many instances of online learning in which the loss function only evaluates one of the arguments at the other, which is the case here.

Finally, let $\mathrm{PLAYER}_{\mathrm{OCO}}$ be a player oracle for $\mathcal{C}$, let $W \subseteq \mathbb{R}^d$, and set $\mathcal{H} \coloneqq \{\, x \in \mathbb{R}^d \mapsto w^\mathsf{T} x : w \in W \}$. Then, there exists a player oracle $\mathrm{PLAYER}_{\mathrm{OL}}$ for $\mathcal{P}$ such that, for any $T \in \mathbb{N}$ and any sequences $\boldsymbol{x} \in (\mathbb{R}^d)^T$ and $\boldsymbol{y} \in (\mathbb{R})^T$, there is $\boldsymbol{f} \in \mathcal{F}^T$ such that $\mathrm{Regret}(\boldsymbol{x}, \mathrm{PLAYER}_{\mathrm{OL}}, \boldsymbol{y}, \mathcal{H}) = \mathrm{Regret}(\mathrm{PLAYER}_{\mathrm{OCO}}, \boldsymbol{f}, W)$.

Proof. For every $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$, define the function $f_{(x,y)} \colon \mathbb{R}^d \to \mathbb{R}$ by $f_{(x,y)}(w) \coloneqq L(w^\mathsf{T} x, y)$ for every $w \in \mathbb{R}^d$. Moreover, define the player oracle $\mathrm{PLAYER}_{\mathrm{OL}}$ for $\mathcal{P}$ by

$$\mathrm{PLAYER}_{\mathrm{OL}}(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \mathrm{PLAYER}_{\mathrm{OCO}}\big(\langle f_{(x_1,y_1)}, \dots, f_{(x_{T-1},y_{T-1})}\rangle\big)^\mathsf{T} x_T,$$

for every $T \in \mathbb{N}$, and all sequences $\boldsymbol{x} \in (\mathbb{R}^d)^T$ and $\boldsymbol{y} \in (\mathbb{R})^T$.

Set $h_w(x) \coloneqq w^\mathsf{T} x$ for each $x \in \mathbb{R}^d$. Let $T \in \mathbb{N}$, let $\boldsymbol{x} \in (\mathbb{R}^d)^T$, let $\boldsymbol{y} \in \mathbb{R}^T$, and define

$$\begin{aligned}
d_t &\coloneqq \mathrm{PLAYER}_{\mathrm{OL}}(\boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t-1}) && \text{for each } t \in [T],\\
R_{\mathrm{OL}} &\coloneqq \mathrm{Regret}(\boldsymbol{x}, \mathrm{PLAYER}_{\mathrm{OL}}, \boldsymbol{y}, h_w, L),\\
f_t &\coloneqq f_{(x_t,y_t)} && \text{for each } t \in [T],\\
w_t &\coloneqq \mathrm{PLAYER}_{\mathrm{OCO}}(\boldsymbol{f}_{1:t-1}) && \text{for each } t \in [T].
\end{aligned}$$

Let $w \in \mathbb{R}^d$. In this case, we have,

$$\begin{aligned}
R_{\mathrm{OL}} &= \sum_{t=1}^T L(d_t, y_t) - \sum_{t=1}^T L(h_w(x_t), y_t) = \sum_{t=1}^T L(w_t^\mathsf{T} x_t, y_t) - \sum_{t=1}^T L(w^\mathsf{T} x_t, y_t)\\
&= \sum_{t=1}^T f_{(x_t,y_t)}(w_t) - \sum_{t=1}^T f_{(x_t,y_t)}(w) = \mathrm{Regret}(\mathrm{PLAYER}_{\mathrm{OCO}}, \boldsymbol{f}_{(\boldsymbol{x},\boldsymbol{y})}, w).
\end{aligned}$$

□

Let us look at one final example of an online learning problem which can be easily modeled as an online convex optimization problem. One may note that in this case convexity is fundamental for the reduction to yield an interesting relation between the regret of both instances. Consider an instance of the prediction with expert advice problem $\mathcal{P} \coloneqq (A^E, A, Y, L)$ such that A is convex and L is convex w.r.t. its first argument. This case is interesting because the player can pick a convex combination of the experts' advice and still have some information about the loss incurred by this point. Without any structure on A, the player is virtually forced to decide to follow only one of the experts at each round (unless the player has some kind of prior information about the game), and the enemy can exploit this fact, as we have seen earlier in the impossibility results. The next proposition shows that player oracles to a closely related online convex optimization problem yield player oracles for this convex version of the prediction with expert advice problem. Recall from Section 1.1 that if E is a finite set, then $\Delta_E \coloneqq \{\, p \in [0,1]^E : \mathbb{1}^\mathsf{T} p = 1\}$ denotes the simplex on the space $\mathbb{R}^E$.

Proposition 2.5.5. Let $\mathcal{P} \coloneqq (A^E, A, Y, L)$ be an instance of prediction with expert advice such that $A \subseteq \mathbb{E}$ is a convex set and $L \colon A \times Y \to \mathbb{R}$ is convex w.r.t. its first argument[18], and let $\mathcal{C} \coloneqq (\Delta_E, \mathcal{F})$ be an OCO problem where

$$\mathcal{F} \coloneqq \{\, p \in \mathbb{R}^E \mapsto p^\mathsf{T} c : c \in [-1,1]^E \}.$$

Finally, let $\mathrm{PLAYER}_{\mathrm{OCO}}$ be a player oracle for $\mathcal{C}$, let $U \coloneqq \{\, e_i \in \{0,1\}^E : i \in E\}$, and define the hypothesis set $\mathcal{H} \coloneqq \{\, x \in A^E \mapsto x(i) : i \in E\}$. Then, there exists a player oracle $\mathrm{PLAYER}_{\mathrm{OL}}$ for $\mathcal{P}$ such that, for any $T \in \mathbb{N}$ and any sequences $\boldsymbol{x} \in (A^E)^T$ and $\boldsymbol{y} \in Y^T$, there is $\boldsymbol{f} \in \mathcal{F}^T$ such that $\mathrm{Regret}(\boldsymbol{x}, \mathrm{PLAYER}_{\mathrm{OL}}, \boldsymbol{y}, \mathcal{H}) \leq \mathrm{Regret}(\mathrm{PLAYER}_{\mathrm{OCO}}, \boldsymbol{f}, U)$.

[18] That is, $L(\cdot, y)$ is convex for any $y \in Y$.

Proof. For every $x \in A^E$ and $y \in Y$, define $c(x,y) \in [-1,1]^E$ by

$$(c(x,y))_e \coloneqq L(x(e), y), \qquad \forall e \in E.$$

Define the player oracle $\mathrm{PLAYER}_{\mathrm{OL}}$ for $\mathcal{P}$ given for every $T \in \mathbb{N}$, $\boldsymbol{x} \in X^T$, and $\boldsymbol{y} \in Y^{T-1}$ by

$$\begin{aligned} \mathrm{PLAYER}_{\mathrm{OL}}(\boldsymbol{x}, \boldsymbol{y}) \coloneqq \mathrm{PLAYER}_{\mathrm{OCO}}(\boldsymbol{f}')^{\mathsf{T}} x_T, \text{ where } \boldsymbol{f}' \in \mathcal{F}^{T-1} \text{ is given by} \\ f'_t(z) \coloneqq c(x_t, y_t)^{\mathsf{T}} z \text{ for each } z \in \mathbb{R}^E \text{ and } t \in \{1, \dots, T-1\}. \end{aligned}$$

Let $T \in \mathbb{N}$, and let both $\boldsymbol{x} \in (A^E)^T$ and $\boldsymbol{y} \in Y^T$ be arbitrary sequences of length T. Moreover, define

$$\begin{aligned} d_t &\coloneqq \mathrm{PLAYER}_{\mathrm{OL}}(\boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t-1}), && \forall t \in [T], \\ f_t(z) &\coloneqq c(x_t, y_t)^{\mathsf{T}} z && \forall z \in \mathbb{R}^E, \forall t \in [T], \\ z_t &\coloneqq \mathrm{PLAYER}_{\mathrm{OCO}}(\langle f_1, \dots, f_{t-1} \rangle) && \forall z \in \mathbb{R}^E, \forall t \in [T]. \end{aligned}$$

Finally, let $i^* \in E$, define $h(x) \coloneqq x(i^*)$ for every $x \in A^E$, and set $R_{\mathrm{OL}} \coloneqq \mathrm{Regret}(\boldsymbol{x}, \mathrm{PLAYER}_{\mathrm{OL}}, \boldsymbol{y}, h)$. Then,

$$\begin{aligned} R_{\mathrm{OL}} &= \sum_{t=1}^{T} [L(d_t, y_t) - L(x_t(i^*), y_t)] \\ &= \sum_{t=1}^{T} [L(z_t^{\mathsf{T}} x_t, y_t) - L(x_t(i^*), y_t)] && \text{by the def. of } \mathrm{PLAYER}_{\mathrm{OL}} \\ &\leq \sum_{t=1}^{T} \left[\Big(\sum_{e \in E} z_t(e) L(x_t(e), y_t) \Big) - L(x_t(i^*), y_t) \right] && \text{by the convexity of } L(\cdot, y_t) \\ &= \sum_{t=1}^{T} [c(x_t, y_t)^{\mathsf{T}} z_t - c(x_t, y_t)^{\mathsf{T}} e_{i^*}] && \text{by the def. of } c(x_t, y_t) \\ &= \sum_{t=1}^{T} [f_t(z_t) - f_t(e_{i^*})] = \mathrm{Regret}(\mathrm{PLAYER}_{\mathrm{OCO}}, \boldsymbol{f}, e_{i^*}) && \text{by the def. of } f_t. \quad \square \end{aligned}$$

2.6 From Online Learning to Online Convex Optimization

As we have seen on the previous section, online optimization is a special case of online learning. However, this remark is of no practical help since in simple online learning instances it is already impossible for the player to attain sublinear regret (see Proposition 2.3.3, for example). Thus, it is more interesting to investigate how to go the other way around: model online learning instances into online optimization instances, or at least use algorithms from the latter to tackle instances from the former. In fact, we want to devise player strategies for OL problems using algorithms from online *convex* optimization since, for the latter, there are algorithms for the player which attain sublinear regret under some mild assumptions. In this section we will look at two major strategies which allow us to use players for OCO instances to obtain players for OL instances. The first is to use, at each round, a convex function to emulate the loss function with the current query point from nature, and use such functions in an OCO player oracle. Such convex functions are sometimes called *surrogate loss functions* [69]. The second technique which we look at is *randomization*, that is, letting the player randomize her choices with random bits which the enemy does not have access to. However, this latter technique brings some subtleties to the setting which need to be handled carefully and which we discuss later in this section.

2.6.1 Surrogate Loss Functions

An online learning instance may be harder to model as an OCO instance if the loss function is not convex w.r.t. its first argument. For example, Proposition 2.5.4 shows that linear regression fits seamlessly into the online optimization framework. However, if the loss function is not convex w.r.t. its first argument, we cannot guarantee that the functions from $\mathcal{F}$ as defined in the statement of that proposition are convex. Thus, the oracles devised for OCO instances would not be usable in this case. Still, not all hope is lost. In Proposition 2.5.4, the functions of the online convex optimization instance are built in a way that the regrets of the OL and OCO instances (each with the proper arguments) are equal. But we do not need equality to hold, because we only want to upper bound the regret of the OL instance by the regret of an OCO instance, as in Proposition 2.5.5, since for the latter there usually are player oracles with low regret guarantees. Therefore, building an OCO instance with functions which upper bound the loss of the OL instance, the *surrogate loss functions*, may already yield good regret guarantees. Before giving a rough idea of this technique in the general case, let us look at an application of this idea to the problem of online binary classification.

Proposition 2.6.1. Let[19] $\mathcal{P} := (X, \{0,1\}, \{0,1\}, L)$ be an online binary classification instance with $L(d,y) := [d \neq y]$ for every $d, y \in \{0,1\}$, and let $\mathcal{H} \subseteq \{0,1\}^X$ be a finite hypothesis set for $\mathcal{P}$. Moreover, define the set $\mathcal{F} := \{\, w \in \mathbb{R}^{\mathcal{H}} \mapsto |v^{\mathsf{T}} w - \alpha| : v \in \{0,1\}^{\mathcal{H}}, \alpha \in \{0,1\}\}$, the online convex optimization instance $\mathcal{C} := (\Delta_{\mathcal{H}}, \mathcal{F})$, and let $\mathrm{PLAYER}_{\mathrm{OCO}}$ be a player oracle for $\mathcal{C}$. Then, there is a player oracle $\mathrm{PLAYER}_{\mathrm{OL}}$ for $\mathcal{P}$ such that, for every $T \in \mathbb{N}$, and all sequences $\boldsymbol{x} \in X^T$ and $\boldsymbol{y} \in \{0,1\}^T$, there is $\boldsymbol{f} \in \mathcal{F}^T$ such that

$$\mathrm{Regret}(\boldsymbol{x}, \mathrm{PLAYER}_{\mathrm{OL}}, \boldsymbol{y}, h) \leq \mathrm{Regret}(\mathrm{PLAYER}_{\mathrm{OCO}}, \boldsymbol{f}, e_h) + \sum_{t=1}^{T} L(h(x_t), y_t), \qquad \forall h \in \mathcal{H}. \quad (2.6)$$

Proof. Define $v \colon X \to \mathbb{R}^{\mathcal{H}}$ by $v(x)_h := h(x)$ for every $x \in X$ and $h \in \mathcal{H}$. Set

$$F(x, y, w) := 2|w^{\mathsf{T}} v(x) - y|, \qquad \forall w \in \mathbb{R}^{\mathcal{H}}, \forall x \in X, \forall y \in Y.$$

Moreover, define the player oracle $\mathrm{PLAYER}_{\mathrm{OL}}$ for every $T \in \mathbb{N} \setminus \{0\}$, and all sequences $\boldsymbol{x} \in X^T$ and $\boldsymbol{y} \in \{0,1\}^{T-1}$ by

$$\begin{aligned} \mathrm{PLAYER}_{\mathrm{OL}}(\boldsymbol{x}, \boldsymbol{y}) &:= \big[\mathrm{PLAYER}_{\mathrm{OCO}}(\boldsymbol{f})^{\mathsf{T}} v(x_T) \geq 1/2\big], \\ \text{where } f_t(w) &:= \big[\mathrm{PLAYER}_{\mathrm{OL}}(\boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t-1}) \neq y_t\big] F(x_t, y_t, \cdot) \qquad \forall t \in \{1, \dots, T-1\}. \end{aligned}$$

Let $T \in \mathbb{N} \setminus \{0\}$, let $\boldsymbol{x} \in X^T$, and let $\boldsymbol{y} \in \{0,1\}^T$. Additionally,define, for every $t \in [T]$,

$$\begin{aligned} d_t &:= \mathrm{PLAYER}_{\mathrm{OL}}(\boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t-1}) \\ f_t &:= [d_t \neq y_t] F(x_t, y_t, \cdot), \\ \text{and } w_t &:= \mathrm{PLAYER}_{\mathrm{OCO}}(\boldsymbol{f}_{1:t-1}). \end{aligned}$$

Let $t \in [T]$. If $d_t = y_t$, we have $L(d_t, y_t) = 0 = f_t(w_t)$. On the other hand, suppose $d_t \neq y_t$. By definition, we have $d_t = [w_t^{\mathsf{T}} v(x_t) \geq 1/2]$. Thus, either $w_t^{\mathsf{T}} v(x_t) \geq 1/2$ and $y_t = 0$, or $w_t^{\mathsf{T}} v(x_t) < 1/2$ and $y_t = 1$. In any of these cases, we have $|w_t^{\mathsf{T}} v(x_t) - y_t| \geq 1/2$, that is, $f_t(w_t) = 2|w_t^{\mathsf{T}} v(x_t) - y_t| \geq 1 = L(d_t, y_t)$. Therefore,

$$f_t(w_t) \geq L(d_t, y_t), \qquad \forall t \in [T]. \quad (2.7)$$

[19] Note that X does not need to be finite in this case.

In words, the functions $f_1, \dots, f_T$ at the points picked by $\text{PLAYER}_{\text{OCO}}$ upper-bound the losses of $\text{PLAYER}_{\text{OL}}$. Moreover, for every $t \in [T]$ we have

$$f_t(e_h) = [d_t \neq y_t] 2 |e_h^{\mathsf{T}} v(x_t) - y_t| = [d_t \neq y_t] 2 |h(x_t) - y_t| \leq 2L(h(x_t), y_t), \qquad \forall h \in \mathcal{H}. \tag{2.8}$$

Finally, let $h \in \mathcal{H}$ and set $R_{\text{OL}} \coloneqq \text{Regret}(\boldsymbol{x}, \text{PLAYER}_{\text{OL}}, \boldsymbol{y}, h)$. Therefore,

$$\begin{aligned} R_{\text{OL}} = \sum_{t=1}^{T} (L(d_t, y_t) - L(h(x_t), y_t)) &\overset{(2.7)}{\leq} \sum_{t=1}^{T} (f_t(w_t) - 2L(h(x_t), y_t)) + \sum_{t=1}^{T} L(h(x_t), y_t) \\ &\overset{(2.8)}{\leq} \text{Regret}(\text{PLAYER}_{\text{OCO}}, \boldsymbol{f}, e_h) + \sum_{t=1}^{T} L(h(x_t), y_t). \end{aligned} \qquad \square$$

One may be wondering if the extraneous loss term on the above bound ruins its usefulness. Note that in the realizable case (see Section 2.1), there is a hypothesis h^* in the hypothesis set which zeroes out this extra loss term. Thus, by taking $h = h^*$ in (2.6), the proposition shows that the regret[20] of the OCO player w.r.t. the point e_{h^*} upper bounds the number of mistakes made by the player from the online learning game. Additionally, for each $h \in \mathcal{H}$ one may note that the bigger the value of $\sum_{t=1}^{T} L(h(x_t), y_t)$ in the above proposition, the harder it is to interpret or use the above bound. Thus, if there is no hypothesis which models reasonably well the queries from nature and the answers from the OL enemy, then the above bound is not very informative.

Let us now give a rough idea of the technique for the general case. One may find it helpful to keep in mind the above proof while reading the remainder of this section. Let $\mathcal{P} \coloneqq (X, D, Y, L)$ be an online learning instance, let $\mathcal{H} \subseteq D^X$ be a hypothesis set, and let $\mathcal{C} \coloneqq (S, \mathcal{F})$ be an online convex optimization instance, where $S \subseteq \mathbb{E}$. A strategy of building surrogate functions for $\mathcal{P}$ on $\mathcal{C}$ goes as follows: first, we know a player oracle $\text{PLAYER}_{\text{OCO}}$ for $\mathcal{C}$. Then, we define a player oracle $\text{PLAYER}_{\text{OL}}$ for $\mathcal{P}$ for every $T \in \mathbb{N} \setminus \{0\}$, every $\boldsymbol{x} \in X^T$, and every $\boldsymbol{y} \in Y^T$, by

$$\text{PLAYER}_{\text{OL}}(\boldsymbol{x}, \boldsymbol{y}_{1:T-1}) \coloneqq G(\text{PLAYER}_{\text{OCO}}(\boldsymbol{f}_{1:T-1}),$$

where $G \colon S \to D$ and $\boldsymbol{f} \coloneqq \boldsymbol{f}_{(\boldsymbol{x},\boldsymbol{y})} \in \mathcal{F}^T$ are carefully crafted to upper bound the loss function on the points picked by the player (and may depend on the hypothesis set $\mathcal{H}$, as we shall see). More specifically, for every $T \in \mathbb{N} \setminus \{0\}$, and all sequences $\boldsymbol{x} \in X^T$ and $\boldsymbol{y} \in Y^T$, we want the following to hold:

$$L(d_t, y_t) \leq f_t(s_t), \qquad \forall t \in [T], \tag{2.9}$$

where, for every $t \in [T]$,

$$\begin{aligned} f_t &\coloneqq (\boldsymbol{f}_{(\boldsymbol{x},\boldsymbol{y})})_t, \\ s_t &\coloneqq \text{PLAYER}_{\text{OCO}}(\langle f_1, \dots, f_{t-1} \rangle), \\ d_t &\coloneqq \text{PLAYER}_{\text{OL}}(\langle x_1, \dots, x_t \rangle, \langle y_1, \dots, y_{t-1} \rangle). \end{aligned} \tag{2.10}$$

Let NATURE and ENEMY_{OL} be nature and enemy oracles for $\mathcal{P}$, define

$$(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}) \coloneqq \text{OL}_{\mathcal{P}}(\text{NATURE}, \text{PLAYER}_{\text{OL}}, \text{ENEMY}_{\text{OL}}),$$

[20] Or, actually, the loss of the OCO player, since $f_t(e_{h^*}) = 0$ for each $t \in \mathbb{N}$, where $T \in \mathbb{N}$, $\mathcal{H}$, and $\boldsymbol{f} \in \mathbb{R}^{\mathcal{H}}$ are as in Proposition 2.6.1.

and define the sequences $\boldsymbol{f} \in \mathcal{F}^T$ and $\boldsymbol{s} \in S^T$ as in (2.10). If (2.9) holds, the loss of $\mathrm{PLAYER_{OL}}$ can be upper bounded by the loss of $\mathrm{PLAYER_{OCO}}$ against an enemy which plays the sequence of functions $\boldsymbol{f}$ (which depends on the sequences $\boldsymbol{x}$ and $\boldsymbol{y}$ from the OL game), that is,

$$\sum_{t=1}^{T} L(d_t, y_t) \leq \sum_{t=1}^{T} f_t(s_t).$$

Although this guarantees a bound on the raw cumulative loss of $\mathrm{PLAYER_{OL}}$, the property on (2.9) is not enough to bound the regret since we do not have a bound on the cumulative loss of the hypotheses in $\mathcal{H}$, that is, a lower bound on $\sum_{t=1}^{T} L(h(x_t), y_t))$ for each $h \in \mathcal{H}$. To obtain a bound on the regret, the functions in the sequence $\boldsymbol{f}$ must *lower bound* the loss of the hypotheses from $\mathcal{H}$ when evaluated in some set $U_{\mathcal{H}} \subseteq \mathbb{E}$. That is, for each $h \in \mathcal{H}$, there must be $u \in \mathbb{E}$ such that

$$L(h(x_t), y_t) \geq f_t(u), \qquad \forall t \in [T]. \tag{2.11}$$

If the above inequality holds, then, it together with (2.9) imply that there is $U_{\mathcal{H}} \subseteq \mathbb{E}$ such that

$$\mathrm{Regret}(\boldsymbol{x}, \mathrm{PLAYER_{OL}}, \boldsymbol{y}, \mathcal{H}) \leq \mathrm{Regret}(\mathrm{PLAYER_{OCO}}, \boldsymbol{f}, U_{\mathcal{H}}).$$

For example, in Proposition 2.6.1, we have (2.7), which is analogous to (2.9). However, in the proposition we were not able to cleanly obtain an inequality like (2.11). In fact, note that the extraneous cumulative loss term on the bound from Proposition 2.6.1 originates from (2.8), which is a weaker version of (2.11).

2.6.2 Randomization

Recall that Proposition 2.3.3 shows that for simple instances of prediction with expert advice it is impossible to find a player oracle which attains regret sublinear in the number of rounds. That proposition relies on the possibility of the enemy predicting exactly which is the next player prediction in the online learning setting. What if, instead of making predictions deterministically, the player decided only on a probability distribution over her possible predictions, and left the actual choice for randomness to take care of? Let us take the online binary classification case as an example. Given a query, the player may think that there is a 60% chance of it being from the class 1, for example. If the player decides to deterministically pick the class in which she is more confident, the enemy will be able to exploit that. If, instead, we flip a biased coin which the enemy *does not have access to*, we take away part of this advantage. Randomizing the choices seems even more appealing when we have a greater number of choices and the confidence that the player has about each choice is small. In view of this discussion, we can change the model to allow the player oracle in the online learning setting to randomize its predictions, and restrict the access to information of the enemy oracle: it will not have access to the "random bits" played. For example, consider a player oracle in the prediction with expert advice problem that, instead of choosing an expert deterministically, samples one from some probability distribution. The key here is that the enemy is able to simulate this player oracle and see the probability of each expert being sampled that round, but the enemy does not know which expert gets sampled before making a decision.

Formally, let $\mathcal{P} := (X, Y, D, L)$ be an online learning instance such that Y and D are each equipped with a σ-algebra[21] and such that L is measurable. Additionally, let $(\Omega, \Sigma, \mathbb{P})$ be a probability space.

[21] If D is a finite set or $\mathbb{R}$, there are natural σ-algebras over them. Namely, the power set of D and the (Borel) σ-algebra generated by the open intervals in the real line, respectively. Whenever we are in one of these cases, we assume D is equipped with such a σ-algebra, unless stated otherwise.

A **randomized player oracle** for $\mathcal{P}$ is a function $\overline{\text{PLAYER}}\colon \text{Seq}(X) \times \text{Seq}(Y) \to D^{\Omega}$ such that every function $F\colon \Omega \to D$ in the image of $\overline{\text{PLAYER}}$ is measurable, that is, F is a random variable over $(\Omega, \Sigma, \mathbb{P})$ that takes values in D. In Algorithm 2.4 we overload the definition of $\text{OL}_{\mathcal{P}}$ for randomized player oracles, making it clear which information each oracle has access to. Moreover, in the definition of $\text{OL}_{\mathcal{P}}$ for randomized player oracles we also naturally suppose that ENEMY is a **measurable enemy oracle** for $\mathcal{P}$, that is, we assume that for each $T \in \mathbb{N} \setminus \{0\}$ and every $\boldsymbol{x} \in X^T$ be have that $\text{ENEMY}(\boldsymbol{x}, \cdot)$ is a measurable function from Y^{T-1} to D. Finally, the definition of regret for randomized player oracles is similar to the definition of regret for deterministic player oracles seen earlier. Note, however, that the function Regret for online learning becomes a random variable in the case of randomized player oracles (given that the enemy oracle is a measurable enemy oracle and that the loss function is measurable).

Algorithm 2.4 Definition of $[\text{OL}_{\mathcal{P}}(\text{NATURE}, \overline{\text{PLAYER}}, \text{ENEMY}, T)](\omega)$ (overloading $\text{OL}_{\mathcal{P}}$)

Input:

(i) An OL instance $\mathcal{P} = (X, D, Y, L)$ such that D and Y are each equipped with a σ-algebra and L is Borel measurable,

(ii) nature and measurable enemy oracles for $\mathcal{P}$ denoted, respectively, by NATURE and ENEMY,

(iii) a randomized player oracle $\overline{\text{PLAYER}}$ on a probability space $(\Omega, \Sigma, \mathbb{P})$,

(iv) a number $T \in \mathbb{N}$ of rounds, and

(v) an elemente $\omega \in \Omega$ (the "random bits").

Output: $(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}) \in X^T \times D^T \times Y^T$.

```
for $t = 1$ to $T$ do
    $x_t \leftarrow \text{NATURE}(t)$
    $D_t \leftarrow \overline{\text{PLAYER}}\big(\langle x_1, \dots, x_t \rangle, \langle Y_1(\omega), \dots, Y_{t-1}(\omega)\rangle\big)$
    $Y_t(\omega) \leftarrow \text{ENEMY}\big(\langle x_1, \dots, x_t \rangle, \langle D_1(\omega), \dots, D_{t-1}(\omega)\rangle\big)$
return $(\boldsymbol{x}, \langle D_1(\omega), \dots, D_T(\omega)\rangle, \langle Y_1(\omega), \dots, Y_T(\omega)\rangle)$
```

Let us revisit the prediction with expert advice problem. On Proposition 2.5.5, we have looked at the expert instances on which the loss function was convex w.r.t. its first argument and the advice set was convex. In this case, it was enough to have a player oracle for the OCO instance $\mathcal{C} := (\Delta_E, \mathcal{F})$, where E is the set of experts and $\mathcal{F}$ is a set of linear functions, to build a player oracle for the original experts problem with the same regret guarantees. The next proposition shows how to build a randomized player oracle for the experts problem from a player oracle for the OCO instance $\mathcal{C}$ such that the expected regret on the experts' problem is the same as the regret of the player for $\mathcal{C}$ against a properly chosen enemy.

Proposition 2.6.2. Let $\mathcal{P} := (A^E, A, Y, L)$ be a prediction with expert advice problem such that A and Y are each equipped with a σ-algebra and L is measurable. Moreover, let $\mathcal{C} := (\Delta_E, \mathcal{F})$ be an OCO instance where

$$\mathcal{F} := \{\, p \in \mathbb{R}^E \mapsto p^{\mathsf{T}} c : c \in [-1, 1]^E \}.$$

Finally, let $\text{PLAYER}_{\text{OCO}}$ be a player oracle for $\mathcal{C}$ and let $U := \{\, e_i \in \{0,1\}^E : i \in E\}$. Then, there exists a randomized player oracle $\overline{\text{PLAYER}}_{\text{OL}}$ for $\mathcal{P}$ such that, for any $T \in \mathbb{N}$ and any sequences $\boldsymbol{x} \in (A^E)^T$ and $\boldsymbol{y} \in Y^T$, there is $\boldsymbol{f} \in \mathcal{F}^T$ such that $\mathbb{E}[\text{Regret}(\boldsymbol{x}, \text{PLAYER}_{\text{OL}}, \boldsymbol{y}, \mathcal{H})] = \text{Regret}(\text{PLAYER}_{\text{OCO}}, \boldsymbol{f}, U)$, where

Proof. For every $x \in A^E$ and $y \in Y$, define $c(x, y) \in [-1, 1]^E$ by

$$(c(x, y))_e := L(x(e), y), \qquad \forall e \in E.$$

Let $(\Omega, \Sigma, \mathbb{P})$ be a probability space such that, for each $T \in \mathbb{N} \setminus \{0\}$ and $p \in \Delta_E$, there is an independent random variable[22] $I_p \colon \Omega \to E$ such that $\mathbb{P}(I_p = e) = p_e$ for each $e \in E$. Finally, define the randomized player oracle $\text{PLAYER}_{\text{OL}}$ for $\mathcal{P}$ given, for every $T \in \mathbb{N}$, $\boldsymbol{x} \in X^T$, and $\boldsymbol{y} \in Y^{T-1}$ by

$$\begin{aligned}
[\overline{\text{PLAYER}}_{\text{OL}}(\boldsymbol{x}, \boldsymbol{y})](\omega) &\coloneqq x_T(I_{p_T}(\omega)) \text{ for each } \omega \in \Omega, \text{ where} \\
f_t(z) &\coloneqq c(x_t, y_t)^\mathsf{T} z \text{ for all } z \in \mathbb{R}^E \text{ and } t \in \{1, \dots, T-1\} \text{ and} \\
p_T &\coloneqq \text{PLAYER}_{\text{OCO}}(\boldsymbol{f}).
\end{aligned}$$

Let $T \in \mathbb{N}$, $\boldsymbol{x} \in X^T$, and $\boldsymbol{y} \in Y^T$. Moreover, for each $t \in [T]$ define

$$\begin{aligned}
f_t(z) &\coloneqq c(x_t, y_t)^\mathsf{T} z, \qquad \forall z \in \mathbb{R}^E, \\
p_t &\coloneqq \text{PLAYER}_{\text{OCO}}(\boldsymbol{f}_{1:t-1}), \\
D_t &\coloneqq \overline{\text{PLAYER}}_{\text{OL}}(\boldsymbol{x}_{1:t}, y_{1:t-1}).
\end{aligned}$$

Let $t \in [T]$. Note that $D_t = x_t(I_{p_t}(\cdot))$. Thus, $\mathbb{P}(D_t = x_t(e)) = \mathbb{P}(I_{p_t} = e) = p_t(e)$ for each $e \in E$. Since $\boldsymbol{y}$ is fixed, we have

$$\mathbb{E}[L(D_t, y_t)] = \sum_{e \in E} p_t(e) L(x_t(e), y_t) = \sum_{e \in E} p_t(e)[c(x_t, y_t)](e) = p_t^\mathsf{T} c(x_t, y_t) = f_t(p_t).$$

With that, by setting $R_{\text{OL}} \coloneqq \text{Regret}(\boldsymbol{x}, \overline{\text{PLAYER}}_{\text{OL}}, \boldsymbol{y}, \mathcal{H})$ we have

$$\begin{aligned}
\mathbb{E}[R_{\text{OL}}] = \mathbb{E}\Big[\sum_{t=1}^T L(D_t, y_t) - \min_{i \in E} \sum_{t=1}^T L(x_t(i), y_t)\Big] &= \sum_{t=1}^T \mathbb{E}[L(D_t, y_t)] - \min_{i \in E} \sum_{t=1}^T L(x_t(i), y_t) \\
= \sum_{t=1}^T f_t(p_t) - \min_{i \in E} \sum_{t=1}^T f_t(e_i) = \text{Regret}(\text{PLAYER}_{\text{OCO}}, \boldsymbol{f}, U). & \qquad \square
\end{aligned}$$

We have claimed that the above proposition shows that a player oracle for an OCO instance with low regret guarantees was enough to build a randomized player oracle for the experts problem with good guarantees on the expected regret. However, as one might have noticed, there is a catch. The above proposition proves an upper bound on the expected regret of the randomized player oracle against *fixed* sequences of enemy choices. In other words, the above proposition only proves that such a player oracle has low expected regret against enemies which are oblivious to the choices of the player. This nuance might be often overlooked, but we find extremely insightful to deeply understand it. Thus, let us discuss this issue more formally.

Let $\mathcal{P} \coloneqq (X, Y, D, L)$ be an online learning instance such that Y and D are each equipped with a σ-algebra and L is measurable, and let NATURE, PLAYER, and ENEMY be nature, player, and measurable enemy oracles for $\mathcal{P}$, respectively. Let $\mathcal{H} \subseteq D^X$, let $T \in \mathbb{N}$, and suppose there is $\alpha \in \mathbb{R}$ such that

$$\text{Regret}(\boldsymbol{x}, \text{PLAYER}, \boldsymbol{y}, \mathcal{H}) \leq \alpha, \qquad \forall \boldsymbol{x} \in X^T, \forall \boldsymbol{y} \in Y^T. \tag{2.12}$$

The above bound, which is of the same type as the bound on Proposition 2.5.5, guarantees an upper bound of α on the regret of PLAYER against NATURE and ENEMY. To see this, note that if we set

$$(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}) \coloneqq \text{OL}_{\mathcal{P}}(\text{NATURE}, \text{PLAYER}, \text{ENEMY}, T),$$

we can plug such sequences $\boldsymbol{x}$ and $\boldsymbol{y}$ into (2.12) to obtain a bound on the regret of the player oracle in this game. Let us now look at what happens if we have a bound as the one in Proposition 2.6.2.

[22] Since E is finite, it is a measure space when equipped with its powerset as its σ-algebra.

Namely, let $\overline{\mathrm{PLAYER}}$ be a randomized player oracle for $\mathcal{P}$ on a probability space $(\Omega, \Sigma, \mathbb{P})$, and suppose there is $\beta \in \mathbb{R}$ such that

$$\mathbb{E}[\mathrm{Regret}(\boldsymbol{x}, \overline{\mathrm{PLAYER}}, \boldsymbol{y}, \mathcal{H})] \leq \beta, \qquad \forall \boldsymbol{x} \in X^T, \forall \boldsymbol{y} \in Y^T. \tag{2.13}$$

Even though (2.12) and (2.13) are similar, the latter does not directly yield a bound on the expected regret of $\overline{\mathrm{PLAYER}}$ against NATURE and ENEMY. To see this, define

$$(\boldsymbol{x}, \boldsymbol{D}, \boldsymbol{Y}) \coloneqq \mathrm{OL}_{\mathcal{P}}(\mathrm{NATURE}, \overline{\mathrm{PLAYER}}, \mathrm{ENEMY}, T)$$

and let $t \in \{2, \ldots, T\}$. Note that Y_t is a function of $\boldsymbol{x}_{1:t}$ and $\boldsymbol{D}_{1:t-1}$, and since the latter are random variables (since $t > 1$), we have that Y_t is a random variable. That is, we cannot plug $\boldsymbol{Y}$ in the place of $\boldsymbol{y}$ in (2.13). Thus, (2.13) only applies directly in the cases where ENEMY is **oblivious** (to the choices of the player), that is, when

$$\mathrm{ENEMY}(\boldsymbol{x}, \boldsymbol{d}) = \mathrm{ENEMY}(\boldsymbol{x}, \boldsymbol{d}') \qquad \forall T \in \mathbb{N} \setminus \{0\}, \forall \boldsymbol{x} \in X^T, \forall \boldsymbol{d}, \boldsymbol{d}' \in D^{T-1}.$$

In words, an oblivious enemy oracle only depends on the nature queries, without taking into account in his decisions the choices of the player. An extreme case of obliviousness is when the enemy oracle only depends on the *length* of the sequence of queries from nature (i.e. the number of the current round). In this latter case, the oblivious enemy oracle already knows which points it is going to pick at each round even before the game begins. Oblivious enemies are not any longer functions of the specific choices of the player, only of the nature queries, which are deterministic. Thus, the bound on (2.13) can be applied for games where the randomized player oracle plays against oblivious enemies.

At first sight, devising randomized player oracles with low expected regret against oblivious enemies seems much easier than devising such oracles to work against general/adaptive enemies. Surprisingly, it seems that in most online learning problems, a player oracle which is guaranteed to attain low expected regret against any oblivious enemy is also guaranteed to attain low expected regret against adaptive enemies. For example, [24, Lemma 4.1] states that in the prediction with expert advice, if a randomized player oracle which only chooses actions suggested by one of the experts (which is the usual case) has expected regret against any oblivious enemy bounded by $\beta \in \mathbb{R}$, then the expected regret against adaptive enemies is also bounded by β. However, it is not clear how to adapt the arguments from the proof of [24, Lemma 4.1] to the framework we present here since, for example, the lemma By Cesa-Bianchi and Lugosi focuses on the experts' problem.

Still, it seems that the claim that a player oracle which performs well against any oblivious enemy also performs well against any adaptive enemy should hold for more general online learning problems. For example, if the set Y from where the enemy makes his choices is countable, then using expectations conditioned on the choices of the enemy and the law of total expectation seems to prove this claim. Additionally, in [28, Section 3], the authors prove that randomized player oracles for the *online optimization setting* whose random variables picked by the player are all independent[23] and that have low expected regret against oblivious enemies also have low expected regret against adaptive enemies. It seems that since the nature oracle for any OL instance is deterministic, this result should be extensible to the online learning setting almost seamlessly. However, some arguments used in the proof of [28, Theorem 3.1] do not seamlessly translate to the framework presented in this text (in particular, the application of their induction hypothesis).

Finally, this "equivalence" between the power of players for adaptive and oblivious enemies does not seem to hold for an online learning framework with *bandit feedback* [10], that is, the player oracle only receives the loss it suffered on past rounds, not the points picked by the enemy as in the classical online learning setting. Thus, having a clear proof for our framework of the claim discussed above and looking at the arguments that do not hold for the bandit setting may be very insightful.

[23] Which we suppose for the randomized player oracles of this section.

2.7 A Closer Look at Regret

On Section 2.3, we have shown that looking only at the raw cumulative loss of the player oracle in general does not give us much information about the quality of its predictions and proposed regret as a better quality measure. On that same section, we also discussed the intuitive meaning of the regret of a player oracle w.r.t. a comparison hypothesis $h\colon X \to D$: it measures how "sorry" the player is for not using h to pick his decisions throughout the game. However, this intuition is not entirely accurate.

Let us formalize our discussion before continuing. Let $\mathcal{P} \coloneqq (X, D, Y, L)$ be an online learning instance. Moreover, let NATURE, PLAYER, and ENEMY be nature, enemy, and player oracles for $\mathcal{P}$, respectively, let $T \in \mathbb{N}$, and define

$$(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}) \coloneqq \mathrm{OL}_{\mathcal{P}}(\mathrm{NATURE}, \mathrm{PLAYER}, \mathrm{ENEMY}, T).$$

Then, the regret of PLAYER w.r.t. a hypothesis $h\colon : X \to D$ is

$$\mathrm{Regret}_T(\mathrm{NATURE}, \mathrm{PLAYER}, \mathrm{ENEMY}\ h) = \sum_{t=1}^{T} L(d_t, y_t) - \sum_{t=1}^{T} L(h(x_t), y_t).$$

Note that, when we say "(...) [regret] measures how sorry the player is for not using h to pick his decisions throughout the game", we are assuming that, even if the player had used h to make her decisions, the enemy would still pick $\boldsymbol{y}$ as he sequence of the points for the game. In other words, this intuition assumes that

$$\mathrm{ENEMY}(\boldsymbol{x}_{1:t}, \boldsymbol{d}_{1:t-1}) = y_t = \mathrm{ENEMY}(\boldsymbol{x}_{1:t}, \langle h(x_1), h(x_2), \dots, h(x_{t-1})\rangle), \qquad \forall t \in [T].$$

The above holds in particular if the enemy oracle is *oblivious*, i.e. if the enemy oracle does not adapt to the player's choices (see Section 2.6.2 for the formal definition and further discussion about oblivious enemy oracles). Still, the above equation does not hold for general enemy oracles. This does not nullify the search of low-regret player oracles, which has been a strong focus of research in online learning and online convex optimization [36, 67]. Besides, the classic notion of regret yields interesting connections with other areas, such as game theory [24] and statistical learning theory [36, Chapter 9]. Nevertheless, it is still useful to look at the idea of *policy regret*, a stronger notion of regret introduced in [5] which better reflects the intuition we gave previously.

Let $\mathcal{P} \coloneqq (X, D, Y, L)$ be an online learning instance. Moreover, let NATURE, PLAYER, and ENEMY be nature, enemy, and player oracles for $\mathcal{P}$, respectively. Finally, let $T \in \mathbb{N}$ and define

$$(\boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}) \coloneqq \mathrm{OL}_{\mathcal{P}}(\mathrm{NATURE}, \mathrm{PLAYER}, \mathrm{ENEMY}, T).$$

Then, the **policy regret** of PLAYER against ENEMY in T rounds (w.r.t. NATURE and a hypothesis $h\colon X \to D$) is

$$\mathrm{PRegret}_T(\mathrm{NATURE}, \mathrm{PLAYER}, \mathrm{ENEMY}, h, L) \coloneqq \sum_{t=1}^{T} L(d_t, y_t) - \sum_{t=1}^{T} L(h(x_t), \mathrm{ENEMY}(\boldsymbol{x}_{1:t}, \boldsymbol{u}_{1:t-1})),$$

where $\boldsymbol{u} \coloneqq \langle h(x_1), h(x_2), \dots, h(x_T)\rangle$. One may note that the main distinction between the definitions of regret and policy regret is in the second summation in the above definition. As we have discussed, we are now comparing the loss of the player with the actual loss she would have suffered had she made her predictions according to the hypothesis h. Unfortunately, this notion of regret is too strong.

From Cover's impossibility result[24] (Proposition 2.3.3), we know that there is no deterministic player oracle which attains sublinear policy regret in general. However, even *randomized* player oracles have no hope of attaining sublinear (in the number of rounds) expected policy regret.

Theorem 2.7.1 (Based on [5, Theorem 1])**.** Let $\mathcal{P} \coloneqq (\{1,2\}^2, \{1,2\}, [-1,1]^2, L)$ be a prediction with expert advice problem, where[25] $L(d,y) \coloneqq y_d$ for every $d \in \{1,2\}$ and $y \in Y$. Let $\overline{\text{PLAYER}}$ be a randomized player oracle for $\mathcal{P}$ on a probability space $(\Omega, \Sigma, \mathbb{P})$. Then, there are nature and measurable enemy oracles NATURE and ENEMY for $\mathcal{P}$, respectively, such that, for any $T \in \mathbb{N} \setminus \{0\}$ and for the hypothesis set $\mathcal{H} \coloneqq \{\, x \in \{1,2\}^2 \mapsto x_i : i \in \{1,2\}\}$, we have

$$\max_{h\in\mathcal{H}} \mathbb{E}[\text{PRegret}_T(\text{NATURE}, \overline{\text{PLAYER}}, \text{ENEMY}, h)] \geq \frac{T-1}{2}.$$

Proof. Let $y \in \{1,2\}$ be such that

$$\mathbb{P}\big(\overline{\text{PLAYER}}(\langle(1,2)^\mathsf{T}\rangle, \langle\rangle) = y\big) \geq \frac{1}{2}.$$

One exists since the player is bound to make a choice from a pool of 2 options. Moreover, define nature and enemy oracles NATURE and $\text{ENEMY}^*_{\text{PLAYER}}$ for $\mathcal{P}$, respectively, by

$$\begin{aligned} \text{NATURE}(t) &\coloneqq (1,2)^\mathsf{T} && \text{for each } t \in \mathbb{N}, \\ \text{ENEMY}^*_{\text{PLAYER}}(\boldsymbol{x}, \boldsymbol{d}) &\coloneqq [t > 1 \text{ and } d_1 = y]\mathbb{1} && \text{for each } t \in \mathbb{N},\ \boldsymbol{x} \in (\{1,2\}^2)^t, \text{ and } \boldsymbol{d} \in (\{1,2\}^2)^{t-1}. \end{aligned}$$

Note that for any $T \in \mathbb{N} \setminus \{0\}$ and any $\boldsymbol{x} \in (A^E)^T$, we have that $\text{ENEMY}^*_{\text{PLAYER}}(\boldsymbol{x}, \cdot)$ is a constant function on A^{T-1} and, thus, measurable. Let $T \in \mathbb{N}$, and define

$$(\boldsymbol{x}, \boldsymbol{D}, \boldsymbol{Y}) \coloneqq \text{OCO}_{\mathcal{P}}(\text{NATURE}, \overline{\text{PLAYER}}, \text{ENEMY}^*_{\text{PLAYER}}, T).$$

By the definition of the enemy oracle $\text{ENEMY}^*_{\text{PLAYER}}$, the player suffers a loss of 1 at every round (except for the first) if she picks y on the first round, and suffers a loss of 0 otherwise. Thus,

$$\mathbb{E}\Big[\sum_{t=1}^{T} L(D_t, Y_t)\Big] = (T-1)\mathbb{P}(D_1 = y) \geq \frac{T-1}{2}.$$

Let $z \in \{1,2\} \setminus \{y\}$, set $h_z(x) \coloneqq x_z$ for every $x \in \{1,2\}^2$, and define

$$\boldsymbol{u} \coloneqq \langle h_z(x_1), h_z(x_2), \ldots, h_z(x_T)\rangle.$$

Then, $\boldsymbol{u}_1 = h_z(x_1) = x_1(z) = z \neq y$. Hence, by definition,

$$\begin{gathered} \text{ENEMY}^*_{\text{PLAYER}}(\boldsymbol{x}_{1:t}, \boldsymbol{u}_{1:t-1}) = 0 \text{ for each } t \in [T] \\ \implies \sum_{t=1}^{T} L(h_z(x_t), \text{ENEMY}^*_{\text{PLAYER}}(\boldsymbol{x}_{1:t}, \boldsymbol{u}_{1:t-1})) = 0. \end{gathered}$$

Since $h_z \in \mathcal{H}$, we are done. □

[24] Although Cover's result is about regret, the enemy built in the statement of the proposition is oblivious. Thus, one can check that regret and policy regret are equal for such an enemy, and the result holds for policy regret.

[25] Note that L is measurable.

Even though the original result from [5] is slightly more general, this version of the theorem which looks at a specific instance of the prediction with experts' problem can be easily compared to Cover's impossibility result (Proposition 2.3.3). Additionally, by Proposition 2.6.2 we know that the randomized experts problem can be reduced, in some sense, to an online convex optimization instance over the simplex. For this latter OCO problem, we are going to see that there are player oracles which attain regret sublinear in the number of rounds. This shows that policy regret is way harder to handle than the traditional notion of regret.

Policy regret is not the focus of our text, so we limit its discussion to this section. Still, we point the reader who is interested in policy regret minimization to [5] and [22]. Interestingly, in both of these papers the authors obtain more interesting results when the enemy is oblivious to "old" rounds, that is, the enemy is a function only of the last c rounds, where c is some constant. Finally, there are some other variations of regret which are useful for some problems and in some applications of online learning to other fields. For more information on different regret variations, see [24, Sections 4.4 and 4.6].

Chapter 3

Convex Analysis, Optimization, and Duality Theory

It is not surprising that the description and analysis of most algorithms for online convex optimization, which is the focus of the remainder of the text, heavily relies on ideas from convex analysis and optimization. Even though the early developments in OCO could be seen in a mostly self-contained way, the field has recently been experiencing a more coherent and unified progress, mainly due to the use of powerful ideas from convex analysis and optimization, chiefly the ideas from convex duality theory based on the Hyperplane Separation Theorem and on Fenchel conjugates of functions. The presentation of the algorithms on next chapters follows this latter trend since it reveals interesting insights about the inner-working and connections among OCO algorithms. However, simply requiring the reader to have a background on convex analysis heavily restricts the accessibility of this text.

In this chapter we overview the main concepts from convex analysis which we use throughout the remainder of the text, with a focus on building intuition. The presentation of this chapter aims to be of use for both proficient *and* inexperienced readers when it comes to convex analysis. For the former group, this chapter serves as a revision of the main concepts from convex analysis we use throughout the text, together with some proofs of more specific results which are used on later chapters. For those who are having one of their first contacts with convex analysis through this text, we aim to build most of the intuition necessary to understand the descriptions and analyses of the main algorithms presented in this text. For the sake of conciseness and simplicity, we do not prove many results that we state in this chapter, although leave references the interested reader in such cases.

Recall from Section 1.1 that, throughout the whole text, $\mathbb{E}$ denotes an euclidean space (finite-dimensional real vector space equipped with an inner product) whose inner product we denote by $\langle \cdot, \cdot \rangle$. Moreover, throughout the remainder of the text we equip $\mathbb{R}^d$ with the **euclidean inner product** $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d \mapsto x^\mathsf{T} y$, and we equip $\mathbb{S}^d$ with the **trace inner product** $(X, Y) \in \mathbb{S}^d \times \mathbb{S}^d \mapsto \mathrm{Tr}(XY)$. Finally, this chapter is mainly based on [59], although it also draws from many other sources such as [15, 17, 18, 55]

3.1 Convex Sets and Functions

A set $C \subseteq \mathbb{E}$ is **convex** if $\lambda x + (1 - \lambda) y \in C$ for any $\lambda \in [0, 1]$ and any $x, y \in C$, and C is **affine** if $\lambda x + (1 - \lambda) y \in C$ for any $\lambda \in \mathbb{R}$ and for any $x, y \in C$. That is, a set $C \subseteq \mathbb{E}$ is convex if and only if, for any $x, y \in C$, the **line segment** (between x and y) given by $[x, y] \coloneqq \{\, \lambda x + (1 - \lambda) y : \lambda \in [0, 1] \}$ is entirely contained in C. Moreover, a set $C \subseteq \mathbb{E}$ is affine if the *line* that passes through any two distinct points $x, y \in C$ is contained in C. Note that intersections of convex (affine) sets are convex (affine).

Let us now define *convex functions*. We do so by looking at the set formed by the graph of the function, in some sense. This way of looking at functions is useful since results and concepts for convex sets can often be translated into analogous results about convex functions almost seamlessly. Formally, let $f\colon S \to \mathbb{R}$ where $S \subseteq \mathbb{E}$. The **epigraph** of f is the set

$$\operatorname{epi} f \coloneqq \{\, x \oplus \mu \in S \oplus \mathbb{R} : f(x) \leq \mu \},$$

and f is **convex** if $\operatorname{epi} f$ is convex. That is, the epigraph of a function f is the set built by taking the graph of f and extruding it upwards. On Figure 3.1 we present a graphic representation of the epigraph of a two-dimensional function.

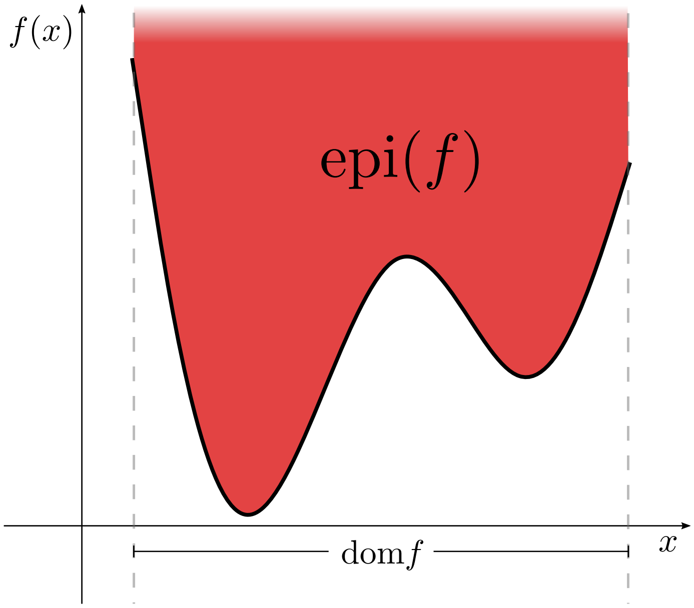

Figure 3.1: Illustration of the epigraph of a (non-convex) function f.

In this text we will follow the same convention used in [59]: all functions we deal with can be evaluated everywhere in $\mathbb{E}$, even though they can take on infinite values. The arithmetic properties of $+\infty$ (which we often denote simply by ∞) and $-\infty$ that we use are the same the author of [59] uses. Namely,

$$\begin{aligned}
\alpha + \infty = +\infty + \alpha &= +\infty &&\text{and} & \alpha - \infty = -\infty + \alpha &= -\infty && \text{for all } \alpha \in \mathbb{R},\\
\alpha(+\infty) = (+\infty)\alpha &= (+\infty) &&\text{and} & \alpha(-\infty) = (-\infty)\alpha &= -\infty && \text{for all } \alpha \in \mathbb{R}_{++},\\
\alpha(+\infty) = (+\infty)\alpha &= -\infty &&\text{and} & \alpha(-\infty) = (-\infty)\alpha &= +\infty && \text{for all } \alpha \in -\mathbb{R}_{++},\\
0(+\infty) = (+\infty)0 &= 0 &&\text{and} & 0(-\infty) = (-\infty)0 &= 0, &&\\
+\infty + \infty &= (+\infty) &&\text{and} & -\infty - \infty &= -\infty, &&\\
\inf \varnothing &= (+\infty) &&\text{and} & \sup \varnothing &= -\infty. &&
\end{aligned}$$

We note that the expressions $+\infty - \infty$ and $-\infty + \infty$ are not define and, thus, are utterly avoided.

We do not lose any generality with this assumption over the values functions can take since a convex function f defined only in a subset $S \subseteq \mathbb{E}$ can be extended by setting $f(x) \coloneqq +\infty$ for every $x \in \mathbb{E} \setminus S$. This extension preserves the epigraph and, thus, convexity. The usefulness of this convention is that it makes many proofs and results less technical, just needing, in some cases, some care with the (non-)finiteness at some points. The **(effective) domain** of $f \colon \mathbb{E} \to [-\infty, +\infty]$ is $\operatorname{dom} f \coloneqq \{\, x \in \mathbb{E} : f(x) \neq +\infty \,\}$ (see Figure 3.1 for an example of effective domain of a function). While $+\infty$ is used to indicate places outside the domain of f, functions which take the value $-\infty$ somewhere are, in some sense, pathological. Not only that, we want to avoid dealing with functions which are infinite everywhere. Thus, we will almost always deal with *proper functions*: a function $f \colon \mathbb{E} \to [-\infty, +\infty]$ is **proper** if $\operatorname{epi} f$ is nonempty, that is, $\operatorname{dom} f$ is nonempty and $f(x) \neq -\infty$ for every $x \in \mathbb{E}$.

Finally, one can prove that a function $f \colon \mathbb{E} \to (-\infty, +\infty]$ is convex if and only if it satisfies the more familiar condition

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda) f(y), \qquad \forall x, y \in \mathbb{E},\, \forall \lambda \in [0,1].$$

Even though intuitive, the above characterization of convex functions may be hard to show convexity of some given function. The next lemma will be useful later on to show convexity of two-times continuously differentiable functions.

Lemma 3.1.1 ([59, Theorem 4.5] or [15, Corollary 2.1])**.** Let $f \colon \mathbb{R}^d \to (-\infty, +\infty]$ be a proper two-times continuously differentiable on $\operatorname{dom} f$ function such that $\operatorname{dom} f$ is convex. Then f is convex if and only if $\nabla^2 f(x) \succeq 0$ for any $x \in \operatorname{dom} f$.

Some functions are very useful when stating or proving convex analysis results. For any set $C \subseteq \mathbb{E}$, define

- the **indicator function** $\delta(\cdot \,|\, C)$ of C by $\delta(x \,|\, C) \coloneqq 0$ for every $x \in C$ and $\delta(x \,|\, C) \coloneqq +\infty$ for every $x \in \mathbb{E} \setminus C$, and
- the **support function** $\delta^*(\cdot \,|\, C)$ of $C \subseteq \mathbb{E}$ by $\delta^*(x \,|\, C) \coloneqq \sup\{\, \langle y, x \rangle : y \in C \,\}$.

The best way to build intuition about indicator functions is by looking at its epigraph. For example, the epigraph of $\delta(\cdot \,|\, [-1,1])$ is simply the (infinity) rectangle on $\mathbb{R}^2$ formed by taking the segment from $(-1,0)$ to $(1,0)$ and extruding it upwards (i.e., in the direction $(0,1)$). More generally, if $C \subseteq \mathbb{E}$, then the epigraph of $\delta(\cdot \,|\, C)$ it is the set C embedded in the hyperplane $\mathbb{E} \oplus 0$ and extruded in the direction $0 \oplus 1 \in \mathbb{E} \oplus \mathbb{R}$.

The intuition on the support function for a convex set $C \subseteq \mathbb{E}$, on the other hand, is better pictured in a different way. Namely, let $a \in C$ be such that $\bar{\beta} \coloneqq \delta^*(a \,|\, C)$ is finite. For every $\beta \in \mathbb{R}$ we have the *hyperplane* $H(\beta) \coloneqq \{\, x \in \mathbb{E} : \langle a, x \rangle = \beta \,\}$, and two associated *(closed) half-spaces*

$$H^{\leq}(\beta) \coloneqq \{\, x \in \mathbb{E} : \langle a, x \rangle \leq \beta \,\} \qquad \text{and} \qquad H^{\geq}(\beta) \coloneqq \{\, x \in \mathbb{E} : \langle a, x \rangle \geq \beta \,\}$$

which, in some sense, divide $\mathbb{E}$ in two almost disjoint sets. In particular, $H(\bar{\beta})$ is also a hyperplane, and by the definition of support function, we have $\langle a, x \rangle \leq \delta^*(a \,|\, C) = \bar{\beta}$ for every $x \in C$, that is, we have $C \subseteq H^{\leq}(\bar{\beta})$. Thus, $H(\bar{\beta})$ is such that the set C is entirely contained in one of its half-spaces. Not only that, by the definition of support function we have $\bar{\beta} = \inf\{\, \beta \in \mathbb{R} : C \subseteq H(\beta) \,\}$. In words, $\bar{\beta}$ is the minimum value of $\beta \in \mathbb{R}$ such that C in contained into one of the half-spaces associated with $H(\beta)$. Finally, on Section 3.4 it will become clear why the notations for indicator and support function are similar.

Given a set $S \subseteq \mathbb{E}$, we are sometimes interested in the smallest set with some property that contains S. For example, the smallest affine set that contains S tells us, in some sense, if we could fit the points of S in a space of smaller dimension. For any $S \subseteq \mathbb{E}$, define

- $\operatorname{aff} S \coloneqq \bigcap\{\, M \subseteq \mathbb{E} : S \subseteq M \text{ and } M \text{ is affine}\}$, called the **affine hull** of S;
- $\operatorname{conv} S \coloneqq \bigcap\{\, C \subseteq \mathbb{E} : S \subseteq C \text{ and } C \text{ is convex}\}$, called the **convex hull** of S.

Even though the above operations will not be used very often in the remainder of the text, they are important in some results and definitions that we see in this chapter. However, one may find it hard to have much intuition on the hull operations with the definitions given above. The following result shows an easier way to see the hull operations: the affine (convex) hull of a set $S \subseteq \mathbb{E}$ is the set of all affine (convex) combinations of finite subsets of points in S.

Proposition 3.1.2 (see [59, Chapter 1] and [59, Theorem 2.3])**.** For any $S \subseteq \mathbb{E}$, we have

$$\operatorname{aff} S = \Big\{ \sum_{i=1}^{m} \lambda_i x_i \in \mathbb{E} : \text{for every } m \in \mathbb{N},\ \text{every } \{x_i\}_{i=1}^{m} \subseteq S,\ \text{and every } \lambda \in \mathbb{R}^m \text{ s.t. } \sum_{i=1}^{m} = 1 \Big\}$$

and

$$\operatorname{conv} S = \Big\{ \sum_{i=1}^{m} \lambda_i x_i \in \mathbb{E} : \text{for every } m \in \mathbb{N},\ \text{every } \{x_i\}_{i=1}^{m} \subseteq S,\ \text{and every } \lambda \in \mathbb{R}^m_+ \text{ s.t. } \sum_{i=1}^{m} = 1 \Big\}.$$

For concreteness, let us look at a small example. Define $S \coloneqq \{(1,0),(2,1)\} \subseteq \mathbb{R}^2$. With the above proposition, one can easily see that $\operatorname{conv} S$ is the line segment between the points $(1,0)$ and $(2,1)$ and that $\operatorname{aff} S$ is the line that passes through $(1,0)$ and $(2,1)$. By setting $S' \coloneqq S \cup \{(2,0)\}$, we have that $\operatorname{conv} S'$ is the region enclosed by the triangle formed by the points in S' and, interestingly, $\operatorname{aff} S'$ is the entire space $\mathbb{R}^2$.

3.2 Topological Properties of Convex Sets and Functions

Many results of convex analysis depend on some topological properties of the sets or functions considered. Thus, in this section we state and try to give intuition on the main results and definitions regarding topological properties of convex sets and functions. For this section we suppose that the reader is mildly familiar with the basic concepts from topology, such as closed and open sets.

Set $\mathbb{B} \coloneqq \{\, x \in \mathbb{E} : \langle x, x\rangle \leq 1\}$. For every $C \subseteq \mathbb{E}$, define

- the **closure** of C by $\operatorname{cl} C \coloneqq \bigcap_{\varepsilon > 0}(C + \varepsilon\mathbb{B})$;
- the **interior** of C by $\operatorname{int} C \coloneqq \{\, x \in C : \text{there exists } \varepsilon > 0 \text{ s.t. } x + \varepsilon\mathbb{B} \subseteq C\}$;
- the **relative interior** of C by $\operatorname{ri} C \coloneqq \{\, x \in C : \text{there exists } \varepsilon > 0 \text{ s.t. } (x + \varepsilon\mathbb{B}) \cap \operatorname{aff} C \subseteq C\}$.

Moreover, a set $C \subseteq \mathbb{E}$ is **relatively open** if $\operatorname{ri} C = C$. The concepts of closure and interior of a set are classic topology ideas. Loosely saying, the closure of a set C is the set C with just enough points added so that it becomes closed, and the interior of a set is the set C with the sets from the "boundary" (i.e., points which do not lie in the interior) which prevent C from being open removed. However, the notion of relative interior is not a standard topology concept. Thus, let us look a little bit closer at it.

Loosely saying, the relative interior of a set $C \subseteq \mathbb{E}$ is the interior C would have if it lived in the "correct" dimension. For example, the interior of the line segment $[0,1]$ is $(0,1)$. However, if we place this same line segment into $\mathbb{R}^2$, its interior becomes empty. Namely, the set $C \coloneqq [(0,0),(0,1)] \subseteq \mathbb{R}^2$ has empty interior. This is unfortunate if one hoped to obtain the analogous of the set $(0,1)$ on the 2-dimensional case. This is where the notion of relative interior comes in handy, since it is built exactly for the purpose of looking at the interior the set would have if it were "full-dimensional", that is, in the case it lived in a space in which it had non-empty interior. In the example of the line segment, we have $\operatorname{ri} C = 0 \oplus (0,1) = \{\, (0,\lambda) : \lambda \in (0,1) \}$. On Figure 3.2 we illustrate a 3-dimensional example.

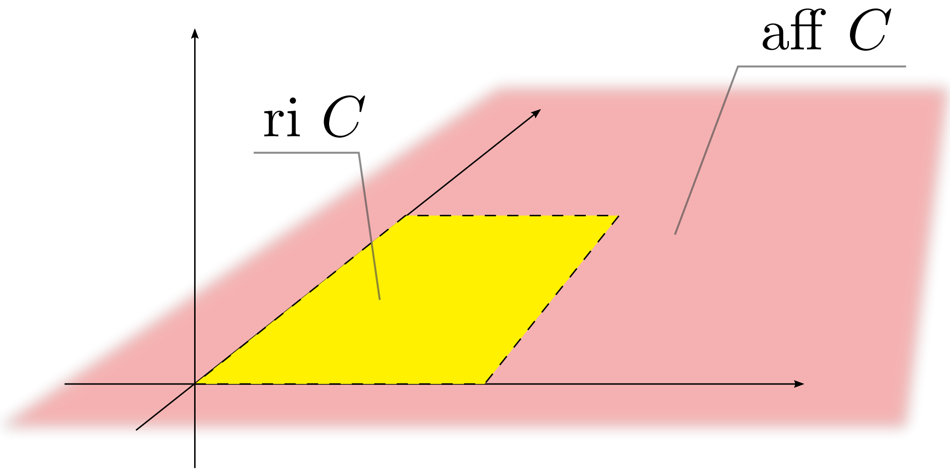

Figure 3.2: Illustration of the relative interior (yellow) and affine hull (red) of the set C, a closed two-dimensional rectangle in $\mathbb{R}^3$, whose interior is empty.

It is worth warning that the operation of taking relative interior of sets can sometimes behave in unexpected ways. For example, if $C \subseteq \mathbb{E}$ and $D \subseteq \mathbb{E}$ are such that $C \subseteq D$, then from the definitions of closure and interior one can see that $\operatorname{cl} C \subseteq \operatorname{cl} D$ and that $\operatorname{int} C \subseteq \operatorname{int} D$. However, $\operatorname{ri} C$ is not necessarily contained in $\operatorname{ri} D$. In fact, it may happen that $\operatorname{ri} C$ and $\operatorname{ri} D$ are disjoint, and such cases are not so pathological as one might think. For example, take $C \coloneqq 0 \oplus [0,1]$ and $D \coloneqq \{\, (\lambda,\mu) \in \mathbb{R}^2 : \lambda,\mu \in [0,1] \}$. That is, D is a square on $\mathbb{R}^2$, and C is its bottom edge. In this case we have $C \subseteq D$. Yet, we have seen in the previous paragraph that $\operatorname{ri} C = 0 \oplus (0,1)$. Moreover, $\operatorname{aff} D = \mathbb{R}^2$, which implies that $\operatorname{ri} D = \operatorname{int} D = \{\, (\lambda,\mu) \in \mathbb{R}^2 : \lambda,\mu \in (0,1) \}$, that is, $\operatorname{ri} C$ and $\operatorname{ri} D$ are disjoint in this case.

Still, there are some tools which allows us to work with relative interiors without much trouble. The next theorem states, for a convex set $C \subseteq \mathbb{E}$, that the segment between a point in $\operatorname{ri} C$ with any other point $\bar{x} \in \operatorname{cl} C$ is almost entirely contained in $\operatorname{ri} C$, with the only exception being the point $\bar{x}$ itself.

Theorem 3.2.1 ([59, Theorem 6.1])**.** Let $C \subseteq \mathbb{E}$ be a nonempty convex set, let $\mathring{x} \in \operatorname{ri} C$, and let $\bar{x} \in \operatorname{cl} C$. Then $(1-\lambda)\mathring{x} + \lambda\bar{x} \in \operatorname{ri} X$ for every $\lambda \in [0,1)$.

Even though the above theorem gives us some intuition about points in the relative interior, it is of no help if we want to show that a point from a set lies in its relative interior. The next theorem helps us in this sense, showing a very enlightening characterization of points in the relative interior of *convex* sets: a point $\mathring{x}$ in a convex set $C \subseteq \mathbb{E}$ is in its relative interior if and only if, for every $x \in C$, the line segment between x and $\mathring{x}$ can be slightly extended in the direction of $\mathring{x}$. To obtain a grasp of why this holds, recall that a point $\mathring{x} \in C$ is in $\operatorname{ri} C$ if and only if there is $\varepsilon > 0$ such that $(\mathring{x} + \varepsilon\mathbb{B}) \cap \operatorname{aff} C \subseteq C$ where $\mathbb{B} \coloneqq \{\, x \in \mathbb{E} : \langle x, x \rangle \leq 1 \}$. That is, there is a ball confined in the affine hull of C and centered in $\mathring{x}$ which lies entirely in C. Since $\mathring{x} - x \in \operatorname{aff} C$ for any $x \in C$, one can at least intuitively see that if the segment between $\mathring{x}$ and every point $x \in C$ can be extended in the direction of $\mathring{x}$, there must be such a "ball confined in aff C" from the definition of relative interior.

Theorem 3.2.2 ([59, Theorem 6.4])**.** Let $C \subseteq \mathbb{E}$ be a nonempty convex set and let $\mathring{x} \in C$. Then $\mathring{x} \in \operatorname{ri} C$ if and only if for each $x \in C$ there is $\mu > 1$ such that $(1-\mu)x + \mu\mathring{x} \in C$.

As an application of the above theorem, let us compute the relative interior of some kinds of polyhedra.

Corollary 3.2.3. Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $C \in \mathbb{R}^{k \times n}$, and $d \in \mathbb{R}^k$ be such that there is $\mathring{x} \in P \coloneqq \{\, x \in \mathbb{R}^n : Ax \leq b, Cx = d\}$ such that $A\mathring{x} < b$. Then $\operatorname{ri} P = \{\, x \in \mathbb{R}^n : Ax < b, Cx = d\}$.

Proof. Define $P' \coloneqq \{\, x \in \mathbb{R}^n : Ax < b, Cx = d\}$ and let $\bar{x} \in P$. For any $\mu \in \mathbb{R} \setminus \{0\}$ and $x \in P$ we have $C((1-\mu)x + \mu\bar{x}) = d$. Thus, by Theorem 3.2.2 it suffices to show that for every $x \in P$ there is $\mu > 1$ such that $A((1-\mu)x + \mu\bar{x}) < b$ if and only if $A\bar{x} < b$ (i.e. $\bar{x} \in P'$).

First, suppose that for any $x \in P$ there is $\mu >$ 1such that $A((1-\mu)x + \mu\bar{x}) < b$. In particular, there is $\bar{\mu} > 1$ such that $z \coloneqq (1-\bar{\mu})\mathring{x} + \bar{\mu}\bar{x} \in P$. Since $1 - \bar{\mu} < 0$ we have

$$Az = A((1-\bar{\mu})\mathring{x} + \bar{\mu}\bar{x}) = (1-\bar{\mu})A\mathring{x} + \bar{\mu}A\bar{x} > (1-\bar{\mu})b + \bar{\mu}A\bar{x}.$$

Since $z \in P$, we have $Az \leq b$, which holds if and only if $\mu b > \mu A\bar{x}$, that is, if and only if $A\bar{x} < b$.

Suppose now that $A\bar{x} < b$, and let $x \in P$. Set $r \coloneqq b - A\bar{x} > 0$ and $s \coloneqq b - Ax \geq 0$. Note that, for any $\mu > 1$, by setting $x_\mu \coloneqq (1-\mu)x + \mu\bar{x}$ we have

$$Ax_\mu = A((1-\mu)x + \mu\bar{x}) = Ax + \mu(A\bar{x} - Ax) = b - s + \mu(s - r). \tag{3.1}$$

If $s - r \leq 0$, then we are done since, in this case, $Ax_\mu \leq b - s \leq b$ for any $\mu > 1$. Thus, suppose there is $i \in [n]$ with $s_i - r_i > 0$ and set

$$\bar{\mu} \coloneqq \min\left\{ \frac{s_i}{s_i - r_i} : i \in [n], s_i - r_i > 0 \right\} > 1,$$

and let $i^* \in [n]$ such that $s_{i^*}(s_{i^*} - r_{i^*})^{-1}$ attains the above minimum. It only remains to show that $Ax_{\bar{\mu}} \leq b$. Let $i \in [n]$. If $s_i - r_i \leq 0$, then by (3.1) we have $(A\bar{x})_i \leq b_i$. On the other hand, if $s_i - r_i > 0$, then

$$(Ax_{\bar{\mu}})_i = b_i - s_i + \bar{\mu}(s_i - r_i) = b_i - s_i + \frac{s_{i^*}}{s_{i^*} - r_{i^*}}(s_i - r_i) \leq b_i - s_i + \frac{s_i}{s_i - r_i}(s_i - r_i) = b_i. \qquad \square$$

Let us look at the idea of lower semi-continuity for functions which, maybe surprisingly, is just the translation of the closure property of the epigraph. Let $f\colon \mathbb{E} \to (-\infty, +\infty]$. The function f is **lower semi-continuous** at $x \in \mathbb{E}$ if $f(x) = \liminf_{y \to x} f(y)$. One may note that a continuous function is, in particular, lower semi-continuous. The next theorem shows that for a function f to be lower semi-continuous is equivalent to its epigraph epi f being closed.

Theorem 3.2.4 ([59, Theorem 7.1])**.** Let $f\colon \mathbb{E} \to (-\infty, +\infty]$. The following are equivalent:

(i) f is lower semi-continuous on $\mathbb{E}$;

(ii) For every $\alpha \in \mathbb{R}$ the set $\{\, x \in \mathbb{E} : f(x) \leq \alpha\}$ is closed;

(iii) The epigraph of f is a closed set in $\mathbb{E} \oplus \mathbb{R}$.

The above theorem makes it natural to define the closure of a function f as the function whose epigraph is $\operatorname{cl}(\operatorname{epi} f)$. Formally, the **closure** of $f\colon \mathbb{E} \to (-\infty, +\infty]$ is the function $\operatorname{cl} f\colon \mathbb{E} \to (-\infty, +\infty]$ given by

$$(\operatorname{cl} f)(x) \coloneqq \inf\{\, \mu \in \mathbb{R} : x \oplus \mu \in \operatorname{cl}(\operatorname{epi} f)\}, \qquad \forall x \in \mathbb{E}.$$

That is, $\operatorname{cl} f$ for some function f is the function whose epigraph is $\operatorname{cl}(\operatorname{epi} f)$. If $f\colon \mathbb{E} \to [-\infty, +\infty]$ is such that there is $x \in \mathbb{E}$ with $f(x) = -\infty$, we set $(\operatorname{cl} f)(x) \coloneqq -\infty$ for every $x \in \mathbb{E}$. Moreover, we say that $f\colon \mathbb{E} \to [-\infty, +\infty]$ is **closed** if $\operatorname{cl} f = f$. The next theorem shows that the closure operation for convex functions yields closed convex functions and does not change the functions by much. Namely, a function and its closure differ maybe only on the border of the domain.

Theorem 3.2.5 ([59, Theorem 7.4])**.** Let $f\colon \mathbb{E} \to (-\infty, +\infty]$ be a proper convex function. Then $\operatorname{cl} f$ is a proper closed convex function, and $f(x) = (\operatorname{cl} f)(x)$ for every $x \in \operatorname{ri}(\operatorname{dom} f)$.

With the above theorem we know where a function and its closure can differ. Yet, we have not shown, besides taking the inferior limit, a way to discover the value of the closure of the function at one of these boundary points. The next theorem shows a simpler way to obtain the values of the closure of a function.

Theorem 3.2.6 ([59, Theorem 7.5])**.** Let $f\colon \mathbb{E} \to (-\infty, +\infty]$ be a proper convex function and let $\mathring{x} \in \operatorname{ri}(\operatorname{dom} f)$. Then,

$$(\operatorname{cl} f)(x) = \lim_{\lambda \uparrow 1} f(\lambda x + (1 - \lambda)\mathring{x}), \qquad \forall x \in \mathbb{E}.$$

Finally, in the same way that the sum of continuous functions is continuous, we would like the sum of closed convex functions to be a closed convex function as well. The next theorem states exactly this.

Theorem 3.2.7 ([59, Theorem 9.3])**.** Let $f_1, \dots, f_m\colon \mathbb{E} \to (-\infty, +\infty]$ be convex. If f_i is closed for each $i \in [m]$, then $\sum_{i=1}^m f_i$ is a closed convex function.

3.3 Hyperplane Separation and Duality

As pointed out at the beginning of this chapter, many of the results and ideas we shall see in future chapters rely on results and ideas from convex duality theory. The latter is fundamentally based in the well-known *Hyperplane Separation Theorem*, which states that for any two convex set which are "sufficiently disjoint" (i.e., their relative interiors do not meet), there is a hyperplane such that each convex set lies in a different closed half space.

Theorem 3.3.1 (Hyperplane Separation Theorem; see [59, Theorems 11.1 to 11.4])**.** Let $X, Y \subseteq \mathbb{E}$ be nonempty convex sets such that $\operatorname{ri}(X) \cap \operatorname{ri}(Y) = \varnothing$. Then, there is $a \in \mathbb{E} \setminus \{0\}$ which satisfies the following properties:

(i) $\sup_{x \in X} \langle a, x \rangle \leq \inf_{y \in Y} \langle a, y \rangle$, and

(ii) $\inf_{x \in X} \langle a, x \rangle < \sup_{y \in Y} \langle a, y \rangle$.

Moreover, if $0 \notin \operatorname{cl}(X - Y)$, then there is $a \in \mathbb{E} \setminus \{0\}$ as above such that the inequality from (i) is strict.

Let us look a bit closer at the meaning of the above theorem. Let $X, Y \subseteq \mathbb{E}$ and $a \in \mathbb{E} \setminus \{0\}$ be as in Theorem 3.3.1 and define

$$\bar{\beta} := \frac{1}{2}\big(\sup_{x \in X} \langle a, x \rangle + \inf_{y \in Y} \langle a, y \rangle\big).$$

Moreover, define the hyperplane $H := \{ x \in \mathbb{E} : \langle a, x \rangle = \bar{\beta} \}$ and define its two associated closed half-spaces

$$H^{\leq} := \{ x \in \mathbb{E} : \langle a, x \rangle \leq \bar{\beta} \} \qquad \text{and} \qquad H^{\geq} := \{ x \in \mathbb{E} : \langle a, x \rangle \geq \bar{\beta} \}.$$

In the above theorem, condition (i) states that the sets X and Y lie each in a different half-space associated with H. To see this, note that (i) implies $\sup_{x \in X} \langle a, x \rangle \leq \bar{\beta} \leq \inf_{y \in Y} \langle a, y \rangle$, that is, $X \subseteq H^{\leq}$ and $Y \subseteq H^{\geq}$. Condition (ii) states that H is such that at least one of the convex sets is not entirely contained in H, that is, either there is $\bar{x} \in X$ such that $\langle a, \bar{x} \rangle \neq \bar{\beta}$, or there is $\bar{y} \in Y$ such that $\langle a, \bar{y} \rangle \neq \bar{\beta}$. Finally, if $0 \notin \operatorname{cl}(X - Y)$, Theorem 3.3.1 states that we can pick $a \in \mathbb{E} \setminus \{0\}$ in a way a such that the inequality from (i) holds strictly. The latter fact implies that the hyperplane H separates X and Y *strongly*: there is[1] $\varepsilon > 0$ such that $X \subseteq \{ x \in \mathbb{E} : \langle a, x \rangle \leq \bar{\beta} - \varepsilon \}$ and that $Y \subseteq \{ x \in \mathbb{E} : \langle a, x \rangle \geq \bar{\beta} + \varepsilon \}$. In particular, neither X nor Y meet the hyperplane H in this latter case.

The Hyperplane Separation Theorem forms the basis of a powerful duality theory in convex analysis. However, at this point one may find it hard to see how the above theorem induces any kind of duality theory. In fact, it is hard to say what we mean by the last phrase since the term "duality" is loosely used in many different contexts in mathematics. In the introductory remarks of a brief survey on many duality theories in mathematics [9], Atiyah says the following:

> Duality in mathematics is not a theorem, but a "principle". (...) Fundamentally, duality gives two different points of view of looking at the same object.

Here, our objects of interest are convex sets. Usually, any set is described by the points contained in it. This can be seen as a way of describing sets in an internal fashion since we are describing the set by stating which points are contained in it. In the convex case, the Hyperplane Separation Theorem allows us to represent (closed) convex sets by hyperplanes: a closed convex set $C \subseteq \mathbb{E}$ is the intersection of all closed half-spaces which contain C. This can be seen as a way to describe a convex set in an external fashion, since each hyperplane tells us which points are certainly *not* in the set. We formally show this dual description of closed convex sets in the next theorem.

Theorem 3.3.2 ([59, Theorem 11.5])**.** Let $C \subseteq \mathbb{E}$ be a closed convex set and set $H_a^{\leq}(\beta) := \{ x \in \mathbb{E} : \langle a, x \rangle \leq \beta \}$ for every $a \in \mathbb{E} \setminus \{0\}$ and $\beta \in \mathbb{R}$. Then

$$C = \bigcap \big\{ H_a^{\leq}(\beta) : a \in \mathbb{E} \setminus \{0\}, \beta \in \mathbb{R}, C \subseteq H_a^{\leq}(\beta) \big\}. \tag{3.2}$$

Proof. First, suppose $\varnothing \neq C \neq \mathbb{E}$ since the statement holds trivially otherwise. Moreover, by definition we have that C is contained in the set from the right-hand side of (3.2). Let $\bar{x} \in \mathbb{E} \setminus C$. Then $0 \notin C - \bar{x} = \operatorname{cl}(C - \bar{x})$, where the last equation holds since C is closed. Thus, by Theorem 3.3.1 there is $a \in \mathbb{E} \setminus \{0\}$ such that $\sup_{x \in C} \langle a, x \rangle < \langle a, \bar{x} \rangle$. By setting $\beta := (\sup_{x \in C} \langle a, x \rangle + \langle a, \bar{x} \rangle)/2$, we conclude $C \subseteq H_a^{\leq}(\beta)$ and $\bar{x} \notin H_a^{\leq}(\beta)$, that is, $\bar{x}$ is not in the set in the right-hand side of (3.2). □

[1] Namely, one can set $\varepsilon := (\sup_{x \in X} \langle a, x \rangle - \inf_{y \in Y} \langle a, y \rangle)/2$. In this case, we have $\bar{\beta} - \varepsilon = \inf_{y \in Y} \langle a, y \rangle > \sup_{x \in X} \langle a, x \rangle = \bar{\beta} + \varepsilon$.

The concepts from this section are the cornerstone ideas for a beautiful duality theory in convex analysis. We will focus only in a small subset of duality correspondences the above theorems yield since our purpose here is only to introduce ideas and build intuition for the study of OCO algorithms. Namely, our focus will be to look at the application of hyperplane representation of convex sets to epigraphs of functions in Section 3.4 and then to look at the set of hyperplanes which are "tangent" to the epigraphs of convex functions in Section 3.5.

3.4 Fenchel Conjugate

Since our focus in this text is optimization of convex functions, we will focus on the theory and results derived by applying the ideas from the previous section to convex functions. Before jumping to the main definition of this section, let us apply some of the ideas regarding separating hyperplanes to the epigraph of a proper closed convex function $f\colon \mathbb{E} \to (-\infty, +\infty]$.

By Theorem 3.3.2, the set $\operatorname{epi} f \subseteq \mathbb{E} \oplus \mathbb{R}$ is the intersection of a collection of closed half-spaces in $\mathbb{E} \oplus \mathbb{R}$. To study the form of the half-spaces that contain $\operatorname{epi} f$, let us look at the form of hyperplanes in $\mathbb{E} \oplus \mathbb{R}$. If $H \subseteq \mathbb{E} \oplus \mathbb{R}$ is a hyperplane, then there are $y^* \oplus \gamma^* \in \mathbb{E} \oplus \mathbb{R}$ and $\alpha \in \mathbb{R}$ such that

$$H = \{\, x \oplus \mu \in \mathbb{E} \oplus \mathbb{R} : \langle x \oplus \mu, y^* \oplus \alpha \rangle = \gamma^* \}.$$

If we multiply the equation regarding the points in H by a non-zero scalar, the hyperplane H keeps unchanged. With this in mind, let us look at two cases in which the hyperplane H may fit. Either $\alpha = 0$, in which case we say that H is *vertical*, or $\alpha \neq 0$, which we assume is the case for the remainder of this discussion. In this case, define $x^* \oplus \mu^* \coloneqq -\alpha^{-1}(y^* \oplus \gamma^*)$ and multiply the equation in the definition of H by $-\alpha^{-1}$. Then,

$$\begin{aligned} H &= \{\, x \oplus \mu \in \mathbb{E} \oplus \mathbb{R} : \langle x \oplus \mu, x^* \oplus -1 \rangle = \mu^* \} \\ &= \{\, x \oplus \mu \in \mathbb{E} \oplus \mathbb{R} : \langle x, x^* \rangle - \mu^* = \mu \}. \end{aligned}$$

Finally, define the closed half-spaces

$$H^{\leq} \coloneqq \{\, x \oplus \mu \in \mathbb{E} \oplus \mathbb{R} : \langle x, x^* \rangle - \mu^* \leq \mu \}$$

and

$$H^{\geq} \coloneqq \{\, x \oplus \mu \in \mathbb{E} \oplus \mathbb{R} : \langle x, x^* \rangle - \mu^* \geq \mu \}.$$

Since $x \oplus \mu \in \operatorname{epi} f$ for any $x \in \operatorname{dom} f$ and $\mu \in \mathbb{R}$ such that $\mu \geq f(x)$, we have $\operatorname{epi} f \not\subseteq H^{\geq}$. Moreover, by setting $h(x) \coloneqq \langle x, x^* \rangle - \mu^*$, one can readily see that $H^{\leq} = \operatorname{epi} h$. Not only that, we also have $\operatorname{epi} f \subseteq \operatorname{epi} h$ if and only if $h(x) \leq f(x)$ for every $x \in \mathbb{E}$.

From the above discussion, we conclude that the epigraph of a proper closed convex function $f\colon \mathbb{E} \to (-\infty, +\infty]$ is the intersection of half-spaces of two types. In one of these cases, the half-spaces are epigraphs of **affine functions** which lower bound f everywhere, that is, functions of the form $x \in \mathbb{E} \mapsto \langle x^*, x \rangle - \mu^*$ where $x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R}$ such that $\langle x^*, x \rangle - \mu^* \leq f(x)$ for every $x \in \mathbb{E}$. The second type of half-spaces considered are associated with vertical hyperplanes in $\mathbb{E} \oplus \mathbb{R}$. Since f is proper, $f(x) \neq -\infty$ for any $x \in \mathbb{E}$ and, thus, $\operatorname{epi} f$ cannot be the intersection of only half-spaces associated with vertical hyperplanes. The next theorem tell us an interesting fact: half-spaces associated with vertical hyperplanes do not need to be considered at all, that is, $\operatorname{epi} f$ is the intersection of all affine functions which lower bound f everywhere.

Theorem 3.4.1 ([59, Theorem 12.1])**.** Let $f\colon \mathbb{E} \to (-\infty, +\infty]$ be a proper closed convex function and define[2]

$$\mathcal{H} := \{\, x \in \mathbb{E} \mapsto \langle x^*, x\rangle - \mu^* : x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R} \text{ and } \langle x^*, x\rangle - \mu^* \leq f(x) \text{ for each } x \in \mathbb{E}\}.$$

Then $f(x) = \sup_{h \in \mathcal{H}} h(x)$ and $\operatorname{epi} f = \bigcap_{h\in\mathcal{H}} \operatorname{epi} h$ for every $x \in \mathbb{E}$.

The above discussion helps us gain some intuition on how the results from Section 3.3 can be interpreted when applied to epigraphs of convex functions. Maybe more importantly, the above discussion tries to give a bit of the intuition on the *Fenchel conjugate* of a convex function. Take the set

$$F^* := \{\, x^* \oplus \mu^* \in \mathbb{E} \oplus (-\infty, +\infty] : \operatorname{epi} f \subseteq \operatorname{epi} h,\ \text{ where } h := \langle x^*, \cdot\rangle - \mu^*\}, \tag{3.3}$$

that is, that set of points which define the affine functions that lower bound f. In this section we will study the function whose epigraph is the set F^*. This function is know as the *Fenchel conjugate* of f.

Formally, let $f\colon \mathbb{E} \to [-\infty, +\infty]$. The **(Fenchel) conjugate** of f is the function $f^*\colon \mathbb{E} \to (-\infty, +\infty]$ defined by

$$f^*(x^*) := \sup_{x\in\mathbb{E}} (\langle x^*, x\rangle - f(x)), \qquad \forall x^* \in \mathbb{E}.$$

Note that for any function $f\colon \mathbb{E} \to [-\infty, +\infty]$ we have that $\operatorname{epi} f^*$ can be written as follows:

$$\begin{aligned}
\operatorname{epi} f^* &= \{\, x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R} : f^*(x^*) \leq \mu^*\}\\
&= \{\, x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R} : \langle x^*, x\rangle - f(x) \leq \mu^* \text{ for all } x \in \mathbb{E}\}\\
&= \{\, x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R} : \langle x^*, x\rangle - \mu \leq \mu^* \text{ for all } x \oplus \mu \in \operatorname{epi} f\}\\
&= \bigcap_{x\oplus\mu\in\operatorname{epi} f} \{\, x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R} : \langle x^*, x\rangle - \mu \leq \mu^*\}\\
&= \bigcap_{x\oplus\mu\in\operatorname{epi} f} \{\, x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R} : \langle x^* \oplus \mu^*, x \oplus -1\rangle \leq \mu\}.
\end{aligned}$$

That is, $\operatorname{epi} f^*$ is the intersection of closed half-spaces. Thus, $\operatorname{epi} f^*$ is a closed convex set and by Theorem 3.2.4 we have that f^* is a closed convex function. Moreover, note that the epigraph of the conjugate matches the set from (3.3) from our discussion. On Figure 3.3 we give a illustration of the evaluation of the conjugate f^* of a function f at a point $x^* \in \mathbb{E}$.

In spite of the discussion regarding the connections between the Fenchel conjugate and separating hyperplanes, one may still feel that the results from Section 3.3 were not useful. Indeed, the only result so far which relies on the Hyperplane Separation Theorem is Theorem 3.4.1, which was not yet put to use. One may even note that the definition of Fenchel conjugate applies to general functions, not only to convex functions. The importance of Theorem 3.4.1 and, thus, of the Hyperplane Separation Theorem, is to show that the conjugate of the conjugate of a closed convex function is the function itself. This result is fundamental for most of the results regarding Fenchel conjugates. Additionally, the property that the dual of the dual of some object is the object itself is usually one of the most important properties of many duality theories (see [9] for some examples).

Theorem 3.4.2 ([59, Theorem 12.2])**.** Let $f\colon \mathbb{E} \to (-\infty, +\infty]$ be a convex function. Then f^* is a closed convex function and proper if and only if f is proper. Moreover, $(\operatorname{cl} f)^* = f^*$ and $f^{**} = \operatorname{cl} f$.

[2] In words, $\mathcal{H}$ is the set of affine functions which lower bound f.

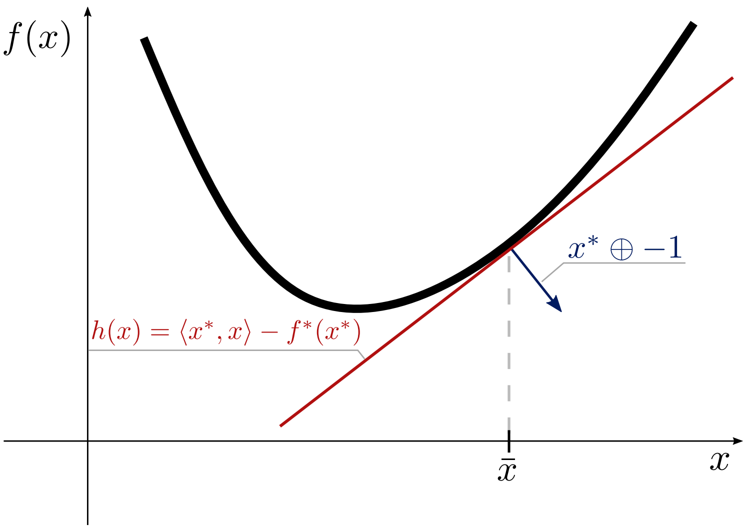

Figure 3.3: Illustration of the Fenchel conjugate of a function f evaluated at a point $x^* \in \mathbb{E}$. One can think of the supremum in the definition of f^* as sliding the red line (the graph of the affine function) vertically up to the point where it touches the graph of the function.

Proof. As we have already discussed, $\operatorname{epi} f^*$ is the intersection of closed half-spaces. Thus, $\operatorname{epi} f^*$ is a closed convex set and f^* is a closed function by Theorem 3.2.4. For the remainder of the claims in the statement, let us first look at the case when f is improper. If there is $\bar{x} \in \mathbb{E}$ such that $f(\bar{x}) = -\infty$, then $f^*(x) = +\infty$ for any $x \in \mathbb{E}$. In this case we have $(\operatorname{cl} f)^* = f^*$ since $\operatorname{cl} f$ is the constant $-\infty$ function by definition. Moreover, f^{**} is the constant $-\infty$ function, that is, $f^{**} = \operatorname{cl} f$. If $\operatorname{dom} f = \varnothing$, then $f = \operatorname{cl} f$, we clearly have $f^*(x) = -\infty$ for every $x \in \mathbb{E}$, and, thus, f^{**} is the constant $-\infty$ function.

Suppose now that f is proper. In this case, $\operatorname{cl} f$ is the function whose epigraph is $\operatorname{cl}(\operatorname{epi} f)$. With this in mind, we have

$$\begin{aligned}
\operatorname{epi} f^* &= \{\, x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R} : f^*(x^*) \leq \mu^* \} \\
&= \{\, x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R} : \langle x^*, x\rangle - f(x) \leq \mu^* \text{ for every } x \in \mathbb{E} \} \\
&= \{\, x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R} : \langle x^*, x\rangle - \mu^* \leq f(x) \text{ for every } x \in \mathbb{E} \} \\
&= \{\, x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R} : h(x) \leq f(x) \text{ for every } x \in \mathbb{E}, \text{ where } h := \langle x^*, \cdot\rangle - \mu^* \} \qquad (3.4) \\
&= \{\, x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R} : \operatorname{epi} f \subseteq \operatorname{epi} h \text{ where } h := \langle x^*, \cdot\rangle - \mu^* \} \\
&= \{\, x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R} : \operatorname{cl}(\operatorname{epi} f) \subseteq \operatorname{epi} h \text{ where } h := \langle x^*, \cdot\rangle - \mu^* \} \qquad (3.5) \\
&= \{\, x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R} : \operatorname{epi}(\operatorname{cl} f) \subseteq \operatorname{epi} h \text{ where } h := \langle x^*, \cdot\rangle - \mu^* \} \\
&= \operatorname{epi}((\operatorname{cl} f)^*),
\end{aligned}$$

where in (3.5) we used that $\operatorname{epi} h$ is closed since it is a closed half-space in $\mathbb{E} \oplus \mathbb{R}$. Thus, $f^* = (\operatorname{cl} f)^*$. By Theorem 3.4.1, $\operatorname{cl} f$ is the pointwise supremum of all affine functions $h := \langle x^*, \cdot\rangle - \mu^*$ with $x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R}$ such that $h(x) \leq (\operatorname{cl} f)(x)$ holds for every $x \in \mathbb{E}$. By (3.4), the latter holds for $x^* \oplus \mu^* \in \mathbb{E} \oplus \mathbb{R}$ if and only if we have $x^* \oplus \mu^* \in \operatorname{epi} f^*$. Therefore, for every $x \in \mathbb{E}$ we have

$$\operatorname{cl} f(x) = \sup\{\, \langle x^*, x\rangle - \mu^* : x^* \oplus \mu^* \in \operatorname{epi} f^* \} = \sup\{\, \langle x^*, x\rangle - f^*(x) : x^* \in \mathbb{E} \} = f^{**}(x). \qquad \square$$

If $f \colon \mathbb{E} \to (-\infty, +\infty]$ is a proper function, then by the definition of conjugate one can easily get the **Fenchel-Young inequality**:

$$\langle x^*, x\rangle \leq f^*(x^*) + f(x), \qquad \forall x^*, x \in \mathbb{E}.$$

In spite of the simplicity of the above inequality, the case when this inequality holds as an equation will be very important when we look at subgradients in Section 3.5.

For the sake of concreteness, let us quickly compute the conjugates of some functions. As a warm-up, let us compute the conjugate of the indicator function of a convex set $C \subseteq \mathbb{E}$. For each $x^* \in \mathbb{E}$, we have

$$(\delta(\cdot \,|\, C))^*(x^*) = \sup_{x\in\mathbb{E}}\big(\langle x^*, x\rangle - \delta(x \,|\, C)\big) = \sup_{x\in C}\langle x^*, x\rangle = \delta^*(x^* \,|\, C).$$

That is, the conjugate of the indicator function is the support function! This is one of the reason of the similarity between the notation for both of these functions.

Fortunately, we will not need to compute the conjugate of very complex functions. Still, if we know the conjugate of a convex function $f \colon \mathbb{E} \to (-\infty, +\infty]$, we often need to deal with the conjugate of the functions λf for any $\lambda \in \mathbb{R}_{++}$. The following theorem shows how to compute such conjugates and, even though we skip the proof for the sake of conciseness, the proof follows easily from the definition of conjugate.

Theorem 3.4.3 ([59, Theorem 16.1])**.** If $f \colon \mathbb{E} \to (-\infty, +\infty]$ is a proper convex function, then for any $\lambda \in \mathbb{R}_{++}$ and $x^* \in \mathbb{E}$ we have $(\lambda f)^*(x^*) = \lambda f^*(\lambda^{-1} x^*)$.

Let us now compute the conjugate of the *negative entropy* function. This function will be used extensively in the text and understanding the behavior of its conjugate shall be very useful later on.

Proposition 3.4.4. Define $R(x) \coloneqq \frac{1}{\eta}\sum_{i=1}^d [x_i > 0] x_i \ln x_i + \delta(x \,|\, \mathbb{R}^d_+)$ for each $x \in \mathbb{R}^d$, where $\eta \in \mathbb{R}$ is some positive constant. Then R is a proper closed convex function and

$$R^*(y) = \frac{1}{\eta}\sum_{i=1}^d e^{\eta y_i - 1}, \qquad \forall y \in \mathbb{R}^d.$$

Proof. Define $\psi(\alpha) \coloneqq [\alpha > 0]\alpha \ln \alpha + \delta(\alpha \,|\, \mathbb{R}_+)$ for every $\alpha \in \mathbb{R}$. Note that $R(x) = \sum_{i=1}^d \frac{1}{\eta}\psi(x_i)$. First, let us show that *psi* is a closed convex function, which implies that so is R. Define the function $\phi \coloneqq \psi + \delta(\cdot \,|\, \mathbb{R}_{++})$. Since $\phi''(\alpha) = \alpha^{-1} > 0$ for any $\alpha \in \mathbb{R}_{++}$ by Lemma 3.1.1 we conclude that ϕ is a convex function. Moreover, $\lim_{\alpha \downarrow 0} \phi(\alpha) = 0 = \psi(\alpha)$. Thus, $\operatorname{cl} \phi = \psi$ and we conclude that ψ is a closed convex function. Thus, R is a closed convex function and one can easily see that R is proper.

Let us now show that,

for any $\beta \in \mathbb{R}$, the supremum $\sup_{\alpha\in\mathbb{R}}(\beta\alpha - \psi(\alpha)) = \psi^*(\beta)$ is attained by $e^{\beta-1}$ and $\psi^*(\beta) = e^{\beta-1}$. (3.6)

To see this, let $\beta \in \mathbb{R}$ and define $h(\alpha) \coloneqq \beta\alpha - \psi(\alpha)$ for each $\alpha \in \mathbb{R}$. Note that for any $\alpha \in \mathbb{R}_{++}$ we have

$$h'(\alpha) = \beta - \psi'(\alpha) = \beta - 1 - \ln \alpha.$$

Therefore, for $\alpha \in \mathbb{R}_{++}$ we have that $h'(\alpha) = 0$ if and only if $\alpha = e^{\beta-1}$, that is, $e^{\beta-1}$ is a critical point of ψ. Since $\lim_{\alpha\to+\infty} \psi(\alpha) = +\infty$, we have $\inf_{\alpha\in\mathbb{R}_{++}}(\beta\alpha - \psi(\alpha)) = -\infty$. Thus, $\sup_{\alpha\in\mathbb{R}_{++}}(\beta\alpha - \psi(\alpha))$ is attained by $e^{\beta-1}$. Moreover, one can check that $h(e^{\beta-1}) = e^{\beta-1}$. Since $\psi(0) = 0$, we have that $e^{\beta-1} > 0 = 0\beta - \psi(0)$. Finally, noting that $\beta\alpha - \psi(\alpha) = -\infty$ for $\alpha \in \mathbb{R}$ with $\alpha < 0$ finishes the proof of (3.6).

Therefore, for every $y \in \mathbb{R}^d$

$$R^*(y) = \sup_{z \in \mathbb{R}^d} \Big(y^\mathsf{T} z - R(z) \Big) = \sup_{z \in \mathbb{R}^d} \Big(\sum_{i \in E} \Big(y_i z_i - \frac{1}{\eta}\psi(z_i) \Big) \Big) = \sum_{i=1}^d \Big(\frac{1}{\eta}\psi \Big)^*(y_i)$$

$$\overset{\text{Thm. 3.4.3}}{=} \sum_{i=1}^d \frac{1}{\eta}\psi^*(\eta y_i) \overset{(3.6)}{=} \frac{1}{\eta} \sum_{i=1}^d e^{\eta y_i - 1}. \qquad \square$$

3.5 Subgradients

Let us look now at the idea of *subgradients*, a generalization of gradients for non-differentiable functions which is specially fruitful for convex functions. Let $f \colon \mathbb{E} \to \mathbb{R}$. A point $x^* \in \mathbb{E}$ is a **subgradient** of f at x if x^* satisfies, for every $z \in \mathbb{E}$, the *subgradient inequality*

$$f(z) \geq f(x) + \langle x^*, z - x \rangle.$$

The **subdifferential** of f at x is the set $\partial f(x)$ comprised of all the subgradients of f at x, and the **subdifferential** of f is the mapping $\partial f \colon x \in \mathbb{E} \mapsto \partial f(x)$. Moreover, f is **subdifferentiable** at a point $x \in \mathbb{E}$ if $\partial f(x) \neq \varnothing$.

Let us try to understand the subgradient inequality for a subgradient $x^* \in \partial f(x)$ of a function $f \colon \mathbb{E} \to (-\infty, +\infty]$ at a point $x \in \mathbb{E}$. On Figure 3.4 we have an representative illustration of some subgradients. Note that the function given by $h(z) \coloneqq f(x) + \langle x^*, z - x \rangle$ for every $z \in \mathbb{E}$ is an affine function. Thus, the graph of this function is a hyperplane. Namely, it is the hyperplane H given by $H \coloneqq \{\, z \oplus h(z) \in \mathbb{E} \oplus \mathbb{R} : z \in \mathbb{E} \}$. In this case, note that H has some special properties. First of all, by the definition of subgradient we have that $\operatorname{epi} f$ is contained in one of the closed half-spaces associated with H (namely, $\operatorname{epi} h$). Not only that, we have that the hyperplane and the epigraph of the function meet on at least one point: $x \oplus f(x) \in \operatorname{epi} f \cap H$. A hyperplane $H \subseteq \mathbb{E}$ such that a set $C \subseteq \mathbb{E}$ is entirely contained in one of its closed half-spaces and such that $H \cap C \neq \varnothing$ is a **supporting hyperplane** of C. Thus, subgradients of a function f at a point x are associated with supporting hyperplanes of $\operatorname{epi} f$ which meet the later at least at $x \oplus f(x)$.

Although we are able to picture a little better the meaning of the subgradient inequality, we have never shown that subgradients exist. Indeed, there may be points on the domain of a function at which there are no subgradients. The next theorem shows that this may only happen at the (relative) boundary of the effective domain.

Theorem 3.5.1 ([59, Theorem 23.4])**.** Let $f \colon \mathbb{E} \to \mathbb{R}$ be a proper convex function and let $x \in \mathbb{E}$. If $x \notin \operatorname{dom} f$, then $\partial f(x) = \varnothing$, and if $x \in \operatorname{ri}(\operatorname{dom} f)$, then $\partial f(x) \neq \varnothing$. Finally, $\partial f(x)$ is nonempty and bounded if and only if $x \in \operatorname{int}(\operatorname{dom} f)$.

From the previous discussion, the reader may already be guessing that the Fenchel conjugate and subgradients have deep connections due to their relation with separating hyperplanes. The Fenchel conjugate of a function f is built based on all hyperplanes which have $\operatorname{epi} f$ entirely contained in one of its closed half-spaces and subgradients are associated with only the *supporting* hyperplanes which have $\operatorname{epi} f$ contained in one of its closed half-spaces. Indeed, on Figure 3.3 one may note that, at least graphically, the hyperplane in red is a supporting hyperplane at the point $\bar{x}$ and, in this case, $x^* \in \partial f(\bar{x})$. The intuition is that the supremum from the definition of $f^*(x^*)$ is, in some sense, sliding a hyperplane until it touches the graph of f, and at the point of contact $\bar{x} \in \mathbb{E}$ the point x^* is actually a subgradient of f at $\bar{x}$. Finally, such a point $\bar{x}$ where the "sliding hyperplane" touches the graph of f is exactly the point which attains the supremum from the definition of $f^*(x^*)$.

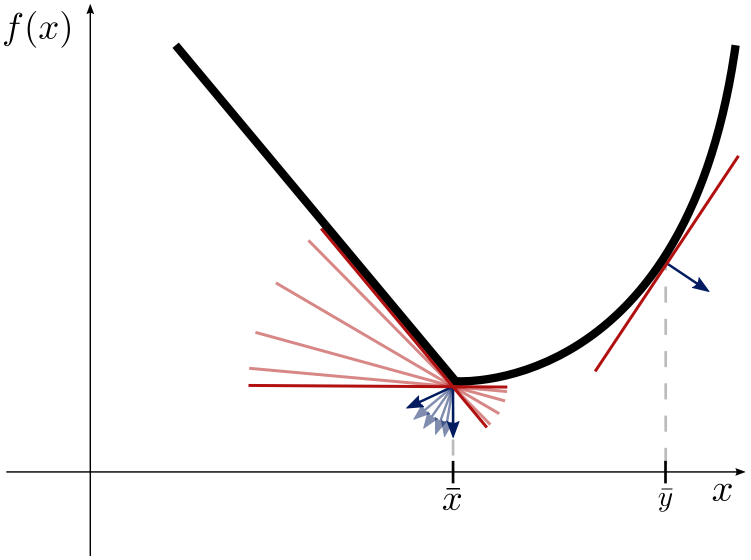

Figure 3.4: Illustration of the subgradients of f at the points $\bar{x}$, where f has multiple subgradients, and $\bar{y}$, where f is differentiable.

The next theorem shows this and other relations between conjugate functions and subgradients. In spite of its simple proof, this is one of the most used theorems throughout the whole text since it establishes useful relations regarding subgradients and Fenchel conjugates. As one will see, the proof basically shows that the points which attain the supremum in the definition of conjugate (that is, the point of contact when the sliding hyperplane hits the graph of the function in our intuitive discussion) are subgradients of the function.

Theorem 3.5.2 ([59, Theorem 23.5])**.** Let $f\colon \mathbb{E} \to (-\infty, +\infty]$ be proper and convex and let $x, x^* \in \mathbb{E}$. Then, the following are equivalent:

(i) $x^* \in \partial f(x)$;

(ii) x attains the supremum $\sup_{z \in \mathbb{E}}(\langle x^*, z\rangle - f(z)) = f^*(x^*)$;

(iii) $f^*(x^*) + f(x) \leq \langle x^*, x\rangle$;

(iv) $f^*(x^*) + f(x) = \langle x^*, x\rangle$.

Moreover, if $(\operatorname{cl} f)(x) = f(x)$, the following can be added to the list:

(v) $x \in \partial f^*(x^*)$;

(vi) x^* attains the supremum $\sup_{z^* \in \mathbb{E}}(\langle z^*, x\rangle - f^*(z^*)) = f(x)$;

(vii) $x^* \in \partial(\operatorname{cl} f)(x)$.

Proof. [(i) $\iff$ (ii)] Note that $x^* \in \partial f(x)$ if and only if $f(z) \geq f(x) + \langle x^*, z - x\rangle$ for every $z \in \mathbb{E}$, that is, if and only if $\langle x^*, x\rangle - f(x) \geq \langle x^*, z\rangle - f(z)$ for every $z \in \mathbb{E}$. Since equality holds for $z = x$, we are done.

[(iii) $\iff$ (iv)] By the Fenchel-Young inequality, $f(x^*) + f(x) \geq \langle x^*, x\rangle$. Thus, $f(x^*) + f(x) \leq \langle x^*, x\rangle$ if and only if equality holds.

[(ii) $\iff$ (iv)] we have $\langle x^*, x\rangle - f(x) = f^*(x^*) = \sup_{z \in \mathbb{E}}(\langle x^*, z\rangle - f(z))$ if and only if the latter supremum is attained at $z = x$.

For the remainder of the proof, suppose $(\operatorname{cl} f)(x) = f(x)$. By Theorem 3.4.2, this implies that $f^{**}(x) = f(x)$

[(iv) $\iff$ (v)] By the equivalence of (i) and (iv) applied to f^*, we have $x \in \partial f(x^*)$ if and only if $\langle x, x^*\rangle = f^{**}(x) + f^*(x^*) = f(x) + f^*(x)$.

[(v) $\iff$ (vi)] By the equivalence of (i) and (ii) applied to f^*, we have $x \in \partial f^*(x^*)$ if and only if x attains $\sup_{z^* \in \mathbb{E}}(\langle z^*, x\rangle - f^*(z^*)) = f^{**}(x) = f(x)$.

[(v) $\iff$ (vii)] By the equivalence between (i) and (v), we have $x^* \in \partial(\operatorname{cl} f)(x)$ if and only if $x \in \partial(\operatorname{cl} f)^*(x^*) = \partial f^*(x^*)$, where the last equation holds since $(\operatorname{cl} f)^* = f^*$ by Theorem 3.4.2. □

For the sake of concreteness, let us compute the subdifferential of the indicator function of a convex set. In order to do so, we need to define *normal cones*. The **normal cone** of C at $x \in C$ is the set $N_C(x) \coloneqq \{\, d \in \mathbb{E} : \langle d, z - x\rangle \leq 0 \text{ for every } z \in C\}$ and it is the empty set for every $c \in \mathbb{E} \setminus C$. That is, a direction $d \in \mathbb{E}$ is in the normal cone of a set $C \subseteq \mathbb{E}$ at $x \in C$ is the inner product of d with any vector starting at x and point into the set C is non-positive. In the two-dimensional case (i.e., $C \subseteq \mathbb{R}^2$), this is equivalent to saying that the direction d forms a degree of at least 90 degrees with any vector starting at x and pointing into C. The idea of normal cones of convex sets and the following lemma will be fundamental for the study of optimality conditions in the next section.

Lemma 3.5.3. Let $C \subseteq \mathbb{E}$ be a convex set. Then, for every $x \in C$ we have

$$N_C(x) = \partial(\delta(\cdot \,|\, C))(x).$$

Proof. Let $x \in C$ and $x^* \in \mathbb{E}$. By definition, $x^* \in (\partial\delta(\cdot \,|\, C))(x)$ if and only if $\delta(z \,|\, C) \geq \delta(x \,|\, C) + \langle x^*, z - x\rangle = \langle x^*, z - x\rangle$ for every $z \in \mathbb{E}$, that is, if and only if $0 \geq \langle x^*, z - x\rangle$ for every $z \in C$. □

If we have a collection of m functions from $\mathbb{E}$ to $(-\infty, +\infty]$, all of them differentiable at a point $\bar{x} \in \mathbb{E}$, then the gradient of the sum of the m functions at $\bar{x}$ is just the sum of the gradients of each function at $\bar{x}$. In the case where the functions are subdifferentiable at $\bar{x}$ instead of differentiable, things are not that simple. The subdifferential of the sum of functions at a point may be equal to the sum of the subdifferentials, but this depends on some conditions on the relative interior of the effective domains of the functions we are looking at.

Theorem 3.5.4 ([59, Theorem 23.8])**.** Let $f_1, \dots, f_m \colon \mathbb{E} \to (-\infty, +\infty]$ be proper convex functions and set $f \coloneqq \sum_{i=1}^m f_i$. Then

$$\partial f_1(x) + \cdots + \partial f_m(x) \subseteq \partial f(x),$$

and equality holds if $\bigcap_{i \in [m]} \operatorname{ri}(\operatorname{dom} f_i) \neq \varnothing$.

It is high time we relate subgradients and gradients. As one may already guess, if a convex function $f \colon \mathbb{E} \to (-\infty, +\infty]$ is differentiable at a point $x \in \mathbb{E}$, then $\nabla f(x)$ is a subgradient of f. The next theorem shows that $\nabla f(x)$ is the only subgradient of f at x and, maybe more importantly, shows that if the subdifferential of f at x is a singleton, then f is actually differentiable at x.

Theorem 3.5.5 ([59, Theorem 25.1])**.** Let $f \colon \mathbb{E} \to (-\infty, +\infty]$ be convex, and let $x \in \operatorname{dom} f$. If f is differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$. Moreover, if $\partial f(x)$ is a singleton, then f is differentiable at x.

From Theorem 3.5.2, we know that in the case of a proper closed convex function $f \colon \mathbb{E} \to (-\infty, +\infty]$, the point $x^* \in \mathbb{E}$ is a subgradient of f at $x \in \mathbb{E}$ if and only if x is a subgradient of f^*

at x^*. That is, the conjugacy operation switches the roles of subgradient and point of evaluation of the function. This together with the above theorem shows a fact in the case where both f and f^* are differentiable which will be extremely important for Chapter 5: the gradient map of f is the inverse of the gradient map of f^*

Corollary 3.5.6. Let $f\colon \mathbb{E} \to (-\infty, +\infty]$ be a closed proper convex function. Moreover, let $D, D^* \subseteq \mathbb{E}$ be open convex sets such that f is differentiable on D and f^* is differentiable on D^*. Then, for every $x \in D$ such that $\nabla f(x) \in D^*$, we have $\nabla f^*(\nabla f(x)) = x$, and for every $x^* \in D^*$ such that $\nabla f^*(x^*) \in D$ we have $\nabla f(\nabla f^*(x^*)) = x^*$.

Proof. Let $x \in D$ be such that $\nabla f(x) \in D^*$. By Theorem 3.5.5, we have $\partial f^*(\nabla f(x)) = \{\nabla f^*(\nabla f(x))\}$. Thus, by Theorem 3.5.2 items (i) and (v) we have $x \in \partial f^*(\nabla f(x)) = \{\nabla f^*(\nabla f(x))\}$, that is, $\nabla f(\nabla f^*(x)) = x$. Since f is closed, $f = f^{**}$ by Theorem 3.4.2, which together with what we have just proved yields that if $x^* \in D^*$ is such that $\nabla f^*(x^*) \in D$, then $\nabla f(\nabla f^*(x^*)) = x^*$. □

3.6 Optimality Conditions

Since the focus of this text is the optimization of convex functions (often over convex sets), it is of no surprise that we are interested in necessary and sufficient conditions for $x \in \mathbb{E}$ to attain $\inf_{x \in C} f(x)$, where $f\colon \mathbb{E} \to (-\infty, +\infty]$ is a proper convex function and $C \subseteq \mathbb{E}$ is a convex set. First, let us look at one of the nicest facts about minimizers of convex functions: if a point minimizes a convex function "locally", then it is a (global) minimizer of the function.

Lemma 3.6.1. Let $f\colon \mathbb{E} \to (-\infty, +\infty]$ be a proper convex function and set $\mathbb{B} := \{\, x \in \mathbb{E} : \langle x, x \rangle \leq 1\}$. If $\bar{x} \in \mathbb{E}$ is such that there is $\varepsilon > 0$ such that $f(\bar{x}) \leq f(z)$ for any $z \in \varepsilon\mathbb{B} + \bar{x}$, then $f(\bar{x}) \leq f(z)$ for any $z \in \mathbb{E}$.

Proof. Let $\bar{x} \in \mathbb{E}$ and $\varepsilon > 0$ be as in the statement of the lemma, and let $z \in \mathbb{E} \setminus \{\bar{x}\}$. By the convexity of f, for any $\lambda \in (0, 1]$ we have

$$f(\bar{x} + \lambda(z - \bar{x})) - f(\bar{x}) \leq \lambda(f(\bar{x}) - f(z)) \implies \frac{f(\bar{x} + \lambda(z - \bar{x})) - f(\bar{x})}{\lambda} \leq f(z) - f(\bar{x}).$$

Set

$$\delta := \frac{\varepsilon}{\langle z - \bar{x}, z - \bar{x} \rangle} > 0.$$

Then, for any $\lambda \in (0, \delta]$, we have $\lambda(z - \bar{x}) \in \varepsilon\mathbb{B}$. Thus, for any $\lambda \in (0, \delta]$,

$$0 \leq \frac{f(\bar{x} + \lambda(z - \bar{x})) - f(\bar{x})}{\lambda} \leq f(z) - f(\bar{x}) \implies f(\bar{x}) \leq f(z). \qquad \square$$

The above lemma is one of the reasons why convex functions are so ubiquitous in optimization: if we find a point $\bar{x}$ which minimizes a convex function f locally, we know that it is also a global minimizer of f.

When devising algorithms for optimization, sufficient (and hopefully necessary) conditions of optimality are extremely helpful. There are well-known optimality conditions from Calculus, but to hold they usually rely on the functions to be differentiable. Fortunately, subgradients are specially useful as tools to check optimality in the case of convex and not necessarily differentiable functions. Indeed, let $f\colon \mathbb{E} \to (-\infty, +\infty]$ be a proper convex function and $\bar{x} \in \mathbb{E}$ be such that $f(\bar{x}) \leq f(z)$ for any $z \in \mathbb{E}$. Equivalently,

$$f(\bar{x}) + \langle 0, z - \bar{x} \rangle = f(\bar{x}) \leq f(z), \qquad \forall z \in \mathbb{E}.$$

That is, $\bar{x}$ minimizes f if and only if 0 is a subgradient of f at $\bar{x}$! Even though this condition does not apply for the optimization problem $\inf_{x \in C} f(x)$ where $C \subseteq \mathbb{E}$ is convex, it forms the basis for the optimality condition we look next. The proof of the next theorem relies on the fact that $\inf_{x \in C} f(x) = \inf_{x \in \mathbb{E}}(f(x) + \delta(x \,|\, C))$ and then applies the subgradient condition we have just seen with Theorem 3.5.4 which shows the form of the subdifferential of the sum of functions.

Theorem 3.6.2 ([59, Theorem 27.4])**.** Let $C \subseteq \mathbb{E}$ be a nonempty convex set and let $f\colon \mathbb{E} \to (-\infty, +\infty]$ be proper convex function. A sufficient condition for a point $x \in \mathbb{E}$ to attain $\inf_{z \in C} f(z)$ is that $\partial f(x) \cap (-N_C(x))$ is nonempty. If $\operatorname{ri}(\operatorname{dom} f) \cap \operatorname{ri} C \neq \varnothing$, this condition is also necessary.

Proof. Define $F \coloneqq f + \delta(\cdot \,|\, C)$ and let $\bar{x} \in C$. In this way we have $\inf_{x \in C} f(x) = \inf_{x \in \mathbb{E}} F(x)$ and $\bar{x}$ attains the former infimum if and only if it attains the second infimum. Moreover, by the definition of subgradient we have that $\bar{x}$ attains $\inf_{x \in \mathbb{E}} F(x)$ if and only if $0 \in \partial F(\bar{x})$. By Lemma 3.5.3 we have $\partial(\delta(\cdot \,|\, C))(\bar{x}) = N_C(\bar{x})$. Thus, by Theorem 3.5.4,

$$\partial f(\bar{x}) + N_C(\bar{x}) \subseteq \partial F(\bar{x}), \tag{3.7}$$

and equality holds above if $\operatorname{ri}(\operatorname{dom} f) \cap \operatorname{ri} C \neq \varnothing$. Finally, note that $0 \in \partial f(\bar{x}) + N_C(\bar{x})$ if and only if $\partial f(x) \cap (-N_C(x))$ is nonempty. Thus, if $\partial f(x) \cap (-N_C(x)) \neq \varnothing$, then $0 \in \partial F(\bar{x})$ and, thus, $\bar{x}$ attains $\inf_{x \in \mathbb{E}} F(x)$. Moreover, if $\operatorname{ri}(\operatorname{dom} f) \cap \operatorname{ri} C \neq \varnothing$, then (3.7) holds as an equation and, thus, $0 \in \partial F(\bar{x})$ in this case if and only if $\partial f(x) \cap (-N_C(x))$ is nonempty. □

3.7 Convex Spectral Functions

On some parts of the text, mainly on Chapter 6 and on Section 7.3, we will deal with convex functions on the space of $d \times d$ symmetric matrice $\mathbb{S}^d$ equipped with the trace inner product given by $\langle X, Y \rangle \coloneqq \operatorname{Tr}(XY)$ for every $X, Y \in \mathbb{S}^d$. Luckily, the matrix functions we meet in this text are special: they depend only on the eigenvalues of the matrix given as input, which we call by *spectral functions*. In this section we develop tools to compute conjugates, gradients, and subgradients of such spectral functions based on the results from previous sections.

First, let us define spectral functions. Let $f\colon \mathbb{R}^d \to [-\infty, +\infty]$. Define the function $f_{\mathbb{S}}\colon \mathbb{S}^d \to [-\infty, +\infty]$ by

$$f_{\mathbb{S}}(X) = f(\lambda^{\uparrow}(X)), \qquad \forall X \in \mathbb{S}^d,$$

where $\lambda^{\uparrow}\colon \mathbb{S}^d \to \mathbb{R}^d$ extracts the eigenvalues of the matrix given as input in non-decreasing order (this and other definitions can be found on Section 1.1.3). Moreover, the function f is **symmetric** if for every permutation matrix $P \in \mathbb{R}^{d \times d}$ and $x \in \mathbb{R}^d$ we have $f(Px) = f(x)$. Functions on $\mathbb{S}^d$ which can be expressed as $h_{\mathbb{S}}$ for some symmetric function $h\colon \mathbb{E} \to (-\infty, +\infty]$ are said to be **spectral**. We require the functions over which spectral functions are based on to be symmetric since the order of the eigenvalues should not change the function's behavior, even though in our definition of spectral functions we use the eigenvalues in increasing order.

Before looking at the Fenchel conjugates of spectral functions, we need a result which connects the inner products from $\mathbb{S}^d$ and from $\mathbb{R}^d$. We skip the proof of this inequality for the sake of conciseness, but it is of fundamental importance for the remainder of the results of this section.

Theorem 3.7.1 ([45, Theorem 2.2])**.** Let $X, Y \in \mathbb{S}^d$. Then

$$\operatorname{Tr}(XY) \leq \lambda^{\uparrow}(X)^{\mathsf{T}} \lambda^{\uparrow}(Y).$$

If equality holds, then there is an orthogonal matrix $Q \in \mathbb{R}^{d \times d}$ such that $Q^{\mathsf{T}} X Q = \operatorname{Diag}(\lambda^{\uparrow}(X))$ and $Q^{\mathsf{T}} Y Q = \operatorname{Diag}(\lambda^{\uparrow}(Y))$.

Let us look now at the Fenchel conjugate of spectral functions. Let $f\colon \mathbb{E} \to (-\infty, +\infty]$ be a symmetric function. Intuitively, if we know the conjugate f^* of f, we should be able to compute the conjugate $(f_{\mathbb{S}})^*$ of $f_{\mathbb{S}}$. Not only that, ideally one would hope $(f_{\mathbb{S}})^* = (f^*)_{\mathbb{S}}$ to hold. This is indeed the case, holding even for not necessarily convex functions.

Theorem 3.7.2 ([17, Theorem 5.2.2])**.** Let $f\colon \mathbb{R}^d \to (-\infty, +\infty]$ be a symmetric function. Then

$$(f_{\mathbb{S}})^* = (f^*)_{\mathbb{S}}.$$

Proof. Let $X^* \in \mathbb{S}^d$. Then

$$\begin{aligned}
(f_{\mathbb{S}})^*(X^*) &= \sup_{X \in \mathbb{S}^d} (\langle X^*, X\rangle - f_{\mathbb{S}^d}(X)) \\
&= \sup_{X \in \mathbb{S}^d} (\langle X^*, X\rangle - f(\lambda^{\uparrow}(X))) \\
&\leq \sup_{X \in \mathbb{S}^d} (\lambda^{\uparrow}(X^*)^{\mathsf{T}} \lambda^{\uparrow}(X) - f(\lambda^{\uparrow}(X))) \\
&= \sup_{x \in \mathbb{R}^d} (\lambda^{\uparrow}(X^*)^{\mathsf{T}} x - f(x)) \\
&= f^*(\lambda^{\uparrow}(X^*)) = (f^*)_{\mathbb{S}}(X^*).
\end{aligned}$$

Moreover, by the Spectral Decomposition Theorem (Theorem 1.1.1), there is an orthogonal matrix $Q \in \mathbb{R}^d$ such that $X^* = Q \operatorname{Diag}(\lambda^{\uparrow}(X^*))Q^{\mathsf{T}}$. Since $Q^{\mathsf{T}} Q = I$ and $\operatorname{Tr}(AB) = \operatorname{Tr}(BA)$ for any matrices real matrices A and B of appropriate dimensions, we have

$$\begin{aligned}
(f^*)_{\mathbb{S}}(X^*) &= \sup_{x \in \mathbb{R}^d} (\lambda^{\uparrow}(X^*)^{\mathsf{T}} x - f(x)) \\
&= \sup_{x \in \mathbb{R}^d} \big(\operatorname{Tr}(\lambda^{\uparrow}(X^*)^{\mathsf{T}} x) - f(\lambda^{\uparrow}(Q \operatorname{Diag}(x) Q^{\mathsf{T}}))\big) \\
&= \sup_{x \in \mathbb{R}^d} \big(\operatorname{Tr}(\operatorname{Diag}(\lambda^{\uparrow}(X^*)) \operatorname{Diag}(x)) - f_{\mathbb{S}}(Q \operatorname{Diag}(x) Q^{\mathsf{T}})\big) \\
&= \sup_{x \in \mathbb{R}^d} \big(\operatorname{Tr}(Q^{\mathsf{T}} Q \operatorname{Diag}(\lambda^{\uparrow}(X^*)) Q^{\mathsf{T}} Q \operatorname{Diag}(x)) - f_{\mathbb{S}}(Q \operatorname{Diag}(x) Q^{\mathsf{T}})\big) \\
&= \sup_{x \in \mathbb{R}^d} \big(\operatorname{Tr}(Q^{\mathsf{T}} X^* Q \operatorname{Diag}(x)) - f_{\mathbb{S}}(Q \operatorname{Diag}(x) Q^{\mathsf{T}})\big) \\
&= \sup_{x \in \mathbb{R}^d} \big(\operatorname{Tr}(X^* Q \operatorname{Diag}(x) Q^{\mathsf{T}}) - f_{\mathbb{S}}(Q \operatorname{Diag}(x) Q^{\mathsf{T}})\big) \\
&\leq \sup_{X \in \mathbb{S}^d} \big(\operatorname{Tr}(X^* X) - f_{\mathbb{S}}(X)\big) \\
&= \sup_{X \in \mathbb{S}^d} \big(\langle X^*, X\rangle - f_{\mathbb{S}}(X)\big) \\
&= (f_{\mathbb{S}})^*(X^*).
\end{aligned}$$

□

With the above theorem in hands, many results for spectral functions can be almost directly obtained from results from sections Sections 3.4 and 3.5. For example, the next corollary states that a spectral function based on a closed convex function is also closed convex function. The proof follows almost directly from the above theorem and Theorem 3.4.2 (the "double conjugacy theorem").

Corollary 3.7.3 ([17, Corollary 5.2.3])**.** Let $f\colon \mathbb{R}^d \to (-\infty, +\infty]$ be proper symmetric closed convex function. Then $f_{\mathbb{S}}$ is a proper closed convex function.

Proof. The fact that $f_{\mathbb{S}}$ is a proper convex function is trivial. Thus, we need only o show that $\mathrm{cl}(f_{\mathbb{S}}) = f_{\mathbb{S}}$ holds. Using Theorem 3.4.2 and Theorem 3.7.2 we have

$$\mathrm{cl}(f_{\mathbb{S}}) = (f_{\mathbb{S}})^{**} = ((f^*)_{\mathbb{S}})^* = (f^{**})_{\mathbb{S}} = (\mathrm{cl}\, f)_{\mathbb{S}} = f_{\mathbb{S}}. \qquad \square$$

Since we will often need look at infima regarding spectral functions, it is of no surprise that we will need to compute subgradients or even gradients of these functions at some points. Although we skip the proofs for the sake of conciseness, the proofs of the following corollaries follow from the results from Section 3.5 together with Theorem 3.7.2 and, sometimes, the conditions for Theorem 3.7.1 to hold as an equation.

Corollary 3.7.4 ([17, Corollary 5.2.4])**.** Let $f\colon \mathbb{R}^d \to (-\infty, +\infty]$ be proper, symmetric, closed, and convex. Then, for any $X, Y \in \mathbb{S}^d$, the following are equivalent:

(i) $Y \in \partial f_{\mathbb{S}}(X)$;

(ii) $\lambda^{\uparrow}(Y) \in \partial f(\lambda^{\uparrow}(X))$ and there is an orthogonal matrix $Q \in \mathbb{R}^{d \times d}$ such that $Q^{\mathsf{T}} X Q = \mathrm{Diag}(\lambda^{\uparrow}(X))$ and $Q^{\mathsf{T}} Y Q = \mathrm{Diag}(\lambda^{\uparrow}(Y))$.

Corollary 3.7.5 ([17, Corollary 5.2.5])**.** Let $f\colon \mathbb{R}^d \to (-\infty, +\infty]$ be proper, symmetric, closed, and convex, and let $X \in \mathbb{S}^d$. Then $f_{\mathbb{S}}$ is differentiable at X if and only if f is differentiable at $\lambda^{\uparrow}(X)$. Moreover, if $f_{\mathbb{S}}$ is differentiable at X, then, for any orthogonal matrix $Q \in \mathbb{R}^{d \times d}$ such that $Q^{\mathsf{T}} X Q = \lambda^{\uparrow}(X)$, we have

$$\nabla f_{\mathbb{S}}(X) = Q\, \mathrm{Diag}(\nabla f(\lambda^{\uparrow}(X))) Q^{\mathsf{T}}.$$

3.8 Norms

On the Euclidean space $\mathbb{E}$, norms are functions which assign non-negative "lengths" or "sizes" to each point in $\mathbb{E}$, assigning length zero only to the zero vector. In the case of $\mathbb{R}^d$, one is most used with the *euclidean norm*

$$x \in \mathbb{R}^d \mapsto \left(\sum_{i=1}^{d} x_i^2 \right)^{\frac{1}{2}}.$$

In this text we will be interested in cases where we use a bit less standard norms. Thus, let us define what properties a function needs to satisfy to be a norm and then let us see some concepts and results related to convex analysis duality applied to norms.

Let $\|\cdot\|\colon \mathbb{E} \to \mathbb{R}$. We say that $\|\cdot\|$ is a **norm** on $\mathbb{E}$ if, for all $u, v \in \mathbb{E}$,

(i) $\|v\| \geq 0$, and equality holds if and only if $v = 0$,

(ii) $\|\alpha v\| = |\alpha| \|v\|$ for every $\alpha \in \mathbb{R}$,

(iii) $\|u + v\| \leq \|u\| + \|v\|$, also known as **triangle inequality**.

If $\|\cdot\|$ satisfies only conditions (ii) and (iii) and $\|v\| \geq 0$ for any $v \in \mathbb{E}$, then $\|\cdot\|$ is a **semi-norm** on $\mathbb{E}$. Condition (ii) and the non-negativity imply that semi-norms and norms are convex functions. Moreover, one may verify that if $\|\cdot\|$ is a norm on $\mathbb{E}$, then it is a continuous[3] function on $\mathbb{E}$.

[3] The definition of continuous function itself (at least at first sight) depends on a norm, so say that all norms are continuous without defining continuity is a somewhat circular statement. Thus, we use the following definition of continuity: a function $f\colon \mathbb{E} \to [-\infty, +\infty]$ is **continuous** at a point $\bar{x} \in \mathbb{E}$ if for every $\varepsilon > 0$ there is $\delta > 0$ such that, for any $x \in \mathbb{E}$ with $\langle \bar{x} - x, \bar{x} - x \rangle \leq \delta$, we have $|f(\bar{x}) - f(x)| \leq \varepsilon$. That is, we use the (squared) euclidean norm in $\mathbb{E}$ to define continuity.

The **euclidean norm** or **ℓ_2-norm** on $\mathbb{E}$ is the norm $\|\cdot\|_2$ given by $\|x\|_2 \coloneqq \sqrt{\langle x, x\rangle}$ for every $x \in \mathbb{E}$. Additionally, throughout the text we may use some special and known norms for $\mathbb{R}^d$:

- the **ℓ_1-norm** given by $\|x\|_1 \coloneqq \sum_{i=1}^d |x_i|$ for every $x \in \mathbb{R}^d$,
- the **ℓ_∞-norm** given by $\|x\|_\infty \coloneqq \max_{i\in[d]} |x_i|$ for every $x \in \mathbb{R}^d$,
- the **ℓ_p-norm** for $p \in (1, \infty)$ given by

$$\|x\|_p \coloneqq \Big(\sum_{i=1}^d |x_i|^p\Big)^{\frac{1}{p}}, \qquad \forall x \in \mathbb{R}^d.$$

As we have already said, our main focus in this chapter is to define and gain intuition about duality relations and concepts in convex analysis. One very interesting dual object related to norms are the *dual norms*. If $\|\cdot\|$ is a norm on $\mathbb{E}$, the **dual norm** of $\|\cdot\|$ is the norm[4] $\|\cdot\|_*$ on $\mathbb{E}$ defined by

$$\|x^*\|_* \coloneqq \max\{\, \langle x^*, x\rangle : x \in \mathbb{E}, \|x\| \leq 1\}, \qquad \forall x^* \in \mathbb{E}.$$

At first sight, the definition of the dual norm $\|\cdot\|_*$ of a norm $\|\cdot\|$ on $\mathbb{E}$ hardly has any intuitive meaning. There are two ways of looking at $\|\cdot\|_*$ which may be helpful in gaining some intuition. One way is to see dual norms as special cases of support functions. Set $\mathbb{B}_{\|\cdot\|} \coloneqq \{\, x \in \mathbb{E} : \|x\| \leq 1\}$, that is, $\mathbb{B}_{\|\cdot\|}$ is the *unit ball* (w.r.t. the norm $\|\cdot\|$). Then, by the definition of conjugate function we have

$$\|x^*\|_* = \delta^*(x^* \mid \mathbb{B}_{\|\cdot\|}), \qquad \forall x^* \in \mathbb{E}.$$

This way of seeing the dual norm as the support function of the unit ball of the original norm may be useful in some cases, but it is arguably yet too abstract. A more concrete way of seeing norms is as norms in the space of linear functionals on $\mathbb{E}$, that is, linear functions from $\mathbb{E}$ to $\mathbb{R}$. Let $x^* \in \mathbb{E}$ and fix a norm $\|\cdot\|$ on $\mathbb{E}$. The point x^* can be seen as representing the linear functional T_{x^*} given by $T_{x^*}(x) \coloneqq \langle x^*, x\rangle$ for every $x \in \mathbb{E}$. Given that we have the norm $\|\cdot\|$ to measure the sizes of elements in $\mathbb{E}$, it would be interesting to have a related way to measure the "size" of T_{x^*}. Intuitively, we want a measure such that the bigger the norm of $T_{x^*}(x)$ when compared to the norm of $x \in \mathbb{E}$, the bigger is the size of T_{x^*}. That is, we want to measure how much T_{x^*} stretches the vectors when we measure lengths with $\|\cdot\|$. Of course, for distinct non-zero vectors $x, y \in \mathbb{E}$ the ratios $T_{x^*}(x)/\|x\|$ and $T_{x^*}(y)/\|y\|$ may differ. Thus, we measure a linear functional by the direction which it stretches the most, that is,

$$\sup_{x\in\mathbb{E}\setminus\{0\}} \frac{T_{x^*}(x)}{\|x\|} = \sup_{x\in\mathbb{E}\setminus\{0\}} \frac{\langle x^*, x\rangle}{\|x\|} = \sup_{x\in\mathbb{E}\,:\,\|x\|\leq 1} \langle x^*, x\rangle = \|x^*\|_*.$$

Let us look at some properties and interesting special cases of dual norms. One interesting fact that we will use repeatedly during the remainder of the text, usually without reference, is that the ℓ_2-norm on $\mathbb{E}$ is *self-dual*, that is, we have $(\|\cdot\|_2)_* = \|\cdot\|_2$.

Lemma 3.8.1. The dual norm of $\|\cdot\|_2$ on $\mathbb{E}$ is $\|\cdot\|_2$.

Proof. Let $\|\cdot\|_{2,*}$ be the norm dual to $\|\cdot\|_2$. By the Cauchy–Schwarz inequality we have, for any $x^* \in \mathbb{E} \setminus \{0\}$,

$$\|x^*\|_{2,*} = \max\{\, {x^*}^{\mathsf{T}} x : x \in \mathbb{E}, \|x\|_2 \leq 1\} \leq \max\{\, \|x^*\|_2 \|x\|_2 : x \in \mathbb{E}, \|x\|_2 \leq 1\} = \|x^*\|_2,$$

and since the above inequality holds as an equation for $x \coloneqq \|x^*\|_2^{-1} x^*$, we have $\|\cdot\|_{2,*} = \|\cdot\|_2$. □

[4] We skip the proof that the dual norm is indeed a norm for the sake of conciseness.

As expected, we show in the next theorem that the dual norm of a dual norm is the original norm. Maybe more interestingly, the proof follows relatively easily when we use the results about Fenchel conjugates, mainly the fact that the conjugate of the conjugate of a closed function if the function itself. Since norms are continuous, and thus closed, functions, we are guaranteed that the conjugate of the conjugate of (the square of) a norm is the norm itself.

Theorem 3.8.2. Let $\|\cdot\|$ be a norm on $\mathbb{E}$. Then $(\frac{1}{2}\|\cdot\|^2)^* = \frac{1}{2}\|\cdot\|_*^2$. In particular, $\|\cdot\|_{**} = \|\cdot\|$.

Proof. Let $x^* \in \mathbb{E}$. Note that

$$\big(\tfrac{1}{2}\|\cdot\|^2\big)^*(x^*) = \sup_{x\in\mathbb{E}}\big(\langle x^*, x\rangle - \tfrac{1}{2}\|x\|^2\big) \leq \sup_{x\in\mathbb{E}}\big(\|x^*\|_*\|x\| - \tfrac{1}{2}\|x\|^2\big) = \tfrac{1}{2}\|x^*\|_*^2, \tag{3.8}$$

where in the last inequality we used that $\sup_{\alpha\in\mathbb{R}}(\alpha\|x^*\|_* - \alpha^2/2)$ is attained by $\|x^*\|_*$. Let $\bar{y} \in \mathbb{E}$ attain $\max\{\, \langle x^*, x\rangle : x \in \mathbb{E}, \|x\| \leq 1\} = \|x^*\|_*$, and set $\bar{x} := \|x^*\|_*\bar{y}$. We have

$$\langle x^*, \bar{x}\rangle - \tfrac{1}{2}\|\bar{x}\|^2 = \|x^*\|_*\langle x^*, \bar{y}\rangle - \tfrac{1}{2}\|x^*\|_*^2\|\bar{y}\|^2 = \|x^*\|_*^2 - \tfrac{1}{2}\|x^*\|_*^2\|\bar{y}\|^2 = \tfrac{1}{2}\|x^*\|_*^2.$$

Hence, (3.8) holds as an equation. Finally, since $\frac{1}{2}\|\cdot\|^2$ is continuous (and, thus, closed), by what we have just proved and by Theorem 3.4.2 we have

$$\tfrac{1}{2}\|\cdot\|_{**}^2 = (\tfrac{1}{2}\|\cdot\|_*^2)^* = (\tfrac{1}{2}\|\cdot\|^2)^{**} = \tfrac{1}{2}\|\cdot\|^2,$$

that is, $\|\cdot\|_{**} = \|\cdot\|$. □

One result that we will use extensively in this text is the fact that ℓ_1 and ℓ_∞-norms are dual to each other.

Lemma 3.8.3. The dual norm of $\|\cdot\|_1$ on $\mathbb{R}^d$ is $\|\cdot\|_\infty$.

Proof. Let $x^* \in \mathbb{R}^d$ and let $x \in \mathbb{R}^d$ such that $\|x\|_1 \leq 1$. We have

$$(x^*)^\mathsf{T} x = \sum_{i=1}^d x_i^* x_i \leq \sum_{i=1}^d |x_i^*||x_i| \leq \|x^*\|_\infty \sum_{i=1}^d |x_i| \leq \|x^*\|_\infty.$$

Since the above chain of inequalities holds as an equation for $x := |x_{i^*}^*|e_{i^*}$, where $i^* \in \arg\max_{i\in[d]}|x_i^*|$, we are done. □

When we start to look at regret bounds for OCO algorithms, most of them will depend on the norms of the subgradients of the functions used by the enemy. Thus, one may already imagine that if the player is able to have any control on the norm which measures the sizes of the subgradients, she could pick a norm under which the enemy functions' subgradients have small norm. Still, to make such a choice, the players needs to have some information on the functions the enemy is allowed to pick. In optimization problems one usually assumes that the functions which one has to handle are Lipschitz continuous, that is, the functions cannot change too much between points which are close to each other (w.r.t. to some fixed norm). For differentiable functions, Lipschitz continuity means that the derivative in any direction is bounded by a constant. Interestingly, Lipschitz continuity and the (dual) norms of the subgradients of the function are deeply connected. Before proving this result, let us define Lipschitz continuity.

Let $\rho > 0$. A function $f\colon \mathbb{E} \to (-\infty, +\infty]$ is **ρ-Lipschitz continuous** on a set $X \subseteq \operatorname{dom} f$ w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$ if

$$|f(x) - f(y)| \leq \rho\|x - y\|, \qquad \forall x, y \in X,$$

and when X is not explicitly stated, assume $X = \operatorname{dom} f$.

Theorem 3.8.4 (Based on [67, Lemma 2.6])**.** Let $X \subseteq \mathbb{E}$ be a convex set with nonempty interior, and let $f \colon \mathbb{E} \to (-\infty, +\infty]$ be a proper closed convex function which is ρ-Lipschitz continuous on X w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$. Then, for every $x \in X$ there is $g \in \partial f(x)$ such that $\|g\|_* \leq \rho$. Additionally, for every $x \in \operatorname{int} X$ we have $\partial f(x) \subseteq \{\, g \in \mathbb{E} : \|g\|_* \leq \rho \}$.

Proof. First, let us show that

$$\varnothing \neq \partial f(\mathring{x}) \subseteq \{\, y \in \mathbb{E} : \|y\|_* \leq \rho \}, \qquad \forall \mathring{x} \in \operatorname{int} X. \tag{3.9}$$

Let $\mathring{x} \in \operatorname{int} X$ and let $u \in \partial f(\mathring{x})$, which exists by Theorem 3.5.1 since $X \subseteq \operatorname{dom} f$ and, thus, $\operatorname{int} X \subseteq \operatorname{int}(\operatorname{dom} f)$. Moreover, since the set $\mathbb{B} \coloneqq \{\, v \in \mathbb{E} : \|v\| \leq 1 \}$ is compact[5], $\sup_{v \in \mathbb{B}} \langle v, u \rangle = \|u\|_*$ is attained. Let

$$y \in \mathring{x} + \operatorname*{arg\,max}_{v \in \mathbb{B}} \langle u, v \rangle. \tag{3.10}$$

Additionally, for every $\lambda \in [0,1]$ define $z_\lambda \coloneqq \mathring{x} + \lambda(y - \mathring{x})$. Since $\mathring{x} \in \operatorname{int} X$, there is $\varepsilon > 0$ such that $z_\varepsilon \in X$. Therefore,

$$\varepsilon \|u\|_* \overset{(3.10)}{=} \varepsilon \langle u, y - \mathring{x} \rangle = \langle u, z_\varepsilon - \mathring{x} \rangle \leq f(z_\varepsilon) - f(\mathring{x}) \leq \rho \|z_\varepsilon - \mathring{x}\| = \rho\varepsilon \|y - \mathring{x}\| \overset{(3.10)}{\leq} \rho\varepsilon,$$

where in the first inequality we just used the subgradient inequality. Hence, $\|u\|_* \leq \rho$. This completes the proof of (3.9).

Let $\bar{x} \in X$, let $\mathring{x} \in \operatorname{int} X$, and define

$$x_k \coloneqq \mathring{x} + \left(1 - \frac{1}{k+1}\right)(\bar{x} - \mathring{x}), \qquad \forall k \in \mathbb{N}.$$

By Theorem 3.2.1, we have $x_k \in \operatorname{int} X$ for every $k \in \mathbb{N}$. Thus, by (3.9), for every $k \in \mathbb{N}$ there is $u_k \in \partial f(x_k)$ with $\|u_k\|_* \leq \rho$. That is, $\{u_k\}_{k \in \mathbb{N}}$ is a bounded sequence and therefore it has a convergent subsequence. Namely, there is an increasing injection $\pi \colon \mathbb{N} \to \mathbb{N}$ such that $\lim_{k \to \infty} u_{\pi(k)} = \bar{u}$ for some $\bar{u} \in \mathbb{E}$ with $\|\bar{u}\|_* \leq \rho$. Moreover, since f is closed, by Theorem 3.2.6 we have

$$\lim_{k \to \infty} f(x_{\pi(k)}) = f(\bar{x}). \tag{3.11}$$

Finally, by the subgradient inequality we have, for every $k \in \mathbb{N}$ and $z \in \mathbb{N}$,

$$f(z) \geq f(x_{\pi(k)}) + \langle u_{\pi(k)}, z - x_{\pi(k)} \rangle.$$

Taking the limit for k tending to $+\infty$ in the above inequality together with (3.11) (and since the inner product= is a continuous function) yields

$$f(z) \geq f(\bar{x}) + \langle \bar{u}, z - \bar{x} \rangle, \qquad \forall z \in \mathbb{E},$$

that is, $\bar{u} \in \partial f(\bar{x})$. □

On Chapter 6, we will look at (semi-)norms on $\mathbb{R}^d$ which have a special form: they are the ℓ_2-norm skewed by a positive semi-definite matrix $A \in \mathbb{S}^d_+$. Formally, for every $A \in \mathbb{S}^d_+$, define the **(semi-)norm induced by** A by

$$\|x\|_A \coloneqq \sqrt{x^\mathsf{T} A x}, \qquad \forall x \in \mathbb{R}^d.$$

The next lemma shows us that the functions that we have just defined are indeed semi-norms. Moreover, it shows that in the case of positive definite matrices, the above functions are indeed norms and that the dual norm of a norm induced by $A \in \mathbb{S}^d_{++}$ is the norm induced by the inverse matrix A^{-1}.

[5] Recall that any norm in $\mathbb{E}$ is a continuous function.

Lemma 3.8.5. Let $A \in \mathbb{S}^d_+$. Then $\|\cdot\|_A$ is a semi-norm, and if $A \succ 0$, then $\|\cdot\|_A$ is actually a norm whose dual norm is $\|\cdot\|_{A^{-1}}$.

Proof. Since $A \succeq 0$, we have that by Proposition 1.1.4 $A^{1/2} \in \mathbb{S}^d_+$ exists and is unique. Thus, $\|v\|_A = \|A^{1/2}v\|_2$ for any $v \in \mathbb{R}^d$. Since $v \in \mathbb{R}^d \mapsto A^{1/2}v$ is a linear function, it is clear that $v \in \mathbb{R}^d \mapsto \|A^{1/2}v\|_2$ is non-negative everywhere on $\mathbb{R}^d$ and that it satisfies properties (iii) and (ii) from the definition of norm. Suppose $A \succ 0$. Then $A^{1/2}$ is invertible, and for any $v \in \mathbb{R}^d$ we have $A^{1/2}v = 0$ if and only if $v = A^{-1/2}0 = 0$. Thus, in this case, $\|\cdot\|_A$ is a norm on $\mathbb{R}^d$. Finally, by Proposition 1.1.4 we have $(A^{-1})^{1/2} = A^{-1/2} = (A^{1/2})^{-1}$. With this, note that for every $x \in \mathbb{R}^d$ there is $y \coloneqq A^{1/2}x \in \mathbb{R}^d$ such that $x = A^{-1/2}y$. Thus, $\mathbb{R}^d = \{\, A^{-1/2}y : y \in \mathbb{R}^d\}$. Therefore, by Theorem 3.8.2 and since the ℓ_2-norm is dual to itself we have, for any $x^* \in \mathbb{R}^d$,

$$\begin{aligned}
(\tfrac{1}{2}\|\cdot\|_A^2)^*(x^*) &= \sup_{x \in \mathbb{R}^d} ((x^*)^\mathsf{T} x - \tfrac{1}{2}\|x\|_A^2) = \sup_{y \in \mathbb{R}^d} \Big((x^*)^\mathsf{T} A^{-1/2}y - \tfrac{1}{2}\|A^{-1/2}y\|_A^2\Big) \\
&= \sup_{y \in \mathbb{R}^d} \Big((A^{-1/2}x^*)^\mathsf{T} y - \tfrac{1}{2}y^\mathsf{T}(A^{-1/2}AA^{-1/2})y\Big) = \sup_{y \in \mathbb{R}^d} \Big((A^{-1/2}x^*)^\mathsf{T} y - \tfrac{1}{2}y^\mathsf{T} y\Big) \\
&= \sup_{y \in \mathbb{R}^d} \Big(\Big(A^{-1/2}x^*\Big)^\mathsf{T} y - \tfrac{1}{2}\|y\|_2^2\Big) \overset{\text{Thm 3.8.2}}{=} \|A^{-1/2}x^*\|_2^2 = \|x^*\|_{A^{-1}}^2.
\end{aligned}$$

Thus, by Theorem 3.8.2 we conclude that the norm dual to $\|\cdot\|_A$ is $\|\cdot\|_{A^{-1}}$. □

Finally, at some points of the text we shall need some norms for the space of real square matrices. One of the better-known norms for matrices are the *operator norms*, which are based on norms for $\mathbb{R}^d$. In this text we shall restrict our attention only to the operator norm induced by the ℓ_2-norm. Formally, the **operator norm** (induced by the ℓ_2-norm) of $A \in \mathbb{R}^{d \times d}$ is

$$\|A\|_2 \coloneqq \max\{\, \|Ax\|_2 : x \in \mathbb{R}^d, \|x\|_2 \leq 1\}.$$

The next lemma shows useful connections between the operator norm of a matrix and its eigenvalues. We skip its proof for the sake of conciseness.

Lemma 3.8.6 ([39, Example 5.6.6])**.** If $A \in \mathbb{S}^d$, then $\|A\|_2 = \max\{|\lambda_1^\uparrow(A)|, |\lambda_d^\uparrow(A)|\}$. In particular, if $A \succeq 0$, then $\|A\|_2 \leq \operatorname{Tr}(A)$.

3.9 Strong Convexity

In some situations, we need convex functions whose graphs have at least some kind of "curvature". In other words, we need convex functions $R \colon \mathbb{E} \to (-\infty, +\infty]$ such that, for any two points $x, y \in \mathbb{E}$, the line segment between $R(x)$ and $R(y)$ lies strictly above the graph of the functions (except, of course, for the points $R(x)$ and $R(y)$). In this section we study two classes of convex functions which satisfy this condition: strictly and strongly convex function. Even though the focus of this section is strong convexity, looking at strict convexity first helps to build intuition.

A convex function $R \colon \mathbb{E} \to (-\infty, +\infty]$ is **strictly convex** on a convex set $X \subseteq \mathbb{E}$ if, for any distinct $x, y \in X$ and any $\lambda \in (0,1)$, we have

$$R(\lambda x + (1-\lambda)y) < \lambda R(x) + (1-\lambda)R(y). \tag{3.12}$$

It is immediate from the definition that the sum of a strictly convex and a convex function is strictly convex and that strict convexity is preserved under multiplication of positive scalars. The above inequality basically translates the intuition from the previous paragraph: the relative interior of the segment between two points of the graph of a strictly convex function R lies strictly above the graph of R.

As an example of a non-strictly convex function, consider the affine function $x \in \mathbb{R}^d \mapsto \langle a, x\rangle + \beta$ where $a \in \mathbb{R}^d$ and $\beta \in \mathbb{R}$. Thus function clearly satisfies (3.12) as an equation everywhere. Thus, affine functions are not strictly convex. Indeed, the graphs of affine functions have exactly the form that we want graphs of strict convex to avoid: they are flat everywhere.

To look at some concrete examples of strictly convex functions, we will use the following lemma[6] (without proving it). It gives a criterion to detect convexity through a function's hessian similar to the convexity criterion from Lemma 3.1.1. Still, one should note that Lemma 3.1.1 gives a characterization of convexity, while the following lemma only gives a sufficient condition for strict convexity.

Lemma 3.9.1 ([16, Proposition 1.2.6])**.** *Let $X \subseteq \mathbb{R}^d$ be a convex set and let $R\colon \mathbb{R}^d \to (-\infty, +\infty]$ be twice continuously-differentiable on $\mathbb{R}^d$. If $\nabla^2 R(x) \succ 0$ for every $x \in X$, then R is strictly convex on X.*

The above lemma allows us to quickly detect strict convexity of many functions. A classic 1-dimensional example of strictly convex function is the exponential function given by $\alpha \in \mathbb{R} \mapsto e^{\alpha}$. Its second derivative at $\alpha \in \mathbb{R}$ is e^{α} and, by Lemma 3.9.1, we conclude that the exponential function is strictly convex. Another interesting example is the squared ℓ_2-norm. Namely, define $R(x) \coloneqq \frac{1}{2}\|x\|_2^2 = \frac{1}{2}\langle x, x\rangle$ for every $x \in \mathbb{R}^d$ (the $\frac{1}{2}$ factor is not necessary but it is a nice normalization factor, as we are going to see). In this case we have, for every $x \in \mathbb{R}^d$ we have

$$\nabla R(x) = x \qquad \text{and} \qquad \nabla^2 R(x) = I,$$

where one should recall from Section 1.1.3 that I denotes the identity matrix with fitting dimensions. By Lemma 3.9.1 we conclude that the squared ℓ_2-norm is strictly convex on $\mathbb{R}^d$.

Hopefully the above examples helped the reader gain some intuition about strictly convex functions. Although we have looked at some examples, we have not yet seen any properties of strictly convex functions which may be of use in optimization. The following lemma shows the main property of strictly convex functions that we shall use: if a minimizer of a strictly convex function exists, it is unique.

Lemma 3.9.2. *Let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a proper strictly convex function on a convex set $X \subseteq \mathbb{E}$. Then $\inf_{x \in X} R(x)$ is attained by at most one point.*

Proof. Suppose there are distinct $x, y \in \mathbb{E}$ such that both x and y attain $\inf_{x\in X} R(x)$. Then $\frac{1}{2}x + \frac{1}{2}y \in X$ and, by the strict convexity of R,

$$R(\tfrac{1}{2}x + \tfrac{1}{2}y) < \tfrac{1}{2}R(x) + \tfrac{1}{2}R(y) = \inf_{x \in X} R(x),$$

a contradiction. □

As discussed earlier, the graph of a strictly convex function $R\colon \mathbb{E} \to (-\infty, +\infty]$ is curved, that is, a line segment joining any two distinct points of its graph lies strictly above the graph. However, the definition of strict convexity does not state *how* curved R is. That is, for any distinct $x, y \in \mathbb{E}$ and $\lambda \in (0, 1)$, we know that

$$\lambda R(x) + (1-\lambda)R(y) - R(\lambda x + (1-\lambda)y) \tag{3.13}$$

[6] Although the lemma states the result only for $\mathbb{R}^d$, it holds for an arbitrary euclidean space with minor changes in terminology (such as positive-definiteness). We state it for $\mathbb{R}^d$ only for the sake of simplicity.

is positive, but we do not know how far away from zero it is, no matter the distance between x and y. For example, the following lemma shows that in the case where R is the squared ℓ_2-norm on $\mathbb{E}$, the value of (3.13) is fully determined by the distance (w.r.t. the ℓ_2-norm) between x and y besides, of course, the value of $\lambda \in (0,1)$.

Lemma 3.9.3. Let $x, y \in \mathbb{E}$ and $\lambda \in [0,1]$. Then,

$$\lambda\|x\|_2^2 + (1-\lambda)\|y\|_2^2 - \|\lambda x + (1-\lambda)y\|_2^2 = \lambda(1-\lambda)\|x-y\|_2^2$$

Proof. We have

$$\begin{aligned}
&\lambda\|x\|_2^2 + (1-\lambda)\|y\|_2^2 - \|\lambda x + (1-\lambda)y\|_2^2\\
&= \lambda\|x\|_2^2 + (1-\lambda)\|y\|_2^2 - \lambda^2\|x\|_2^2 - (1-\lambda)^2\|y\|_2^2 - 2\lambda(1-\lambda)\langle x, y\rangle\\
&= \lambda(1-\lambda)\|x\|_2^2 + \lambda(1-\lambda)\|y\|_2^2 - 2\lambda(1-\lambda)\langle x, y\rangle\\
&= \lambda(1-\lambda)(\|x\|_2^2 + \|y\|_2^2 - 2\langle x, y\rangle)\\
&= \lambda(1-\lambda)\|x-y\|^2.
\end{aligned}$$

□

With the above lemma, we know that the value of (3.13) for the squared ℓ_2-norm is invariant to translations, that is, the value of (3.13) for some$x, y \in \mathbb{E}$ is the same for the points $x+d$ and $y+d$ for any direction $d \in \mathbb{E}$. This is not the case for the exponential function, for example.

Lemma 3.9.4. Let $\lambda \in (0,1)$ and let $\delta \in \mathbb{R}$. Then

$$\lim_{\alpha \to -\infty} \lambda e^{\alpha} + (1-\lambda)e^{(\alpha+\delta)} - e^{\lambda\alpha + (1-\lambda)(\alpha+\delta)} = 0 \tag{3.14}$$

Proof. Let $\alpha \in \mathbb{R}$. Since e^{α} is the gradient of the exponential function at α, it is also a subgradient by Theorem 3.5.5. Thus, by the subgradient inequality we have

$$e^{\lambda\alpha + (1-\lambda)(\alpha+\delta)} \geq e^{\alpha} + e^{\alpha}(\lambda\alpha + (1-\lambda)(\alpha+\delta) - \alpha) = e^{\alpha} + e^{\alpha}(1-\lambda)\delta = \lambda e^{\alpha} + (1-\lambda)e^{\alpha}(1+\delta).$$

Therefore,

$$\lambda e^{\alpha} + (1-\lambda)e^{\alpha+\delta} - e^{\lambda\alpha+(1-\lambda)(\alpha+\delta)} \leq (1-\lambda)(e^{\alpha+\delta} - e^{\alpha}(1+\delta)) = (1-\lambda)e^{\alpha}(e^{\delta} - 1 - \delta).$$

Since the limit of the right-hand side for α tending to $-\infty$ is 0 and since the limit in (3.14) is nonnegative due to the convexity of the exponential function, we are done. □

That is, for scalars $\alpha, \beta \in \mathbb{R}$, the bigger the value of $\delta \in \mathbb{R}_+$, the better the line segment between the points $e^{\alpha-\delta}$ and $e^{\beta-\delta}$ approximates the graph of the exponential in the interval $[\alpha-\delta, \beta-\delta]$. In other words, the further one goes in the direction of $-\infty$ in the real line, the flatter the graph of the exponential function is.

As one may expect, there are cases which we need a function which has a certain curvature throughout the space we are considering, such as in the case of the squared ℓ_2-norm in $\mathbb{E}$. Additionally, how much curved the function is should matter in these cases. This idea of more "uniformly curved" functions is translated into the definition of *strongly convex functions*.

Let $\sigma > 0$. A function $R\colon \mathbb{E} \to (-\infty, +\infty]$ is **σ-strongly convex** on a convex set $X \subseteq \mathbb{E}$ (with respect to a norm $\|\cdot\|$ on $\mathbb{E}$) if for every $x, y \in X$ and $\lambda \in [0,1]$ we have

$$R(\lambda x + (1-\lambda)y) \leq \lambda R(x) + (1-\lambda)R(y) - \lambda(1-\lambda)\frac{\sigma}{2}\|x-y\|^2.$$

If the set X is not explicitly stated, consider $X \coloneqq \mathbb{E}$. Moreover, we may omit σ and/or the norm $\|\cdot\|$ whenever their knowledge is not necessary.

Let $R \colon \mathbb{E} \to (-\infty, +\infty]$ be a σ-strongly convex function on a convex set $X \subseteq \mathbb{E}$ w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$. Then, in particular, R is strictly convex on X. Still, the definition of strong convexity requires, in some sense, for the function to be curved at least by a factor of σ throughout X. Formally, for any $x, y \in X$ and $\lambda \in (0,1)$, the difference in (3.13) in the case where R is σ-strongly convex w.r.t. $\|\cdot\|$ is at least $\frac{\sigma}{2}\lambda(1-\lambda)\|x-y\|^2$.

There is another property of σ-strongly convex functions w.r.t. a norm $\|\cdot\|$ which may help the reader to gain some intuition: it is a function which has, at each point, a quadratic (on $\|\cdot\|$) lower bound. For simplicity's sake, let us first look at the case where $\|\cdot\|$ is the euclidean norm on $\mathbb{E}$. The next lemma shows a characterization of strong convexity for this latter case which helps us show the existence of such quadratic lower bounds

Lemma 3.9.5. Let $R \colon \mathbb{E} \to (-\infty, +\infty]$ be proper and let $\sigma \in \mathbb{R}_{++}$. Then R is σ-strongly convex w.r.t. $\|\cdot\|_2$ if and only if $R - \frac{\sigma}{2}\|\cdot\|_2^2$ is convex.

Proof. For any $x, y \in \mathbb{E}$ and $\lambda \in (0,1)$ we have

$$\begin{aligned}
&R(\lambda x + (1-\lambda)y) - \frac{\sigma}{2}\|\lambda x + (1-\lambda)y\|_2^2 \leq \lambda\Big(R(x) - \frac{\sigma}{2}\|x\|_2^2\Big) + (1-\lambda)\Big(R(y) - \frac{\sigma}{2}\|y\|_2^2\Big)\\
\iff\ &R(\lambda x + (1-\lambda y)) \leq \lambda R(x) + (1-\lambda)R(y) - \frac{\sigma}{2}\big(\lambda\|x\|_2^2 + (1-\lambda)\|y\|_2^2 - \|\lambda x + (1-\lambda)y\|_2^2\big)\\
\overset{\text{Le. 3.9.3}}{\iff} &R(\lambda x + (1-\lambda y)) \leq \lambda R(x) + (1-\lambda)R(y) - \frac{\sigma}{2}\lambda(1-\lambda)\|x-y\|_2^2. \qquad \square
\end{aligned}$$

It directly follows from Lemma 3.9.5 that $\frac{1}{2}\|\cdot\|_2^2$ is 1-strongly convex on $\mathbb{E}$ w.r.t. $\|\cdot\|_2$. More importantly, the above lemma allows us to derive a stronger version of the subgradient inequality for strongly convex functions w.r.t. the ℓ_2-norm. Namely, let $\sigma \in \mathbb{R}_{++}$ and let $R \colon \mathbb{E} \to (-\infty, +\infty]$ be a proper σ-strongly convex w.r.t. $\|\cdot\|_2$ function. Moreover, let $x \in \mathbb{E}$ be such that there is $g \in \partial R(x)$. By Lemma 3.9.5 the function $R' \coloneqq R - \frac{\sigma}{2}\|\cdot\|_2^2$ is convex, and by Theorems 3.5.4 and 3.5.5 we have the subgradient $g - \sigma x \in \partial R'(x)$. Thus, by the subgradient inequality, for every $z \in \mathbb{E}$ we have

$$\begin{aligned}
&R'(z) \geq R'(x) + \langle g - \sigma x, z - x\rangle\\
\iff\ &R(z) - \tfrac{\sigma}{2}\|z\|_2^2 \geq R(x) - \tfrac{\sigma}{2}\|x\|_2^2 + \langle g, z - x\rangle - \sigma\langle x, z\rangle + \sigma\|x\|_2^2\\
\iff\ &R(z) \geq R(x) + \langle g, z - x\rangle + \tfrac{\sigma}{2}\|z\|_2^2 - \sigma\langle x, z\rangle + \tfrac{\sigma}{2}\|x\|_2^2\\
\iff\ &R(z) \geq R(x) + \langle g, z - x\rangle + \tfrac{\sigma}{2}\|z - x\|_2^2.
\end{aligned}$$

That is, at each point $x \in \mathbb{E}$ on which R is subdifferentiable there is a quadratic function which lower bounds R and meets it at least at x. On $\mathbb{R}^2$, such lower bounds as simply parabolas, and in this simpler case one is able to see how strong convexity implies that the function is curved: having parabolas as lower bounds makes it impossible for the function to be linear (or “flat”) anywhere.

Although we are looking only at the ℓ_2-norm case, it happens that the above “strong version” of the subgradient inequality holds for strongly convex function w.r.t. arbitrary norms on $\mathbb{E}$. Even more surprisingly, such strong subgradient inequality characterizes strongly convex functions. That is, a function R is strongly convex if and only if such strong subgradient inequality holds on every point where R is subdifferentiable. To prove this claim, we will first show an interesting fact: to check if a *closed* convex function is strongly convex or not, it suffices to look at the relative interior of its domain.

Lemma 3.9.6. Let $R \colon \mathbb{E} \to (-\infty, +\infty]$ be a proper closed convex function and let $X \subseteq \mathbb{E}$ be a convex set such that $X \cap \operatorname{ri}(\operatorname{dom} R)$ is nonempty. If R is σ-strongly convex w.r.t. a norm $\|\cdot\|$ on $X \cap \operatorname{ri}(\operatorname{dom} R)$, then R is σ-strongly convex on X w.r.t. $\|\cdot\|$.

Proof. Suppose R is σ-strongly convex w.r.t. a norm $\|\cdot\|$ on $X \cap \operatorname{ri}(\operatorname{dom} R)$. We need to show that, for every $x, y \in X$,

$$R(\lambda x + (1-\lambda)y) \leq \lambda R(x) + (1-\lambda)R(y) - \lambda(1-\lambda)\frac{\sigma}{2}\|x-y\|^2, \qquad \forall \lambda \in (0,1). \tag{3.15}$$

By assumption, we know that the above inequality holds for $x, y \in X \cap \operatorname{ri}(\operatorname{dom} R)$. Let us show that

> If (3.15) holds for every $x \in X \cap \operatorname{ri}(\operatorname{dom} R)$ and for $y = \bar{y}$ for some $\bar{y} \in X \cap (\operatorname{dom} R)$, then (3.15) actually holds for every $x \in X$ and for $y = \bar{y}$. (3.16)

Let $\bar{y} \in X \cap (\operatorname{dom} R)$ be as in the above claim. Since (3.15) holds trivially for $x \in X \setminus (\operatorname{dom} R)$, we only need to show that it holds for every $x \in X \cap (\operatorname{dom} R)$. Let $\bar{x} \in X \cap (\operatorname{dom} R)$, let $\mathring{x} \in X \cap \operatorname{ri}(\operatorname{dom} R)$, let $\mu \in [0,1)$, and define $x_\mu := (1-\mu)\mathring{x} + \mu\bar{x}$. By Theorem 3.2.1, $x_\mu \in \operatorname{ri}(\operatorname{dom} R)$ and $x_\mu \in X$ since X is convex. Let $\lambda \in (0,1)$. By setting $\mathring{z}_\lambda := \lambda\mathring{x} + (1-\lambda)\bar{y}$ and $\bar{z}_\lambda := \lambda\bar{x} + (1-\lambda)\bar{y}$ we have by assumption (since $x_\mu \in X \cap \operatorname{ri}(\operatorname{dom} R)$)

$$\begin{aligned} R((1-\mu)\mathring{z}_\lambda + \mu\bar{z}_\lambda) &= R((1-\mu)(\lambda\mathring{x} + (1-\lambda)\bar{y}) + \mu(\lambda\bar{x} + (1-\lambda)\bar{y})) \\ &= R(\lambda x_\mu + (1-\lambda)\bar{y}) \\ &\leq \lambda R(x_\mu) + (1-\lambda)R(\bar{y}) - \lambda(1-\lambda)\frac{\sigma}{2}\|x_\mu - \bar{y}\|^2. \end{aligned} \tag{3.17}$$

By Theorem 3.2.1, we have $\mathring{z}_\lambda = \lambda\mathring{x} + (1-\lambda)\bar{y} \in \operatorname{ri}(\operatorname{dom} R)$ since $\lambda > 0$. Thus, since R is closed, by Theorem 3.2.6 we have

$$\lim_{\mu \uparrow 1} R((1-\mu)\mathring{z}_\lambda + \mu\bar{z}_\lambda) = R(\bar{z}_\lambda) \qquad \text{and} \qquad \lim_{\mu \uparrow 1} R(x_\mu) = R(\bar{x}).$$

Therefore, taking the limit as μ tends to 1 from below on (3.17) yields

$$R(\lambda\bar{x} + (1-\lambda)\bar{y}) = R(\bar{z}_\lambda) \leq \lambda R(\bar{x}) + (1-\lambda)R(\bar{y}) - \lambda(1-\lambda)\frac{\sigma}{2}\|\bar{x} - \bar{y}\|^2.$$

This proves (3.16).

To ease the notation, for every $x, y \in \mathbb{E}$ and $\lambda \in (0,1)$, define $F_\lambda(x,y) := \lambda R(x) + (1-\lambda)R(y) - \lambda(1-\lambda)\frac{\sigma}{2}\|x-y\|^2$. Now, note that

$$\begin{aligned} & R(\lambda x + (1-\lambda)y) \leq F_\lambda(x,y), && \forall \lambda \in (0,1), \forall x, y \in X \cap \operatorname{ri}(\operatorname{dom} R) \\ \overset{(3.16)}{\implies} & R(\lambda x + (1-\lambda)y) \leq F_\lambda(x,y), && \forall \lambda \in (0,1), \forall y \in X \cap \operatorname{ri}(\operatorname{dom} R), \forall x \in X \\ \implies & R(\lambda y + (1-\lambda)x) \leq F_\lambda(y,x), && \forall \lambda \in (0,1), \forall y \in X \cap \operatorname{ri}(\operatorname{dom} R), \forall x \in X \\ \overset{(3.16)}{\implies} & R(\lambda y + (1-\lambda)x) \leq F_\lambda(y,x), && \forall \lambda \in (0,1), \forall x, y \in X. \end{aligned}$$

□

Let us now show that strongly convex functions are characterized by the "strong subgradient inequality" about which we discussed earlier.

Theorem 3.9.7. Let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a proper closed convex function and let $X \subseteq \mathbb{E}$ be a convex set such that $X \cap \operatorname{ri}(\operatorname{dom} R)$ is nonempty. Then R is σ-strongly convex on X w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$ if and only if for every $x \in \operatorname{ri}(\operatorname{dom} R) \cap X$, every $y \in X$, and every $g \in \partial R(x)$ we have

$$R(y) \geq R(x) + \langle g, y - x\rangle + \frac{\sigma}{2}\|y - x\|^2. \tag{3.18}$$

Proof. Suppose R is σ-strongly convex on $X \subseteq \mathbb{E}$ w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$, let $x \in \operatorname{ri}(\operatorname{dom} R) \cap X$, and let $y \in X$. Moreover, let $\lambda \in (0,1)$. By the definition of strong convexity, we have

$$R(x) + \lambda(R(y) - R(x)) - (1-\lambda)\lambda\frac{\sigma}{2}\|x-y\|^2 \geq R(x + \lambda(y-x)).$$

Thus, rearranging the above inequality and using the subgradient inequality for any $u \in \partial R(x)$ (which exists by Theorem 3.5.1),

$$R(y) - R(x) - (1-\lambda)\frac{\sigma}{2}\|x-y\|^2 \geq \frac{R(x+\lambda(y-x)) - R(x)}{\lambda} \geq \frac{\lambda\langle u, y-x\rangle}{\lambda} = \langle u, y-x\rangle.$$

Taking the limit with λ tending to zero on the above expression yields (3.18).

For the converse, let us first show that

$$\text{if (3.18) holds for every } x, y \in \operatorname{ri} X, \text{ then } R \text{ is } \sigma\text{-strongly convex on } \operatorname{ri}(\operatorname{dom} R) \cap X \text{ w.r.t. } \|\cdot\|. \tag{3.19}$$

Suppose that (3.18) holds for every $x, y \in \operatorname{ri} X$, let $\mathring{x}, \mathring{y} \in X \cap \operatorname{ri}(\operatorname{dom} R)$, let $\lambda \in (0,1)$, and define

$$z_\lambda \coloneqq \lambda\mathring{x} + (1-\lambda)\mathring{y}.$$

Since $\operatorname{ri}(\operatorname{dom} R)$ is convex, $z_\lambda \in \operatorname{ri}(\operatorname{dom} R)$. Hence, by Theorem 3.5.1 there is $v_\lambda \in \partial f(z_\lambda)$. By (3.18), we have

$$\langle v_\lambda, \mathring{x} - z_\lambda\rangle \leq R(\mathring{x}) - R(z_\lambda) - \frac{\sigma}{2}\|\mathring{x} - z_\lambda\|^2$$

and

$$\langle v_\lambda, \mathring{y} - z_\lambda\rangle \leq R(\mathring{y}) - R(z_\lambda) - \frac{\sigma}{2}\|\mathring{y} - z_\lambda\|^2.$$

Taking a convex combination of $\langle v_\lambda, \mathring{x} - z_\lambda\rangle$ and $\langle v_\lambda, \mathring{y} - z_\lambda\rangle$ with weights given by λ yields

$$\langle v_\lambda, \lambda(\mathring{x} - z_\lambda) + (1-\lambda)(\mathring{y} - z_\lambda)\rangle = \langle v_\lambda, \lambda\mathring{x} + (1-\lambda)\mathring{y} - z_\lambda\rangle = \langle v_\lambda, z_\lambda - z_\lambda\rangle = 0.$$

Therefore,

$$\begin{aligned}
0 &= \lambda\langle v_\lambda, \mathring{x} - z_\lambda\rangle + (1-\lambda)\langle v_\lambda, \mathring{y} - z_\lambda\rangle \\
&\leq \lambda\Big(R(\mathring{x}) - R(z_\lambda) - \frac{\sigma}{2}\|\mathring{x} - z_\lambda\|^2\Big) + (1-\lambda)\Big(R(\mathring{y}) - R(z_\lambda) - \frac{\sigma}{2}\|\mathring{y} - z_\lambda\|^2\Big) \\
&= \lambda R(\mathring{x}) + (1-\lambda)R(\mathring{y}) - R(z_\lambda) - \frac{\sigma}{2}\big(\lambda\|\mathring{x} - z_\lambda\|^2 + (1-\lambda)\|\mathring{y} - z_\lambda\|^2\big) \\
&= \lambda R(\mathring{x}) + (1-\lambda)R(\mathring{y}) - R(z_\lambda) - \frac{\sigma}{2}\big(\lambda(1-\lambda)^2\|\mathring{x} - \mathring{y}\|^2 + (1-\lambda)\lambda^2\|\mathring{x} - \mathring{y}\|^2\big) \\
&= \lambda R(\mathring{x}) + (1-\lambda)R(\mathring{y}) - R(z_\lambda) - \lambda(1-\lambda)\frac{\sigma}{2}\|\mathring{x} - \mathring{y}\|^2.
\end{aligned}$$

This finishes the proof of (3.19), and by Lemma 3.9.6 we are done. □

The following proposition, which we state without proof, gives another characterization of strongly convex functions and shows that the conjugate of a strongly convex function is very well-behaved: it is finite and differentiable everywhere. On the next section we shall see that this properties on the conjugate of strongly convex functions is part of a dual relation regarding strong convexity with *strong smoothness.*

Proposition 3.9.8 ([60, Section 12H])**.** Let $\sigma > 0$ and $R \colon \mathbb{E} \to (-\infty, +\infty]$ be a proper, closed, and convex function. Moreover, let $\|\cdot\|$ be a norm on $\mathbb{E}$ and let X be a nonempty convex set. Then, the following are equivalent:

(i) R is σ-strongly convex on X w.r.t. $\|\cdot\|$,

(ii) for every $x, y \in X \cap \operatorname{ri}(\operatorname{dom} R)$ and every $u \in \partial R(x)$, we have $R(y) \geq R(x) + \langle u, y-x\rangle + \frac{\sigma}{2}\|x-y\|^2$;

(iii) for every $x, y \in X \cap \operatorname{ri}(\operatorname{dom} R)$, every $u \in \partial R(x)$, and every $v \in \partial R(y)$, we have $\langle u - v, x - y\rangle \geq \sigma\|x-y\|^2$;

Moreover, if R is strongly convex on $\mathbb{E}$, then R^* is finite and differentiable on $\mathbb{E}$.

As an application of the above properties, let us show that the negative entropy on $\mathbb{R}^d$ is 1-strongly convex w.r.t. $\|\cdot\|_1$ on (multiples of) the unit ball of the ℓ_1-norm. Interestingly, the negative entropy is only strictly convex on $\mathbb{R}^d$. Thus, we need to restrict the negative entropy to get strong convexity. First we will show a known inequality regarding the exponential of a scalar and then we shall prove the strong convexity of the negative entropy.

Lemma 3.9.9. For each $\alpha \in \mathbb{R}$,

$$e^{\alpha} \geq 1 + \alpha.$$

Moreover, for every $\beta \in \mathbb{R}_{++}$,

$$\ln \beta \leq \beta - 1.$$

Proof. Let us divide the proof in three cases. For $\alpha \leq -1$ the statement is trivially true. Suppose that $\alpha \geq 0$. By definition,

$$e^{\alpha} = \sum_{i=0}^{\infty} \frac{\alpha^i}{i!}.$$

Since $\alpha \geq 0$, each term of this series is nonnegative. Hence,

$$e^{\alpha} = \sum_{i=0}^{\infty} \frac{\alpha^i}{i!} = 1 + \alpha + \sum_{i=2}^{\infty} \frac{\alpha^i}{i!} \geq 1 + \alpha.$$

Suppose now that $-1 < \alpha < 0$ and define $\beta \coloneqq -\alpha$. Hence, $0 < \beta < 1$ and

$$e^{\alpha} = e^{-\beta} = \sum_{i=0}^{\infty} \frac{(-\beta)^i}{i!}.$$

It is easy to see that the terms of this alternating serie decreases in modulus since $0 < \beta < 1$. Hence,

$$\sum_{i=2}^{\infty} \frac{(-\beta)^i}{i!} \geq 0 \implies e^{-\beta} = 1 - \beta + \sum_{i=2}^{\infty} \frac{(-\beta)^i}{i!} \geq 1 - \beta.$$

Finally, note that due to what we have just proved, $e^{\beta - 1} \geq \beta$ for every $\beta \in \mathbb{R}_{++}$, which implies that $\ln \beta \leq \beta - 1$ holds. □

Lemma 3.9.10 ([68, Lemma 16])**.** Define $R(x) \coloneqq \sum_{i=1}^{d} [x_i > 0] x_i \ln x_i + \delta(x \,|\, \mathbb{R}^d_+)$ for every $x \in \mathbb{R}^d$ and let $\theta \in \mathbb{R}_{++}$. Then R is closed and $(1/\theta)$-strongly convex on $B_\theta \coloneqq \{\, x \in \mathbb{R}^d : \|x\|_1 \leq \theta \,\}$ w.r.t. $\|\cdot\|_1$.

Proof. First, note that $R(x)$ is continuous and differentiable on $\operatorname{int}(\operatorname{dom} R) = \mathbb{R}^d_{++}$. Moreover, by Proposition 3.4.4 R is closed. Let us now show that R is $(1/\theta)$-strongly convex w.r.t. $\|\cdot\|_1$ on B_θ. By Proposition 3.9.8, we just need to show that

$$\langle \nabla R(x) - \nabla R(y), x - y \rangle \geq \tfrac{1}{\theta}\|x - y\|_1^2, \qquad \forall x, y \in B_\theta \cap \mathbb{R}^d_{++}.$$

Let $x, y \in B_\theta \cap \mathbb{R}^d_{++}$, and define $w \in \mathbb{R}^d$ by

$$w_i := (\nabla R(x)_i - \nabla R(y)_i)(x_i - y_i) = (\ln x_i - \ln y_i)(x_i - y_i), \qquad \forall i \in [d].$$

Since the logarithm is an increasing function on $(0, +\infty)$, we have $w \geq 0$ and $w_i = 0$ if and only if $x_i = y_i$ for each $i \in [d]$. Set $\operatorname{supp}(w) := \{\, i \in [d] : w_i \neq 0\} = \{\, i \in [d] : x_i \neq y_i\}$. Then, by the Cauchy-Schwarz inequality and the definition of w,

$$\begin{aligned}
\|x - y\|_1^2 = \Big(\sum_{i=1}^d (x_i - y_i)\Big)^2 &= \Big(\sum_{i \in \operatorname{supp}(w)} (x_i - y_i)\Big)^2 = \Big(\sum_{i \in \operatorname{supp}(w)} \sqrt{w_i}\frac{(x_i - y_i)}{\sqrt{w_i}}\Big)^2 \\
&\leq \Big(\sum_{i \in \operatorname{supp}(w)} w_i\Big)\Big(\sum_{i \in \operatorname{supp}(w)} \frac{(x_i - y_i)^2}{w_i}\Big) \\
&= \langle \nabla R(x) - \nabla R(y), x - y\rangle \Big(\sum_{i \in \operatorname{supp}(w)} \frac{x_i - y_i}{\ln x_i - \ln y_i}\Big).
\end{aligned}$$

Thus, it only remains to prove

$$\Big(\sum_{i \in \operatorname{supp}(w)} \frac{x_i - y_i}{\ln x_i - \ln y_i}\Big) \leq \theta.$$

In order to prove the above inequality, let us first show that

$$\frac{x_i - y_i}{\ln x_i - \ln y_i} \leq \frac{x_i + y_i}{2}, \qquad \forall i \in \operatorname{supp}(w). \tag{3.20}$$

Let $i \in \operatorname{supp}(w)$ and, without loss of generality, suppose $x_i > y_i$. Define

$$\phi(\alpha) := [\alpha > 0](2(\alpha - y_i) - (\alpha + y_i)(\ln \alpha - \ln y_i)), \qquad \forall \alpha \in \mathbb{R}.$$

Note that $\phi(y_i) = 0$. Moreover, for every $\alpha \in \mathbb{R}_{++}$ we have

$$\phi'(\alpha) = 2 - (\ln \alpha - \ln y_i) - \frac{\alpha + y_i}{\alpha} = 1 + \ln \frac{y_i}{\alpha} - \frac{y_i}{\alpha} \leq \frac{y_i}{\alpha} - \frac{y_i}{\alpha} = 0,$$

where in the last inequality we used $\ln \beta \leq \beta - 1$ for any $\beta \in \mathbb{R}_{++}$ (see Lemma 3.9.9). Therefore, ϕ is a non-increasing function, and since $\phi(y_i) \leq 0$ we have $\phi(\alpha) \leq 0$ for every $\alpha \in [y_i, +\infty)$. Since $y_i < x_i$ we conclude that $\phi(x_i) \leq 0$, which finally implies (3.20). Therefore,

$$\Big(\sum_{i \in \operatorname{supp}(w)} \frac{x_i - y_i}{\ln x_i - \ln y_i}\Big) \leq \Big(\sum_{i \in \operatorname{supp}(w)} \frac{x_i + y_i}{2}\Big) \leq \frac{\|x\|_1 + \|y\|_1}{2} \leq \theta. \qquad \square$$

The focus of this text is ultimately optimization of convex functions. As we have seen on Section 3.6, convex functions are easier to optimize since (among other facts) their local minima on convex sets are also global minima. At the beginning of this section we have seen that when we restrict our attention to strictly convex function, we have the additional property that, if the infimum over a convex set is attained, it is actually attained by a unique point (this uniqueness will be important in Section 3.11). One natural question is: when we further restrict our attention to strongly convex function, do we have additional properties regarding their infima over convex sets? Fortunately, the answer is yes and the additional property we earn will be fundamental in many algorithms seen in the next chapters.

We will show in the next results that the infimum of a proper closed strongly convex function R over a closed convex set $X \subseteq \mathbb{E}$ is guaranteed to be attained. The intuition behind this results is that the level sets of R are bounded and, thus, compact. The boundedness is an implication of the "curvature" of the graph of these functions, as we shall see soon. Then, since the intersection of non-empty nested compact sets is non-empty (Cantor's Intersection Theorem), the intersection of all level-sets of the function is non-empty, that is, there is a point which attains $\inf_{x \in \mathbb{E}} R(x)$. Finally, since $R + \delta(\cdot \,|\, X)$ is also a closed (usually proper) strongly convex function, then $\inf_{x \in X} R(x)$ is also attained. First, let us show that the level sets of R are compact. To do so, we will use the following theorem, which we state without proof.

Theorem 3.9.11 ([59, Thm. 8.4])**.** Let $C \subseteq \mathbb{E}$ be a nonempty closed convex set. Then C is unbounded if and only if there is $d \in \mathbb{E} \setminus \{0\}$ such that $C + \mu d \subseteq C$ for every $\mu \in \mathbb{R}_+$.

The idea of the above theorem has a very nice intuition. If a set $C \subseteq \mathbb{E}$ is convex and unbounded, there must be a direction $d \in \mathbb{E} \setminus \{0\}$ such that, starting at any point in C, we can go in the direction d as much as we want. With this result, we are able to prove boundedness of the level sets of strongly convex functions.

Lemma 3.9.12. Let $R \colon \mathbb{E} \to (-\infty, +\infty]$ be a proper closed strongly convex on $\mathbb{E}$ function. Then, for every $\alpha \in \mathbb{R}$ such that $\inf_{x \in \mathbb{E}} R(x) < \alpha$ the set $L_\alpha \coloneqq \{\, x \in \mathbb{E} : f(x) \leq \alpha \}$ is nonempty and compact.

Proof. Since R is closed, by Theorem 3.2.4 we have that L_α is closed for any $\alpha \in \mathbb{R}$. Thus, we only need to prove boundedness. Suppose there is $\alpha \in \mathbb{R}$ with $\inf_{x \in \mathbb{E}} R(x) < \alpha$ such that L_α is unbounded. By Theorem 3.9.11, there is $d \in \mathbb{E} \setminus \{0\}$ such that $x + \mu d \in L_\alpha$ for any $x \in L_\alpha$ and any $\mu \in \mathbb{R}_+$. Let $\bar{x} \in L_\alpha$ and for every $\gamma \in \mathbb{R}_+$ with $\gamma \geq 1/\|d\|$ define

$$x_\gamma \coloneqq \bar{x} + \gamma d \in L_\alpha \qquad \text{and} \qquad \lambda_\gamma \coloneqq \frac{1}{\|\bar{x} - x_\gamma\|} = \frac{1}{\gamma \|d\|} \in (0, 1].$$

Thus, by the definition of strong convexity, for every $\gamma \in \mathbb{R}_+$ with $\gamma \geq 1/\|d\|$ we have

$$\begin{aligned} R((1 - \lambda_\gamma)\bar{x} + \lambda_\gamma x_\gamma) &\leq (1 - \lambda_\gamma) R(\bar{x}) + \lambda_\gamma R(x_\gamma) - \lambda_\gamma (1 - \lambda_\gamma) \frac{\sigma}{2} \|\bar{x} - x_\gamma\|^2 \\ &= (1 - \lambda_\gamma) R(\bar{x}) + \lambda_\gamma R(x_\gamma) - (1 - \lambda_\gamma) \frac{\sigma}{2} \|\bar{x} - x_\gamma\| \\ &\leq \alpha - (1 - \lambda_\gamma) \frac{\sigma}{2} \gamma \|d\|. \end{aligned}$$

Note that for any $\gamma \in \mathbb{R}_{++}$ we have $(1 - \lambda_\gamma)\bar{x} + \lambda_\gamma x_\gamma = \bar{x} + \|d\|^{-1} d \in L_\alpha$. This together with the fact that R is proper implies that $R((1 - \lambda_\gamma)\bar{x} + \lambda_\gamma x_\gamma) = R(\bar{x} + \|d\|^{-1} d) \neq -\infty$ for any $\gamma \in \mathbb{R}_{++}$. However, $\lim_{\gamma \to \infty} \alpha - (1 - \lambda_\gamma) \frac{\sigma}{2} \gamma \|d\| = -\infty$, a contradiction. □

Finally, let us show that infima of closed strongly convex function over closed convex sets are attained. As we have discussed earlier, the proof relies mainly on Cantor's Intersection Theorem, which we state next. Following it, we prove the attainability of the infima.

Theorem 3.9.13 (Cantor's Intersection Theorem [64, Theorem 2.36])**.** Let $\{K_i\}_{i=0}^{\infty}$ be a collection of nonemtpy compact sets on $\mathbb{E}$ such that $K_{i+1} \subseteq K_i$ for every $i \in \mathbb{N}$. Then $\bigcap_{i=0}^{\infty} K_i$ is nonempty.

Lemma 3.9.14. Let $R \colon \mathbb{E} \to (-\infty, +\infty]$ be a proper closed strongly convex on $\mathbb{E}$ function. Then $\inf_{x \in \mathbb{E}} R(x)$ is attained by a unique point. In particular, if $X \subseteq \mathbb{E}$ is convex such that $(\operatorname{epi} R) \cap (X \oplus \mathbb{R})$ is a closed nonempty set, then $\inf_{x \in X} R(x)$ is attained by a unique point. In particular, if X is closed and $(\operatorname{dom} R) \cap X$ is nonempty, then $\inf_{x \in X} R(x)$ is attained by a unique point.

Proof. For every $n \in \mathbb{N}$, define $\rho_n := \inf_{x \in \mathbb{E}} R(x) + \frac{1}{n+1}$ and $L_n := \{ x \in \mathbb{E} : R(x) \leq \rho_n \}$, where the latter sets are nonempty and compact for every $n \in \mathbb{N}$ by Lemma 3.9.12. Since $L_{n+1} \subseteq L_n$ holds for every $n \in \mathbb{N}$, by Cantor's Intersection Theorem (Theorem 3.9.13), we have that $\{ x \in \mathbb{E} : R(x) \leq \inf_{x \in \mathbb{E}} R(x) \} = \bigcap_{n=0}^{\infty} L_n$ is nonempty, that is, $\inf_{x \in \mathbb{E}} R(x)$ is attained. The uniqueness of the minimizer comes from the fact that R is strictly convex on $\mathbb{E}$ (since R is strongly convex in $\mathbb{E}$) and by Lemma 3.9.2.

In particular, let $X \subseteq \mathbb{E}$ be a convex set such that $(\operatorname{epi} R) \cap (X \oplus \mathbb{R})$ is closed and nonempty. Note that if we define $R' := R + \delta(\cdot \,|\, X)$, then $\operatorname{epi} R' = (\operatorname{epi} R) \cap (X \oplus \mathbb{R})$ is a closed nonempty set. Thus, we have that R' is a proper strongly convex function which is closed by Theorem 3.2.4. Thus, $\inf_{x \in \mathbb{E}} R'(x) = \inf_{x \in X} R(x)$ is attained by a unique point by the result we have just proved. Finally, note that if $X \subseteq \mathbb{E}$ is closed and $(\operatorname{dom} R) \cap X$ is nonempty, then $X \oplus \mathbb{R}$ is closed and $(X \oplus \mathbb{R}) \cap (\operatorname{epi} R)$ is a closed nonempty set. □

3.10 Strong Convexity and Smoothness Duality

As we have discussed on the previous section, the fact from Proposition 3.9.8 that the conjugate of a strongly convex function is finite and differentiable everywhere is, in fact, a corollary of a much more fundamental result. We investigate such a result in this section. Namely, we study the duality relation between strongly convex and *strongly smooth* functions.

Let $\beta \in \mathbb{R}_{++}$. A function $R \colon \mathbb{E} \to \mathbb{R}$ is **β-strongly smooth** (with respect to a norm $\|\cdot\|$ on $\mathbb{E}$) if R is everywhere differentiable and if for every $x, y \in \mathbb{E}$ we have[7]

$$\|\nabla R(x) - \nabla R(y)\|_* \leq \beta \|x - y\|.$$

From the definition, one can see that strong smoothness restricts how rapidly the slope of a function can change from one point to another. As an example of a smooth function, one may easily see that $R := \frac{1}{2}\|\cdot\|_2^2$ is 1-strongly smooth w.r.t. the ℓ_2-norm. This holds since $\nabla R(x) = x$ for any $x \in \mathbb{E}$ and since the ℓ_2-norm is self-dual. Another example of strongly smooth function (this time with respect to any norm) is any affine function, that is, any function of the form $R(x) := \langle a, x \rangle + \alpha$ for every $x \in \mathbb{E}$ for some $a \in \mathbb{E}$ and $\alpha \in \mathbb{R}$. The latter function is in fact ε-strongly smooth w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$ for any $\varepsilon > 0$ since $\nabla R(x) - \nabla R(y) = 0$ for any $x, y \in \mathbb{E}$. Thus, in some sense, affine functions are the most extreme example of strongly smooth functions. This was expected since the slope of affine functions are constant throughout $\mathbb{E}$.

At this point, it is not clear yet how strong convexity and strong smoothness may be related at all. The following lemma, which we state without proof, shows that strong smoothness implies a condition dual to Theorem 3.9.7: at each point $x \in \mathbb{E}$ a strongly smooth function R is upper-bounded by a quadratic function which touches the graph of R at least at x.

Lemma 3.10.1 ([55, Lemma 1.2.3])**.** If $R \colon \mathbb{E} \to (-\infty, +\infty]$ is a β-strongly smooth function w.r.t. a norm $\|\cdot\|$ in $\mathbb{E}$, then, for every $x, y \in \mathbb{E}$ we have

$$R(y) \leq R(x) + \langle \nabla R(x), y - x \rangle + \frac{\beta}{2}\|y - x\|^2.$$

Another way to interpret the above result is as a bound for the error of approximating a strongly smooth function by its first-order Taylor expansion. It shows that β-strongly smooth functions can be well-approximated by its first-order Taylor expansions, and the smaller the value of β the better the approximation.

[7] That is, ∇R is β-Lipschitz continuous with respect to $\|\cdot\|$.

Finally, the next theorem shows a dual relation regarding strongly convex and strongly smooth functions. Namely, the conjugate of a strongly convex function is strongly smooth and, maybe surprisingly, the converse also holds. It is important to notice that the theorem only holds for functions which are strongly convex throughout $\mathbb{E}$. Thus, if a function $R\colon \mathbb{E} \to (-\infty, +\infty]$ is strongly convex only on a closed convex set $X \subseteq \mathbb{E}$, one may apply the lemma to $R + \delta(\cdot \,|\, X)$ but not to R itself.

Theorem 3.10.2 ([41, Theorem 3])**.** Let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a closed convex function, let $\|\cdot\|$ be a norm on $\mathbb{E}$, and let $\sigma > 0$. Then R is σ-strongly convex on $\mathbb{E}$ w.r.t. $\|\cdot\|$ if and only if R^* is $(1/\sigma)$-strongly smooth w.r.t. $\|\cdot\|_*$.

3.11 Bregman Divergence and Projection

In the algorithms we will see in the next chapters, we shall use different norms in order to obtain algorithms with better guarantees. However, this may force us to change other aspects of the algorithm, such as the way we measure distances. For example, if one needs to project a point $x \in \mathbb{E}$ onto a (convex) set $X \subseteq \mathbb{E}$, a natural way to do so is by using euclidean projection. Namely, to pick the (unique) point which attains $\min_{z \in X} \|x - z\|_2$. This clearly depends on the euclidean norm, which will not be ideal in some applications we shall see. Moreover, simply replacing the ℓ_2-norm by any norm of our choice often does not work as intended. For example, consider the situation where we have the all ones vector $\mathbb{1} \in \mathbb{R}^2$ and want to project it onto the two-dimensional simplex $\Delta_2 = \{\, x \in \mathbb{R}^2_+ : x_1 + x_2 = 1\}$ using distances based on the ℓ_1-norm. This does not work as intended since the point which attains $\inf_{z \in \Delta_2} \|\mathbb{1} - z\|_1$ is not unique. In fact, $\|\mathbb{1} - z\|_1 = 1$ for any $z \in \Delta_2$, rendering the idea of projecting $\mathbb{1}$ onto the simplex meaningless. A notion which works better as a way of measuring "distances"[8] when we use a norm different of the ℓ_2-norm as the "reference norm" is the idea of *Bregman Divergences*.

Let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a function which is differentiable on an open set $D \subseteq \mathbb{E}$. The **Bregman divergence** associated with R is the function

$$B_R(x, y) \coloneqq R(x) - R(y) - \langle \nabla R(y), x - y \rangle, \qquad \forall x \in \mathbb{E}, y \in D.$$

On Figure 3.5 we give a illustration of the Bregman Divergence with respect to R. By looking at the figure the main idea behind the Bregman Divergence at $(\bar{x}, \bar{y}) \in \mathbb{E} \times D$ is made clear: it is the difference between the value of R at $\bar{x}$ and value of the first-order Taylor expansion of R around $\bar{y}$ evaluated at $\bar{x}$. Although we define Bregman Divergence for arbitrary differentiable functions, when the function R is convex at least one nice property arises: $B_R(x, y) \geq 0$ for any $x \in \mathbb{E}$ and $y \in \mathbb{E}$ and equality holds at least when $x = y$. Intuitively, we want the divergence between two points $x, y \in \mathbb{E}$ to be a measure of how different these points are when one looks at them through the lens of R.

From the above discussion, one may have noticed that the usefulness of the Bregman Divergence depends on the function R being curved throughout $\mathbb{E}$. As an extreme example, on may note that if R is of the form $\langle a, \cdot \rangle + \beta$ for some $a \in \mathbb{E}$ and $\beta \in \mathbb{R}$, then B_R is identically zero. In words, the Bregman Divergence associated with a flat function is zero everywhere. As we shall see, the divergence function starts to look nicer and have more interesting properties when R is strictly or even strongly convex, for example.

Before looking at some of the properties of Bregman Divergences, let us look at a classic example of Bregman Divergence. Namely, set $R \coloneqq \frac{1}{2}\|\cdot\|_2^2$. In this case, R is differentiable on $\mathbb{E}$ with $\nabla R(y) = y$

[8] Saying that we measure distances with Bregman Divergences is misleading since Bregman Divergences do not satisfy the conditions of a distance function.

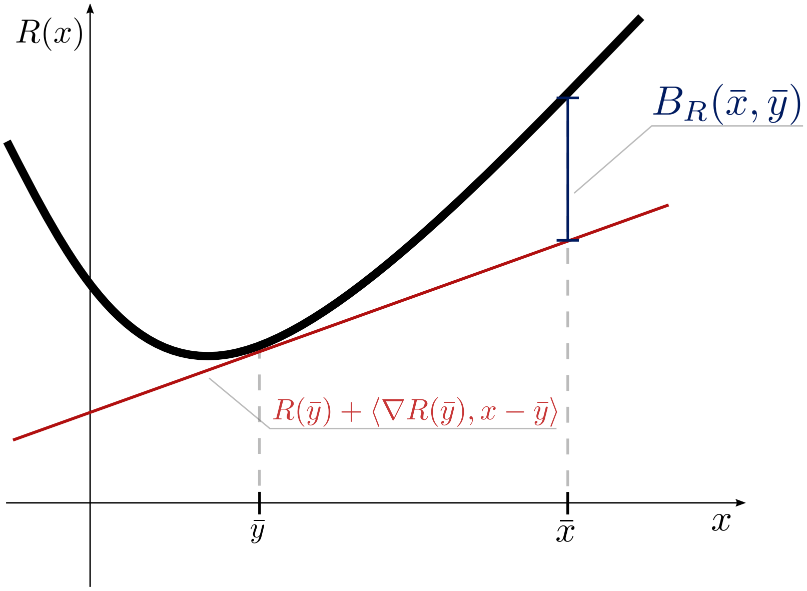
$R(x)$
$B_R(\bar{x}, \bar{y})$
$R(\bar{y}) + \langle \nabla R(\bar{y}), x - \bar{y} \rangle$
$\bar{y}$
$\bar{x}$
x

Figure 3.5: Illustration of the Bregman Divergence associated with R.

for every $y \in \mathbb{E}$. Thus,

$$B_R(x, y) = \frac{1}{2}\big(\|x\|_2^2 - \|x\|_2^2 - 2\langle y, x - y\rangle\big) = \frac{1}{2}\big(\|x\|_2^2 + \|y\|_2^2 - 2\langle y, x\rangle\big) = \frac{1}{2}\|x - y\|_2^2.$$

That is, the Bregman Divergence associated with the squared euclidean norm is the squared euclidean distance. Later we shall look at a more interesting example of Bregman Divergence. Namely, we shall show that the Bregman Divergence associated with the negative entropy $x \in \mathbb{R}^d \mapsto -\sum_{i=1}^d [x_i > 0] x_i \ln x_i + \delta(\cdot \,|\, \mathbb{R}^d_+)$ is (practically) the *Kullback–Leibler divergence* (also known as relative entropy).

Let us now look at simple but useful properties of Bregman Divergences. First, let us see the form of the (sub)gradients of Bregman Divergences.

Lemma 3.11.1. Let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a proper convex function which is differentiable on a convex set $D \subseteq \mathbb{E}$. Then,

$$\partial(B_R(\cdot, y))(x) = \partial R(x) - \nabla R(y), \qquad \forall x \in \mathbb{E}, \forall y \in D.$$

In particular, $\nabla(B_R(\cdot, y))(x) = \nabla R(x) - \nabla R(y)$ for every $x, y \in D$.

Proof. Let $y \in D$ and define $L_y(x) \coloneqq R(y) + \langle \nabla R(y), y - x\rangle$ for every $x \in \mathbb{E}$. For every $x \in \mathbb{E}$ we have $\partial L_y(x) = \{\nabla L_y(x)\} = \{-\nabla R(y)\}$ by Theorem 3.5.5. Thus, by Theorem 3.5.4 we have, for every $x \in \mathbb{E}$,

$$\partial(B_R(\cdot, y))(x) = \partial R(x) + \partial L_y(x) = \partial R - \nabla R(y).$$

In particular, if $x \in D$, then by Theorem 3.5.5 we have $\partial R(x) = \{\nabla R(x)\}$. □

Let us look now at the Bregman Divergence associated with a strongly convex function $R\colon \mathbb{E} \to (-\infty, +\infty]$ which is differentiable on a convex set $D \subseteq \mathbb{E}$. By definition of Bregman Divergence, for every $y \in D$ we have that $B_R(\cdot, y)$ is the sum of R, which is strongly convex, and the linear function

$x \in \mathbb{E} \mapsto -R(y) - \langle \nabla R(y), x - y \rangle$. Thus, $B_R(\cdot, y)$ is also strongly convex for any $y \in \mathbb{E}$. Although trivial, we state this result in the next lemma for future reference since it shall be important in the next results and chapters.

Lemma 3.11.2. Let $\|\cdot\|$ be a norm on $\mathbb{E}$, let $\sigma \in \mathbb{R}_{++}$, and let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a proper σ-strongly convex on X w.r.t. $\|\cdot\|$ function which is differentiable on a convex set $D \subseteq \mathbb{E}$. Then, for every $y \in D$, the function$B_R(\cdot, y)$ is also σ-strongly convex w.r.t. $\|\cdot\|$.

Proof. Let $y \in D$ and define $L_y(x) := R(y) + \langle \nabla R(y), x - y \rangle$ for every $x \in \mathbb{E}$. Then $B_R(\cdot, y) = R - L_y$. Since $-L_y$ is convex and R is σ-strongly convex on X w.r.t. $\|\cdot\|$, we conclude that $B_R(\cdot, y)$ is also σ-strongly convex on X w.r.t. $\|\cdot\|$. □

For the sake of brevity, we do not show more properties regarding Bregman Divergences since they will not be used in the remainder of the text. Still, many properties (and applications) of Bregman Divergences can be found, for example, in [24, Section 11.2] and in [19, Section 5.1].

Now, let us look at the idea of *Bregman projetions* which is extensively used in the algorithms seen in Chapter 5. Let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a proper convex function which is differentiable on a convex set D. Further, let $X \subseteq \mathbb{E}$ be a closed convex set and let $\bar{x} \in D$. If $\inf_{x \in X} B_R(x, \bar{x})$ is attained by a unique point, then the **Bregman projection** $\Pi_X^R(\bar{x})$ of $\bar{x}$ onto X (with respect to R) is given by

$$\{\Pi_X^R(\bar{x})\} := \operatorname*{arg\,min}_{x \in X} B_R(x, \bar{x}).$$

The operator Π_X^R is called the **Bregman projector** onto X w.r.t. R. As an example, if $R = \frac{1}{2}\|\cdot\|_2^2$, then Π_X^R recovers the well-known euclidean projection onto X since B_R becomes (half of) the squared euclidean distance.

As we briefly discussed at the beginning of this section, one of our motivations for using Bregman Divergences is to project points onto (convex) sets in ways which do not rely on the euclidean norm. The above definition of Bregman projection assumes that the projection is unique at every point where the function R is differentiable. The following lemma shows that the existence and uniqueness of the projection are guaranteed when the function R is strongly convex. Thus, in applications we usually will have Bregman projections w.r.t. strongly convex function. Additionally, it is worth noting that the norm associated with the strong convexity will usually play a major role in the analyses of the algorithms we shall see in future chapters.

Lemma 3.11.3. Let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a closed and strongly convex function which is differentiable on D and let $X \subseteq \mathbb{E}$ be such that $(\operatorname{dom} R) \cap X$ is nonempty. Then $\inf_{x \in X} B_R(x, y)$ is attained by a unique point for any $y \in D$.

Proof. Let $y \in D$. By Lemma 3.11.2, $B_R(\cdot, y)$ is strongly convex. Since $(\operatorname{dom} R) \cap X$ is nonempty, by Lemma 3.9.14 we conclude that $\inf_{x \in X} B_R(x, y)$ is attained. Uniqueness of the minimizer follows directly from the strict convexity of R and Lemma 3.9.2. □

Finally, it shall be very useful to have sufficient and necessary conditions for a point to be the Bregman projection of another onto some convex set. The next lemma derives some conditions of this form by basically applying the optimality conditions from Section 3.6 and writing them in a more palatable way.

Lemma 3.11.4. Let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a proper strictly convex function which is differentiable on an open set $D \subseteq \mathbb{E}$ and let $X \subseteq \mathbb{E}$ be a convex set. Finally, suppose $\operatorname{ri} X \cap \operatorname{ri}(\operatorname{dom} R) \neq \varnothing$. Then, for any $x, z \in D$ we have

$$x = \Pi_X^R(z) \iff \nabla R(z) - \nabla R(x) \in N_X(x). \tag{3.21}$$

In particular, if $y \in \mathbb{E}$ is such that R^* is differentiable at y and $\nabla R^*(y) \in D$, then, for any $x \in \operatorname{dom} R$,

$$x = \Pi_X^R(\nabla R^*(y)) \iff y - \nabla R(x) \in N_X(x). \tag{3.22}$$

Proof. Let $x, z \in D$. Recall that, by definition of the Bregman projector, $x = \Pi_X^R(z)$ if and only if x (is the unique point which) attains $\inf_{u \in X} B_R(u, z)$. Since $\operatorname{dom} R = \operatorname{dom} B_R(\cdot, x)$ and $\operatorname{ri} X \cap \operatorname{ri}(\operatorname{dom} R) \neq \varnothing$, by the optimality conditions given by Theorem 3.6.2,

$$x \in \operatorname*{arg\,min}_{u \in X} B_R(u, z) \iff \nabla R(z) - \nabla R(x) = -\nabla(B_R(\cdot, z))(x) \in N_X(x),$$

and the uniqueness of the projection follows from the strict convexity of R (which implies that $B_R(\cdot, z)$ is strictly convex as well). Let $y \in \mathbb{E}$ be such that R^* is differentiable at y and that $\nabla R^*(y) \in D$. By Corollary 3.5.6 we have $\nabla R(\nabla R^*(y)) = y$. Therefore, taking $z = \nabla R^*(y)$ on (3.21) yields (3.22). □

Finally, let us look at another classic example about which we have briefly talked about earlier on this section. The next proposition shows that the Bregman Divergence associated with the negative entropy on $\mathbb{R}^d$ is almost equal to the well-known Kullback-Leibler divergence (where equality holds when we look at the divergence among points in the simplex). Moreover, we also show that the Bregman projection w.r.t. the negative entropy in the simplex boils down to a normalization w.r.t. the ℓ_1-norm. Thus, Bregman projection associated with the negative entropy might be seen as a better way of computing projections when one has as a "reference norm" the ℓ_1-norm (one may recall the failed attempt of projecting onto the simplex by minimizing the distance w.r.t. the ℓ_1-norm from the beginning of this section).

Proposition 3.11.5. Define $R(x) \coloneqq \sum_{i=1}^d [x_i > 0] x_i \ln x_i + \delta(x \,|\, \mathbb{R}^d_+)$ for each $x \in \mathbb{R}^d$. Then, for any $x \in \mathbb{R}^d_+$ and $y \in \mathbb{R}^d_{++}$,

$$B_R(x, y) = \sum_{i=1}^d \left([x_i > 0] x_i \ln\left(\frac{x_i}{y_i}\right)\right) + \|y\|_1 - \|x\|_1.$$

In particular, if $X \coloneqq \Delta_d$ is the $(d-1)$-dimensional simplex, then

$$\Pi_X^R(y) = \frac{1}{\|y\|_1} y, \qquad y \in \mathbb{R}^d_{++}. \tag{3.23}$$

Proof. Let $x \in \mathbb{R}^d_+$ and $y \in \mathbb{R}^d_{++}$. Then,

$$\begin{aligned}
B_R(x, y) &= R(x) - R(y) - \langle \nabla R(y), x - y \rangle \\
&= \sum_{i=1}^d \big([x_i > 0] x_i \ln x_i - y_i \ln y_i - (1 + \ln y_i)(x_i - y_i) \big) \\
&= \sum_{i=1}^d \big([x_i > 0] x_i (\ln x_i - \ln y_i) - (x_i - y_i) \big) \\
&= \sum_{i=1}^d \left([x_i > 0] x_i \ln\left(\frac{x_i}{y_i}\right)\right) + \|y\|_1 - \|x\|_1.
\end{aligned}$$

In particular, set $X \coloneqq \Delta_d$. Then, since all points in X have same same ℓ_1-norm,

$$\begin{aligned}\{\Pi_X^R(y)\} &= \operatorname*{arg\,min}_{x \in X}\left(\sum_{i=1}^d \left([x_i > 0]x_i \ln\left(\frac{x_i}{y_i}\right)\right) + \|y\|_1 - \|x\|_1\right)\\ &= \operatorname*{arg\,min}_{x \in X}\sum_{i=1}^d \left([x_i > 0]x_i \ln\left(\frac{x_i}{y_i}\right)\right).\end{aligned}$$

Define $\bar{y} \coloneqq \|y\|_1^{-1}y$. By Corollary 3.2.3, $\operatorname{ri}\Delta_d = \{\, x \in \mathbb{R}^d_{++} : \|x\|_1 = 1\} \subseteq \mathbb{R}^d_{++} = \operatorname{ri}\mathbb{R}^d_{+}$. Therefore, by Lemma 3.11.4 we have $\bar{y} = \Pi_X^R(y)$ if and only if $\nabla R(y) - \nabla R(\bar{y}) \in N_X(\bar{y})$. We have

$$\nabla R(y) - \nabla R(\bar{y}) = \sum_{i=1}^d e_i(1 + \ln y_i - 1 - \ln \bar{y}_i) = \sum_{i=1}^d e_i\left(\ln y_i - \ln\left(\frac{y_i}{\|y\|_1}\right)\right) = \ln(\|y\|_1)\mathbb{1},$$

and, for every $u \in \Delta_d$, we have

$$\ln(\|y\|_1)\langle \mathbb{1}, u - \bar{y}\rangle = \ln(\|y\|_1)(\|u\|_1 - \|\bar{y}\|_1) = 0.$$

Therefore, $\nabla R(y) - \nabla R(\bar{y}) \in N_X(\bar{y})$ and, thus, $\bar{y} = \Pi_X^R(y)$. □

Chapter 4

The Follow The Regularized Leader Algorithm

Intuitively, the best point for a player to pick in an online convex optimization game is one that performs best on the already-seen functions. This strategy in its purest form, known as *Follow the Leader* (FTL), a name first coined in [42], fails to attain sublinear regret even in simple cases (see Section 4.1 or [67, Example 2.2]). Still, it is possible to tweak it to make it work well on a rich class of OCO problems. One issue with FTL is that simply picking a point that minimizes the cumulative loss of the past functions makes the player's choices overly susceptible to enemies' strategies, e.g. consecutive points from the player may be forced to be far from one another. As a matter of fact, a point picked following the FTL strategy does not leave any room for the player to skew this choice in any way, leaving the possibility for the enemy to direct the player's decisions through a smart choice of functions. A better strategy for the player is to pick a point that minimizes the cost of the already-seen functions plus a (convex) regularization term, which makes the choices of the player change less wildly between consecutive rounds. This idea is the basis of the *Follow the Regularized Leader* (FTRL) algorithm, which we describe and discuss in this chapter.

We first describe the algorithm in a more general and adaptive form, with a time-dependent regularizer. Later, we derive its classical version, that is, the one with a static regularizer, and apply both the classical and adaptive FTRL strategies to some problems. Even though the non-adaptive FTRL algorithm in the OCO context was first presented by Shalev-Shwartz and Singer in [69] (and by Shalev-Shwartz in [68]) its adaptive version and the analysis techniques here presented are due to McMahan [48].

4.1 The Follow the Leader and Follow the Regularized Leader Algorithms

As a warm-up and to build intuition for most of the algorithms of this chapter, let us look at the *Follow the Leader* algorithm for online convex optimization. For the sake of simplicity, let us first look at the randomized prediction with expert problem. As we have seen on Proposition 2.6.2, to obtain a good (in expectation) randomized player oracle for the expert's problem $\mathcal{P} := (A^E, A, Y, L)$, where E is a finite set of experts and A is a set of actions, it suffices to build a player oracle for the OCO instance $\mathcal{C} := (\Delta_E, \mathcal{F})$ which attains sub-linear regret, where Δ_E is the simplex on $\mathbb{R}^E$ and

$$\mathcal{F} := \{\, p \in \mathbb{R}^E \mapsto z^{\mathsf{T}} p : z \in [-1, 1]^E \}.$$

Consider the situation in which a certain player oracle for $\mathcal{C}$ has already played $T \in \mathbb{N} \setminus \{0\}$ rounds against an enemy oracle ENEMY for $\mathcal{C}$. Moreover, suppose that $\boldsymbol{z} \in ([-1,1]^E)^T$ is such that, for every $t \in [t]$ the function $f_t \colon \mathbb{R}^E \to \mathbb{R}$ given by $f_t(p) \coloneqq z_t^{\mathsf{T}} p$ for every $p \in \mathbb{R}^E$ is the function chosen by ENEMY on round t. In this case, at least intuitively, which is a good choice in Δ_E for the player at round $T+1$?

All the information the player has at round $T+1$ to make her choice is comprised in the sequence $\boldsymbol{z}$. In particular, for each expert $e \in E$ she can compute its *cumulative cost* $\sum_{t=1}^{T} z_t(e)$, i.e., the sum of all the costs attributed to this expert throughout the game. Instinctively, the lower the cumulative loss of an expert, the better its advice have been up to round T. Thus, an idea for an strategy for the player at round $T+1$ is for her to follow the expert with minimum cumulative cost. More generally, the player may pick at round $T+1$ the probability distribution over the experts which minimizes the sum of the expected losses from rounds 1 to T, that is,

$$p_{T+1} \in \operatorname*{arg\,min}_{p \in \Delta_E} \sum_{t=1}^{T} \langle z_t, p \rangle. \tag{4.1}$$

That is the main idea behind the Follow the Leader (FTL) algorithm: at round $t+1$ the player picks point which minimizes the sum of the already-seen functions. On Algorithm 4.1 we define an oracle which formally implements the FTL algorithm.

Algorithm 4.1 Definition of $\text{FTL}^X(\langle f_1, \dots, f_T \rangle)$

```
Input:
   (i) Functions $f_1, \dots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$,
  (ii) A set $X \subseteq \mathbb{E}$ such that $\arg\min_{x \in X} \sum_{t=1}^{T} f_t(x)$ is attained.
Output: $x_{T+1} \in X$
  Compute $x_{T+1} \in \arg\min_{x \in X} \sum_{t=1}^{T} f_t(x)$
  return $x_{T+1}$
```

One may be wondering how good the FTL algorithm performs in certain classes of OCO problems. Although there are some certain classes of OCO problems in which the FTL algorithm is able to attain sub-linear regret (we shall look at this case in details in Section 4.8), there are quite natural OCO problems, such as the experts' problem, in which FTL fails to attains sub-linear regret in the worst-case scenario. Note that when discussing the idea of the algorithm in the expert's problem, we ended up with the formula on (4.1). Since in the latter we are minimizing a linear function on the simplex, we have that for some expert $i \in E$ the vector e_i attains the minimum. That is, at each round the FTL algorithm may choose to follow one expert deterministically and, as we have seen on Proposition 2.3.3, this is bound to fail in the worst-case scenario. For the sake of concreteness, let us build an enemy oracle for an instance of the expert's problem (with only two experts) against which the FTL algorithm suffers linear regret.

Proposition 4.1.1. Define $E \coloneqq \{1,2\}$ and define the OCO instance $\mathcal{C} \coloneqq (\Delta_E, [-1,1]^E)$. Moreover, define

$$z_t \coloneqq \begin{cases} \frac{1}{2} e_2 & \text{if } t = 1, \\ e_1 - e_2 & \text{if } t \text{ is even}, \\ -e_1 + e_2 & \text{if } t \text{ is odd and } t \neq 1, \end{cases} \qquad \forall t \in \mathbb{N} \setminus \{0\},$$

and define the enemy oracle ENEMY for $\mathcal{C}$ by

$$\text{ENEMY}(\langle p_1, \dots, p_T \rangle) \coloneqq (x \in \mathbb{R}^E \mapsto (z_{T+1})^{\mathsf{T}} x), \qquad \forall T \in \mathbb{N}, \forall \boldsymbol{p} \in (\Delta_E)^T.$$

Then,

$$\mathrm{Regret}_T(\mathrm{FTL}^{\Delta_E}, \mathrm{ENEMY}, \{e_1, e_2\}) \geq T - 2.$$

Proof. Let $T \in \mathbb{N}$ and define

$$(\boldsymbol{p}, \boldsymbol{f}) \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{FTL}^{\Delta_E}, \mathrm{ENEMY}, T).$$

First, let us derive a closed formula for the player's choices. Namely, let us show that

$$p_t = \begin{cases} e_1, & \text{if } t \text{ is even} \\ e_2, & \text{if } t \text{ is odd,} \end{cases} \qquad \forall t \in [T] \setminus \{1\}. \tag{4.2}$$

From the definition of z_t, by an easy induction one may see that

$$\sum_{i=1}^{t} z_i = \begin{cases} e_1 - \frac{1}{2}e_2, & \text{if } t \text{ is even} \\ \frac{1}{2}e_2, & \text{if } t \text{ is odd,} \end{cases} \qquad \forall t \in [T]. \tag{4.3}$$

Therefore, for every $t \in \{1, \ldots, T-1\}$

$$p_{t+1} \in \operatorname*{arg\,min}_{p \in \Delta_E} \Big(\sum_{i=1}^{t} z_i\Big)^{\mathsf{T}} p = \operatorname*{arg\,min}_{p \in \Delta_E} \big([t \text{ is even}]p_1 + (-1)^{t+1}p_2\big) = \begin{cases} \{e_1\} & \text{if } t \text{ is odd} \\ \{e_2\} & \text{if } t \text{ is even.} \end{cases}$$

This proves (4.2). Using (4.2) together with the definition of z_t for each $t \in [T]$, we have

$$\sum_{t=1}^{T} f_t(p_t) = z_1^{\mathsf{T}} p_1 + \sum_{t=2}^{T} z_t^{\mathsf{T}} p_t = z_1^{\mathsf{T}} p_1 + \sum_{t=2}^{T} 1 \geq T - 1.$$

Finally, by the definition of z_t for each $t \in [T]$ together with (4.3) we have, for any $i \in E$,

$$\sum_{t=1}^{T} f_t(e_i) = \sum_{t=1}^{T} z_t(i) \leq 1.$$ □

As one may have noticed, one of the problems of the FTL algorithm is that the enemy has too much power over the player's decisions. In the above proposition, the enemy is able to make the player switch from one expert to another while making, at the same time, the expert chosen by the player suffer the maximum amount of loss possible. It seems that if we could make the choices of the player change less widely from one round to another, the algorithm would be less susceptible to such kind of malicious enemy. One way to stabilize the player's choices is to add to the functions being minimized an extra function $R \colon \mathbb{E} \to (-\infty, +\infty]$, a *regularizer*. Intuitively, when a regularizer with certain properties is added to the functions being minimized in the Follow the Leader algorithm, it is harder for the iterates from one round to the next to be very different from one another. On Algorithm 4.2 we define an oracle which formally implements the FTL algorithm with the addition of a regularizer, strategy known as the *Follow the Regularized Leader* algorithm.

We shall leave the discussion about which functions work well as a regularizer for later sections. For now, let us try to understand the FTRL oracle and look at some nice properties for regularizer functions to have.

One may have noticed that while on Algorithm 4.1 the oracle is parameterized by the set on which the minimization shall happen, on Algorithm 4.2 the algorithm is only parameterized by the regularizer function. This is possible since we can embed the set restriction into the regularizer

Algorithm 4.2 Definition of $\mathrm{FTRL}_R(\langle f_1, \dots, f_T \rangle)$

```
Input:
    (i) Functions $f_1, \dots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$,
    (ii) $R\colon \mathbb{E} \to (-\infty, +\infty]$ such that, for every $t \in \{0, \dots, T\}$, the function $R + \sum_{i=1}^{t} f_i$ is proper
         and its infimum over $\mathbb{E}$ is attained.
Output: $x_{T+1} \in \operatorname{dom} R \subseteq \mathbb{E}$
  Compute $x_{T+1} \in \arg\min_{x \in \mathbb{E}} \Big( R(x) + \sum_{t=1}^{T} f_t(x) \Big)$
  return $x_{T+1}$
```

through the use of indicator functions. For example, for any set $X \subseteq \mathbb{E}$ we have $\mathrm{FTRL}_{\delta(\cdot \,|\, X)} = \mathrm{FTL}^X$. More generally, given a function $R\colon \mathbb{E} \to (-\infty, +\infty]$, we can use the function $R + \delta(\cdot \,|\, X)$ as a regularizer for FTRL to enforce the points chosen by the algorithm to lie in a set $X \subseteq \mathbb{E}$.

Finally, the biggest issue now is to properly choose regularizer functions for FTRL for a given OCO instance $\mathcal{C} := (X, \mathcal{F})$. Without trying to think about which conditions the regularizer ought to have in order for the FTRL algorithm to attain sub-linear regret, there are some basic properties which it should have to either make the analysis simpler and/or to certify that the FTRL algorithm with such a function as a regularizer is properly defined. These conditions as encapsulated in the concept of *classical FTRL regularizer*.

Definition 4.1.2 (Classical FTRL regularizer)**.** Let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a convex function and let $\mathcal{C} := (X, \mathcal{F})$ be an OCO instance. Then R is a **(classical) FTRL regularizer** for $\mathcal{C}$ if

(4.4.i) R is closed, proper, and convex,

(4.4.ii) $\operatorname{dom} R \subseteq X$, and

(4.4.iii) for any $T \in \mathbb{N}$ and any $\boldsymbol{f} \in \mathcal{F}^T$, the function $R + \sum_{i=1}^{T} f_i$ is closed, proper, convex, and its infimum over $\mathbb{E}$ is attained.

Property (4.4.i) ensures that a FTRL regularizer is well-behaved, which shall allow us to use many of the tools developed on the previous chapter. Property (4.4.ii) is only to ensure that the points picked by the FTRL oracle are valid points for the player to pick at the OCO game. Finally, property (4.4.iii) ensures that minima which the FTRL algorithm computes are well-behaved.

4.2 The Adaptive FTRL Algorithm

In the previous section we have briefly looked at the Follow the Regularized algorithm. In a nutshell, the algorithm picks at round a given round $t \in \mathbb{N}$ the point which minimizes the sum of the already-seen functions plus a regularizer function. However, no matter which functions the enemy play, the algorithm is stuck with the same regularizer throughout the whole game. In this section we look at a generalization of the FTRL algorithm which works similarly to the FTRL algorithm with the key difference that, at each round, it chooses a different regularizer.

Namely, let $f_1, \dots, f_t\colon \mathbb{E} \to (-\infty, +\infty]$ be closed proper convex functions played by an enemy oracle until round t in an OCO instance. That idea is that, at round $t+1$, the player u picks a function R_{t+1}, the regularizer at round $t+1$, and then picks a point that minimizes $R_{t+1} + \sum_{i=1}^{t} f_i$ as her choice for the round. One important aspect of R_{t+1} is that it may depend on $f_1, \dots, f_t$ and on the past player choices. In order to derive bounds on the regret and define the algorithm more concisely, it is convenient for us to write R_{t+1} as the sum of $t+1$ functions[1] $r_1, r_2, \dots, r_{t+1}$, where r_i is chosen by the player on round i for each $i \in \{1, \dots, t+1\}$. In this way, at round $t+1$ the player picks the point that minimizes $\sum_{i=1}^{t+1} r_i + \sum_{i=1}^{t} f_i$. This algorithm is known as *Adaptive Follow the Regularized Leader* (Adaptive FTRL or AdaFTRL), and an oracle that implements it is formally defined in Algorithm 4.3.

Algorithm 4.3 Definition of $\mathrm{AdaFTRL}_{\mathcal{R}}(\langle f_1, \dots, f_T \rangle)$

```
Input:
    (i) Functions $f_1, \dots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$,
    (ii) $\mathcal{R}\colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ such that, for every $t \in \{1, \dots, T+1\}$, the function
         $\sum_{i=1}^{t+1} \mathcal{R}(\langle f_1, \dots, f_{i-1} \rangle) + \sum_{i=1}^{t} f_i$ is proper and its infimum over $\mathbb{E}$ is attained.
Output: $x_{T+1} \in \operatorname{dom} \mathcal{R}(\langle\rangle) \subseteq \mathbb{E}$
  for $t = 1$ to $T+1$ do
      $r_t \leftarrow \mathcal{R}(\langle f_1, \dots, f_{t-1} \rangle)$
      $R_t \leftarrow [t > 1] R_{t-1} + r_t$
  Compute $x_{T+1} \in \arg\min_{x \in \mathbb{E}} \Big( R_{T+1}(x) + \sum_{t=1}^{T} f_t(x) \Big)$
  return $x_{T+1}$
```

To formalize the dependence of the regularizer function r_{t+1} on the past functions $f_1, \dots, f_t$, the AdaFTRL oracle is parameterized by a function $\mathcal{R}$ from $\mathrm{Seq}(\mathcal{F})$ to $(-\infty, +\infty]^{\mathbb{E}}$ for some $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$. Such $\mathcal{R}$ is what defines the strategy behind the choice of the regularizers for the player. In other words, at a round $t \in \mathbb{N}$ the regularizer increment r_t of the player is given by $\mathcal{R}(\langle f_1, \dots, f_{t-1} \rangle)$, where $f_1, \dots, f_{t-1}$ were the functions played by the enemy in past rounds. Moreover, the regularizer function used in the minimization by the player is the sum of all the regularizer increments computed up to round t.

To use AdaFTRL in games for an OCO instance $\mathcal{C} \coloneqq (X, \mathcal{F})$, we want the algorithm to behave exactly as the FTRL algorithm but with a different regularizer at each round. Thus, we usually want to use AdaFTRL with a function $\mathcal{R}\colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]$ which, for any $T \in \mathbb{N}$ and any $\boldsymbol{f} \in \mathcal{F}^T$, the function $R_{T+1} \coloneqq \sum_{t=1}^{T+1} \mathcal{R}(\langle f_1, \dots, f_{t-1} \rangle)$ is a classical FTRL regularizer. Moreover, to make the analysis easier it is good that $\mathcal{R}(\boldsymbol{f})$ is a well-behaved (proper closed convex) function for any $\boldsymbol{f} \in \mathrm{Seq}(\mathcal{F})$.

Definition 4.2.1 (FTRL regularizer strategy)**.** If $\mathcal{C} \coloneqq (X, \mathcal{F})$ is an OCO instance, we say that a function $\mathcal{R}\colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]$ is a **FTRL regularizer strategy** for $\mathcal{C}$ if, for any $T \in \mathbb{N}$ and any $\boldsymbol{f} \in \mathcal{F}^T$,

(4.5.i) $\mathcal{R}(\boldsymbol{f})$ is closed, proper, and convex,

(4.5.ii) $\sum_{t=1}^{T+1} \mathcal{R}(\boldsymbol{f}_{1:t-1})$ is a classical FTRL regularizer for $\mathcal{C}$

Property (4.5.i) from the definition of FTRL regularizer strategy guarantees that the regularizer increments are well-behaved. Property (4.4.ii) from the definition of classical FTRL regularizer together with (4.5.ii) from the definition of FTRL regularizer strategy ensures that the iterates picked by $\mathrm{AdaFTRL}_{\mathcal{R}}$ are valid points for the player, that is, $\mathrm{AdaFTRL}_{\mathcal{R}}(\boldsymbol{f}) \in X$ for any $\boldsymbol{f} \in \mathrm{Seq}(\mathcal{F})$.

[1] We can usually do that since, given functions $R_1, R_2, \dots, R_{t+1}$, we may define $r_i \coloneqq R_i - [i > 1] R_{i-1}$ for $i \in \{1, \dots, t+1\}$. However, we will usually go the other way around, defining the "increment" functions r_i first. In this way, we can guarantee some properties on these increment functions, such as convexity, which will be useful mainly on the next chapter.

Finally, property (4.4.iii) together with (4.5.ii) from the definition of FTRL regularizer strategy guarantees that the infimum of the function minimized by the AdaFTRL oracle when applied to any sequence from $\mathrm{Seq}(\mathcal{F})$ is attained. Most of the convexity properties stated above will not be used until Section 4.4 (and this is made clear in the statements of the results). Still, it is worth standardizing these conditions for FTRL regularizers from the outset since they will be ubiquitous throughout most of the remaining of the text.

At first sight, one might think that $\mathcal{R}$ should also receive the points chosen by the player in the past rounds as input, since it may be a good idea for the player to choose regularizers based also on the points she picked in the past rounds. Note, however, that for every round $t \in \mathbb{N}$, the point $x_{t+1} \coloneqq \mathrm{AdaFTRL}_{\mathcal{R}}(\langle f_1, \dots, f_t \rangle)$ depends deterministically only on $f_1, \dots, f_t$ and $\mathcal{R}$. Thus, at round $t+1$, which is when $\mathcal{R}$ receives as input $f_1, \dots, f_t$, the regularizer strategy $\mathcal{R}$ knows (or can compute) the points picked on past rounds by the player, even without receiving such points explicitly as input.

Before we continue, we need to address a technical problem. The definition of $\mathrm{AdaFTRL}_{\mathcal{R}}$ is ambiguous, since different choices of points in the set of minimizers could lead to different evaluations of $\mathrm{AdaFTRL}_{\mathcal{R}}$ given the same arguments. In order to fix that, not only for this definition but for other oracles that we will define throughout this chapter, we assume that every set A is equipped with an arbitrary relation $\leq_A$ which defines a well-ordering of the elements on A. Thus, every time we have to pick a point from a nonempty set A, we will use the convention that such a point is the minimum with respect to $\leq_A$. This solves the technical problem and, since the order is arbitrary, it does not restrict us in any way. We stress here that the sole purpose of this assumption is to guarantee that, if we give the same input at different points in the text to any oracle that we define, the outputs of the oracle in these cases are the same, i.e., the oracle is a function. This assumption is used for no other purpose besides this one.

Finally, let us look at some regularizer strategies examples for the sake of concreteness. Let $\mathcal{C} \coloneqq (\mathbb{R}^n, \mathcal{F})$ be an OCO instance such that every function in $\mathcal{F}$ is differentiable on $\mathbb{R}^n$. Probably the most classical example of FTRL regularizer strategy is the (static) ℓ_2-norm squared. Namely, let $\eta \in \mathbb{R}_{++}$ be some fixed positive constant, define $R \coloneqq \frac{1}{2\eta}\|\cdot\|_2^2$, and define the FTRL regularizer strategy $\mathcal{R}$ by

$$\mathcal{R}(\boldsymbol{h}) \coloneqq [\boldsymbol{h} = \langle\rangle] R = [\boldsymbol{h} = \langle\rangle] \frac{1}{2\eta}\|\cdot\|_2^2, \qquad \forall \boldsymbol{h} \in \mathrm{Seq}(\mathcal{F}),$$

That is, the regularizer increment in the first round is the ℓ_2-norm squared (scaled by a constant), and all the other increments are 0. It is easy to see that R satisfies the conditions of a classical FTRL regularizer for $\mathcal{C}$, which implies that $\mathcal{R}$ is indeed an FTRL regularizer strategy for $\mathcal{C}$. Indeed, since the ℓ_2-norm is convex and continuous on $\mathbb{R}^d$, we have that R satisfies (4.4.i), and (4.4.ii) is easily satisfied since we are in the unconstrained case (i.e., the player's decision space is $\mathbb{R}^d$). Finally, since the squared ℓ_2-norm is strongly convex on $\mathbb{R}^d$ (see Lemma 3.9.5) and since $\mathrm{dom}\, f = \mathbb{R}^d$, we have that $R + \sum_{t=1}^T f_t$ is also strongly convex on $\mathbb{R}^d$ for any $T \in \mathbb{N}$ and any $\boldsymbol{f} \in \mathcal{F}^T$. Thus, condition (4.4.iii) for R is a consequence of Lemma 3.9.14, which states that the infimum of a closed strongly convex function over the entire space is always attained.

Let us now look if we can obtain a closed formula for the iterates of AdaFTRL with this regularizer. Let $T \in \mathbb{N}$ and $\boldsymbol{f} \in \mathcal{F}^T$ be a sequence of linear functions, that is, there is $\boldsymbol{g} \in (\mathbb{R}^d)^T$ such that $f_t = g_t^{\mathsf{T}} \cdot$ for each $t \in [T]$. By definition we have

$$x_t \coloneqq \mathrm{AdaFTRL}_{\mathcal{R}}(\boldsymbol{f}_{1:t-1}) \in \operatorname*{arg\,min}_{x \in \mathbb{R}^n} \Big(\frac{1}{2\eta}\|x\|_2^2 + \sum_{i=1}^{t-1} f_i(x) \Big), \qquad \forall t \in [T].$$

Since the whole function being minimized above is differentiable, with $\nabla f_t(x) = g_t$ and $\partial f_t(x) =$

$\{\nabla f_t(x)\}$ (the latter by Theorem 3.5.5) for any $x \in \mathbb{E}$ and $t \in [T]$, optimality conditions (Theorem 3.6.2) yield, for each $t \in [T]$,

$$x_t = -\eta \sum_{i=1}^{t-1} \nabla f_i(x_i) = -\eta \sum_{i=1}^{t-1} g_i \implies x_t = [t > 1](x_{t-1} - \eta g_{t-1}),$$

where the implication follows by a simple induction since $x_1 = 0$ in this case. That is, in this unconstrained case the Adaptive FTRL with squared ℓ_2 regularization is exactly the well-known gradient descent algorithm! A reader familiar with gradient descent is probably thinking how to use time-varying values for η, the *step size*. For this case, define $\mathcal{R}$ by

$$\mathcal{R}(\boldsymbol{h}) := \left(\frac{1}{\eta_{t+1}} - [t > 0]\frac{1}{\eta_t}\right)\frac{1}{2}\|\cdot\|_2^2, \qquad \forall t \in \mathbb{N}, \boldsymbol{h} \in \mathcal{F}^t,$$

where $\eta \colon \mathbb{N} \setminus \{0\} \to \mathbb{R}_{++}$. Then, following similar steps to the static ℓ_2-norm case, we get the update rule $x_t = [t > 1](x_{t-1} - \eta_t g_{t-1})$. On Section 4.5 we will look at the general case with static regularizers, and on Section 4.6 we will look at time-varying step sizes. Moreover, the connections among FTRL and different variants of gradient descent will be further investigated in Chapter 5. There are several other regularizer strategy examples which we will look at throughout the remaining of the text. For example, in both of the previous examples we have always used squared norm as the regularizer at each round, changing only maybe the scaling factor. One option is too look at the squared distance from the previous iterate, that is, use at round t a regularizer increment of the type $x \in \mathbb{E} \mapsto \|x - x_t\|^2$. We will look at these types of regularizers on Section 4.7. Another interesting option is, for each $t \in \mathbb{N} \setminus \{0\}$, to have a positive definite matrix $A_t \in \mathbb{R}^{n \times n}$, and to use at round t a regularizer of the type $x \in \mathbb{R}^n \mapsto x^\mathsf{T} A_t x$. That is, at each round use a different (squared) norm induced by some matrix. This is a path which we investigate on Chapter 6.

4.3 Fundamental Lemmas for Regret Bounds

Our goal now is to prove general upper bounds on the worst-case regret of AdaFTRL. The main ingredients for our bounds will be proved in this section. Namely, we will prove Lemmas 4.3.1 and 4.3.2. The first is also known as the *Strong FTRL Lemma*, a very general result which does not depend on convexity, and even though it is not that useful by itself, it highlights which quantities we should focus on to obtain meaningful bounds. On Section 4.9 we will compare this lemma to the similar Follow the Leader–Be the Leader Lemma by Kalai and Vempala [42], which is often used to bound the regret of non-adaptive FTRL algorithms. Lemma 4.3.2 is the second ingredient, whose proof shown here relies heavily on convex analysis concepts and duality, and is the tool that makes the bounds given by the Strong FTRL Lemma more concrete by connecting these guarantees with the convexity parameters of the functions played by the enemy.

Lemma 4.3.1 (Strong FTRL Lemma [48, Lemma 5])**.** Let $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$ and $\mathcal{R} \colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$. Let $T \in \mathbb{N}$ and $\boldsymbol{f} \in \mathcal{F}^T$. Moreover, define

$$\begin{aligned}
x_t &:= \mathrm{AdaFTRL}_{\mathcal{R}}(\langle f_1, \ldots, f_{t-1}\rangle) && \text{for each } t \in \{1, \ldots, T+1\},\\
r_t &:= \mathcal{R}(\langle f_1, \ldots, f_{t-1}\rangle) && \text{for each } t \in \{1, \ldots, T+1\},\\
H_t &:= \sum_{i=1}^{t+1} r_i + \sum_{i=1}^{t} f_i && \text{for each } t \in \{0, \ldots, T\},
\end{aligned}$$

and set $x_0 \coloneqq x_1$. If H_t is proper and its infimum over $\mathbb{E}$ is attained for every $t \in \{0, \dots, T\}$, then, for every $u \in \mathbb{E}$,

$$\mathrm{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, u) \leq \sum_{t=1}^{T+1} (r_t(u) - r_t(x_{t-1})) + \sum_{t=1}^{T} (H_t(x_t) - H_t(x_{t+1})). \tag{4.6}$$

Proof. For each $t \in \{0, \dots, T\}$, define $h_t \coloneqq r_{t+1} + [t > 0] f_t$. In this way, we have

$$x_t \in \operatorname*{arg\,min}_{x \in \mathbb{E}} H_{t-1}(x) = \operatorname*{arg\,min}_{x \in \mathbb{E}} \sum_{i=0}^{t-1} h_i(x), \qquad \forall t \in \{1, \dots, T+1\}. \tag{4.7}$$

Let us now bound the regret of the points $x_1, \dots, x_T$ with respect to the functions $h_1, \dots, h_T$ and to a comparison point $u \in \mathbb{E}$ (plus a $-h_0(u)$ term):

$$\begin{aligned}
\sum_{t=1}^{T} (h_t(x_t) - h_t(u)) - h_0(u) &= \sum_{t=1}^{T} h_t(x_t) - H_T(u) = \sum_{t=1}^{T} (H_t(x_t) - H_{t-1}(x_t)) - H_T(u) \\
&\overset{(4.7)}{\leq} \sum_{t=1}^{T} (H_t(x_t) - H_{t-1}(x_t)) - H_T(x_{T+1}) \\
&= \sum_{t=1}^{T} (H_t(x_t) - H_t(x_{t+1})) - H_0(x_1),
\end{aligned}$$

where in the last equation we just re-indexed the summation, placing $H_{T+1}(x_{T+1})$ inside the summation, and leaving $H_0(x_1)$ out. Re-arranging the terms and using $H_0 = h_0 = r_1$ and $x_0 = x_1$ yield

$$\begin{aligned}
\sum_{t=1}^{T} (f_t(x_t) + r_{t+1}(x_t) - f_t(u) - r_{t+1}(u)) &= \sum_{t=1}^{T} (h_t(x_t) - h_t(u)) \\
&\leq r_1(u) - r_1(x_0) + \sum_{t=1}^{T} (H_t(x_t) - H_t(x_{t+1})),
\end{aligned}$$

which implies

$$\mathrm{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, u) = \sum_{t=1}^{T} (f_t(x_t) - f_t(u)) \leq \sum_{t=1}^{T+1} (r_t(u) - r_t(x_{t-1})) + \sum_{t=1}^{T} (H_t(x_t) - H_t(x_{t+1})). \qquad \square$$

The above lemma has a quite straightforward proof, so much so that one may finish reading it with a feeling that we have not done much by proving this lemma. Indeed, most of the proof boils down to rewriting the regret expression in a way in which the terms are displayed in a more palatable way. Interestingly, the only inequality used in the whole proof of the lemma is due to (4.7), which holds by the definition of the AdaFTRL algorithm.

The Strong FTRL Lemma bounds the regret of AdaFTRL by two sums. The first is usually bounded by some kind of per-round diameter, as measured by the regularizer, of the set $X \subseteq \mathbb{E}$ where the player is making her predictions (assuming $u \in X$ as well). This already shows that the choice of a regularizer will be heavily influenced by the set X. The second sum translates the intuition we talked about in the beginning of the chapter: it measures the stability of consecutive iterates. The player then has to balance two competing factors. On the one hand, she wants to minimize the raw values of the functions H_t by the definition of the AdaFTRL oracle. On the other hand, her choices from one round to another should not change too abruptly, so the terms of the form $H_t(x_t) - H_t(x_{t+1})$ on the bound do not become too high.

As one may have noticed, consecutive iterates from the AdaFTRL algorithm are minimizers of functions which are, in some sense, similar. More explicitly, let $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$ be nonempty, let $\boldsymbol{f} \in \mathcal{F}^t$ for some $t \in \mathbb{N} \setminus \{0\}$, and let $\mathcal{R}$ be any appropriate regularizer strategy for $\mathcal{F}$. Define consecutive iterates $x_t \coloneqq \mathrm{AdaFTRL}_{\mathcal{R}}(\langle f_1, \dots, f_{t-1}\rangle)$ and $x_{t+1} \coloneqq \mathrm{AdaFTRL}_{\mathcal{R}}(\langle f_1, \dots, f_t\rangle)$. Note that x_t minimizes $H \coloneqq \sum_{i=1}^t r_i + \sum_{i=1}^{t-1} f_i$, and x_{t+1} minimizes $H + f_t + r_{t+1}$, where $r_i \coloneqq \mathcal{R}(\langle f_1, \dots, f_i\rangle)$ for each $i \in \{0, \dots, t\}$. Looking from this perspective, one may wonder if we can say something about the distance between x_t and x_{t+1} (or the difference between the values of $H + f_t + r_{t+1}$ at these points). For example, in the case where $f_t + r_{t+1}$ does not vary much throughout $\mathbb{E}$ we can guess that x_t and x_{t+1} and the values of $H + f_t + r_{t+1}$ at these points are close. Unfortunately, if we allow the functions from $\boldsymbol{f}$ and the regularizers delivered by $\mathcal{R}$ to be arbitrary, we cannot guarantee much. This is a point where convexity starts to play a major role in the analysis of AdaFTRL. The next lemma shows that if $H + f_t$ is strongly convex[2], then the distance between x_t and x_{t+1} is bounded by the dual norm of the subgradients of f_t at x_t. Even though a bound which depends on the dual norm of the subgradient may seem to lack any intuitive meaning at first, recall that such a quantity is deeply connected with the Lipschitz continuity constant of f_t (see Theorem 3.8.4). Thus, the following lemma tell us that if f_t is ρ-Lipschitz continuous for small ρ and H is strongly convex (for example), then adding f_t to H does not move much the points which attain the minimum. It is worth noting that, when we apply this result, it is not always that case that F from the statement is of the same form as the function H from our current discussion (though usually F will only be slightly different from H). As a matter of fact, the different functions the we plug into F in different applications of the lemma when bounding terms from the Strong FTRL Lemma yield similar bounds, but with important "off-by-one" differences which will be discussed in the next section.

Lemma 4.3.2 ([48, Lemma 7])**.** Let $F, f \colon \mathbb{E} \to (-\infty, +\infty]$ be closed proper convex functions such that $F + f$ is σ-strongly convex with respect to a norm $\|\cdot\|$ on $\mathbb{E}$ and such that $\inf_{x \in \mathbb{E}} F(x)$ is attained, and let $\bar{x} \in \arg\min_{x \in \mathbb{E}} F(x)$. If $\mathrm{ri}(\mathrm{dom}\, F) \cap \mathrm{ri}(\mathrm{dom}\, f)$ is nonempty, then $\inf_{x \in \mathbb{E}}(F(x) + f(x))$ is attained and, for any $g \in \partial f(\bar{x})$,

$$\|\bar{x} - \bar{y}\| \leq \tfrac{1}{\sigma}\|g\|_* \qquad \forall \bar{y} \in \operatorname*{arg\,min}_{x \in \mathbb{E}}(F(x) + f(x))$$

and

$$F(\bar{x}) + f(\bar{x}) - (F(u) + f(u)) \leq \frac{1}{2\sigma}\|g\|_*^2, \qquad \forall u \in \mathbb{E}.$$

Proof. Let $g \in \partial f(\bar{x})$ and define $\phi \coloneqq F + f - \langle g, \cdot\rangle$. Since $F + f$ is σ-strongly convex w.r.t. $\|\cdot\|$ and $-\langle g, \cdot \rangle$ is convex, we have that ϕ is also σ-strongly convex w.r.t. $\|\cdot\|$. Thus, by the strong convexity/smoothness duality (Theorem 3.10.2), we have that

$$\phi^* \text{ is } \tfrac{1}{\sigma}\text{-strongly smooth with respect to } \|\cdot\|_*. \tag{4.8}$$

By Theorem 3.2.7, the sum of closed convex functions is itself a closed function. Thus, $F + f$ is closed, and since $F + f$ is strongly convex, by Lemma 3.9.14 there is $\bar{y} \in \arg\min_{x \in \mathbb{E}}(F(x) + f(x))$.

[2] We do not add r_{t+1} here since we want to use bounds which depend on the subgradients of f_t, not of $f_t + r_{t+1}$. How we deal with this extra r_{t+1} term will become clear when we apply Lemma 4.3.2 to derive regret bounds in the next section. One example of a case where it is easy to deal with this term is in the classical FTRL case, where we use a static regularizer (i.e., one that does not change throughout the game). In this case, $\mathcal{R}(\boldsymbol{f}) = 0$ for any nonempty $\boldsymbol{f} \in \mathrm{Seq}(\mathcal{F})$. Thus, $H + f_t + r_{t+1} = H + f_t$ for any $t \in [T]$ since $r_t = 0$ for $t \in \{2, \dots, T+1\}$ in this case.

Assume for now that

$$\bar{x} = \nabla\phi^*(0) \qquad \text{and} \qquad \bar{y} = \nabla\phi^*(-g). \tag{4.9}$$

We will prove the above claim later. With that, since ϕ^* is $(1/\sigma)$-strongly smooth, by the definition of strong smoothness and since $\|\cdot\|_{**} = \|\cdot\|$ by Theorem 3.8.2 we have

$$\|\bar{x} - \bar{y}\| \overset{(4.9)}{=} \|\nabla\phi^*(0) - \nabla\phi^*(-g)\| \leq \frac{1}{\sigma}\|g\|_*.$$

To prove the second inequality from the statement, note that by Theorem 3.5.2 (items (iv) and (v)) together with (4.9), we have

$$\begin{aligned} \langle 0, \bar{x}\rangle &\overset{(4.9)}{=} \langle 0, \nabla\phi^*(0)\rangle \overset{\text{Thm. 3.5.2}}{=} \phi^*(0) + \phi(\nabla\phi^*(0)) \overset{(4.9)}{=} \phi^*(0) + \phi(\bar{x}) \\ &\implies F(\bar{x}) + f(\bar{x}) - \langle g, \bar{x}\rangle = \phi(\bar{x}) = -\phi^*(0) \end{aligned} \tag{4.10}$$

and

$$\begin{aligned} \langle -g, \bar{y}\rangle &\overset{(4.9)}{=} \langle -g, \nabla\phi^*(-g)\rangle \overset{\text{Thm. 3.5.2}}{=} \phi^*(-g) + \phi(\nabla\phi^*(-g)) \overset{(4.9)}{=} \phi^*(-g) + \phi(\bar{y}) \\ &\implies F(\bar{y}) + f(\bar{y}) = \phi(\bar{y}) + \langle g, \bar{y}\rangle = -\phi^*(-g). \end{aligned} \tag{4.11}$$

Moreover, (4.8) together with Lemma 3.10.1 implies

$$\phi^*(y) \leq \phi^*(x) + \langle y - x, \nabla\phi^*(x)\rangle + \frac{1}{2\sigma}\|y - x\|_*^2, \qquad \forall x, y \in \mathbb{E}. \tag{4.12}$$

Therefore, for every $u \in \mathbb{E}$,

$$\begin{aligned} &F(\bar{x}) + f(\bar{x}) - \langle g, \bar{x}\rangle - (F(u) + f(u)) \\ &\qquad \leq F(\bar{x}) + f(\bar{x}) - \langle g, \bar{x}\rangle - (F(\bar{y}) + f(\bar{y})) && \text{since } \bar{y} \in \arg\min_{x \in \mathbb{E}}(F(x) + f(x)) \\ &\qquad = -\phi^*(0) + \phi^*(-g) && \text{by (4.10) and (4.11)} \\ &\qquad \leq \langle -g, \nabla\phi^*(0)\rangle + \frac{1}{2\sigma}\|g\|_*^2 && \text{by (4.12)} \\ &\qquad = -\langle g, \bar{x}\rangle + \frac{1}{2\sigma}\|g\|_*^2 && \text{by (4.9).} \end{aligned}$$

Since the $\langle g, \bar{x}\rangle$ terms above cancel out, this yields the second inequality from the statement.

Finally, it only remains to prove (4.9). Since $\operatorname{ri}(\operatorname{dom} F) \cap \operatorname{ri}(\operatorname{dom} f)$ is nonempty, by Theorem 3.5.4 we have

$$\partial\phi(x) = \partial F(x) + \partial f(x) - g, \qquad \forall x \in \mathbb{E}. \tag{4.13}$$

Since $\bar{x}$ minimizes F, we have $0 \in \partial F(\bar{x})$ by the definition of subgradient. Thus, $g \in \partial f(\bar{x})$ together with (4.13) implies $0 \in \partial\phi(\bar{x})$. Since F, f, and $\langle g, \cdot\rangle$ are closed we have that ϕ is closed as well by Theorem 3.2.7. Thus, by Theorem 3.5.2, we have $\bar{x} \in \partial\phi^*(0)$.

Similarly, since $\bar{y}$ minimizes $F + f$, we have $0 \in \partial(F + f)(\bar{y}) = \partial F(\bar{y}) + \partial f(\bar{y})$, where the equality holds by Theorem 3.5.4. This with (4.13) yields $-g \in \partial\phi(\bar{y})$, and again by Theorem 3.5.2 we have $\bar{y} \in \partial\phi^*(-g)$ since ϕ is closed by Theorem 3.2.7.

To complete the proof of (4.9), note that since ϕ^* is strongly smooth, it is differentiable by the definition of strong smoothness. Therefore, by Theorem 3.5.5 we have $\partial\phi^*(x) = \{\nabla\phi^*(x)\}$ for every $x \in \mathbb{E}$, which completes the proof of (4.9). □

We stress here that the condition over the relative interior of the domains of the functions on the above lemma is extremely fundamental for its proof, but normally one need not worry (much) about it. As we are going to see on the next section, we will need this condition to be satisfied, for every $t \in \mathbb{N}$, by the domain of functions whose form[3] is usually the sum of enemy choices and regularizer increments up to round t with the domain of the regularizer increment r_{t+1} of round $t+1$. In many applications, such as the problems described in Section 2.5, the functions played by the enemy have all the same domain (usually $\mathbb{E}$), and the player has control over the regularizers. Therefore, we will hardly meet an OCO instance in the next sections and chapters where such conditions are not satisfied. Still, every time we need this type of condition on the domains of the functions for some result, we clearly describe it in the statement of the result. In this way, every condition needed for the results to hold, even if met by most or all of the problems we see in this text, are always clearly stated. Thus, one may keep in mind that the sole purpose of most of the stated conditions about the intersection of the relative interiors of function domains in the results of this chapter are meant to enable us to apply the above lemma.

4.4 Regret Bounds for the Adaptive FTRL Algorithm

We may finally derive useful regret bounds for AdaFTRL for a rich class of problems by using the results from the last section. Theorems 4.4.3 and 4.4.4 below both bound the regret of AdaFTRL, yet the latter holds only for specific kinds of regularizer strategies. Namely, Theorem 4.4.4 holds for proximal regularizer strategies.

Definition 4.4.1 (Proximal FTRL regularizer strategy)**.** We say that a FTRL regularizer strategy $\mathcal{R} \colon \mathcal{F} \to (-\infty, +\infty]^{\mathbb{E}}$ is **proximal** if

$$\mathrm{AdaFTRL}_{\mathcal{R}}(\langle f_1, \dots, f_{t-1}\rangle) \in \arg\min_{x \in \mathbb{E}} [\mathcal{R}(\langle f_1, \dots, f_{t-1}, f_t\rangle)](x), \qquad \forall \boldsymbol{f} \in \mathcal{F}^t, \forall t \in \mathbb{N} \setminus \{0\}.$$

In words, the point chosen by the player (using the AdaFTRL oracle) on round $t \in \mathbb{N} \setminus \{0\}$ minimizes the value of the regularizer increment of the round $t+1$. The name comes from the connection of this definition with *proximal operators*. We will briefly discuss more about proximal operators and algorithms on Section 5.3.

Before diving into the theorems and their proofs, it is worth noting that one of the main features of AdaFTRL (as of some of the other algorithms that we shall see in later chapters) is its ability to adapt as time goes on. In the regret bounds, this adaptiveness is translated through different norms for each round of the game. Therefore, throughout the remainder of the text, we often manipulate norms with indices, such as $\|\cdot\|_{(t)}$, whose dual norm we denote by $\|\cdot\|_{(t),*}$. It is worth noting that we use the parentheses to differentiate indices which enumerate different norms from indices which define p-norms, such as the ℓ_2-norm $\|\cdot\|_2$ or the ℓ_1-norm $\|\cdot\|_1$. Even though this notation may seem cumbersome and visually cluttered, it follows the notation from [48], and is one of the few we found which still makes norms visually recognizable, as opposed to naming norms with non-standard notation.

As hinted at the end of the previous section, the slightly different ways in which we apply Lemma 4.3.2 in the general and proximal cases are going to lead to the differences in the regret bounds. Since this lemma requires some assumptions on the strong convexity and the relative interiors of the effective domains of some functions, one can guess that the general and the proximal

[3] Not all results require the intersection of the relative interior of functions which are exactly of this form. Still, all such conditions on the next results are similar.

cases will need slightly different assumptions. These assumptions are encapsulated in the concept of strong regularizer strategy.

Definition 4.4.2 ($\boldsymbol{\sigma}$-strong and $\boldsymbol{\sigma}$-proximally strong FTRL regularizer strategies)**.** Let $T \in \mathbb{N}$, let $\boldsymbol{\sigma} \in \mathbb{R}^T_{++}$, let $\mathcal{C} := (X, \mathcal{F})$ be an OCO instance, let $\boldsymbol{f} \in \mathcal{F}^T$, and let $\mathcal{R}$ be an FTRL regularizer strategy for $\mathcal{C}$. Define

$$r_t := \mathcal{R}(\langle f_1, \dots, f_{t-1} \rangle) \qquad \text{for each } t \in \{1, \dots, T+1\} \text{ and}$$
$$H_t := \sum_{i=1}^{t+1} r_i + \sum_{i=1}^{t} f_i \qquad \text{for each } t \in \{0, \dots, T\}.$$

We say that $\mathcal{R}$ is **$\boldsymbol{\sigma}$-strong** for $\boldsymbol{f}$ (with respect to norms $\|\cdot\|_{(1)}, \|\cdot\|_{(2)}, \dots, \|\cdot\|_{(T)}$ on $\mathbb{E}$) if

(i) $H_{t-1} + f_t$ is σ_t-strongly convex[4] w.r.t. $\|\cdot\|_{(t)}$ for each $t \in [T]$ and

(ii) $\operatorname{ri}(\operatorname{dom} H_{t-1}) \cap \operatorname{ri}(\operatorname{dom} f_t)$ is nonempty for each $t \in [T]$.

If $\|\cdot\| := \|\cdot\|_{(1)} = \|\cdot\|_{(2)} = \cdots = \|\cdot\|_{(T)}$, we may say that $\mathcal{R}$ is $\boldsymbol{\sigma}$-strong w.r.t. $\|\cdot\|$. Likewise, we say that $\mathcal{R}$ is **$\boldsymbol{\sigma}$-proximally strong** for $\boldsymbol{f}$ (with respect to norms $\|\cdot\|_{(1)}, \|\cdot\|_{(2)}, \dots, \|\cdot\|_{(T)}$ on $\mathbb{E}$) if

(i) H_{t-1} is σ_t-strongly convex w.r.t. $\|\cdot\|_{(t)}$ for each $t \in [T]$,

(ii) $\operatorname{ri}(\operatorname{dom}(H_{t-1} + r_{t+1})) \cap \operatorname{ri}(\operatorname{dom} f_t)$ is nonempty for each $t \in [T]$, and

(iii) $\mathcal{R}$ is proximal.

If $\|\cdot\| := \|\cdot\|_{(1)} = \|\cdot\|_{(2)} = \cdots = \|\cdot\|_{(T)}$, we may say that $\mathcal{R}$ is $\boldsymbol{\sigma}$-proximally strong w.r.t. $\|\cdot\|$.

The second condition in each of the above definitions is a technical assumption to apply Lemma 4.3.2 which is usually easily satisfied. Even though it is important to know that such a condition on the effective domains is needed, in the most common OCO instances this condition will be easily satisfied. Usually, the functions from the set $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$ from where the enemy picks his functions have all the same domain. In this way, it is not hard to design regularizers which satisfy either (ii) or (ii) for the proximal case. We shall see some examples soon. The first condition on each of these definitions deals with the the strong convexity parameters $\boldsymbol{\sigma}$ of the functions, which will play a major role in the following regret bounds. Moreover, it is worth saying that even though the strong convexity assumption is over the "cumulative functions" H_t, since the functions played by the enemy are convex, $\boldsymbol{\sigma}$ will usually be determined single-handedly by the regularizer strategy.

For example, consider the OCO instance $\mathcal{C} := (\mathbb{R}^d, \mathcal{F})$, where $\mathcal{F}$ is a set of closed proper convex functions on $\mathbb{R}^d$ with effective domain equal to $\mathbb{R}^d$. Recall from Lemma 3.9.5 that the function $(1/2)\|\cdot\|_2^2$ is 1-strongly convex w.r.t. $\|\cdot\|_2$. Thus, for any $T \in \mathbb{N}$ and $\boldsymbol{f} \in \mathcal{F}$ the function $(1/2)\|\cdot\|_2^2 + \sum_{t=1}^T f_t$ is also 1-strongly convex w.r.t. $\|\cdot\|_2$ since summing convex functions to a strongly convex function preserves strong convexity. Finally, since $\operatorname{dom} f = \mathbb{R}^d$ for each $f \in \mathcal{F}$, we conclude that the FTRL regularizer strategy given by $\mathcal{R}(\boldsymbol{f}) := [\boldsymbol{f} = \langle\rangle](1/2)\|\cdot\|_2^2$ for each $\boldsymbol{f} \in \operatorname{Seq}(\mathcal{F})$ is $\mathbb{1}$-strong for any function sequence in $\operatorname{Seq}(\mathcal{F})$, where $\mathbb{1}$ is a properly sized sequence with all entries equal to 1. One example of proximally strong FTRL regularizer strategy is one with the regularizer increment on round $t \in \mathbb{N}$ of the type $x \mapsto \|x - [t > 1]x_{t-1}\|$, where x_{t-1} is the iterate from the previous round. We will look at these type of regularizers in details on Section 4.7. Let us now prove general regret bounds for the Adaptive FTRL oracle with strong and proximally strong FTRL regularizer strategies. Again, we note that the following proofs rely mainly on the lemmas from the previous section.

[4] Note that, for each $t \in [T]$, the norm $\|\cdot\|_{(t)}$ may be influenced by the regularizer increments $r_1, \dots, r_t$, i.e. the ones chosen up to round t.

Theorem 4.4.3 (General AdaFTRL Regret Bound)**.** Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that each $f \in \mathcal{F}$ is proper and closed. Let $\mathcal{R}\colon \operatorname{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ be a FTRL regularizer strategy, let $T \in \mathbb{N}$, and let ENEMY be an enemy oracle for $\mathcal{C}$. Moreover, define

$$\begin{aligned}(\boldsymbol{x}, \boldsymbol{f}) &\coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaFTRL}_{\mathcal{R}}, \mathrm{ENEMY}, T),\\ r_t &\coloneqq \mathcal{R}(\langle f_1, \dots, f_{t-1}\rangle) \qquad \text{for each } t \in \{1, \dots, T+1\},\end{aligned}$$

Finally, suppose there exists[5] $g_t \in \partial f_t(x_t)$ for each $t \in [T]$. If $\boldsymbol{\sigma} \in \mathbb{R}^T_{++}$ and $\mathcal{R}$ is $\boldsymbol{\sigma}$-strong for $\boldsymbol{f}$ w.r.t. norms $\|\cdot\|_{(1)}, \dots, \|\cdot\|_{(T)}$ on $\mathbb{E}$, then $\boldsymbol{x} \in \operatorname{Seq}(X)$ and

$$\operatorname{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, u) \leq \sum_{t=1}^{T}(r_t(u) - r_t(x_t)) + \frac{1}{2}\sum_{t=1}^{T}\frac{1}{\sigma_t}\|g_t\|^2_{(t),*}.$$

Proof. Define

$$H_t \coloneqq \sum_{i=1}^{t+1} r_i + \sum_{i=1}^{t} f_i \qquad \text{for each } t \in \{0, \dots, T\}.$$

First of all, since $\inf_{x \in \mathbb{E}} H_t(x) = \inf_{x \in \mathbb{E}}(\sum_{i=1}^{t+1} r_i(x) + \sum_{i=1}^{t} f_i(x))$ is attained for every $t \in \{0, \dots, T\}$, we have that $\mathrm{AdaFTRL}_{\mathcal{R}}(\langle f_1, \dots, f_t\rangle)$ is properly defined for each $t \in \{0, \dots, T\}$. Moreover, since $\operatorname{dom} r_1 \subseteq X$, we have $\boldsymbol{x} \in \operatorname{Seq}(X)$ by the definition of $\mathrm{AdaFTRL}_{\mathcal{R}}$.

Define $x_0 \coloneqq x_1$ and $x_{T+1} \coloneqq \mathrm{AdaFTRL}_{\mathcal{R}}(\langle f_1, \dots, f_T\rangle)$. By the Strong FTRL Lemma (Lemma 4.3.1), we have

$$\begin{aligned}\operatorname{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, u) &\leq \sum_{t=1}^{T+1}(r_t(u) - r_t(x_{t-1})) + \sum_{t=1}^{T}(H_t(x_t) - H_t(x_{t+1}))\\ &= -r_1(x_0) + \sum_{t=1}^{T+1} r_t(u) + \sum_{t=1}^{T}(H_t(x_t) - H_t(x_{t+1}) - r_{t+1}(x_t)) \qquad (4.14)\\ &= -r_1(x_1) + \sum_{t=0}^{T} r_{t+1}(u) + \sum_{t=1}^{T}(H_t(x_t) - H_t(x_{t+1}) - r_{t+1}(x_t)).\end{aligned}$$

Let $t \in [T]$. By assumption, $\operatorname{ri}(\operatorname{dom} H_{t-1}) \cap \operatorname{ri}(\operatorname{dom} f_t)$ is nonempty and $H_{t-1} + f_t$ is σ_t-strongly convex w.r.t. $\|\cdot\|_{(t)}$. Thus, since $x_t \in \arg\min_{x \in \mathbb{E}} H_{t-1}(x)$, by Lemma 4.3.2 with $F \coloneqq H_{t-1}$ (which is closed since the sum of closed functions is closed by Theorem 3.2.7) and $f \coloneqq f_t$ we have

$$\begin{aligned}H_t(x_t) - H_t(x_{t+1}) - r_{t+1}(x_t) &= H_{t-1}(x_t) + f_t(x_t) + r_{t+1}(x_t) - H_{t-1}(x_{t+1}) - f_t(x_{t+1})\\ &\quad - r_{t+1}(x_{t+1}) - r_{t+1}(x_t)\\ &\leq \frac{1}{2\sigma_t}\|g_t\|^2_{(t),*} - r_{t+1}(x_{t+1}).\end{aligned}$$

Plugging the above inequality for every $t \in [T]$ into (4.14) yields

$$\begin{aligned}\operatorname{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, u) &\leq \sum_{t=0}^{T}(r_{t+1}(u) - r_{t+1}(x_{t+1})) + \frac{1}{2}\sum_{t=1}^{T}\frac{1}{\sigma_t}\|g_t\|^2_{(t),*}\\ &= \sum_{t=1}^{T+1}(r_t(u) - r_t(x_t)) + \frac{1}{2}\sum_{t=1}^{T}\frac{1}{\sigma_t}\|g_t\|^2_{(t),*}.\end{aligned} \qquad (4.15)$$

[5] From Theorem 3.5.1, we know that a convex function is always subdifferentiable on the relative interior of its domain. Thus, it is usually hard to find a case where the functions played by the enemy are not subdifferentiable at the iterates from the player.

We are almost done: there is still an extra term in the first summation when compared to the bound on the statement. Note, however, that if we set $r_{T+1} \coloneqq 0$, then the iterates delivered by the $\mathrm{AdaFTRL}_{\mathcal{R}}$ oracle over the sub-sequences of $\boldsymbol{f}$ would still be $x_1, \dots, x_T$. We can do such a modification formally by defining $\mathcal{R}'$ by $\mathcal{R}'(\boldsymbol{f}) \coloneqq 0$, and by making it equal to $\mathcal{R}$ on $\mathrm{Seq}(\mathcal{F}) \setminus \{\boldsymbol{f}\}$. In this way, we have $x_t = \mathrm{AdaFTRL}_{\mathcal{R}'}(\langle f_1, \dots, f_{t-1}\rangle)$ for each $t \in [T]$, as argued. Therefore,

$$\begin{aligned}\mathrm{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, u) &= \mathrm{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}'}, \boldsymbol{f}, u)\\ &\overset{(4.15)}{\leq} \sum_{t=1}^{T}(r_t(u) - r_t(x_t)) + \frac{1}{2}\sum_{t=1}^{T}\frac{1}{\sigma_t}\|g_t\|_{(t),*}^2. \qquad \square\end{aligned}$$

Theorem 4.4.4 (Proximal AdaFTRL Regret Bound)**.** Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that each $f \in \mathcal{F}$ is proper and closed. Let $\mathcal{R}\colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ be a proximal FTRL regularizer strategy, let $T \in \mathbb{N}$, and let ENEMY be an enemy oracle for $\mathcal{C}$. Moreover, define

$$\begin{aligned}(\boldsymbol{x}, \boldsymbol{f}) &\coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaFTRL}_{\mathcal{R}}, \mathrm{ENEMY}, T),\\ r_t &\coloneqq \mathcal{R}(\langle f_1, \dots, f_{t-1}\rangle) \qquad \text{for each } t \in \{1, \dots, T+1\},\end{aligned}$$

Finally, suppose there exists $g_t \in \partial f_t(x_t)$ for each $t \in [T]$. If $\boldsymbol{\sigma} \in \mathbb{R}^T_{++}$ and $\mathcal{R}$ is $\boldsymbol{\sigma}$-proximally strong for $\boldsymbol{f}$ w.r.t. norms $\|\cdot\|_{(1)}, \dots, \|\cdot\|_{(T)}$ on $\mathbb{E}$, then $\boldsymbol{x} \in \mathrm{Seq}(X)$ and

$$\mathrm{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, u) \leq \sum_{t=0}^{T}(r_{t+1}(u) - r_{t+1}(x_t)) + \frac{1}{2}\sum_{t=1}^{T}\frac{1}{\sigma_{t+1}}\|g_t\|_{(t+1),*}^2.$$

Proof. Define

$$H_t \coloneqq \sum_{i=1}^{t+1} r_i + \sum_{i=1}^{t} f_i \qquad \text{for each } t \in \{0, \dots, T\}.$$

First of all, since $\inf_{x\in\mathbb{E}} H_t(x) = \inf_{x\in\mathbb{E}}(\sum_{i=1}^{t+1} r_i(x) + \sum_{i=1}^{t} f_i(x))$ is attained for every $t \in \{0, \dots, T\}$, we have that $\mathrm{AdaFTRL}_{\mathcal{R}}(\langle f_1, \dots, f_t\rangle)$ is properly defined for each $t \in \{0, \dots, T\}$. Moreover, since $\mathrm{dom}\, r_1 \subseteq X$, we have $\boldsymbol{x} \in \mathrm{Seq}(X)$ by the definition of $\mathrm{AdaFTRL}_{\mathcal{R}}$.

Define $x_0 \coloneqq x_1$ and $x_{T+1} \coloneqq \mathrm{AdaFTRL}(\boldsymbol{f})$. By the Strong FTRL Lemma (Lemma 4.3.1), we have

$$\mathrm{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, u) \leq \sum_{t=0}^{T}(r_{t+1}(u) - r_{t+1}(x_t)) + \sum_{t=1}^{T}(H_t(x_t) - H_t(x_{t+1})). \tag{4.16}$$

Let[6] $t \in [T]$. By assumption, $\mathrm{ri}(\mathrm{dom}(H_{t-1} + r_{t+1})) \cap \mathrm{ri}(\mathrm{dom}\, f_t)$ is nonempty and $H_t = H_{t-1} + r_{t+1} + f_t$ is σ_{t+1}-strongly convex w.r.t. $\|\cdot\|_{(t+1)}$. Moreover, we have $x_t \in \arg\min_{x\in\mathbb{E}} H_{t-1}(x)$ and $x_t \in \arg\min_{x\in\mathbb{E}} r_{t+1}(x)$ (recall that $\mathcal{R}$ is proximal). Thus, $x_t \in \arg\min(H_{t-1}(x) + r_{t+1}(x))$. Finally, we can apply Lemma 4.3.2 with $F \coloneqq H_{t-1} + r_{t+1}$ (which is closed since the sum of closed functions is closed by Theorem 3.2.7) and $f \coloneqq f_t$, which yields

$$H_t(x_t) - H_t(x_{t+1}) = F(x_t) + f(x_t) - F(x_{t+1}) - f(x_{t+1}) \leq \frac{1}{2\sigma_{t+1}}\|g_t\|_{(t+1),*}^2, \qquad \forall g_t \in \partial f_t(x_t).$$

Plugging the above inequality for every $t \in [T]$ into (4.16) yields the bound from the statement. $\square$

[6] Up to this point, the proof is identical to the one from Theorem 4.4.3. The main differences appear from now on, which is when we use Lemma 4.3.2.

Let $f_1, \dots, f_T \in \mathcal{F}$ for some $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$, and let $\mathcal{R} \colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$. When applying $\mathrm{AdaFTRL}_{\mathcal{R}}$, we usually choose regularizer functions which are strongly convex. That is, we choose $\mathcal{R}$ such that the sum of regularizer increments $\sum_{i=1}^{t} r_i$ is strongly convex for each $t \in [T]$, where $r_i \coloneqq \mathcal{R}(\langle f_1, \dots, f_{i-1} \rangle)$ for each $i \in [T]$. However, on the regret bounds stated above, we make assumptions on the strong convexity of the functions $H_t \coloneqq \sum_{i=1}^{t+1} r_i + \sum_{i=1}^{t} f_i$. The reason for that is to capture the case where the functions f_t themselves are strongly convex, sometimes making AdaFTRL have low-regret guarantees without any regularization at all.

Let us now compare both of these theorems. Note that they are very similar, with the main difference appearing on the indices of the norms on the second summation on each of the bounds. Let us try to understand better what are the implications of these "off-by-one" differences. Let $f_1, \dots, f_T$ and $r_1, \dots, r_{T+1}$ be as in Theorem 4.4.3, and let $t \in \{1, \dots, T-1\}$. Recall that, at round t, the player and the enemy choose, respectively, x_t and f_t simultaneously. Since x_t as defined by the AdaFTRL oracle is the first iterate which depends on r_t, it is at round t that the player has to choose the regularizer r_t.

Note that on both theorems the norm $\|\cdot\|_{(t)}$ is related to the regularizers $r_1, \dots, r_t$. Since these regularizer increments are up to the player to choose, then the player partially[7] chooses the parameters of strong convexity and the norms $\|\cdot\|_{(t)}$. With that in mind, the player will probably want to choose a regularizer strategy which yields norms and parameters that give better guarantees on the regret. That is, at round t the player will try to come up with a regularizer increment r_t strongly convex w.r.t. a norm $\|\cdot\|_{(t)}$ which makes small the terms measured with its dual norm. Note, however, that on the general case (Theorem 4.4.3) we measure the norm of the subgradients of f_t, which the player does not know *until round* $t+1$, with the norm $\|\cdot\|_{(t)}$, which the player has control over only *up to round* t. That is, the player has to pick a regularizer on round t aiming to control the norm of the subgradient of the function she will get to know only on the *next* round, i.e., round $t+1$. In contrast, on Theorem 4.4.4 the subgradients of f_t are measured with the norm $\|\cdot\|_{(t+1)}$, which the player has some control over up to round $t+1$, since $\|\cdot\|_{(t+1)}$ on Theorem 4.4.4 depends on $r_1, \dots, r_{t+1}$, and r_{t+1} is chosen at round $t+1$ by the player. Thus, on the case of a proximal regularizer strategy, the player can craft the norm $\|\cdot\|_{(t+1)}$ with knowledge of f_t whose subgradient norms she wants to control in order to get good regret guarantees. The implications of this, as we are going to see later in applications, is that AdaFTRL algorithms with proximal regularizer strategies may need less prior information about the functions the enemy will play in order to get good regret bounds.

With these bounds, one can already expect the bounds on the regret of FTRL algorithms to depend heavily on the (dual) norms of the subgradients of the functions given by the enemy. Thus, for these bounds to be meaningful, we may need to assume a bound on the norms of the subgradients. Although such an assumption may seem artificial at first glance, it happens to be somewhat natural due to an interesting connection with Lipschitz continuity (see Theorem 3.8.4), the latter being a traditional hypothesis in convergence proofs of many optimization algorithms. One may note at least one of the reasons why Lipschitz continuity may be a sensible assumption: if the function can change drastically between two close points, intuitively, the algorithm will have a harder time optimizing over this function. Thus, if the functions played by the enemy are ρ-Lipschitz continuous, and this is usually the case in the cases studied in the next sections, most of the subgradients of the functions have dual norm bounded by ρ.

[7] We say "partially" here because the strong convexity parameter depends also on the functions played by the enemy. However, if the regularizer increments are strongly convex (which is usually the case), summing convex functions to these regularizers preserves the strong convexity property. Thus, one may ignore the "partially" in this sentence to build intuition.

4.5 The Classical FTRL Algorithm

Using the tools from the previous sections, let us now analyze the performance of the FTRL algorithm as define in Section 4.1. As expected, the classical FTRL algorithm is a special case of AdaFTRL: for any function $R\colon \mathbb{E} \to (-\infty, +\infty]$ we have $\mathrm{FTRL}_R = \mathrm{AdaFTRL}_{\mathcal{R}}$ where $\mathcal{R}$ is given by $\mathcal{R}(\boldsymbol{f}) \coloneqq [\boldsymbol{f} = \langle\rangle]R$ for every $\boldsymbol{f} \in \mathrm{Seq}((-\infty,+\infty]^{\mathbb{E}})$. A natural choice for a regularizer function $R\colon \mathbb{E} \to (-\infty, +\infty]$ is the squared ℓ_2-norm, that is, $R \coloneqq \frac{1}{2}\|\cdot\|_2^2$. As we have briefly seen at the end of Section 4.2, in the unconstrained case against linear functions, FTRL_R boils down to steps in the direction of minus gradient at each round. Still, we do not yet know in which OCO instances and with which kind of regularizers FTRL performs well, that is, attains sublinear regret.

Looking at the bounds given by the theorems from the last section, we know that the dual norms of the subgradients play a major role our FTRL regret guarantees. As discussed at the end of the last section, this naturally directs our attention towards OCO instances with Lipschitz continuous functions. Consider the OCO instance for the randomized experts problem $\mathcal{C} \coloneqq (\Delta_E, \mathcal{F})$, for example, and set $d \coloneqq |E|$. As we are going to see later on this section, the functions from $\mathcal{F}$ are $\sqrt{d}$-Lipschitz continuous w.r.t. $\|\cdot\|_2$, and this together with the theorems from the last section yield a $\sqrt{dT}$ bound on the regret. Besides, later we will see that choosing a different regularizer which is strongly convex w.r.t. the ℓ_1-norm will yield an regret bound which has an exponentially better dependence on the dimension.

Naturally, the functions $R\colon \mathbb{E} \to (-\infty, +\infty]$ we use in FTRL_R will usually be classical FTRL regularizers for the OCO instances in which we are going to use the FTRL oracle. Additionally, we want FTRL regularizers which allow us to apply Lemma 4.3.2 so that we can use the regret bounds from Section 4.4. Thus, in a way similar to the adaptive case, we encapsulate the usually required assumptions on the regularizer function R for FTRL_R for the application of Lemma 4.3.2 in the concept of *σ-strong classical regularizers.*

Definition 4.5.1 (σ-strong classical FTRL regularizer)**.** Let $\sigma \in \mathbb{R}$ be such that $\sigma > 0$, and let $\|\cdot\|$ be a norm on $\mathbb{E}$. We say that a classical FTRL regularizer R for $\mathcal{C}$ is **σ-strong** for $\mathcal{C}$ if

(i) R is σ-strongly convex w.r.t. $\|\cdot\|$, and

(ii) for any $\boldsymbol{f} \in \mathcal{F}^T$, we have that $\mathrm{ri}(\mathrm{dom}(R + \sum_{t=1}^{T-1} f_t)) \cap \mathrm{ri}(\mathrm{dom}\, f_T)$ is nonempty.

Again, one may note that the notion of strong FTRL regularizer is closely related to the notion of strong regularizer strategy. If a classical FTRL regularizer R is σ-strong for $\mathcal{C}$, then, the regularizer strategy $\mathcal{R}\colon\ \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ given by $\mathcal{R}(\boldsymbol{f}) \coloneqq [\boldsymbol{f} = \langle\rangle]R$ for each $\boldsymbol{f} \in \mathrm{Seq}(\mathcal{F})$ is a $\boldsymbol{\sigma}$-strong regularizer strategy for any $\boldsymbol{f} \in \mathrm{Seq}(\mathcal{F})$, where $\boldsymbol{\sigma} \in \mathrm{Seq}(\mathbb{R})$ is a sequence of appropriate size with each entry equal to σ. Note, however, that we require σ-strong classical FTRL regularizers to be strongly convex. Thus, this definition does not cover cases in which the player could use non-strongly convex regularizers (such as in cases against enemies which play strongly convex functions). Still, it is insightful to look at this simpler and quite common class of FTRL regularizers.

Moreover, it is worth noticing that the notion of strong classical FTRL regularizer is defined with respect to all possible sequences of functions the enemy can play, while the concept of strong FTRL regularizer strategy was defined only with respect to a single sequence. The reason for this difference is that, in future chapters, some FTRL regularizer strategies will be $\boldsymbol{\sigma}$-strong with varying values of $\boldsymbol{\sigma}$ depending on the functions played by the enemy, that is, the regularizer strategies will be adaptive. Since the classical FTRL regularizer is not adaptive by definition, we do not need to specify the parameter of strong convexity for each of the possible function sequences separately.

Before deriving a regret bound for the classical FTRL oracle, let us prove that if R is a σ-strong classical FTRL regularizer for some OCO instance $\mathcal{C}$, then for any positive constant $\mu \in \mathbb{R}_{++}$, we have that μR is a $(\mu\sigma)$-strong classical FTRL regularizer strategy. This will be useful to obtain optimal constants in the final regret bounds for the classical FTRL. The only condition from the definition of classical FTRL regularizer which might not hold for a positive multiple of the regularizer is condition (4.4.iii). In general, it is not clear if multiplying the regularizer by a positive constant affects the attainability of the infimum in some case of (4.4.iii). Fortunately, if R is strongly convex, the infimum will still be attained due to Lemma 3.9.14, which states that the infimum of closed strongly convex functions is always attained.

Lemma 4.5.2. Let $\mathcal{C} := (X, \mathcal{F})$ be an OCO instance such that each $f \in \mathcal{F}$ is proper and closed, let $\sigma \in \mathbb{R}_{++}$, and let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a σ-strong classical FTRL regularizer for $\mathcal{C}$. Then, for any $\mu \in \mathbb{R}_{++}$ we have that μR is a $(\mu\sigma)$-strong FTRL regularizer strategy for $\mathcal{C}$.

Proof. Let $\mu \in \mathbb{R}_{++}$ and set $R' := \mu R$. Let us first show that R' is a classical FTRL regularizer for $\mathcal{C}$. Since R is closed, proper, and convex (property (4.4.i) of a classical FTRL regularizer strategy), then so is R' since $\mu > 0$. Moreover, $\operatorname{dom} R' = \operatorname{dom} R \subseteq X$, that is, R' satisfies property (4.4.ii). Let $T \in \mathbb{N}$ and let $\boldsymbol{f} \in \mathcal{F}^T$. Note that $F := R' + \sum_{t=1}^T f_t$ is closed since the sum of convex and closed functions is also closed by Theorem 3.2.7, and it is proper since $R + \sum_{t=1}^T f_t$ is proper and since $\operatorname{dom} F = \operatorname{dom}(R + \sum_{t=1}^T f_t)$. Finally, since R is strongly convex and μ is positive, F is strongly convex. Thus, by Lemma 3.9.14 we have that $\inf_{x\in\mathbb{E}} F(x)$ is attained, that is, R' satisfies condition (4.4.iii) from the definition of classical FTRL regularizer or $\mathcal{C}$.

Let us now show that R' is $(\mu\sigma)$-strong. Since μ is positive, it is clear that R' is $(\mu\sigma)$-strongly convex. Moreover, since $\operatorname{dom} R' = \operatorname{dom} R$, we have that R' clearly satisfies condition (ii) from the definition of $(\mu\sigma)$-strongness for a classical FTRL regularizer of $\mathcal{C}$. □

Corollary 4.5.3 (Derived from Theorem 4.4.3)**.** Let $\mathcal{C} := (X, \mathcal{F})$ be an OCO instance such that each $f \in \mathcal{F}$ is proper and closed. Let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a σ-strong classical FTRL regularizer for $\mathcal{C}$. Let $T \in \mathbb{N}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and define

$$(\boldsymbol{x}, \boldsymbol{f}) := \mathrm{OCO}_{\mathcal{C}}(\mathrm{FTRL}_R, \mathrm{ENEMY}, T).$$

Finally, let $g_t \in \partial f_t(x_t)$ for each $t \in [T]$. Then $\boldsymbol{x} \in \mathrm{Seq}(X)$ and

$$\mathrm{Regret}(\mathrm{FTRL}_R, \boldsymbol{f}, u) \leq R(u) - \min_{x\in\mathbb{E}} R(x) + \frac{1}{2\sigma}\sum_{t=1}^{T} \|g_t\|_*^2, \qquad \forall u \in \mathbb{E}. \tag{4.17}$$

In particular, if every function in $\mathcal{F}$ is ρ-Lipschitz continuous w.r.t. $\|\cdot\|$ on a convex set $D \supseteq X$ with nonempty interior[8] and there is $\theta \in \mathbb{R}_{++}$ such that[9] $\theta \geq \sup\{\, R(x) - R(y) : x \in X, y \in X \cap \operatorname{dom} R\}$, then

$$\mathrm{Regret}_T(\mathrm{FTRL}_{R'}, \mathrm{ENEMY}, X) \leq \rho\sqrt{\frac{2\theta T}{\sigma}},$$

where $R' := \big(\rho\sqrt{T}/\sqrt{2\sigma\theta}\big)R$.

Proof. Note that $\mathrm{FTRL}_R = \mathrm{AdaFTRL}_{\mathcal{R}}$ where $\mathcal{R}$ is given by $\mathcal{R}(\boldsymbol{f}) := [\boldsymbol{f} = \langle\rangle]R$ for every $\boldsymbol{f} \in \mathrm{Seq}((-\infty, +\infty]^{\mathbb{E}})$. Moreover, since R is a σ-strong FTRL regularizer, $\mathcal{R}$ is a $\boldsymbol{\sigma}$-strong regularizer strategy for $\boldsymbol{f}$ w.r.t. $\|\cdot\|$, where $\boldsymbol{\sigma} := \langle \sigma, \ldots, \sigma\rangle \in \mathbb{R}^T$. Therefore, the first inequality is a direct application of Theorem 4.4.3 together with the fact that $R(x_1) = \min_{x\in\mathbb{E}} R(x)$.

[8] Nonempty interior is need only for us to apply Theorem 3.8.4 to bound the dual norms of the subgradients.

[9] One may think of this value as the diameter of the set X measured through the lens of R.

If each $f \in \mathcal{F}$ is ρ-Lipschitz continuous w.r.t. $\|\cdot\|$ on a convex set $D \supseteq X$ with nonempty interior, then by Theorem 3.8.4 we have that, for each $f \in \mathcal{F}$ and $x \in X$, there is $g \in \partial f(x)$ such that $\|g\|_* \leq \rho$. Using such subgradients with bounded dual norm in (4.17) and the fact that $\min_{x \in \mathbb{E}} R(x) = \min_{x \in X} R(x)$ yields

$$\operatorname{Regret}_T(\mathrm{FTRL}_R, \mathrm{ENEMY}, u) \leq R(u) - \min_{x \in X} R(x) + \frac{T\rho^2}{2\sigma}. \tag{4.18}$$

Moreover, suppose there is $\theta \in \mathbb{R}_{++}$ such that $\theta \geq \sup\{\, R(x) - R(y) : x \in X, y \in X \cap \operatorname{dom} R\}$, and define

$$R' := \frac{\rho\sqrt{T}}{\sqrt{2\sigma\theta}} R.$$

Note that R' is a $(\rho\sqrt{\sigma T}/\sqrt{2\theta})$-strong classical FTRL regularizer by Lemma 4.5.2. Thus, plugging R' into the above inequality yields, for every $u \in X$,

$$\operatorname{Regret}_T(\mathrm{FTRL}_{R'}, \mathrm{ENEMY}, u) \leq \frac{\rho\sqrt{T}}{\sqrt{2\sigma\theta}}(R(u) - \min_{x \in X} R(x)) + \frac{\rho\sqrt{\theta T}}{\sqrt{2\sigma}} \leq \frac{\rho\sqrt{\theta T}}{\sqrt{2\sigma}} + \frac{\rho\sqrt{\theta T}}{\sqrt{2\sigma}} = \rho\sqrt{\frac{2\theta T}{\sigma}},$$

where in the second inequality we took the supremum over $u \in X$. □

Let us look at the problem of prediction with expert advice. As we have seen on Chapter 2 (namely, on Proposition 2.6.2), in order to obtain a low-expected-regret randomized player oracle for the experts' problem instance (A^E, Y, A, L), it suffices to devise a player oracle for the OCO instance $\mathcal{C} := (\Delta_E, \mathcal{F})$, where the set $\mathcal{F}$ is given by $\mathcal{F} := \{\, p \in \mathbb{R}^E \mapsto y^\mathsf{T} p : y \in [-1,1]^E\}$. The next proposition shows that FTRL with ℓ_2-regularization in any instance of the randomized experts has low expected regret.

Proposition 4.5.4. Define the OCO instance $\mathcal{C} := (\Delta_E, \mathcal{F})$, where E is a finite set and $\mathcal{F} := \{\, p \in \mathbb{R}^E \mapsto y^\mathsf{T} p : y \in [-1,1]^E\}$. Set $d := |E|$ and define $R := \frac{1}{2}\|\cdot\|_2^2 + \delta(\cdot \,|\, \Delta_E)$. Moreover, let $T \in \mathbb{N}$ and define $R' := \sqrt{dT}R$. Then, for every enemy oracle ENEMY for $\mathcal{C}$ we have

$$\operatorname{Regret}_T(\mathrm{FTRL}_R, \mathrm{ENEMY}, \Delta_E) \leq \sqrt{dT}.$$

Proof. First, let us show that

$$R \text{ is a 1-strong FTRL regularizer for } \mathcal{C} \text{ w.r.t. } \|\cdot\|_2. \tag{4.19}$$

We have that $\frac{1}{2}\|\cdot\|$ is closed (in fact, continuous) and that $\delta(\cdot \,|\, \Delta_E)$ is closed, the latter since Δ_E is closed. Thus, R is a sum of closed functions and, hence, closed by Theorem 3.2.7, which means that R satisfies condition (4.4.i) of a FTRL regularizer. Moreover, clearly $\operatorname{dom} R \subseteq \Delta_E$, that is, R satisfies condition (4.4.ii). Let $T' \in \mathbb{N}$ and $\boldsymbol{f} \in \mathcal{F}^{T'}$. Let us show that $\inf_{x \in \mathbb{R}^d}(R(x) + \sum_{t=1}^{T'} f_t(x))$ is attained. First, notice that since the ℓ_2-norm is induced by the euclidean inner product, by Lemma 3.9.5 we know that $\frac{1}{2}\|\cdot\|_2^2$ is 1-strongly convex on $\mathbb{R}^E$ w.r.t. $\|\cdot\|_2$, which implies that so is $\frac{1}{2}\|\cdot\|_2^2 + \delta(\cdot \,|\, X) = R$. Therefore, $R + \sum_{t=1}^{T'} f_t$ is also strongly convex. It is also proper and closed, the latter by Theorem 3.2.7 since it is the sum of closed functions. Thus, by Lemma 3.9.14 we have that the infimum of $R + \sum_{t=1}^{T'} f_t$ over $\mathbb{R}^E$ is attained. Therefore, R satisfies condition (4.4.iii), and we conclude that R is a classical FTRL regularizer. To see that R is a 1-strong FTRL regularizer, note first that R is 1-strongly convex w.r.t. $\|\cdot\|_2$ again by Lemma 3.9.5. Finally, since every function in $\mathcal{F}$ is linear, we have that $\operatorname{dom} f$ is the entire space for any $f \in \mathcal{F}$. Since R is proper, this implies that

$\mathrm{ri}(\mathrm{dom}(R + \sum_{t=1}^{T'-1} f_t)) \cap \mathrm{ri}(\mathrm{dom}\, f_{T'})$ is nonempty for every $\boldsymbol{f} \in \mathcal{F}^{T'}$ and any $T' \in \mathbb{N}$. We conclude that R is 1-strong w.r.t. $\|\cdot\|_2$, which proves (4.19).

Let show now that

$$\text{every function in } \mathcal{F} \text{ is } \sqrt{d}\text{ -Lipschitz continuous on } \mathbb{R}^E \text{ w.r.t. } \|\cdot\|_2. \tag{4.20}$$

Let $y \in [-1,1]^E$ and define $f_y(x) := y^\mathsf{T} x$ for every $x \in [-1,1]^E$. By the definition of dual norm, for every $u, v \in \mathbb{R}^E$ and for every norm $\|\cdot\|$ on $\mathbb{R}^E$ we have

$$|f_y(u) - f_y(v)| = |y^\mathsf{T}(u - v)| \leq \|y\|_* \|u - v\|.$$

Since the ℓ_2-norm is self-dual and since $\|y\|_2 \leq \sqrt{d}$ for every $y \in [-1,1]^E$, from the above inequality we conclude that every function in $\mathcal{F}$ is $\sqrt{d}$-Lipschitz continuous w.r.t. $\|\cdot\|_2$ on $\mathbb{R}^E$, which proves (4.20). Finally, let us show that

$$\sup_{x,y \in \Delta_E} (R(x) - R(y)) \leq \frac{1}{2}. \tag{4.21}$$

Indeed, note that

$$\sup_{x \in \Delta_E} R(x) = \frac{1}{2} \sup_{x \in \Delta_E} x^\mathsf{T} x \leq \frac{1}{2} \sup_{x \in \Delta_E} \mathbb{1}^\mathsf{T} x = \frac{1}{2}.$$

This together with the fact that $R(x) \geq 0$ for any $x \in \mathbb{R}^E$ proves (4.21). Since by definition we have

$$R' = \sqrt{dT} R = \frac{\rho\sqrt{T}}{\sqrt{2\theta}} R,$$

where $\rho := \sqrt{d}$ is the Lipschitz constant from (4.20) and $\theta := \frac{1}{2}$ is from (4.21), by Corollary 4.5.3 we have, for every enemy oracle ENEMY for $\mathcal{C}$,

$$\mathrm{Regret}_T(\mathrm{FTRL}_{R'}, \mathrm{ENEMY}, \Delta_E) \leq \sqrt{dT}. \tag{4.22}$$

□

It is natural to ask if this regret bound is optimal, especially since our choice of regularizer was mainly due to the self-duality of the ℓ_2-norm for the sake of simplicity. It turns out that the dependence on T on the bound given by Corollary 4.5.3 is optimal: there is a class of OCO instances of the type $(X, \mathcal{F}')$, with each function in $\mathcal{F}'$ being Lipschitz continuous, such that the worst-case regret in T rounds of any player oracle in such instance is no better than $\Omega(\sqrt{T})$, where the constants hidden by the asymptotic notation may depend on other parameters of the instance, such as the dimension [2]. The fact that such an intuitive algorithm already attains optimal regret asymptotically (w.r.t. T) is surprising. Still, this lower bound says nothing about the dependence on the dimension, which can be high, even more so in machine learning applications.

However, a smarter choice of regularizer already improves exponentially the dependence of the regret bound for FTRL on the number of the experts, which can be seen as the dimension of the problem.

Proposition 4.5.5. Define the OCO instance $\mathcal{C} := (\Delta_E, \mathcal{F})$, where E is a finite set and $\mathcal{F} := \{\, p \in \mathbb{R}^E \mapsto y^\mathsf{T} p : y \in [-1,1]^E \}$. Set $d := |E|$, define $R(x) := \sum_{i \in E} [x_i \neq 0] x_i \ln x_i + \delta(x \,|\, \Delta_E)$ for every $x \in \mathbb{R}^E$, let $T \in \mathbb{N}$, and set $R' := (\sqrt{T/(2\ln d)})R$. Then, for every enemy oracle ENEMY for $\mathcal{C}$ we have

$$\mathrm{Regret}_T(\mathrm{FTRL}_{R'}, \mathrm{ENEMY}, \Delta_E) \leq \sqrt{2(\ln d)T}.$$

Proof. First, let us show that

$$R \text{ is a 1-strong FTRL regularizer for } \mathcal{C} \text{ w.r.t. } \|\cdot\|_1. \tag{4.23}$$

By Lemma 3.9.10 and since Δ_E is closed, we know that R is proper, closed, and convex, that is, it satisfies condition (4.4.i) from the definition of FTRL regularizer. Moreover, we clearly have $\operatorname{dom} R \subseteq \Delta_E$, which means that R satisfies condition (4.4.ii). To show that (4.4.iii) holds, let $T' \in \mathbb{N}$ and let $\boldsymbol{f} \in \mathcal{F}^{T'}$. Since each function in $\mathcal{F}$ is closed, we have that $F := R + \sum_{t=1}^{T'} f_t$ is the sum of closed functions and, thus, closed by Theorem 3.2.7. Moreover, by Lemma 3.9.10 we know that R, and thus F, are strongly convex. Finally, by Lemma 3.9.14 we have that $\inf_{x \in \mathbb{R}^d} F(x)$ is attained, which proves that R is a classical FTRL regularizer for $\mathcal{C}$. Let us prove that it is a 1-strong regularizer for $\mathcal{C}$ w.r.t. the ℓ_1-norm. Indeed, by Lemma 3.9.10 one more time we know that R is 1-strongly convex w.r.t. $\|\cdot\|_1$. Additionally, by the definition of $\mathcal{F}$ we have $\operatorname{dom} f_t = \mathbb{R}^d$ for any $t \in [T']$. Since R is proper, this implies that $\operatorname{ri}(\operatorname{dom}(R + \sum_{i=1}^{t-1} f_i)) \cap \operatorname{ri}(\operatorname{dom} f_t)$ is nonempty for every $t \in [T']$. This completes the proof of (4.23).

Note now that, since the dual norm of $\|\cdot\|_1$ is $\|\cdot\|_\infty$, by Hölder's inequality we have, for every $y \in [-1,1]^E$ and every $u, v \in \mathbb{R}^E$,

$$|y^\mathsf{T} u - y^\mathsf{T} v| = |y^\mathsf{T}(u - v)| \leq \|y\|_\infty \|u - v\|_1 \leq \|u - v\|_1.$$

Thus, we conclude that every function in $\mathcal{F}$ is 1-Lipschitz continuous w.r.t. $\|\cdot\|_1$ on $\mathbb{R}^E$. To conclude, let us show that

$$\sup_{x,y \in \Delta_E} (R(x) - R(y)) \leq \ln d. \tag{4.24}$$

First, since $[\alpha > 0]\alpha \ln \alpha \leq 0$ for every $\alpha \in [0,1]$, we have $\sup_{x \in \Delta_E} R(x) \leq 0$. Thus, we need only show that $\inf_{y \in \Delta_E} R(y)$ is attained by $d^{-1}\mathbb{1}$. Indeed, note that for every $x \in \Delta_E$ we have

$$\begin{aligned} -\nabla R(d^{-1}\mathbb{1})^\mathsf{T}(x - d^{-1}\mathbb{1}) &= -\Big(\mathbb{1} + \sum_{i \in E} e_i \ln d^{-1}\Big)^\mathsf{T}(x - d^{-1}\mathbb{1}) \\ &= -(\mathbb{1} - (\ln d)\mathbb{1})^\mathsf{T}(x - d^{-1}\mathbb{1}) \\ &= (1 - \ln d)(1 - 1) = 0. \end{aligned}$$

That is, $-\nabla R(d^{-1}\mathbb{1}) \in N_{\Delta_E}(d^{-1}\mathbb{1})$. By the optimality conditions from Theorem 3.6.2 we conclude that $\inf_{y \in \Delta_E} R(y) = R(d^{-1}\mathbb{1}) = -\ln d$, which proves (4.24). Since

$$R' = \sqrt{\frac{T}{2\ln d}} = \frac{\rho\sqrt{T}}{\sqrt{2\theta}} R,$$

where $\rho := 1$ and $\theta := \ln d$, by Corollary 4.5.3 we have, for every enemy oracle ENEMY for $\mathcal{C}$,

$$\operatorname{Regret}_T(\mathrm{FTRL}_{R'}, \mathrm{ENEMY}, \Delta_E) \leq \sqrt{2(\ln d)T}. \qquad \square$$

It is interesting to try to understand the intuition behind the difference between the regret bounds given by the entropic regularizer (which is strongly convex w.r.t. the ℓ_1-norm) and the squared ℓ_2 regularizer on the prediction with expert advice problem with d experts. Notice that in the case of ℓ_2 regularization, even though the "diameter" of the set where the player is making her choices (i.e., the simplex in this case) is less than $1/2$, the functions the enemy can pick behave badly under the lens of the the ℓ_2 norm. Namely, the functions played by the enemy on the experts' problem are $\sqrt{d}$-Lipschitz continuous w.r.t. $\|\cdot\|_2$. In the case of the entropic regularizer, we have that the functions behave way better w.r.t. the ℓ_1 norm: they are 1-Lipschitz continuous on the simplex w.r.t. the ℓ_1-norm. However, this improvement on the Lipschitz constant is not for free: the diameter of the simplex through the lens of the entropic regularizer is $\ln d$, not a constant anymore if compared to d. Still, in this case the trade-off is quite advantageous. Thus, when looking for FTRL regularizers R for an OCO instance $\mathcal{C} \coloneqq (X, \mathcal{F})$, the intuition that one should balance two factors. The first is the diameter of X through the lens of R, that is, any two points inside X should not have values of R which are too far away. At the same time, this regularizer is usually associated with a norm $\|\cdot\|$ with respect to which R is strongly convex. In this case, one wants the functions played by the enemy to be "well-behaved" under $\|\cdot\|$, that is, to have small Lipschitz constant w.r.t. $\|\cdot\|$. To study the Lipschitz constant of the functions from $\mathcal{F}$, it is usually useful to look at the dual norms of the subgradients since, for any $x, y \in \mathbb{E}$ and any convex function $f \colon \mathbb{E} \to (-\infty, +\infty]$ which is subdifferentiable at x, the subgradient inequality yields, for any $g \in \partial f(x)$,

$$f(x) - f(y) \leq \langle g, x - y \rangle \leq \|g\|_* \|x - y\|.$$

4.6 Regularization Regardless of the Number of Rounds

At the end of the last section, we have seen how to attain good regret guarantees in the experts' problem with the classical FTRL algorithm. However, one may note that to attain these bounds the player needs to know beforehand the total numbers of rounds T of the game she is going to play. If she uses a regularizer designed for a game with T rounds in a game with significantly more (or less) rounds, the regret guarantees would change, and the dependence on the number of rounds would not necessarily be sublinear anymore. One way to circumvent this problem is to use the Doubling Trick (see [67, Section 2.3.1]), where one re-starts the algorithm, not necessarily FTRL, at increasing time intervals, adjusting the player oracle on each interval to the proper time horizon of that interval. We define a player oracle which implements this strategy formally on Algorithm 4.4, parameterized by a function PFAMILY from $\mathbb{N}$ to the set of possible player oracles. That is, PFAMILY is a family of player oracles, where for each $T \in \mathbb{N}$ the oracle PFAMILY_T is guaranteed to perform well in a game with T rounds.

Algorithm 4.4 Definition of $\text{DOUBLING}^{\mathcal{C}}_{\text{PFAMILY}}(\langle f_1, \dots, f_T \rangle)$

```
Input:
    (i) An OCO instance $\mathcal{C} \coloneqq (X, \mathcal{F})$ (which we omit if clear from context).
    (ii) A function PFAMILY from $\mathbb{N}$ to $X^{\mathrm{Seq}(\mathcal{F})}$ such that for each $T' \in \mathbb{N}$ the function $\text{PFAMILY}_{T'}$
         is a player oracle for $\mathcal{C}$.
    (iii) Functions $f_1, \dots, f_T \in \mathcal{F}$.
Output: $x_{T+1} \in X$
  $T' \leftarrow 2^{\lfloor \lg T \rfloor}$.
  Let $x_{T+1} \leftarrow \text{PLAYER}_{T'}(\boldsymbol{f}_{T':T})$
  return $x_{T+1}$
```

Let us look at an example of an example of a "player family" oracle as used by the DOUBLING oracle. Consider a instance $\mathcal{C} \coloneqq (\Delta_E, \mathcal{F})$ of the randomized experts' problem, define $R \coloneqq \frac{1}{2}\|\cdot\|_2^2$, and set $d \coloneqq |E|$. By Proposition 4.5.4, the FTRL oracle with regularizer $\sqrt{dT}R$ performs well in games against any enemy for $\mathcal{C}$ in a game with T rounds. Thus, a good example of a player family oracle to be used in the doubling oracle is

$$\text{PFAMILY}_T \coloneqq \text{FTRL}_{\sqrt{dT}R}, \qquad \forall T \in \mathbb{N}.$$

Soon we will prove that $\text{DOUBLING}^{\mathcal{C}}_{\text{PFAMILY}}$ performs well in games with any number of rounds.

Let us look more closely at Algorithm 4.4. The idea of the algorithm is to divide the rounds of the game into intervals, each with double the size of the previous one. The sizes of these intervals are, in some sense, estimates of the duration of the game. With that, the DOUBLING oracle starts to play with a player oracle for a small number of rounds (the number of rounds of the first interval). If the game goes beyond the timeframe given by the interval of rounds the DOUBLING oracle defined last, the oracle doubles its estimate of the number of rounds, and starts to play *from scratch*, using a brand new player oracle. That is, the oracle does not use the information of the functions given by the enemy in the past round intervals.

The reason to not use the oracle with full information is that, in order to use the player oracle for T functions $f_1, \dots, f_T$, one usually needs to compute the point given by the player oracle for $f_1, \dots, f_t$ for each $t \in [T]$. In an OCO game this is natural since the oracle receives only one new function per round. However, the DOUBLING oracle picks a brand new player oracle on each different section, and it would be inefficient[10] to compute all the points this new player oracle would have played in previous rounds in order to use this new oracle with complete information in the next rounds. In Algorithm 4.4, the number T' is the last round on which the DOUBLING oracle re-started, doubling its estimate of the number of rounds. A nice property of the this strategy is that T' in Algorithm 4.4 is also the size of the current section of rounds which the DOUBLING oracle is considering.

Maybe surprisingly, if the original time-dependent player oracle has a $O(\sqrt{T})$ regret bound in a game with T rounds, the next theorem shows a regret bound for the DOUBLING oracle which is worse by only a constant factor.

Theorem 4.6.1. Let $\mathcal{C} := (X, \mathcal{F})$ be an OCO instance and let $\text{PFAMILY} \colon \mathbb{N} \to \mathbb{E}^{\text{Seq}(\mathcal{F})}$ be a function such that PFAMILY_T is a player oracle for $\mathcal{C}$ for each $T \in \mathbb{N}$. If there are $U \subseteq \mathbb{E}$ and $\alpha \in \mathbb{R}_+$ such that, for every $T \in \mathbb{N}$ and every enemy oracle ENEMY for $\mathcal{C}$,

$$\text{Regret}_T(\text{PFAMILY}_T, \text{ENEMY}, U) \leq \alpha\sqrt{T},$$

then, for every $T \in \mathbb{N}$ and every enemy oracle ENEMY for $\mathcal{C}$ we have

$$\text{Regret}_T(\text{DOUBLING}_{\text{PFAMILY}}, \text{ENEMY}, U) \leq \left(\frac{\sqrt{2}}{\sqrt{2}-1}\right)\alpha\sqrt{T}.$$

Proof. Let ENEMY be an enemy oracle for $\mathcal{C}$. Moreover, let $T \in \mathbb{N}$ and define

$$(\boldsymbol{x}, \boldsymbol{f}) := \text{OCO}_{\mathcal{C}}(\text{DOUBLING}_{\text{PFAMILY}}, \text{ENEMY}, T).$$

Note that $T \leq 2^{\lfloor \lg T \rfloor + 1} - 1$. Set $T' := 2^{\lfloor \lg T \rfloor + 1} - 1$ and define[11] $\boldsymbol{f}' \in (\mathcal{F} \cup \{0\})^{T'}$ by $f'_i := [i \leq T] f_i$ for each $i \in [T']$. In words, $\boldsymbol{f}'$ is just $\boldsymbol{f}$ extended with zeroes. In this case, note that

$$\text{Regret}(\text{DOUBLING}_{\text{PFAMILY}}, \boldsymbol{f}, u) \leq \text{Regret}(\text{DOUBLING}_{\text{PFAMILY}}, \boldsymbol{f}', u), \qquad \forall u \in \mathbb{E}.$$

Thus, we may assume without loss of generality that $T = 2^{\lfloor \lg T \rfloor + 1} - 1$.

[10] Inefficient here is about practical implementations. All oracles in this text are defined in such a way that they "re-compute" all the previous iterates at every round. Still, often these oracles need little effort to generate one iterate given the past ones. This would not be the case for the DOUBLING player if it had to re-compute, even in practice, all the past iterates once he changes his player oracle.

[11] Here we are using 0 to denote the identically zero function on $\mathbb{E}$.

To ease the notation, define $p(n) \coloneqq 2^n$ for every $n \in \mathbb{N}$. Recall that, by our notation definition, $\boldsymbol{f}_{i:j} = \langle f_i, f_{i+1}, \dots, f_j \rangle$. Then, by the definition of the DOUBLING oracle, for any $u \in U$,

$$\begin{aligned}
\mathrm{Regret}(\mathrm{DOUBLING}_{\mathrm{PFAMILY}}, \boldsymbol{f}, u) &= \sum_{i=0}^{\lfloor \lg T \rfloor} \mathrm{Regret}_{p(i)}(\mathrm{PFAMILY}_{p(i)}, \boldsymbol{f}_{p(i)\,:\,p(i+1)-1}, u) \\
&\leq \alpha \sum_{i=0}^{\lfloor \lg T \rfloor} \sqrt{p(i)} = \alpha \sum_{i=0}^{\lfloor \lg T \rfloor} \left(\sqrt{2}\right)^i = \alpha \left(\frac{\sqrt{2}^{\lfloor \lg T \rfloor + 1} - 1}{\sqrt{2} - 1} \right) \\
&\leq \alpha \left(\frac{\sqrt{2T} - 1}{\sqrt{2} - 1} \right) \leq \alpha \sqrt{T} \left(\frac{\sqrt{2}}{\sqrt{2} - 1} \right). \qquad \square
\end{aligned}$$

Even though the Doubling Trick does guarantee regret bounds only a multiplicative constant worse than the bound from Corollary 4.5.3, re-starting the algorithm several times seems wasteful. What we can do instead is to use the AdaFTRL algorithm with a regularizer strategy that uses always the same regularizer function, but with a different constant multiplying it at every round. That is, we are still using a static function as our main regularizer, but at each round we adjust the constants multiplying it so that it takes into account the duration of the game without the need of re-starting the whole algorithm. Before jumping into Corollary 4.5.3, we need to prove a simple lemma.

Lemma 4.6.2. Let $a_1 \dots, a_n \in \mathbb{R}_+$ with $a_1 > 0$. Then,

$$\sum_{i=1}^{n} \left(\frac{a_i}{\sqrt{\sum_{j=1}^{i} a_j}} \right) \leq 2 \sqrt{\sum_{i=1}^{n} a_i}.$$

Proof. The proof is by induction on n. The statement holds trivially for $n = 1$. Let $n > 1$, and define $s \coloneqq \sum_{i=1}^{n} a_i$. By the induction hypothesis,

$$\sum_{i=1}^{n} \frac{a_i}{\sqrt{\sum_{j=1}^{i} a_j}} \leq 2 \sqrt{\sum_{i=1}^{n-1} a_i} + \frac{a_n}{\sqrt{\sum_{j=1}^{n} a_j}} = 2\sqrt{s - a_n} + \frac{a_n}{\sqrt{s}}.$$

Finally, note that

$$\begin{aligned}
2\sqrt{s - a_n} + \frac{a_n}{\sqrt{s}} \leq 2\sqrt{s} &\iff 2\sqrt{s(s - a_n)} \leq 2s - a_n \iff 4s(s - a_n) \leq (2s - a_n)^2 \\
&\iff 4s^2 - 4sa_n \leq 4s^2 - 4sa_n + a_n^2 \iff 0 \leq a_n^2. \qquad \square
\end{aligned}$$

Corollary 4.6.3 (Derived from Theorem 4.4.3)**.** Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that each $f \in \mathcal{F}$ is proper and closed. Let $R \colon \mathbb{E} \to (-\infty, +\infty]$ be a 1-strong FTRL regularizer for $\mathcal{C}$ w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$, and suppose μR is also a classical FTRL strategy for $\mathcal{C}$ for any $\mu \in \mathbb{R}_{++}$. Let $\eta \colon \mathbb{N} \setminus \{0\} \to \mathbb{R}_{++}$ and define the regularizer strategy $\mathcal{R} \colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ by, for each $t \in \mathbb{N}$,

$$\mathcal{R}(\boldsymbol{f}) \coloneqq \left(\frac{1}{\eta_{t+1}} - [t > 0] \frac{1}{\eta_t} \right) R, \qquad \forall \boldsymbol{f} \in \mathcal{F}^t.$$

Let $T \in \mathbb{N}$ and let ENEMY be an enemy oracle for $\mathcal{C}$. Define

$$(\boldsymbol{x}, \boldsymbol{f}) \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaFTRL}_{\mathcal{R}}, \mathrm{ENEMY}, T).$$

Finally, let $g_t \in \partial f_t(x_t)$ for each $t \in [T]$ and define $\boldsymbol{\sigma} \in \mathbb{R}^T$ by $\sigma_t \coloneqq \eta_t^{-1}$ for each $t \in [T]$. Then, $\mathcal{R}$ is a FTRL regularizer strategy for $\mathcal{C}$ which is $\boldsymbol{\sigma}$-strong for $\boldsymbol{f}$ w.r.t. $\|\cdot\|$, $\boldsymbol{x} \in \operatorname{Seq}(X)$ and, for every $u \in X$,

$$\operatorname{Regret}(\operatorname{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, u) \leq \sum_{t=1}^{T} \left(\frac{1}{\eta_t} - [t > 1] \frac{1}{\eta_{t-1}} \right) (R(u) - R(x_t)) + \frac{1}{2} \sum_{t=1}^{T} \eta_t \|g_t\|_*^2. \tag{4.25}$$

In particular, consider the case where every function in $\mathcal{F}$ is ρ-Lipschitz continuous w.r.t. $\|\cdot\|$ on a convex set $D \supseteq X$ with nonempty interior and there is $\theta \in \mathbb{R}_{++}$ such that it holds that $\theta \geq \sup\{\, R(x) - R(y) : x \in X, y \in X \cap \operatorname{dom} R \}$. If we define

$$\eta_t \coloneqq \frac{1}{\rho}\sqrt{\frac{\theta}{t}}, \qquad \forall t \in \mathbb{N} \setminus \{0\}, \tag{4.26}$$

then,

$$\operatorname{Regret}(\operatorname{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, X) \leq 2\rho\sqrt{\theta T}.$$

Proof. Define $r_t \coloneqq \mathcal{R}(\langle f_1, \dots, f_{t-1} \rangle)$ for each $t \in [T]$. Note that, for each $t \in [T]$, the function

$$\sum_{i=1}^{t} r_i + \sum_{i=1}^{t} f_i = \frac{1}{\eta_t} R + \sum_{i=1}^{t} f_i$$

is $(1/\eta_t)$-strongly convex w.r.t. $\|\cdot\|$. Therefore, $\mathcal{R}$ is a $\boldsymbol{\sigma}$-strong FTRL regularizer strategy for $\boldsymbol{f}$ (the other necessary properties are easily implied by the fact that μR is a classical FTRL regularizer for any $\mu \in \mathbb{R}_{++}$). Therefore, (4.25) and $\boldsymbol{x} \in \operatorname{Seq}(X)$ follow directly from Theorem 4.4.3.

Suppose that every $f \in \mathcal{F}$ is ρ-Lipschitz continuous w.r.t. $\|\cdot\|$ on a convex set $D \supseteq X$ with nonempty interior, that there is $\theta \in \mathbb{R}_{++}$ as in the statement, and that η is given by (4.26). By Theorem 3.8.4, for each $t \in [T]$ there is $g_t \in \partial f_t(x_t)$ such that $\|g_t\|_* \leq \rho$. Therefore, by (4.25), for every $u \in X$ we have

$$\begin{aligned}\operatorname{Regret}(\operatorname{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, u) &\leq \theta \sum_{t=1}^{T} \left(\frac{1}{\eta_t} - [t > 1] \frac{1}{\eta_{t-1}} \right) + \frac{\rho^2}{2} \sum_{t=1}^{T} \eta_t \\ &= \rho\sqrt{\theta T} + \frac{\rho\sqrt{\theta}}{2} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq 2\rho\sqrt{\theta T},\end{aligned}$$

where in the last inequality we have used Lemma 4.6.2. □

The above regret bound is a $\sqrt{2}$ multiplicative factor worse than the bound given by Corollary 4.5.3. Yet, this bound holds at every round of the game, without the need of any prior knowledge on the number of rounds. Still, note that we need to know the Lipschitz constants of the functions in order to use the above regularizer strategies.

Even though we know the Lipschitz constant in some important examples, such as in the expert's problem, there are many cases where we do not have this information. Not only that, but even in the cases in which we do know the Lipschitz constant, the enemy may pick many functions that have subgradients with small dual norm, far from the upper bound given by the Lipschitz constant. We may hope that, if the regularizer strategy could "notice" and adapt to subgradients with small norm, the algorithm would perform better in these "easy" cases. On Chapter 6 we will investigate this idea.

4.7 An Adaptive Proximal Example

Up to this point, we have not given application examples of AdaFTRL with regularizer strategies with interesting proximal properties (as opposed to a static regularizers which are trivially proximal). Actually, from the discussion at the end of Section 4.4, one may expect that AdaFTRL with proximal regularizer strategies may be able to adapt better to the enemy choices if compared to a general regularizer strategy, thus yielding better regret guarantees. In this section, we will look at relatively simple proximal regularizer strategies. Despite their simplicity, analyzing their performance and comparing it to the regret bounds from the previous section is insightful.

For example, we have already seen that $(1/2)\|\cdot\|_2^2$ is a good static regularizer for some OCO instances, even more so if properly scaled at each round as in the last section. A natural way to make this regularizer strategy proximal is to change the regularizer increment from round $t \in \mathbb{N} \setminus \{0\}$ to a multiple of $x \in \mathbb{E} \mapsto \|x - [t > 1]x_{t-1}\|_2^2$, where x_{t-1} is the iterate from round $t-1$. Thus, this regularizer strategy is, in some sense, generated by the function $R \colon \mathbb{E} \times \mathbb{E} \to (-\infty, +\infty]$ given by $R(x, y) \coloneqq \|x - y\|_2^2$ for every $x, y \in \mathbb{E}$. As expected, this change in the regularizer increments preserves strong convexity, that is, $R(\cdot, y)$ is strongly convex w.r.t. the ℓ_2-norm for any $y \in \mathbb{E}$. Further, we have the freedom to choose where our regularizer is minimized: $y \in \arg\min_{x \in \mathbb{E}} R(x, y)$ for any $y \in \mathbb{E}$. This regularizer does not yet fully exploit the capabilities of AdaFTRL since all the regularizers are strongly convex w.r.t. the same norm. Still, regularizer strategies of this form, which we call *proximal FTRL regularizers*, already cover some interesting cases, and we prove on Corollary 4.7.3 that they have good regret guarantees. Let us formalize this discussion before jumping to the corollary.

Definition 4.7.1 (Proximal FTRL regularizer)**.** Let $R \colon \mathbb{E} \to (-\infty, +\infty]$ be a convex function and let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance. The function $R \colon \mathbb{E} \times X \to (-\infty, +\infty]$ is a **proximal FTRL regularizer** (for $\mathcal{C}$) if

(i) For every $T \in \mathbb{N} \setminus \{0\}$ and each $\boldsymbol{x} \in X^T$, we have that $\sum_{t=1}^{T} R(\cdot, x_t)$ is a classical FTRL regularizer strategy

(ii) For every $y \in X$ we have that $\inf_{x \in \mathbb{E}} R(x, y)$ is attained by y.

As expected, the above definition builds upon the one of classical FTRL regularizers, significantly adding only the proximal property by (ii). Indeed, it is tailored to naturally build a proximal FTRL regularizer strategy. To see that, let $x_0 \in X$, and define the FTRL regularizer strategy $\mathcal{R}$ for $\mathcal{C}$ in a recursive fashion by

$$\begin{aligned} &\mathcal{R}(\langle\rangle) \coloneqq R(\cdot, x_0), \\ &\mathcal{R}(\boldsymbol{f}) \coloneqq R(\cdot, \mathrm{AdaFTRL}_{\mathcal{R}}(\langle f_1, \dots, f_{t-1}\rangle)), \qquad \forall \boldsymbol{f} \in \mathcal{F}^t, \forall t \in \mathbb{N} \setminus \{0\}. \end{aligned} \tag{4.27}$$

One may check that $\mathcal{R}$ is a proximal FTRL regularizer strategy for $\mathcal{C}$. Finally, let us make a proximal version of the definition of σ-strong classical FTRL regularizer. This time we cannot simply re-use the definition of σ-strong since we need slightly different condition to apply Lemma 4.3.2.

Definition 4.7.2 (σ-proximally strong proximal regularizer strategy)**.** Let $\sigma \in \mathbb{R}_{++}$ and let R be a proximal FTRL regularizer for $\mathcal{C}$. Then R is **σ-proximally strong** for $\mathcal{C}$ w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$ if

(i) $R(\cdot, x)$ is σ-strongly convex for any $x \in X$, and

(ii) $\mathrm{ri}(\mathrm{dom}(\sum_{t=1}^{T+1} R(\cdot, x_t) + \sum_{t=1}^{T-1} f_t)) \cap \mathrm{ri}(\mathrm{dom}\, f_T)$ is nonempty for every $\boldsymbol{x} \in X^{T+1}$, $\boldsymbol{f} \in \mathcal{F}^T$, and $T \in \mathbb{N}$.

Again, the above definition is tailored in a way such that the functions which will be minimized by the Adaptive FTRL oracle satisfy the conditions of Lemma 4.3.2. Indeed, if R used in the definition of $\mathcal{R}$ as in (4.27) is a σ-proximally strong FTRL regularizer for an OCO instance $\mathcal{C}$, then $\mathcal{R}$ is a FTRL regularizer strategy which is $\boldsymbol{\sigma}$-proximally strong for any $\boldsymbol{f} \in \mathrm{Seq}(\mathcal{F})$, where $\boldsymbol{\sigma} \in \mathrm{Seq}(\mathbb{R})$ is a properly sized sequence with all entries equal to σ. This leaves us in position to use Theorem 4.4.4. This discussion outlines the roadmap of the proof of the next corollary, except that in the actual proof we take additional care with constants multiplying the regularizer (which one can interpret as step sizes of the algorithm).

Corollary 4.7.3 (Derived from Theorem 4.4.4)**.** Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that each $f \in \mathcal{F}$ is proper and closed. Let $R\colon \mathbb{E} \times X \to (-\infty, +\infty]$ be a 1-proximally strong[12] FTRL regularizer for $\mathcal{C}$ w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$. Let $\eta\colon \mathbb{N} \setminus \{0\} \to \mathbb{R}_{++}$, let $z \in X$, and define the regularizer strategy $\mathcal{R}\colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ by

$$\begin{aligned}
&\mathcal{R}(\langle\rangle) \coloneqq R(\cdot, z),\\
&\mathcal{R}(\boldsymbol{f}) \coloneqq \left(\frac{1}{\eta_{t+1}} - \frac{1}{\eta_t}\right) R(\cdot, \mathrm{AdaFTRL}_{\mathcal{R}}(\langle f_1, \dots, f_{t-1}\rangle)), \qquad \forall t \in \mathbb{N} \setminus \{0\}, \forall \boldsymbol{f} \in \mathcal{F}^t.
\end{aligned}$$

Let $T \in \mathbb{N}$ and let ENEMY be an enemy oracle for $\mathcal{C}$. Finally, define

$$(\boldsymbol{x}, \boldsymbol{f}) \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaFTRL}_{\mathcal{R}}, \mathrm{ENEMY}, T),$$

set $x_0 \coloneqq x_1$, let $g_t \in \partial f_t(x_t)$ for each $t \in [T]$, and define $\boldsymbol{\sigma} \in \mathbb{R}^T_{++}$ by $\sigma_t \coloneqq \eta_t^{-1}$ for each $t \in [T]$. Then $\mathcal{R}$ is a FTRL regularizer strategy for $\mathcal{C}$ which is $\boldsymbol{\sigma}$-proximally strong for $\boldsymbol{f}$ w.r.t. $\|\cdot\|$, $\boldsymbol{x} \in \mathrm{Seq}(X)$, and, for every $u \in X$,

$$\mathrm{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, u) \leq \sum_{t=0}^{T} \left(\frac{1}{\eta_{t+1}} - [t > 0]\frac{1}{\eta_t}\right)(R(u, x_t) - R(x_t, x_t)) + \frac{1}{2}\sum_{t=1}^{T} \eta_{t+1}\|g_t\|_*^2. \tag{4.28}$$

In particular, consider the case where every function in $\mathcal{F}$ is ρ-Lipschitz continuous w.r.t. $\|\cdot\|$ on a convex set $D \supseteq X$ with nonempty interior and there is $\theta \in \mathbb{R}_{++}$ such that $\theta \geq \sup\{\, R(x,z) - R(y,z) : x, z \in X, y \in X \cap$
is finite. In this case, if we set $\eta_1 \coloneqq 1$ and define

$$\eta_t \coloneqq \frac{1}{\rho}\sqrt{\frac{\theta}{t-1}}, \qquad \forall t \in \mathbb{N} \setminus \{0, 1\}, \tag{4.29}$$

then,

$$\mathrm{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, X) \leq 2\rho\sqrt{\theta T}.$$

Proof. Let us show that

$$\mathcal{R} \text{ is a FTRL regularizer strategy for } \mathcal{C} \text{ which is } \boldsymbol{\sigma}\text{-proximally strong for } \boldsymbol{f} \text{ w.r.t. } \|\cdot\|. \tag{4.30}$$

Since R is a proximal FTRL regularizer, we have that $\mathcal{R}$ is a proximal FTRL regularizer strategy. It only remains to show that it is $\boldsymbol{\sigma}$-proximally strong. Define $r_t \coloneqq \mathcal{R}(\langle f_1, \dots, f_{t-1}\rangle)$ for each $t \in \{1, \dots, T+1\}$. Note that, for each $t \in [T]$, the function

$$\sum_{i=1}^{t} r_i + \sum_{i=1}^{t-1} f_i = \sum_{i=1}^{t}\left(\frac{1}{\eta_i} - [i > 0]\frac{1}{\eta_{i-1}}\right) R(\cdot, x_{i-1}) + \sum_{i=1}^{t} f_i$$

[12] Supposing that R is 1-proximally strong instead of σ-strongly convex for some $\sigma \in \mathbb{R}_{++}$ can be made without loss of generality since we can adjust the strong convexity constant by multiplying the regularizer the a positive constant.

is $(1/\eta_t)$-strongly convex w.r.t. $\|\cdot\|$. Indeed, the sum $\sum_{i=1}^{t} r_i$ of strongly convex functions is also strongly convex, with strong convexity parameter equal to the strong convexity parameter of each one of them added, which is $\sum_{i=1}^{t}(\eta_i^{-1} - [t > 0]\eta_{i-1}^{-1}) = \eta_t^{-1}$. Moreover, since multiplying the regularizers by positive constants does not change their effective domains and since R is a 1-proximally strong FTRL regularizer, we conclude that $\mathcal{R}$ satisfies property (ii) of the definition of $\boldsymbol{\sigma}$-proximally strong. This proves (4.30). With that, we conclude that (4.28) follows directly from Theorem 4.4.4.

Suppose that every function in $\mathcal{F}$ is ρ-Lipschitz continuous w.r.t. $\|\cdot\|$ on a convex set $D \supseteq X$ with nonempty interior, that there is $\theta \in \mathbb{R}_{++}$ as in the statement, and that η is given by (4.29). By Theorem 3.8.4, for each $t \in [T]$ there is $g_t \in \partial f_t(x_t)$ such that $\|g_t\|_* \leq \rho$. Therefore, by (4.28), for every $u \in X$ we have

$$\begin{aligned}\operatorname{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, u) &\leq \theta \sum_{t=0}^{T}\left(\frac{1}{\eta_{t+1}} - [t > 0]\frac{1}{\eta_t}\right) + \frac{\rho^2}{2}\sum_{t=1}^{T}\eta_{t+1} \\ &= \rho\sqrt{\theta T} + \frac{\rho\sqrt{\theta}}{2}\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \leq 2\rho\sqrt{\theta T},\end{aligned}$$

where in the last inequality we have used Lemma 4.6.2. □

The bounds on Corollaries 4.6.3 and 4.7.3 are not much different. At first sight, this seems to point into the opposite direction of the idea that FTRL with proximal regularizer strategies are able to adapt better, as discussed at the end of Section 4.4. Nonetheless, it was expected that the regret bounds of these corollaries were similar. The point is that the regret bound for proximal regularizer strategies from Theorem 4.4.4 differs from the regret bound of the general case (Theorem 4.4.3) on which norm is used to measure the *subgradients*. Thus, if we develop strategies which are oblivious to the subgradients of the functions played by the enemy, as in Corollaries 4.6.3 and 4.7.3, there is no reason for the regret bounds to be significantly different. On Chapter 6 we develop regularizer strategies which take into account the subgradients from the functions played by the enemy on past rounds, and a bigger difference between regret bounds for general and proximal regularizer strategies appears.

For the sake of concreteness, let us apply a proximal regularizer strategy to the (randomized) experts' problem $\mathcal{C} \coloneqq (\Delta_E, \mathcal{F})$, where $\mathcal{F}$ is given by $\mathcal{F} \coloneqq \{\, p \in \mathbb{R}^E \mapsto y^{\mathsf{T}} p : y \in [-1,1]^E \}$. As we have seen at the end of Section 4.5, every function in $\mathcal{F}$ is $\sqrt{d}$-Lipschitz continuous w.r.t. $\|\cdot\|_2$ on $\mathbb{R}^E$, where $d \coloneqq |E|$. Moreover, define $R\colon \mathbb{E} \times \mathbb{E} \to (-\infty, +\infty]$ by $R(x,y) \coloneqq \frac{1}{2}\|x - y\|_2^2$ for every $x, y \in \mathbb{E}$. Note that $R(\cdot, y)$ is 1-strongly convex w.r.t. the ℓ_2-norm for every $y \in \mathbb{E}$ by Lemma 3.9.5 since the ℓ_2-norm is induced by the euclidean inner product. Finally, define $\mathcal{R}\colon \operatorname{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ as in Corollary 4.7.3. Since $R(x,y) \leq \frac{1}{2}(\|x\| + \|y\|)^2 \leq 2$ for every $x, y \in \Delta_E$, we have $\sup_{x,y,z}(R(y,x) - R(z,x)) \leq 2$. Therefore, by Corollary 4.7.3,

$$\operatorname{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}}, \boldsymbol{f}, X) \leq 2\sqrt{2dT} \qquad \forall T \in \mathbb{N}, \forall \boldsymbol{f} \in \mathcal{F}^T.$$

It is interesting to compare the above regret bound with the one on (4.22) derived by using the classical FTRL algorithm with the (squared) ℓ_2-norm. Note that the above bound is a factor of $2\sqrt{2}$ worse than the one on (4.22). A factor of $\sqrt{2}$ of this difference is due to the difference on the bounds given by Corollaries 4.5.3 and 4.7.3. Nevertheless, the remaining difference is due to the different upper bounds given on the value of the regularizers: a bound of $1/2$ for $\frac{1}{2}\|x\|_2^2$ for $x \in \Delta_E$ when using Corollary 4.5.3, and a bound of 2 for $\frac{1}{2}\|x - y\|_2^2$ with $x, y \in \Delta_E$ when using Corollary 4.7.3. Still, this difference might be artificial and not hold in practice because we are loosely bounding $\frac{1}{2}\|x - y\|_2^2$. To see why this bound may be loose in practice, recall that the bound on $\frac{1}{2}\|x - y\|_2^2$ is used to bound the sum $\sum_{t=1}^{T+1} r_t(u) = \sum_{t=0}^{T} \frac{1}{2}\|u - x_t\|_2^2$ from Theorem 4.4.4, where u is the comparison point on the regret formula, and the points x_t are the iterates from of the Adaptive FTRL algorithm. Intuitively, one should expect the iterates x_t to be closer to a "good" comparison point u in the regret formula than to 0 (for example, one might expect in the experts' problem to see iterates attributing a high weight to good experts), which would imply that $\|u - x_t\|_2^2 \leq \|u - 0\|_2^2 = \|u\|_2^2$ for each iterate x_t.

4.8 Logarithmic Regret Against Strongly Convex Functions

As we have seen on Section 4.5, in a game with T rounds FTRL attains a dependence of $O(\sqrt{T})$ on its worst-case regret, which is optimal for OCO instances with Lipschitz continuous functions [2]. Still, if we are dealing with an instance about which we know more about the functions the enemy is allowed to use, we may improve the regret bounds. Indeed, the next corollary shows a bound on the worst case regret bound of FTRL (pratically without a regularizer) for OCO instances with strongly convex functions which is exponentially better than the one from Corollary 4.5.3.

Corollary 4.8.1 (Derived from Theorem 4.4.4)**.** Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that $X \subseteq \mathbb{E}$ is closed and that each $f \in \mathcal{F}$ is σ-strongly convex and ρ-Lipschitz continuous w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$ on a convex set $D \supseteq X$ with nonempty interior. Moreover, suppose $(\operatorname{ri}(\operatorname{dom}(R + \sum_{t=1}^{T} f_t)) \cap \operatorname{ri}(\operatorname{dom} f_{T+1})$ is nonempty[13] for any $\boldsymbol{f} \in \mathcal{F}^{T+1}$ and $T \in \mathbb{N}$, and define $R \coloneqq \delta(\cdot \,|\, X)$. Then, for any enemy oracle ENEMY for $\mathcal{C}$ and $T \in \mathbb{N}$, we have

$$\operatorname{Regret}_T(\mathrm{FTRL}_R, \mathrm{ENEMY}, X) \leq \frac{\rho^2}{2\sigma}(1 + \ln T).$$

Proof. Let $T \in \mathbb{N}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and define

$$(\boldsymbol{x}, \boldsymbol{f}) \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{FTRL}_R, \mathrm{ENEMY}, T).$$

Note that $\mathrm{FTRL}_R = \mathrm{AdaFTRL}_{\mathcal{R}}$ where $\mathcal{R}$ is given by $\mathcal{R}(\boldsymbol{f}') \coloneqq [\boldsymbol{f}' = \langle\rangle]R$ for every $\boldsymbol{f}' \in \operatorname{Seq}((-\infty, +\infty]^{\mathbb{E}})$. Let us show that

$$\begin{gathered}\mathcal{R} \text{ is a proximal FTRL regularizer strategy for } \mathcal{C} \text{ which is } \boldsymbol{\sigma}\text{-proximally strong} \\ \text{for } \boldsymbol{f}, \text{ where } \boldsymbol{\sigma} \in \mathbb{R}^T \text{ is given by}^{14}\ \sigma_t \coloneqq (t-1)\sigma \text{ for each } t \in [T].\end{gathered} \tag{4.31}$$

First, let us show that $\mathcal{R}$ is a FTRL regularizer strategy. Let $t \in [T]$. Since X is closed and convex, $R = \delta(\cdot \,|\, X)$ is closed, proper, and convex. Moreover, $\operatorname{dom} R = X$ by the definition of indicator function. Finally, since each $h \in \mathcal{F}$ is Lipschitz continuous on $D \supseteq X$, we have that $\operatorname{dom} h \supseteq X$. Therefore, we have that $\operatorname{dom}(R + \sum_{i=1}^{t} f_i) = X$, and by Lemma 3.9.14 we conclude that $\inf_{x \in \mathbb{E}}(R(x) + \sum_{i=1}^{t} f_i(x))$ is attained. Thus, to prove (4.31) it only remains to show that $\mathcal{R}$ is $\boldsymbol{\sigma}$-proximally strong for $\boldsymbol{f}$. Note that $\mathcal{R}$ is proximal, and that, since f_t is σ-strongly convex for every $t \in [T]$, we have that

$$\sum_{i=1}^{t} \mathcal{R}(\langle f_1, \ldots, f_{i-1} \rangle) + \sum_{i=1}^{t-1} f_i = \delta(\cdot \,|\, X) + \sum_{i=1}^{t-1} f_i \tag{4.32}$$

is $(t-1)\sigma$-strongly convex for each $t \in [T]$. Moreover, we have that the condition on the intersection of the relative interiors of the domain from the definition of proximally strong FTRL regularizer

[13] The sole purpose of this assumption, which is satisfied in most usual applications, is to enable us to apply Lemma 4.3.2.

[14] One may find weird that $\sigma_1 = 0$, which means that the first regularizer is not strongly convex. Note, however, that σ_1 does not affect the bound from Theorem 4.4.4.

(property (ii)) is satisfied by assumption. This shows that $\mathcal{R}$ is $\boldsymbol{\sigma}$-proximally strong for $\boldsymbol{f}$, which finishes the proof of (4.31).

Let $g_t \in \partial f_t(x_t)$ be such that $\|g_t\|_* \leq \rho$, which exists by Theorem 3.8.4, for each $t \in [T]$. Thus, by (4.31) and Theorem 4.4.4, for every $u \in X$ we have

$$\mathrm{Regret}(\mathrm{FTRL}_R, \boldsymbol{f}, u) \leq \frac{1}{2}\sum_{t=1}^{T} \frac{1}{t\sigma}\|g_t\|_*^2 \leq \frac{\rho^2}{2\sigma}\sum_{t=1}^{T}\frac{1}{t} \leq \frac{\rho^2}{2\sigma}(1 + \ln T). \qquad \square$$

Despite the appeal of such a good regret bound, there is no free lunch. In the application examples we have seen of AdaFTRL, the functions played by the enemy were usually linear, or linearizing them with the use of subgradients did not affect much the regret guarantees. This usually allowed us to derive closed formulas for each application of the AdaFTRL. However, we cannot linearize the functions given by the enemy in this case, otherwise we lose the strong convexity property which yields the above regret bound. Thus, an efficient player oracle for the strongly convex case depends on an efficient way to solve the minimization problem from AdaFTRL.

4.9 Follow the Leader–Be the Leader Lemma

In this chapter we presented the classical FTRL algorithm as a special case of the Adaptive FTRL algorithm, whose regret analysis was based on the Strong FTRL Lemma. However, it is interesting to give a quick look at the main tools for the original analysis of the classical FTRL algorithm, and how the Strong FTRL Lemma yields slightly tighter bounds. The following lemma, originally proved by Kalai and Vempala [42], and known as the Follow the Leader–Be the Leader (FTL–BTL) Lemma, is the classical lemma for the analysis of FTRL–like algorithms.

Lemma 4.9.1 (Follow The Leader-Be The Leader Lemma, [42])**.** Let $T \in \mathbb{N}$, let $R, f_1, \dots, f_T \colon \mathbb{E} \to (-\infty, +\infty]$ be proper and such that, for every $t \in \{1, \dots, T+1\}$, the function $R + \sum_{i=1}^{t-1} f_i$ is proper and its infimum over $\mathbb{E}$ is attained. Moreover, define $x_t \coloneqq \mathrm{FTRL}_R(\langle f_1, \dots, f_{t-1}\rangle)$ for every $t \in \{1, \dots, T+1\}$. Then, for every $u \in \mathbb{E}$,

$$R(x_1) + \sum_{t=1}^{T} f_t(x_{t+1}) \leq R(u) + \sum_{t=1}^{T} f_t(u) \tag{4.33}$$

and

$$\mathrm{Regret}(\mathrm{FTRL}_R, \boldsymbol{f}, u) \leq R(u) - \inf_{x\in\mathbb{E}} R(x) + \sum_{t=1}^{T}(f_t(x_t) - f_t(x_{t+1})). \tag{4.34}$$

Proof. First of all, note that by re-arranging (4.34) we get (4.33). Thus, it suffices to prove that (4.33) holds, and we will do so by induction on T. For $T = 0$ the inequality holds since $x_1 \in \arg\min_{x\in\mathbb{E}} R(x)$ by definition. Suppose $T > 0$. Then, for every $u \in \mathbb{E}$,

$$\begin{aligned} R(x_1) + \sum_{t=1}^{T} f_t(x_{t+1}) &= f_T(x_{T+1}) + R(x_1) + \sum_{t=1}^{T-1} f_t(x_{t+1}) \leq f_T(x_{T+1}) + R(x_{T+1}) + \sum_{t=1}^{T-1} f_t(x_{T+1}) \\ &= R(x_{T+1}) + \sum_{t=1}^{T} f_t(x_{T+1}) \leq R(u) + \sum_{t=1}^{T} f_t(u), \end{aligned}$$

where in the first inequality we used the induction hypothesis with (4.33) specialized to $u = x_{T+1}$, and in the second we used that $x_{T+1} \in \arg\min_{x\in\mathbb{E}}(R(x) + \sum_{t=1}^{T} f_t(x))$ by definition. $\square$

One may note some "intuitive similarities" between the above lemma and Lemma 4.3.1. As in the latter, the FTL–BTL Lemma bounds the regret by two terms. The first is $R(u) - R(x_1)$, which can be seen, when we assume $\operatorname{dom} R \subseteq X$ for $X \subseteq \mathbb{E}$, as the diameter of the set X where the player is making her predictions. The other term translates the idea that, in order for the FTRL algorithm to perform well, the algorithm should be stable: for any round t, the values of the iterates x_t and x_{t+1} on f_t should not be too far away.

Let us quickly derive a regret bound similar to the one from Corollary 4.5.3. Let $R, f_1, \dots, f_T$ and $x_1, \dots, x_{T+1}$ be as in Lemma 4.9.1. Moreover, suppose that R is σ-strongly convex and that, for each $t \in [T]$, the function f_t is closed, convex, and that there is $g_t \in \partial f_t(x_t)$. Finally, let $\|\cdot\|$ be a norm on $\mathbb{E}$. By the subgradient inequality and by the definition of dual norm, we have

$$f(x_t) - f(x_{t+1}) \leq \langle g_t, x_t - x_{t+1} \rangle \leq \|g_t\|_* \|x_t - x_{t+1}\|, \qquad \forall t \in [T].$$

For each $t \in [T]$, we have $x_t \in \arg\min_{x \in \mathbb{E}} (R(x) + \sum_{i=1}^{t-1} f_i(x))$. Thus, Lemma 4.3.2 yields[15]

$$\|x_t - x_{t+1}\| \leq \frac{1}{\sigma} \|g_t\|_*, \qquad \forall t \in [T].$$

Therefore, by Lemma 4.9.1, for every $u \in \mathbb{E}$,

$$\begin{aligned}\operatorname{Regret}(\mathrm{FTRL}_R, \boldsymbol{f}, u) &\leq R(u) - \min_{x \in \mathbb{E}} R(x) + \sum_{t=1}^{T} (f_t(x_t) - f_t(x_{t+1})) \\ &\leq R(u) - \inf_{x \in \mathbb{E}} R(x) + \frac{1}{\sigma} \sum_{t=1}^{T} \|g_t\|_*^2.\end{aligned}$$

Note that the above bound is slightly worse (only by a multiplicative constant) than the one from Corollary 4.5.3. One factor that contributed to this difference is that we used, for a norm $\|\cdot\|$ on $\mathbb{E}$, the inequality

$$\langle u, v \rangle \leq \|u\| \|v\|_*, \qquad \forall u, v \in \mathbb{E}, \tag{4.35}$$

to bound the differences of the form $f(x_t) - f(x_{t+1})$. Note that the inequality from Lemma 4.3.2 that bounds the values of the functions themselves, which is the one used on the adaptive case, is tighter by a constant.

We can modify the FTL–BTL Lemma to bound the regret in the case when we have different constants multiplying the regularizer, as in Section 4.6. However, the analyses start to be slightly more *ad hoc* and technical in cases where the functions that the enemy plays are strongly convex themselves, or when we do not want to use (4.35) in order to obtain tighter bounds. Additionally, it is not clear how to use this lemma when we have a different regularizer at each round. Intuitively, the Strong FTRL Lemma (Lemma 4.3.1) helps us by carrying, for each round, the information of the regularizers and the functions from all the past rounds through the functions H_t on Lemma 4.3.1. This allows us to capture a wider range of cases without needing to change the lemma itself.

[15] Keep in mind that in order to apply Lemma 4.3.2 we need to ensure that $\operatorname{ri}(\operatorname{dom}(R + \sum_{i=1}^{t-1} f_i)) \cap \operatorname{ri}(\operatorname{dom} f_t)$ is nonempty for each $t \in [T]$, which is usually the case.

Chapter 5

The Online Mirror Descent Algorithm

In the previous chapter, we have shown that Follow the Regularized Leader algorithms yield good regret bounds. Not only that, the description of the method is fairly straightforward. In spite of that, it is not clear how to efficiently implement the AdaFTRL oracle in general. Additionally, in many applications one has access to the functions only through "first-order oracles", that is, given a point x and a function f, one may compute $f(x)$ together with a (sub)gradient g of f at x. In these cases, one usually wants to make a constant number of queries to the first-order oracle of the enemy's function at each round and wants the actions of the algorithm at each iteration to use the subgradients and the value of the function at the current point in an efficiently implementable fashion, that is, it should take time close to linear on the sizes of the subgradients to perform the actions of the algorithm at each round.

In this chapter we explore Online Mirror Descent algorithms, which are the online (and adaptive) counterparts of the Mirror Descent algorithms from classical convex optimization. The latter are a generalizations of the well-known Gradient Descent technique, and it was first proposed by Nemirovski and Yudin [54]. The general idea is to perform, at each iteration, a step on the direction of negative subgradient of the current function. However, this subgradient step occurs on the dual space $\mathbb{E}^*$ of linear functionals on $\mathbb{E}$, and the connection between $\mathbb{E}$ and $\mathbb{E}^*$ depends on functions, which we call *mirror maps*, that the player can choose depending on the problem. The intuition behind the choices of mirror maps is that the functions played by the enemy and the set where the optimization is taking place are both better behaved (subgradients with smaller norm and smaller diameter w.r.t. the mirror maps) when we look at them through the lens of the mirror map. One may notice that this intuition is similar to the considerations we had to make when choosing FTRL regularizers, and indeed there are interesting connections between online mirror descent and FTRL algorithms.

In this chapter we first describe the Adaptive Online Mirror Descent (AdaOMD) algorithm, following the presentation from [48], and look at the form of the iterate updates with some static regularizers. We then show connections of AdaOMD with proximal operators and to the AdaFTRL algorithm, and from the latter connection we derive regret bounds of AdaOMD. We also describe the Adaptive Dual Averaging (AdaDA) algorithm, a variation of the AdaOMD algorithm, which can be seen as a version of AdaOMD with lazy projections. We prove the interesting fact that AdaDA is practically equivalent to AdaFTRL when applied to linear functions. Finally, we discuss sufficient conditions under which the non-adaptive versions of Online Mirror Descent and Dual Averaging are equivalent. The content of this chapter is based on many sources [13, 18, 19, 48, 57, 67]. More specific references will be given on each section.

5.1 Adaptive Online Mirror Descent

As described at the beginning of this chapter, the *Adaptive Online Mirror Descent* (AdaOMD or Adaptive OMD) algorithm is an adaptive and online version of the Mirror Descent algorithm for classical convex optimization first proposed by Nemirovski and Yudin [54]. The general idea of Mirror Descent is to start at a point and then perform, at each iteration, a step in the direction of minus subgradient of the current function. The difference on intuitions of this method to the one of the well-known (sub)Gradient Descent algorithm is that, on Mirror Descent, the subgradient step is seen as taking place on the dual space $\mathbb{E}^*$ of $\mathbb{E}$, that is, the space of linear functionals on $\mathbb{E}$, and the connection[1] between primal and dual spaces is made through a "mirror map". In a way similar to the Adaptive FTRL algorithm, Adaptive OMD is parameterized by such mirror maps (which one can imagine as functions analogous to FTRL regularizers), which make the connection between the primal and dual spaces. The choice of mirror map, in a similar way to the choice of FTRL regularizers, deeply affects the performance of the algorithm on different problems. As we show later, the Online Subgradient Descent algorithm is a special case of the Online Mirror Descent with a mirror map based on the ℓ_2-norm. To make the discussion of an intuition for the Online Mirror Descent (OMD) algorithm more concrete, let us first formally define the algorithm, and then discuss an intuitive interpretation of it. In the next section we provide some applications for the sake of concreteness.

As in the case of AdaFTRL, at each round $t \in \mathbb{N}$ the Adaptive Online Mirror Descent algorithm depends on a choice of a function R_t, the *mirror map of round* t, which can be thought of for now as the analogous of a regularizer from AdaFTRL. In spite of the similarities, the conditions necessary over such maps are a bit more delicate when compared to the conditions we have used on FTRL regularizer strategies.

Definition 5.1.1 ((Classical) mirror map)**.** Let $X \subseteq \mathbb{E}$ be convex. A **(classical) mirror map** (for X) is a function $R\colon \mathbb{E} \to (-\infty, +\infty]$ such that

(5.1.i) R is a closed proper strictly convex on $\operatorname{dom} R$ function such that $\operatorname{int}(\operatorname{dom} R)$ is nonempty and such that R differentiable on $\operatorname{int}(\operatorname{dom} R)$,

(5.1.ii) for any $y \in \operatorname{int}(\operatorname{dom} R)$, the infima $\inf_{x \in X} R(x)$ and $\inf_{x \in X} B_R(x, y)$ are attained by a point in $\operatorname{int}(\operatorname{dom} R)$,

(5.1.iii) $(\operatorname{ri}(X)) \cap \operatorname{int}(\operatorname{dom} R) \neq \varnothing$, and

(5.1.iv) $\{\, \nabla R(x) : x \in \operatorname{int}(\operatorname{dom} R)\} = \mathbb{E}$.

The properties of a mirror map may seem cryptic right now, but they become way clearer once we define the AdaOMD oracle (in fact, for the sake of simplicity the above properties were slightly simplified, see [18] for more detailed conditions). For this reason, we defer the discussion on the mirror map properties for later.

In a way similar to the case of FTRL regularizer strategies, we define mirror map strategies, which is basically a way of choosing at each round a different mirror map through "mirror map increments".

[1] In Euclidean (and Hilbert) spaces, such a connection is usually seamless by the Riesz representation theorem, which states that for each $x^* \in \mathbb{E}^*$ there is an unique point $g \in \mathbb{E}$ such that $x^* = \langle g, \cdot \rangle$. However, here we are interested in looking at ways of connecting the primal and dual spaces in different ways. We will discuss this intuition in greater detail later on.

Definition 5.1.2 (Mirror map strategy)**.** Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance and let $D \subseteq \mathbb{E}$ be a nonempty open convex set such that $\operatorname{ri}(X) \cap D \neq \varnothing$. A **mirror map strategy** (for $\mathcal{C}$ which is differentiable on D) is a function $\mathcal{R}\colon \operatorname{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ such that, for each $t \in \mathbb{N}$ and $f \in \mathcal{F}^t$, for $R \coloneqq \sum_{i=0}^{t} \mathcal{R}(f_1, \dots, f_i)$ and $r \coloneqq \mathcal{R}(f_1, \dots, f_t)$, we have

(5.2.i) $D = \operatorname{int}(\operatorname{dom} r) = \operatorname{int}(\operatorname{dom} R)$

(5.2.ii) r is a proper closed convex functions which is differentiable on D, and

(5.2.iii) R is a classical mirror map for X.

In property (5.2.i) from the definition of mirror map strategies we require that the domains of the mirror maps do not change much, that is, they need to have the same interior. Property (5.2.ii) is to avoid cases with pathological mirror map strategies. Finally, (5.2.iii) formally states the main idea of a mirror map strategy: to pick, at each round, a mirror map for the player. A player oracle which implements the Adaptive Online Mirror Descent technique is formally defined on Algorithm 5.1.

Algorithm 5.1 Definition of $\mathrm{AdaOMD}_{\mathcal{R}}^{X}(\langle f_1, \dots, f_T \rangle)$

```
Input:
    (i) A closed convex set $X \subseteq \mathbb{E}$,
    (ii) Convex functions $f_1, \dots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$ such that $f_t$ is subdifferentiable on $X$ for each $t \in [T]$,
    (iii) $\mathcal{R}\colon \operatorname{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ is a mirror map strategy for the OCO instance $(X, \mathcal{F})$ which is differentiable on a nonempty open set $D \subseteq \mathbb{E}$.
Output: $x_{T+1} \in D \cap X$
  $r_1 \leftarrow \mathcal{R}(\langle\rangle)$
  $\{x_1\} \leftarrow \arg\min_{x \in X} r_1(x)$.
  for $t = 1$ to $T$ do
        ▷ Computations for round $t + 1$
     $r_{t+1} \leftarrow \mathcal{R}(\langle f_1, \dots, f_t \rangle)$
     $R_{t+1} \leftarrow \sum_{i=1}^{t+1} r_i$
     Compute $g_t \in \partial f_t(x_t)$
     $y_{t+1} \leftarrow \nabla R_{t+1}(x_t) - g_t$,
     $x_{t+1} \leftarrow \Pi_X^{R_{t+1}}(\nabla R_{t+1}^*(y_{t+1}))$
  return $x_{T+1}$
```

With the AdaOMD oracle defined, we may try to make sense of some of the properties of a mirror map. Note that property (5.1.i) together with (5.2.i) implies that the mirror map of each round is differentiable on the open set $D \subseteq \mathbb{E}$ where the mirror map strategy is differentiable. Thus, if the iterates x_t as in Algorithm 5.1 lie in D, then the points y_t in Algorithm 5.1 are well-defined. This later invariant is guaranteed by (5.1.ii), which ensures the Bregman projections yield points in D. One may still be uneasy, since we do not know yet if the definition of AdaOMD makes any sense. For example, in Algorithm 5.1 we take the gradient of the conjugate of each mirror map on the points y_t, but we do not know yet if the conjugate of the mirror map is differentiable at this point. The reader is probably imagining that this and other problems are solved by looking at the other properties of a mirror map. Indeed, the conditions on a mirror map are built in a way so that the definition of AdaOMD is not ambiguous and, in some sense, the objects we use in the definition of AdaOMD are not pathological. The following proposition shows that AdaOMD is indeed well-defined.

Proposition 5.1.3. Let $X \subseteq \mathbb{E}$ be a nonempty closed convex set and let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a mirror map for X. Then,

(a) R^* is differentiable on $\mathbb{E}$,

(b) $\nabla R^*(x^*) \in \operatorname{int}(\operatorname{dom} R)$ for any $x^* \in \mathbb{E}$, and

(c) $\inf_{x \in X} B_R(x, y)$ is attained by a unique point in D for any $y \in D$.

Proof. Let $x^* \in \mathbb{E}$. By property (5.1.iv), there is $y \in D$ such that $x^* = \nabla R(y)$. By Theorem 3.5.2 (items (i), (ii), and (v)) and Theorem 3.5.5, we have

$$x^* = \nabla R(y) \iff y \in \arg\max_{x \in \mathbb{E}} (\langle x^*, x \rangle - R(x)) \iff y \in \partial R^*(x^*).$$

By (5.1.i) we have that R is strictly convex, which implies by Lemma 3.9.2 that y is the unique point that attains $\inf_{x \in \mathbb{E}}(R(x) - \langle x^*, x \rangle)$. This together with the above equivalences yields $\{y\} = \partial R^*(x^*)$. Hence, Theorem 3.5.5 implies that $y = \nabla R^*(x^*)$, thus proving (a) and (b). Finally, (c) follows directly from properties (5.1.ii) (attainability) and (5.1.i) (uniqueness due to strict convexity) of a mirror map. □

One may have noticed that we have not used in the above proof the relative interior condition between the sets X and D from the definition of mirror map. However, all of the results about Online Mirror Descent algorithms which we describe here (and most in the literature, even if not clearly stated) need this type of condition in one way or another, such as for applying Lemma 4.3.2 to prove good regret bounds or to use optimality conditions for Bregman projections from Lemma 3.11.4. Therefore, cases which do not satisfy these relative interior conditions pratically render most of the results of this chapter useless, and are pathological. For this reason, we have chosen to add this condition to the definition of a mirror map from the beginning.

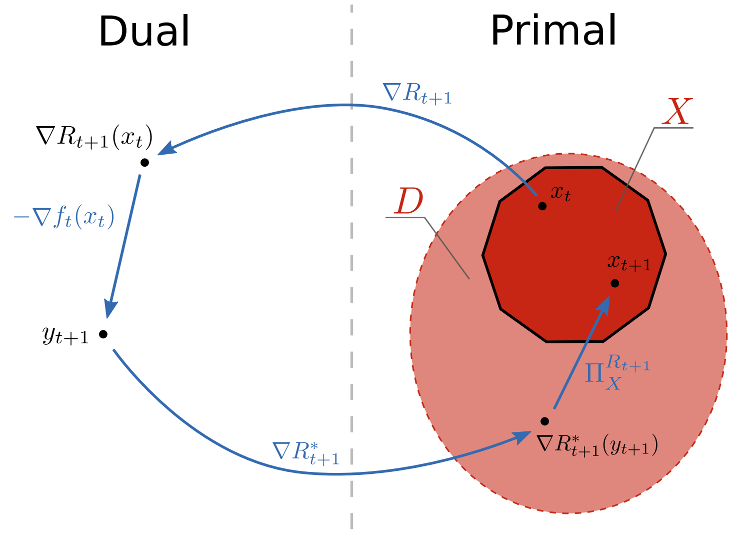

Figure 5.1: Graphic representation of the computations done by AdaOMD on round $t + 1$.

Let us now look at the intuition behind the (Adaptive) Online Mirror Descent. On Figure 5.1 we present an schematic diagram of the computations done by AdaOMD on round $t + 1$. The idea is that gradients are representations of the derivatives, which are linear functionals on $\mathbb{E}$. That is, the gradients represent elements from the dual space $\mathbb{E}^*$ of $\mathbb{E}$, the space of linear functions from $\mathbb{E}$ to $\mathbb{R}$. Thus, a gradient step style update such as $x_t - \nabla f_t(x_t)$, where $x_t \in \mathbb{E}$ and $f_t \colon \mathbb{E} \to (-\infty, +\infty]$ is convex and differentiable[2], can be seen as actually dealing with the functionals $\langle x_t, \cdot \rangle$ and $\langle \nabla f_t(x_t), \cdot \rangle$, and summing x_t and the derivative of f_t at x_t only makes sense due to Riesz representation theorem, which ensures the existence of the representation of the derivative of f_t at x_t (namely, the gradient $\nabla f_t(x_t)$) on the primal space. Without using this theorem, we need to transport the point x_t from the primal to the dual space in some way. Since we are in an Euclidean space, we may correspond $x_t \in \mathbb{E}$ with $\langle x_t, \cdot \rangle \in \mathbb{E}^*$ (which yields the Gradient Descent method), but this choice is more or less arbitrary: for each differentiable function R, we can build the linear function $\langle \nabla R(x_t), \cdot \rangle$. Thus, we should chose a function R_{t+1}, our mirror map at round $t+1$, to make such correspondences. With that connection between $\mathbb{E}$ and $\mathbb{E}^*$ made, the gradient step update becomes $\nabla R_{t+1}(x_t) - \nabla f_t(x_t)$. Yet, this update produces a point y_{t+1} in the dual space $\mathbb{E}^*$, and we need to correspond it back to a point in the primal space. Such a correspondence should, intuitively, be the inverse of the mapping $x \in \mathbb{E} \mapsto \nabla R_{t+1}(x)$. By Proposition 5.1.3 we know that R_{t+1}^* is differentiable everywhere, and by Corollary 3.5.6 we have that ∇R_{t+1}^* is the inverse mapping of ∇R_{t+1}. Thus, we correspond the point y_{t+1} in the dual with the point $\nabla R_{t+1}^*(y_{t+1})$ in the primal space. There is, still, one last factor to discuss: the point $\nabla R_{t+1}^*(y_{t+1})$ may be outside of the set $X \subseteq \mathbb{E}$ where our optimization is actually taking place. Thus, we project $\nabla R_{t+1}^*(y_{t+1})$ onto X with the Bregman projector based on R_{t+1}, yielding a point $x_{t+1} \in X$. We will see why the use of this type of projection is at least partially intuitive on Section 5.4. After this projection, we can again correspond x_{t+1} with a dual point through ∇R_{t+2} and make a gradient step, starting the process again.

5.2 Non-Adaptive Online Mirror Descent and Examples

As we have done in Chapter 4 with the FTRL algorithm, let us look at the non-adaptive version of Online Mirror Descent. Besides the fact that looking at a special and simpler case of the AdaOMD algorithm may help us to understand better how it works, the non-adaptive version is the one which is usually presented by other authors as the Online Mirror Descent algorithm, such as in [19, 36][3], or with time varying constants (step sizes) multiplying a fixed mirror map as in [13].

In Algorithm 5.2 we define the EOMD oracle, which implements the *(Eager) Online Mirror Descent* algorithm. The reason we call this the eager version will become clearer when we present the lazy one on Section 5.5.

As we have mentioned in the beginning of this chapter, Online Mirror Descent with the squared ℓ_2-norm used as a mirror map yields the (Projected) Subgradient Descent algorithm [36, 72].

Lemma 5.2.1. Let $X \subseteq \mathbb{R}^d$ be a nonempty closed convex set, let $\eta \in \mathbb{R}_{++}$, and define the function $R \coloneqq \frac{1}{2\eta}\|\cdot\|_2^2$. Then, R is a 1-strongly convex w.r.t. $\|\cdot\|_2$ classical mirror map for X such that, for any $y \in \mathbb{R}^d$,

$$\Pi_X^R(y) = \operatorname*{arg\,min}_{x \in X} \|x - y\|_2, \qquad \nabla R(y) = \frac{1}{\eta} y, \text{ and } \qquad \nabla R^*(y) = \eta y. \tag{5.3}$$

Proof. We know that R is continuous (and, thus, closed) and, by Lemma 3.9.5, we know that R is strongly convex on $\mathbb{R}^d$ and, in particular, strictly convex on $\mathbb{R}^d$. That is, R satisfies (5.1.i). Since R is

[2] Even though we do not require in the original algorithm differentiability of functions f_t, the intuition discussed here fits almost seamlessly in the non-differentiable case, since subgradients are represententions of the *directional* derivatives of a function. Still, we have chosen to discuss the intuition with differentiable functions for the sake of clarity and simplicity.

[3] We note that in these works, the authors define the algorithm with a positive constant η (a step size) multiplying the subgradient at each step, which is the same as multiplying the mirror map by η^{-1}.

Algorithm 5.2 Definition of $\mathrm{EOMD}_R^X(\langle f_1, \dots, f_T\rangle)$

```
Input:
    (i) A closed convex set $X \subseteq \mathbb{E}$,
    (ii) Convex functions $f_1, \dots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$ such that $f_t$ is
         subdifferentiable on $X$ for each $t \in [T]$,
    (iii) A mirror map $R\colon \mathbb{E} \to (-\infty, +\infty]$ for the OCO instance $(X, \mathcal{F})$.
Output: $x_{T+1} \in \operatorname{int}(\operatorname{dom} R) \cap X$
  $\{x_1\} \leftarrow \arg\min_{x \in X} R(x)$
  $y_1 \leftarrow 0$.
  for $t = 1$ to $T$ do
        ▷ Computations for round $t+1$
      Compute $g_t \in \partial f_t(x_t)$
      $y_{t+1} \leftarrow \nabla R(x_t) - g_t$
      $x_{t+1} \leftarrow \Pi_X^R(\nabla R^*(y_{t+1}))$
  return $x_{T+1}$
```

1-strongly convex w.r.t. the ℓ_2-norm on $\mathbb{R}^d$, then $B_R(\cdot, y)$ is also strongly convex on $\mathbb{R}^d$ for any $y \in \mathbb{R}^d$. Thus, by Lemma 3.9.14 we have that R satisfies (5.1.ii) since X is closed, that is, $\inf_{x \in X} R(x)$ and $\inf_{x \in X} B_R(x, y)$ are both attained for any $y \in \mathbb{R}^d$. We have $\operatorname{int}(\operatorname{dom} R) \cap \operatorname{ri}(X) = \mathbb{R}^d \cap \operatorname{ri}(X) \neq \varnothing$, which implies that (5.1.iii) is satisfied. Finally, since $\nabla R(x) = \eta x$ for every $x \in \mathbb{R}^d$, we have $\{\, \nabla R(x) : x \in \mathbb{R}^d \} = \eta \mathbb{R}^d = \mathbb{R}^d$, that is, R satisfies property (5.1.iv). Therefore, R is a 1-strongly convex w.r.t. $\|\cdot\|_2$ classical mirror map for X. Let us now show that (5.3) holds.

First, note that for every $x, y \in \mathbb{R}^d$,

$$B_R(x, y) = \tfrac{1}{2\eta}(\|x\|_2^2 - \|y\|_2^2 - 2\langle y, x - y\rangle) = \tfrac{1}{2\eta}(\|x\|_2^2 - 2\langle y, x\rangle + \|y\|_2^2) = \tfrac{1}{2\eta}\|x - y\|_2^2.$$

That is, $\Pi_X^R(y) = \arg\min_{x \in X}\|x - y\|_2^2 = \arg\min_{x \in X}\|x - y\|_2$. Moreover, it is clear that $\nabla R(y) = \frac{1}{\eta} y$ for any $y \in \mathbb{R}^d$. Finally, by Theorem 3.8.2, for any norm $\|\cdot\|$ on $\mathbb{R}^d$ (or on any euclidean space) we have $\left(\frac{1}{2}\|\cdot\|^2\right)^* = \frac{1}{2}\|\cdot\|_*^2$. Since the ℓ_2-norm is dual to itself, Theorem 3.4.3 yields, for every $y \in \mathbb{R}^d$,

$$R^*(y) = \frac{1}{2\eta}\|\eta y\|_2^2 \implies \nabla R^*(y) = \eta y. \qquad \square$$

Proposition 5.2.2. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that $X \subseteq \mathbb{R}^d$ is closed and that every function in $\mathcal{F}$ is subdifferentiable on X. Let $T \in \mathbb{N}$, let ENEMY be an enemy oracle for $\mathcal{C}$, let $\eta \in \mathbb{R}_{++}$, define $R \coloneqq \frac{1}{2\eta}\|\cdot\|_2^2$, and set

$$(\boldsymbol{x}, \boldsymbol{f}) \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{EOMD}_R^X, \mathrm{ENEMY}, T).$$

Finally, let $g_t \in \partial f_t(x_t)$ be as in the definition of $\mathrm{EOMD}_R^X(\boldsymbol{f})$ for each $t \in [T]$. Then,

$$x_t = \Pi_X^R\big([t > 1](x_{t-1} - \eta g_{t-1})\big), \qquad \forall t \in [T]. \tag{5.4}$$

Proof. By Lemma 5.2.1, R is a mirror map for X, and thus EOMD_R^X is well-defined. Let us now show that (5.4) holds. First, note that

$$\{x_1\} = \arg\min_{x \in X} R(x) = \arg\min_{x \in X}\|x\|_2 = \Pi_X^R(0).$$

Let $t \in \{2, \dots, T\}$ and let $y_t \in \mathbb{R}^d$ and $g_{t-1} \in \partial f_{t-1}(x_{t-1})$ be as in the definition of $\mathrm{EOMD}_R^X(\boldsymbol{f})$. By the definition of EOMD_R^X, $y_t = \nabla R(x_{t-1}) - g_{t-1} = \frac{1}{\eta} x_{t-1} - g_{t-1}$. Moreover, again by the definition of EOMD_R^X, we have

$$x_t = \Pi_X^R\big(\nabla R^*(y_t)\big) = \Pi_X^R\big(\eta(y_t)\big) = \Pi_X^R\big(x_{t-1} - \eta g_{t-1}\big). \qquad \square$$

The above proposition shows that EOMD with the squared ℓ_2-norm as a mirror map has a neat closed formula. Thus, one may hope that this is the case for other important cases, such as for the negative entropy regularizer used in Section 4.5 to obtain an exponential improvement on the regret guarantees of FTRL on the randomized experts' problem. Let us now show that this is indeed the case: applying EOMD with the negative entropy as a mirror map yields the algorithm known as *Exponentiated Gradient Descent*, *Hedge* [32] or, in the special case of the expert's problem, as *Multiplicative Weights Update Method with exponential updates* (see [6]).

Lemma 5.2.3. Let $\eta \in \mathbb{R}_{++}$ and define $R(x) \coloneqq \frac{1}{\eta}\sum_{i=1}^d [x_i > 0] x_i \ln x_i + \delta(x \,|\, \mathbb{R}^d_+)$ for every $\mathbb{R}^d$. Then R is a mirror map for Δ_d which is differentiable on $\mathbb{R}^d_{++}$ and which is 1-strongly convex on Δ_d w.r.t. $\|\cdot\|_1$.

Proof. By Lemma 3.9.10, we know that for every $\theta \in \mathbb{R}_{++}$ the function R is closed and $\frac{1}{\eta\theta}$-strongly convex w.r.t. $\|\cdot\|_1$ on $B_\theta \coloneqq \{\, x \in \mathbb{R}^d : \|x\|_1 \leq \theta \,\}$. In particular, this implies that R is strictly convex on $\mathbb{R}^d_+$ since, for any distinct $x, y \in \mathbb{R}^d_+$ there is $\theta \in \mathbb{R}_{++}$ such that $\|x\|_1 \leq \theta$ and $\|y\|_1 \leq \theta$, which implies that for any $\lambda \in (0,1)$ we have

$$R(\lambda x + (1-\lambda) y) \leq \lambda R(x) + (1-\lambda) R(y) - \lambda(1-\lambda)\frac{1}{2\eta\theta}\|x - y\|_1^2 < \lambda R(x) + (1-\lambda) R(y).$$

Thus, R satisfies condition (5.1.i).

Moreover, by Proposition 3.11.5 we have that for any $y \in \mathbb{R}^d_{++}$ the infimum $\inf_{x \in \Delta_d} B_R(x, y)$ is attained by $\|y\|_1^{-1} y \in \mathbb{R}^d_{++}$. To show that $\inf_{x \in X} R(x)$ is attained, note that by Proposition 3.11.5 we have

$$B_R(x, \mathbb{1}) = \tfrac{1}{\eta}\sum_{i=1}^d ([x_i > 0] x_i \ln x_i + (x_i - 1)) = R(x) + 1 - d.$$

Thus, $\arg\min_{x \in \Delta_d} R(x) = \arg\min_{x \in \Delta_d} B_R(x, \mathbb{1})$, and $d^{-1}\mathbb{1}$ is contained in the latter again by Proposition 3.11.5. That is, R satisfies condition (5.1.ii) of a mirror map. Additionally, by Corollary 3.2.3, we have $\operatorname{ri}\Delta_d = \{\, x \in (0,1]^d : \mathbb{1}^\mathsf{T} x = 1 \,\}$. Thus, $\mathbb{R}^d_{++} \cap \operatorname{ri}\Delta_d \neq \varnothing$ and, hence, R satisfies (5.1.iii).

Finally, let us show that $\{\, \nabla R(x) : x \in \mathbb{R}^d_{++} \,\} = \mathbb{R}^d$, condition (5.1.iv) from the definition of mirror map. By Proposition 3.4.4, we have that R^* is differentiable on $\mathbb{R}^d$ and that

$$\nabla R^*(x^*)(i) = \frac{1}{\eta} e^{\eta x^*(i) - 1} > 0, \qquad \forall i \in [d], \forall x^* \in \mathbb{R}^d$$

Thus, $\nabla R^*(x^*) \in \mathbb{R}^d_{++}$ for any $x^* \in \mathbb{R}^d$. By Corollary 3.5.6 we have $\nabla R(\nabla R^*(x^*)) = x^*$ for each $x^* \in \mathbb{R}^d$. That is, for every $x^* \in \mathbb{R}^d$ there is $x \in \mathbb{R}^d_{++}$ (namely, $\nabla R^*(x^*)$) such that $\nabla R(x) = x^*$. This completes the proof that R is a mirror map for Δ_d. $\square$

Proposition 5.2.4. Let $\mathcal{C} \coloneqq (\Delta_d, \mathcal{F})$ be an OCO instance such that every function in $\mathcal{F}$ is subdifferentiable on Δ_d. Let $T \in \mathbb{N}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and define $R(x) \coloneqq \frac{1}{\eta}\sum_{i=1}^d [x_i > 0] x_i \ln x_i + \delta(x \,|\, \mathbb{R}^d_+)$ for each $x \in \mathbb{R}^d$, where $\eta \in \mathbb{R}$ is some positive constant. Finally, set

$$(\boldsymbol{x}, \boldsymbol{f}) \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{EOMD}_R^X, \mathrm{ENEMY}, T)$$

and let $g_t \in \partial f_t(x_t)$ be as in the definition of $\mathrm{EOMD}_R^X(\boldsymbol{f})$ for each $t \in [T]$. Then, $x_1 = d^{-1}\mathbb{1}$ and

$$x_{t+1}(i) = \frac{1}{\omega_{t+1}}(x_t(i))e^{-\eta g_t(i)}, \text{ where } \omega_{t+1} \coloneqq \sum_{j=1}^{d}(x_t(j))e^{-\eta g_t(j)} \qquad \forall i \in [d], \forall t \in \{1, \ldots, T-1\}.$$

Proof. By Lemma 5.2.3, we know that R is a mirror map for Δ_d. Thus, it only remains to prove the form of the points x_t for $t \in [T]$. First of all, by Proposition 3.11.5 we have

$$\{d^{-1}\mathbb{1}\} = \operatorname*{arg\,min}_{x \in \Delta_d} B_R(x, \mathbb{1}) = \operatorname*{arg\,min}_{x \in \Delta_d}(R(x) + \|x\|_1 - \|\mathbb{1}\|_1) = \operatorname*{arg\,min}_{x \in \Delta_d}(R(x) + 1 - d) = \operatorname*{arg\,min}_{x \in \Delta_d} R(x),$$

and since $x_1 \in \operatorname{arg\,min}_{x \in \Delta_d} R(x)$ by definition, we conclude that $x_1 = d^{-1}\mathbb{1}$. Let $y_t \in \mathbb{R}^d$ be as in the definition of $\mathrm{EOMD}_R^X(\boldsymbol{f})$ for each $t \in [T]$. For every $t \in \{1, \ldots, T-1\}$ we have by definition of the algorithm that

$$y_{t+1}(i) = \nabla R(x_t)(i) - g_t(i) = \tfrac{1}{\eta}\Big(1 + \ln\big(x_t(i)\big)\Big) - g_t(i), \qquad \forall i \in [d].$$

By Proposition 3.4.4, $R^*(z) = \frac{1}{\eta}\sum_{i=1}^d e^{\eta z(i) - 1}$, we for every $z \in \mathbb{R}^d$, which implies

$$\nabla R^*(z)(i) = e^{\eta z(i) - 1} = \exp(\eta z_i - 1), \qquad \forall i \in [d], \forall z \in \mathbb{R}^d.$$

Therefore, for every $t \in \{1, \ldots, T-1\}$

$$\nabla R^*(y_{t+1})(i) = \exp(\eta(y_{t+1}(i)) - 1) = \exp(1 + \ln(x_t(i)) - \eta g_t(i) - 1) = (x_t(i))e^{-\eta g_t(i)}, \qquad \forall i \in [d].$$

Since $x_{t+1} = \Pi_{\Delta_d}^R(\nabla R^*(y_{t+1}))$, and since by Proposition 3.11.5 the Bregman Projector on Δ_d w.r.t. R boils down to a normalization w.r.t. the ℓ_1-norm, we are done. □

The update rules derived by the mirror maps in the above propositions are simple to implement and well-studied in the optimization literature. Thus, one may hope to unify convergence proofs of many existing algorithms with a general convergence proof of Online Mirror Descent. Indeed, Adaptive OMD happens to have many interesting connections with proximal operators and with Adaptive FTRL. The connections with the former allows us to write the general update rule from AdaOMD in a closed formula which sheds light on its inner workings. The connection with Adaptive FTRL gives yet another perspective on how AdaOMD works, and yields regret bounds through the use of the Strong FTRL Lemma.

5.3 OMD Connection to Proximal Operators

In this section we will look at the connections between OMD algorithms and *proximal operators* and, for that, we will mostly follow the presentations from [13, 57]. Let $f : \mathbb{E} \to (-\infty, +\infty]$ be a proper closed convex function. The **proximal operator** or **proximal map** of f, first studied independently by Moreau [51, 52, 53] and Rockafellar [61] (see [60, Section 1.H] for details and historical notes), is the function $\mathrm{prox}_f : \mathbb{E} \to \mathbb{E}$ given by

$$\{\mathrm{prox}_f(\bar{x})\} \coloneqq \operatorname*{arg\,min}_{x \in \mathbb{E}}\big(f(x) + \tfrac{1}{2}\|x - \bar{x}\|_2^2\big), \tag{5.5}$$

where $\|\cdot\|_2$ is the norm induced by the inner product from $\mathbb{E}$ or the ℓ_2-norm on $\mathbb{E}$. Since f is proper and closed, and since the squared ℓ_2-norm is strongly convex (see Lemma 3.9.5), we know by

Lemma 3.9.14 that the above set is indeed a singleton. The name *proximal* comes from the fact that if we use $f \coloneqq \delta(\cdot \,|\, X)$ for some closed set $X \subseteq \mathbb{E}$, the point given by the proximal operator for $\bar{x} \in \mathbb{E}$ is exactly (one of) the closest point(s) to $\bar{x}$ in X (w.r.t. the ℓ_2-norm).

If we look more carefully at (5.5), we can see a clear intuition of the inner workings of $\operatorname{prox}_f(\bar{x})$: it tries to find a point that approximately minimizes f without going too far away from $\bar{x}$. Interestingly, if we want to find a point which is allowed to be closer (or farther) from $\bar{x}$ when compared to $\operatorname{prox}_f(\bar{x})$ while still approximately minimizing f, it suffices to apply the proximal operator to a positive multiple of f. To see this, note that for any $\lambda \in \mathbb{R}_{++}$ and $\bar{x} \in \mathbb{E}$ we have

$$\{\operatorname{prox}_{\lambda f}(\bar{x})\} = \operatorname*{arg\,min}_{x \in \mathbb{E}} \Big(\lambda f(x) + \frac{1}{2}\|x - \bar{x}\|_2^2\Big) = \operatorname*{arg\,min}_{x \in \mathbb{E}} \Big(f(x) + \frac{1}{2\lambda}\|x - \bar{x}\|_2^2\Big).$$

Thus, if $\lambda \in \mathbb{R}_{++}$ is bigger than 1, the point $\operatorname{prox}_{\lambda f}(\bar{x})$ is allowed to be farther from $\bar{x}$ than $\operatorname{prox}_f(\bar{x})$. In the same way, if $\lambda \in \mathbb{R}_{++}$ is smaller than 1, $\operatorname{prox}_{\lambda f}(\bar{x})$ is probably closer to $\bar{x}$ than $\operatorname{prox}_f(\bar{x})$. The intuition on proximal operator we have discussed is close to the problem of minimizing f. Thus, comes as no surprise that some of the proximal operator's properties are closely related to the the minimizers of f on $\mathbb{E}$ (if any). Indeed, the following proposition[4] shows one (of the many) interesting properties of the proximal map. It shows that the fixed points of prox_f are exactly the minimizers of f.

Proposition 5.3.1. Let $f \colon \mathbb{E} \to (-\infty, +\infty]$ be a proper closed convex function and let $\bar{x} \in \mathbb{E}$. Then $\operatorname{prox}_f(\bar{x}) = \bar{x}$ if and only if $\bar{x}$ attains $\inf_{x \in \mathbb{E}} f(x)$.

Proof. Suppose $\bar{x}$ attains $\inf_{x \in \mathbb{E}} f(x)$. Then, for every $x \in \mathbb{E}$,

$$f(x) + \tfrac{1}{2}\|x - \bar{x}\|_2^2 \geq f(x) \geq f(\bar{x}) = f(\bar{x}) + \tfrac{1}{2}\|\bar{x} - \bar{x}\|_2^2.$$

Suppose now that $\operatorname{prox}_f(\bar{x}) = \bar{x}$ and define $h(x) \coloneqq f(x) + \frac{1}{2}\|x - \bar{x}\|_2^2$ for every $x \in \mathbb{E}$. This way, we have that $\bar{x}$ attain $\inf_{x \in \mathbb{E}} h(x)$. Note that $x \in \mathbb{E} \mapsto \frac{1}{2}\|x - \bar{x}\|_2^2$ is a differentiable function with gradient $x - \bar{x}$ at $x \in \mathbb{E}$. Thus, by Theorems 3.5.4 and 3.5.5, for any $x \in \mathbb{E}$ we have

$$\partial h(x) = \partial f(x) + x - \bar{x}.$$

By the definition of subgradient, we have that $\bar{x}$ attains $\inf_{x \in \mathbb{E}} h(x)$ if and only if $0 \in \partial h(\bar{x}) = \partial f(\bar{x}) + \bar{x} - \bar{x} = \partial f(\bar{x})$, that is, $\bar{x}$ attains $\inf_{x \in \mathbb{E}} f(x)$. □

Since the idea behind prox_f is to (at least approximately) minimize f, it is natural to ask whether it can be applied to optimization. However, the proximal map itself is an optimization problem, and if it is not efficiently solvable, it is of no help in optimizing f efficiently. Luckily, in many applications of proximal operators there are efficient ways to compute this operator. One example is when the function f to be minimized can be written as a sum of two functions, one differentiable but "hard" to optimize over, and one non-smooth function but easier to handle in the proximal map. In this case, one can handle the smooth function with a gradient step, the non-smooth part one handles by computing the proximal map at the current iterate. This line of thought is not our focus here, but one can find more information and related work in [57].

Another idea to make the proximal map prox_f for a proper convex function $f \colon \mathbb{E} \to (-\infty, +\infty]$ on a given point $\bar{x} \in \mathbb{E}$ useful in optimizing f even when we cannot compute prox_f efficiently is to substitute f by an approximation $\tilde{f}$ of f which is easier to handle. If f is subdifferentiable at $\bar{x}$ and

[4] This proposition is not used in the remainder of the text, but it aids to build intuition about proximal operators.

$g \in \partial f(\bar{x})$, then an intuitive approximation candidate is given by $\tilde{f}(x) \coloneqq f(\bar{x}) + \langle g, x - \bar{x} \rangle$ for every $x \in \mathbb{E}$. With this function $\tilde{f}$, we have

$$\{\operatorname{prox}_{\tilde{f}}(\bar{x})\} = \operatorname*{arg\,min}_{x \in \mathbb{E}} \bigl(\langle g, x \rangle + \tfrac{1}{2}\|x - \bar{x}\|_2^2\bigr).$$

By the definition of subgradient, 0 must be a subgradient at $\operatorname{prox}_{\tilde{f}}(\bar{x})$ of the function being minimized above. Since $x \in \mathbb{E} \to \langle g, x \rangle + \frac{1}{2}\|x - \bar{x}\|_2^2$ is a proper closed convex everywhere differentiable function, by Theorem 3.5.5 we know that its only subgradient is its gradient. Thus, the gradient of $x \in \mathbb{E} \to \langle g, x \rangle + \frac{1}{2}\|x - \bar{x}\|_2^2$ at $\operatorname{prox}_{\tilde{f}}(\bar{x})$ is 0, that is,

$$0 = g + \operatorname{prox}_{\tilde{f}}(\bar{x}) - \bar{x} \iff \operatorname{prox}_{\tilde{f}}(\bar{x}) = \bar{x} - g.$$

Therefore, the proximal map in this case is just a step in the direction of minus subgradient! Here we used a step size of 1 for the sake of simplicity, but as we have discussed, one can just scale $\tilde{f}$ to get a subgradient descent with an arbitrary positive step size.

In the above discussion, we have seen how to view a subgradient step from a point x for the function f as the application of a proximal map based on a subgradient of f at x. Moreover, in the previous section we have seen that the online subgradient descent algorithm is just a special case of online mirror descent. Thus, one may ask if there is any generalized proximal operator which generalizes many algorithms in a way similar to the way OMD generalizes many Online Convex Optimization algorithms. Indeed, the first appearance of the proximal operator, which is exactly the one we defined as prox_f, uses the squared ℓ_2-norm as a way to measure distances. One possible generalization replaces this norm with a distance-like function $D\colon \mathbb{E} \times \mathbb{E} \to (-\infty, +\infty]$. In this way, the form of this generalized proximal operator when applied to a linear function[5] $\langle g, \cdot \rangle$ at a point $\bar{x} \in \mathbb{E}$ is

$$\operatorname*{arg\,min}_{x \in \mathbb{E}} \{\langle g, x \rangle + D(x, \bar{x})\}.$$

The next theorem shows that each iterate of the Adaptive OMD is given by a formula as the one above, with the minimization taking place in the set X instead of the whole space. Additionally, the distance-like functions used in the theorem seem incredibly intuitive given what we have already seen about OMD: it is the Bregman divergence w.r.t. the mirror map at the current round. Moreover, the proof of the next theorem follows the lines of the discussion we had above about the subgradient step, that is, we just look at the optimality conditions of the generalized proximal step, and conclude that the generalized proximal is exactly the Adaptive OMD step.

Theorem 5.3.2. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that X is closed and such that each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X. Let $\mathcal{R}\colon \operatorname{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ be a mirror map strategy for $\mathcal{C}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and let $T \in \mathbb{N}$. Define

$$(\boldsymbol{x}, \boldsymbol{f}) \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaOMD}_{\mathcal{R}}^{X}, \mathrm{ENEMY}, T)$$

$$\text{and} \qquad R_t \coloneqq \sum_{i=1}^{t} \mathcal{R}(\langle f_1, \ldots, f_{t-1} \rangle), \qquad \text{for each } t \in [T].$$

Moreover, let $g_t \in \partial f_t(x_t)$ be the same as in the definition of $\mathrm{AdaOMD}_{\mathcal{R}}^{X}(\boldsymbol{f})$ on Algorithm 5.1 for each $t \in [T]$. Set $\{x_1'\} \coloneqq \operatorname{arg\,min}_{x \in X} R_1(x)$ and define

$$\{x_{t+1}'\} \coloneqq \operatorname*{arg\,min}_{x \in X} \bigl(\langle g_t, x \rangle + B_{R_{t+1}}(x, x_t')\bigr), \qquad \forall t \in \{1, \ldots, T-1\}. \tag{5.6}$$

Then $x_t = x_t'$ for each $t \in [T]$.

[5] Recall that we saw in the discussion how the proximal applied to the *linearized function* yields the subgradient step, not the proximal operator applied to the function itself.

Proof. Let us prove the statement by induction on $t \in [T]$. For $t = 1$ the statement holds by the definitions of x_1 and x'_1. Let $t \in \{1, \dots, T-1\}$ and suppose $x_t = x'_t$. Define

$$y_{t+1} \coloneqq \nabla R_{t+1}(x_t) - g_t = \nabla R_{t+1}(x'_t) - g_t.$$

By the definition of AdaOMD$_{\mathcal{R}}$ on Algorithm 5.1, we have that $x_{t+1} = \Pi_X^{R_{t+1}}(\nabla R^*_{t+1}(y_{t+1}))$. Proposition 5.1.3 states that, since R_{t+1} is a mirror map for X, then R^*_{t+1} is differentiable on $\mathbb{E}$ and R_{t+1} is differentiable on $\nabla R^*_{t+1}(y_{t+1}) \in \operatorname{int}(\operatorname{dom} R_{t+1})$. Thus, since $(\operatorname{int}(\operatorname{dom} R_{t+1})) \cap \operatorname{ri}(X) \neq \varnothing$ by the definition of mirror map, Lemma 3.11.4 yields

$$x_{t+1} = \Pi_X^{R_{t+1}}(\nabla R^*_{t+1}(y_{t+1})) \iff y_{t+1} - \nabla R_{t+1}(x_{t+1}) \in N_X(x_{t+1}). \tag{5.7}$$

Note that

$$y_{t+1} - \nabla R_{t+1}(x_{t+1}) = \nabla R_{t+1}(x'_t) - g_t - \nabla R_{t+1}(x_{t+1}) = -g_t - (\nabla B_{R_{t+1}}(\cdot, x'_t))(x_{t+1}),$$

and the latter is minus the gradient of $x \in \mathbb{E} \mapsto g_t^\mathsf{T} x + B_{R_{t+1}}(x, x'_t)$ at x_{t+1}. Therefore, using (5.7) with the above equation and by the optimality conditions from Theorem 3.6.2,

$$\begin{aligned} x_{t+1} = \Pi_X^{R_{t+1}}(\nabla R^*_{t+1}(y_{t+1})) &\iff -(g_t + (\nabla B_{R_{t+1}}(\cdot, x'_t))(x_{t+1})) \in N_X(x_{t+1}) \\ &\iff x_{t+1} \in \operatorname*{arg\,min}_{x \in X}\big(\langle g_t, x\rangle + B_{R_{t+1}}(x, x'_t)\big). \end{aligned} \qquad \square$$

This connection between Adaptive OMD and generalized proximal operators gives us yet another intuitive view of online mirror descent. At round t, the algorithm minimizes, as much as possible, the value of the linearized version of the function f_{t-1} played by enemy on the last round, but tries to not pick a point too far from the last iterate w.r.t. the Bregman divergence based on the current mirror map. In the next section, we will see that the Adaptive OMD can be seen as an application of the Adaptive FTRL algorithm.

5.4 OMD Connection with FTRL and Regret Bounds

The form of a proximal operator is way closer to the minimization done by Follow the Regularized Leader algorithms. Still, there are some key differences. For example, in the connection shown between OMD and proximal operators, at each round we optimize only over the (linearized version of the) *last* function played by the enemy. On the other hand, at each round FTRL algorithms optimize over *all* the past functions played by the enemy. Still, note that OMD in proximal operator form uses the point picked by the player in the last round, while FTRL look only at the functions played by the enemy. Thus, we may still hope reduce one of them to the other.

Another interesting difference is that, unlike from the AdaFTRL oracle, the AdaOMD oracle needs to explicitly receive the set X where the player is allowed to pick her points. In the case of AdaFTRL this is not needed since adding the indicator function of a closed set X to the first regularizer already makes the point picked to lie in X. In the case of the Adaptive OMD oracle, it is not clear what is the effect of adding an indicator function to the mirror map since the gradients of the regularizers play a major role in the iterate update rules of the oracle. Interestingly, the following proposition says that, if we take strongly convex function plus the indicator function of a set X, the gradient of its conjugate boils down to the gradient of the conjugate of the original function projected onto X through a Bregman projection. The latter is exactly one of the steps done by the Adaptive OMD oracle.

Proposition 5.4.1. Let $X \subseteq \mathbb{E}$ be a closed convex set and let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a mirror map for X such that R is strongly convex on X. Finally, define $R_X \coloneqq R + \delta(\cdot \,|\, X)$. Then, for every $x \in \mathbb{E}$,

$$\nabla R_X^*(x) = \Pi_X^R(\nabla R^*(x)).$$

Proof. Note that the functions R^* and R_X^* are both differentiable everywhere, the former[6] by the properties of mirror maps given by Proposition 5.1.3 and the latter by Proposition 3.9.8 since R_X is actually strongly convex on $\mathbb{E}$. Moreover, since $\operatorname{int}(\operatorname{dom} R) \cap \operatorname{ri}(X)$ is nonempty and since $\partial(\delta(\cdot \,|\, X)(z) = N_X(z)$ by Lemma 3.5.3, we have by Theorems 3.5.4 and 3.5.5 that, for every $z \in \operatorname{int}(\operatorname{dom} R) \cap X$,

$$N_X(z) + \nabla R(z) = \partial(\delta(\cdot \,|\, X) + R)(z) = \partial R_X(z). \tag{5.8}$$

Finally, by (5.1.ii) from the mirror map definition, $\Pi_X^R(\nabla R^*(x)) \in \operatorname{int}(\operatorname{dom} R) \cap X$. Therefore, for any $x \in \mathbb{E}$ and $z \in \operatorname{int}(\operatorname{dom} R) \cap X$,

$$\begin{aligned} z = \Pi_X^R(\nabla R^*(x)) &\iff x - \nabla R(z) \in N_X(z) && \text{by Lemma 3.11.4,}\\ &\iff x \in \partial R_X(z) && \text{by (5.8),}\\ &\iff z \in \partial R_X^*(x) = \{\nabla R_X^*(x)\} && \text{by Theorems 3.5.2 and 3.5.5.} \quad \square \end{aligned}$$

The above proposition hints at the possibility of passing the restriction of the set X to the OMD oracles through an indicator function added to the mirror map and make it more similar to a FTRL algorithm. It only remains to look at the updates[7] of the type

$$y_{t+1} = \nabla R(x_t) - g_t \tag{5.9}$$

from Algorithm 5.2, where R is a mirror map for a set X from an OCO instance $\mathcal{C} \coloneqq (X, \mathcal{F})$. However, at this point the universe stops its acts of kindness. If we define $R_X \coloneqq R + \delta(\cdot \,|\, X)$, then, assuming that $(\operatorname{ri}(X)) \cap \operatorname{dom}(\operatorname{ri} R)$ is nonempty, by Theorem 3.5.4 we have $\partial R_X(x) = \nabla R(x) + N_X(x)$ for $x \in \operatorname{dom} R$. Since 0 is in in the normal cone of X at any point of X, we know that $\nabla R(x)$ is a subgradient of R_X at x, and thus EOMD might work as usual with such a mirror map if we use a subgradient of the mirror map instead of its gradient in (5.9). However, without explicit knowledge of the original mirror map R and of the indicator function of X, there is no way to know the "correct" subgradient of R_X to pick.

After the above discussion, one may guess that some connections between Adaptive OMD and Adaptive FTRL will involve points from the normal cone of X. The following theorem shows one connection in which this is exactly what happens.

Theorem 5.4.2. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that X is closed and such that each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X. Let $\mathcal{R}\colon \operatorname{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ be a mirror map strategy for $\mathcal{C}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and let $T \in \mathbb{N}$. Define

$$\begin{aligned} (\boldsymbol{x}, \boldsymbol{f}) &\coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaOMD}_{\mathcal{R}}^{X}, \mathrm{ENEMY}, T) \\ \text{and} \qquad r_t &\coloneqq \mathcal{R}(\langle f_1, \dots, f_{t-1} \rangle) \qquad \text{for each } t \in \{1, \dots, T+1\}. \end{aligned}$$

[6] At first sight, one may think that the differentiability of R^* is a consequence Proposition 3.9.8 as well. Note, however, that R is strongly convex only on a subset X of $\mathbb{E}$, and Proposition 3.9.8 only applies to functions which are strongly convex on the whole euclidean space. For FTRL regularizers this was never an issue since we already needed to add the indicator function of the set $X \subseteq \mathbb{R}^d$ where the player could pick her points and on which, usually, the regularizer was strongly convex.

[7] We will look at the EOMD oracle for the sake of simplicity, but the same discussion holds for the AdaOMD oracle.

Moreover, let $g_t \in \partial f_t(x_t)$ be the same as in the definition of $\mathrm{AdaOMD}^X_{\mathcal{R}}(\boldsymbol{f}_{1:t})$ on Algorithm 5.1 for each $t \in [T]$. Then, there is $p_t \in N_X(x_t)$ for each $t \in [T]$ such that, for every $t \in [T]$, both infima

$$\inf_{x \in X} \Big(\sum_{i=1}^{t-1} \langle g_i, x\rangle + r_1(x) + \sum_{i=1}^{t-1} (B_{r_{i+1}}(x, x_i) + \langle p_i, x \rangle) \Big) \text{ and } \tag{5.10}$$

$$\inf_{x \in \mathbb{E}} \Big(\sum_{i=1}^{t-1} \langle g_i, x\rangle + r_1(x) + \langle p_t, x\rangle + \sum_{i=1}^{t-1} (B_{r_{i+1}}(x, x_i) + \langle p_i, x \rangle) \Big) \tag{5.11}$$

are attained only by x_t. In particular, if we define $\mathcal{R}' \colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ by

$$\mathcal{R}'(\langle\rangle) \coloneqq r_1 + \langle p_1, \cdot \rangle + \delta(\cdot \,|\, X)$$

and by

$$\mathcal{R}'(\boldsymbol{f}) \coloneqq B_{r_{t+1}}(\cdot, x_t) + \langle p_{t+1}, \cdot \rangle, \qquad \forall \boldsymbol{f} \in \mathcal{F}^t, \forall t \in \{1, \dots, T-1\},$$

and define $h_t \coloneqq \langle g_t, \cdot \rangle$ for each $t \in [T]$, then $\mathrm{AdaOMD}^X_{\mathcal{R}}(\boldsymbol{f}_{1:t-1}) = \mathrm{AdaFTRL}_{\mathcal{R}'}(\boldsymbol{h}_{1:t-1})$ for each round $t \in [T]$.

Proof. Set $x_{T+1} \coloneqq \mathrm{AdaOMD}^X_{\mathcal{R}}(\boldsymbol{f})$ and define $p_1, \dots, p_T \in \mathbb{E}$, in order, by

$$p_t = -\Big(\sum_{i=1}^{t-1} g_i + \nabla r_1(x_t) + \sum_{i=1}^{t-1} (\nabla r_{i+1}(x_t) - \nabla r_{i+1}(x_i) + p_i) \Big), \qquad \forall t \in [T].$$

First, let us show that

$p_t \in N_X(x_t)$ for each $t \in [T]$ and x_t is the unique point that attains the infimum in (5.10) for each $t \in \{1, \dots, T+1\}$. (5.12)

Let us prove (5.12) by induction on $T \in \mathbb{N}$. For $T = 0$, we have that (5.10) for $t = 0$ is $\inf_{x \in X} r_1(x)$. By (5.1.ii) the latter infimum is attained, and $x_1 \in \arg\min_{x \in X} r_1(x)$ by the definition of the $\mathrm{AdaOMD}^X_{\mathcal{R}}$ oracle. Let $T \in \mathbb{N} \setminus \{0\}$. By induction hypothesis, (5.12) holds for $T - 1$, that is, the points $p_1, \dots, p_{T-1}$ are in normal cones as described in (5.12). Thus, we have that

$$\{x_T\} = \arg\min_{x \in X} \Big(\sum_{t=1}^{T-1} \langle g_t, x \rangle + r_1(x) + \sum_{t=1}^{T-1} (B_{r_{t+1}}(x, x_t) + \langle p_t, x \rangle) \Big)$$

Define $H(x) \coloneqq \Big(\sum_{t=1}^{T-1} \langle g_t, x\rangle + r_1(x) + \sum_{t=1}^{T-1} (B_{r_{t+1}}(x, x_t) + \langle p_t, x \rangle) \Big)$ for every $x \in \mathbb{E}$. Set $D \coloneqq \mathrm{int}(\mathrm{dom}\, r_T)$ (which is nonempty and is such that $D = \mathrm{int}(\mathrm{dom}\, r_t)$ for $t \in [T]$ by the definition of mirror map strategy). One can easily see that $\mathrm{int}(\mathrm{dom}\, H) = D$ since H is the sum of Bregman divergences w.r.t. the regularizer increments and linear functions. Since $(\mathrm{ri}(X)) \cap D \neq \varnothing$ (by the definition of mirror map strategy) we can use the optimality conditions from Theorem 3.6.2. That is, x_T attains $\inf_{x \in X} H(x)$ if and only if $(-\partial H(x_T)) \cap N_X(x_T)$ is nonempty. Since $x_T \in D$ by the guarantees of the $\mathrm{AdaOMD}^X_{\mathcal{R}}$ oracle, we have that r_t is differentiable at x_T for each $t \in [T]$. Therefore, H is differentiable at x_T, and by Theorem 3.5.5 we have $\partial H(x_T) = \{\nabla H(x_T)\}$. Therefore, x_T attains $\inf_{x \in X} H(X)$ if and only if

$$-\Big(\sum_{t=1}^{T-1} g_t + \nabla r_1(x_T) + \sum_{t=1}^{T-1} (\nabla r_{t+1}(x_T) - \nabla r_{t+1}(x_t) + p_t) \Big) = -\nabla H(x) \in N_X(x_T).$$

Since the left-hand side of the above equation if exactly p_T from (5.4), we conclude that $p_T \in N_X(x_T)$. It only remains to show that x_{T+1} is the unique point that attains the infimum in (5.10) with $t = T+1$. To see this, first note that

$$\begin{aligned}
&\sum_{t=1}^{T-1} g_t + \nabla r_1(x_T) + p_T + \sum_{t=1}^{T-1}(\nabla r_{t+1}(x_T) - \nabla r_{t+1}(x_t) + p_t) = 0 \\
\iff &\sum_{t=1}^{T-1} g_t + \nabla r_1(x_T) + \sum_{t=1}^{T}(\nabla r_{t+1}(x_T) - \nabla r_{t+1}(x_t) + p_t) = 0 \\
\iff &\sum_{t=1}^{T+1} \nabla r_t(x_T) = \sum_{t=1}^{T} \nabla r_{t+1}(x_t) - \sum_{t=1}^{T-1} g_t - \sum_{t=1}^{T} p_t.
\end{aligned}$$

Define

$$R_{T+1} \coloneqq \sum_{t=1}^{T+1} r_t \qquad \text{and} \qquad s \coloneqq \sum_{t=1}^{T} g_t + \sum_{t=1}^{T} p_t.$$

Since $\nabla R_{T+1}(x_T) = \sum_{t=1}^{T+1} \nabla r_t(x_T)$, we have

$$\nabla R_{T+1}(x_T) = \sum_{t=1}^{T} \nabla r_{t+1}(x_t) - \sum_{t=1}^{T-1} g_t - \sum_{t=1}^{T} p_t = \sum_{t=1}^{T} \nabla r_{t+1}(x_t) - s + g_T. \tag{5.13}$$

Moreover, recall that by Theorem 5.3.2,

$$\{x_{T+1}\} = \arg\min_{x \in X}(\langle g_T, x\rangle + B_{R_{T+1}}(x, x_T)).$$

Finally, note that, for any $x \in \mathbb{E}$,

$$\begin{aligned}
&\langle g_T, x\rangle + B_{R_{T+1}}(x, x_T) = \langle g_T, x\rangle + R_{T+1}(x) - R_{T+1}(x_T) - \langle \nabla R_{T+1}(x_T), x - x_T\rangle \\
&\overset{(5.13)}{=} \langle g_T, x\rangle + R_{T+1}(x) - R_{T+1}(x_T) - \sum_{t=1}^{T}\langle \nabla r_{t+1}(x_t), x - x_T\rangle + \langle s, x - x_T\rangle - \langle g_T, x - x_T\rangle \\
&= r_1(x) - r_1(x_T) + \sum_{t=1}^{T}(r_{t+1}(x) - r_{t+1}(x_T) - \langle \nabla r_{t+1}(x_t), x - x_T\rangle) + \langle s, x - x_T\rangle + \langle g_T, x_T\rangle \\
&= r_1(x) - r_1(x_T) + \sum_{t=1}^{T}\big(B_{r_{t+1}}(x, x_t) - r_{t+1}(x_T) - r_{t+1}(x_t) + \langle \nabla r_{t+1}(x_t), x_T - x_t\rangle\big) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad + \langle s, x - x_T\rangle + \langle g_T, x_T\rangle.
\end{aligned}$$

Therefore, by ignoring the terms which do not depend on x in the above equation (since they do not affect which points attain the minimum for $x \in X$), we conclude that

$$\{x_{T+1}\} = \arg\min_{x \in X}(\langle g_T, x\rangle + B_{R_{T+1}}(x, x_T)) = \arg\min_{x \in X}\Big(r_1(x) + \sum_{t=1}^{T} B_{r_{t+1}}(x, x_t) + \langle s, x\rangle\Big),$$

and the right-hand side of the above equation is exactly the set of points which attain (5.10) for $t = T + 1$ by the definition of s. This finishes the proof of (5.12).

Finally, let us show that x_t attains (5.11) for each $t \in [T]$ using $p_1, \dots, p_T \in \mathbb{E}$ as in (5.4). Let $t \in [T]$ and define

$$F(x) \coloneqq \langle p_t, x\rangle + \Big(\sum_{i=1}^{t-1} \langle g_i, x\rangle + r_1(x) + \sum_{i=1}^{t-1} (B_{r_{i+1}}(x, x_i) + \langle p_i, x\rangle)\Big).$$

Let us show that x_t attains $\inf_{x \in X} F(x)$. Since $x_t \in D$, we have that r_i is differentiable at x_t for every $i \in [t]$. Hence, F is also differentiable at x_t. Thus, by the form of p_t from (5.12) we have

$$\nabla F(x_t) = p_t + \Big(\sum_{i=1}^{t-1} g_i + \nabla r_1(x_t) + \sum_{i=1}^{t-1} (\nabla r_{i+1}(x_t) - \nabla r_{i+1}(x_i) + p_i)\Big) \overset{(5.12)}{=} 0,$$

that is $\nabla F(x_t) = 0$. Since $\partial F(x_t) = \{\nabla F(x_t)\}$ by Theorem 3.5.5, we have $0 \in \partial F(x_t)$, which happens if and only if $x_t \in \arg\min_{x \in \mathbb{E}} F(x)$. Moreover, since F is strictly convex (since $r_1, \dots, r_t$ are strongly convex), we have $\{x_t\} = \arg\min_{x \in \mathbb{E}} F(x)$. That is, x_t is the unique point that attains $\inf_{x \in \mathbb{E}} F(x)$.

In particular, if we define $\mathcal{R}'$ and $\boldsymbol{h} \in (\mathbb{R}^{\mathbb{E}})^T$ as in the statement, for every $t \in [T]$ we have

$$\begin{aligned}
\{\mathrm{AdaFTRL}_{\mathcal{R}'}(\boldsymbol{h}_{1:t-1})\} &= \arg\min_{x \in \mathbb{E}} \Big(\sum_{i=1}^{t-1} h_i(x) + \sum_{i=1}^{t} \mathcal{R}'(\boldsymbol{h}_{1:i-1})\Big)\\
&= \arg\min_{x \in X} \Big(\sum_{i=1}^{t-1} \langle g_i, x\rangle + r_1(x) + \langle p_1, x\rangle + \sum_{i=2}^{t} (B_{r_i}(x, x_{i-1}) + \langle p_i, x\rangle)\Big)\\
&= \arg\min_{x \in X} \Big(\sum_{i=1}^{t-1} \langle g_i, x\rangle + r_1(x) + \langle p_t, x\rangle + \sum_{i=1}^{t-1} (B_{r_{i+1}}(x, x_i) + \langle p_i, x\rangle)\Big)\\
&= \{\mathrm{AdaOMD}_{\mathcal{R}}^X(\boldsymbol{f}_{1:t-1})\},
\end{aligned}$$

where in the last equation we used that the infimum in (5.11) is attained by the same point if we add $\delta(\cdot \,|\, X)$ to it. □

The above theorem states that the Adaptive OMD algorithm can be seen as the application of the Adaptive FTRL algorithm with a very interesting proximal regularizer strategy. At each round t, the regularizer increment of the AdaFTRL oracle is a Bregman divergence w.r.t. the t-th increment from the original mirror map strategy. Not only that, there are some special vectors from the normal cone of the set X from where the player picks her points that crawl up in the FTRL formula. Intuitively, they skew a bit the minimization formula of the original AdaFTRL oracle so that the iterates match the ones of the AdaOMD oracle. Intuitively, this is the part that accounts for the possibility of choice of subgradient from the mirror map plus the indicator function we had discussed in our hypothetical version of AdaOMD.

Moreover, note that the infimum (5.11) is unfair in the sense the it "looks into the future". To decide the iterate x_{t+1} from round $t + 1$ it needs the point on the normal cone of X at x_{t+1}. Even though this formula is not pratically implementable, it will help us in deriving regret bounds for AdaOMD from the tools we have developed on Chapter 4. Specifically, the following theorem applies the lemmas from Section 4.3 in a way very similar to the way we did to derive regret bounds for AdaFTRL from Section 4.4. Unfortunately, just blindly applying one of the theorems to the above FTRL regularizer strategies does not yield the regret bounds that we want. If we do so, the points in the normal cone crawl into the bound in undesired ways: either inside the dual norms together with the subgradients, or in the regret formula in not very desirable ways.

Finally, it is worth saying that this section is far from showing one the simplest ways to derive regret bound for the Adaptive OMD algorithm. The reason we are showing these connections (mainly based on the work of McMahan [48]) is two-fold. First, one of the main purposes of this text is to analyze the connections among many algorithms from Online Convex Optimization, a interesting fact which will be better discussed on Chapter 7. Second, some proofs of OMD regret bounds rely on potential functions (e.g., see [11]) or other smart tricks. This proof is, arguably, more "automatic" in the sense that we just apply what we already know, without major secrets and tricks.

Theorem 5.4.3. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that X is closed and such that each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X. Let $\mathcal{R}\colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ be a mirror map strategy for $\mathcal{C}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and let $T \in \mathbb{N}$. Moreover, define

$$\begin{aligned} (\boldsymbol{x}, \boldsymbol{f}) &\coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaOMD}^X_{\mathcal{R}}, \mathrm{ENEMY}, T), \\ r_t &\coloneqq \mathcal{R}(\langle f_1, \dots, f_{t-1}\rangle), & \text{for each } t \in [T], \\ x_0 &\coloneqq x_1, \text{ and } r_0 \coloneqq r_1. \end{aligned}$$

Finally, let $g_t \in \partial f_t(x_t)$ be the same as in the definition of $\mathrm{AdaOMD}^X_{\mathcal{R}}(\boldsymbol{f})$ on Algorithm 5.1 for each $t \in [T]$. If for every $t \in [T]$ there is $\sigma_t \in \mathbb{R}_{++}$ such that $\sum_{i=1}^t r_i$ is σ_t-strongly convex w.r.t. a norm $\|\cdot\|_{(t)}$ on X, then,

$$\mathrm{Regret}(\mathrm{AdaOMD}^X_{\mathcal{R}}, \boldsymbol{f}, u) \leq \sum_{t=1}^{T+1} B_{r_t}(u, x_{t-1}) + \frac{1}{2}\sum_{t=1}^{T} \frac{1}{\sigma_{t+1}} \|g_t\|^2_{(t+1),*}, \qquad \forall u \in X.$$

Proof. Define $h_t \coloneqq \langle g_t, \cdot\rangle$ for each $t \in [T]$, define $x_{T+1} \coloneqq \mathrm{AdaOMD}^X_{\mathcal{R}}(\boldsymbol{f})$, and let $u \in X$. By[8] Theorem 5.4.2, there are $p_t \in N_X(x_t)$ for each $t \in \{1, \dots, T+1\}$ such that, if we define $\mathcal{R}'\colon \mathcal{F} \to (-\infty, +\infty]^{\mathbb{E}}$ by $\mathcal{R}'(\langle\rangle) \coloneqq r_1 + \langle p_1, \cdot\rangle + \delta(\cdot \,|\, X)$ and by

$$\mathcal{R}'(\boldsymbol{f}) \coloneqq B_{r_{t+1}}(\cdot, x_t) + \langle p_{t+1}, \cdot\rangle, \qquad \forall \boldsymbol{f} \in \mathcal{F}^t, \forall t \in [T],$$

then, $\mathrm{AdaOMD}^X_{\mathcal{R}}(\boldsymbol{f}_{1:t-1}) = \mathrm{AdaFTRL}_{\mathcal{R}'}(\boldsymbol{h}_{1:t-1})$ for each $t \in \{1, \dots, T+1\}$. Therefore, using the subgradient inequality we have,

$$\begin{aligned} \mathrm{Regret}(\mathrm{AdaOMD}^X_{\mathcal{R}}, \boldsymbol{f}, u) &= \sum_{t=1}^{T}(f_t(x_t) - f_t(u)) \leq \sum_{t=1}^{T}\langle g_t, x_t - u\rangle = \sum_{t=1}^{T}(h_t(x_t) - h_t(u)) \\ &= \mathrm{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}'}, \boldsymbol{h}, u). \end{aligned}$$

Make the following definitions:

$$\begin{aligned} b_t &\coloneqq B_{r_t}(\cdot, x_{t-1}) & \text{for each } t \in \{2, \dots, T+1\}, \\ b_1 &\coloneqq r_1 + \delta(\cdot \,|\, X), \\ H_t &\coloneqq \sum_{i=1}^{t} h_i + \sum_{i=1}^{t+1}(b_i + \langle p_i, \cdot\rangle) & \text{for each } t \in [T], \\ x_0 &\coloneqq x_1. \end{aligned}$$

[8] We use Theorem 5.4.2 for a game with $T+1$ rounds so that it yields the point $p_{T+1} \in N_X(x_{T+1})$ that we use in this proof.

By Lemma 4.3.1 we have

$$
\begin{aligned}
\mathrm{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}'}, \boldsymbol{h}, u) &\leq \sum_{t=1}^{T+1}(b_t(u) - b_t(x_{t-1})) + \sum_{t=0}^{T}\langle p_{t+1}, u - x_t\rangle + \sum_{t=1}^{T}(H_t(x_t) - H_t(x_{t+1}))\\
&\leq \sum_{t=1}^{T+1}(b_t(u) - b_t(x_{t-1})) + \sum_{t=1}^{T}\langle p_{t+1}, u - x_t\rangle + \sum_{t=1}^{T}(H_t(x_t) - H_t(x_{t+1})),
\end{aligned}
\tag{5.14}
$$

where in the last inequality we have used that $\langle p_1, u - x_0\rangle = \langle p_1, u - x_1\rangle \leq 0$ since $p_1 \in N_X(x_1)$ and $u \in X$. Let us now show that

$$
H_t(x_t) - H_t(x_{t+1}) \leq \frac{1}{2\sigma_{t+1}}\|g_t\|_{(t+1),*}^2 + \langle p_{t+1}, x_t - x_{t+1}\rangle, \qquad \forall t \in [T]. \tag{5.15}
$$

Let $t \in [T]$. Since $\operatorname{dom} h_t = \mathbb{E}$ and since $H_{t-1} + b_{t+1}$ is proper, we have that $\operatorname{ri}(\operatorname{dom}(H_{t-1} + b_{t+1})) \cap \operatorname{ri}(\operatorname{dom} h_t)$ is nonempty. Moreover, $x_t \in \arg\min_{x \in \mathbb{E}}(H_{t-1}(x) + b_{t+1}(x))$ since x_t minimizes H_{t-1} by the definition of AdaFTRL and clearly minimizes $b_{t+1} = B_{r_{t+1}}(\cdot, x_t)$. Finally, since $\sum_{i=1}^{t+1} r_i$ is σ_{t+1}-strongly convex (w.r.t. $\|\cdot\|_{(t+1)}$) on X, the function $\sum_{i=1}^{t+1} b_i$ is also σ_{t+1}-strongly convex on X (see Lemma 3.11.2), but since $\operatorname{dom}(\sum_{i=1}^{t+1} b_i) \subseteq X$, we have that $\sum_{i=1}^{t+1} b_i$ is actually σ_{t+1}-strongly convex on $\mathbb{E}$. This implies that $H_{t-1} + b_{t+1}$ is also is σ_{t+1}-strongly convex. Therefore, by Lemma 4.3.2 and since $\nabla h_t(x) = g_t$ for every $x \in \mathbb{E}$, we have

$$
\begin{aligned}
H_t(x_t) - H_t(x_{t+1}) = H_{t-1}(x_t) + b_{t+1}(x_t) + h_t(x_t) - (H_{t-1}(x_{t+1}) + b_{t+1}(x_{t+1}) + h_t(x_{t+1}))&\\
+ \langle p_{t+1}, x_t - x_{t+1}\rangle&\\
\leq \frac{1}{2\sigma_{t+1}}\|g_t\|_{(t+1),*}^2 + \langle p_{t+1}, x_t - x_{t+1}\rangle.&
\end{aligned}
$$

This ends the proof of (5.15). Putting together (5.14) and (5.15) yields

$$
\begin{aligned}
\mathrm{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}'}, \boldsymbol{h}, u) &\leq \sum_{t=1}^{T+1}(b_t(u) - b_t(x_{t-1})) + \sum_{t=1}^{T}\langle p_{t+1}, u - x_t\rangle + \sum_{t=1}^{T}(H_t(x_t) - H_t(x_{t+1}))\\
&\leq \sum_{t=1}^{T+1}(b_t(u) - b_t(x_{t-1})) + \sum_{t=1}^{T}\langle p_{t+1}, u - x_{t+1}\rangle + \sum_{t=1}^{T}\frac{1}{2\sigma_{t+1}}\|g_t\|_{(t+1),*}\\
&\leq \sum_{t=1}^{T+1}(b_t(u) - b_t(x_{t-1})) + \sum_{t=1}^{T}\frac{1}{2\sigma_{t+1}}\|g_t\|_{(t+1),*}\\
&= \sum_{t=1}^{T+1} B_{r_t}(u, x_{t-1}) + \sum_{t=1}^{T}\frac{1}{2\sigma_{t+1}}\|g_t\|_{(t+1),*},
\end{aligned}
$$

where in the second inequality we have used that, $\langle p_{t+1}, u - x_{t+1}\rangle \leq 0$ for every $t \in [T]$ since $p_{t+1} \in N_X(x_{t+1})$ and $u \in X$, and in the last equation we have used the definition of b_t for $t \in \{1, \dots, T+1\}$ and the fact that $r_1(u) - r_1(x_1) \leq B_{r_1}(u, x_1)$ since $\nabla r_1(x_1) \in N_X(x_1)$ and, thus, $\langle \nabla r_1(x_1), u - x_1\rangle \leq 0$. □

Recall that if R is a mirror map for a convex and closed set X and $\mathcal{C} := (X, \mathcal{F})$ is an OCO instance, then $\mathcal{R}\colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ given by $\mathcal{R}(\boldsymbol{f}) := [\boldsymbol{f} \neq \langle\rangle]R$ is a mirror map strategy for $\mathcal{C}$. Thus, an immediate corollary of the above theorem together with Theorem 3.8.4 is a regret bound for the EOMD oracle which matches the regret bound of the classic FTRL algorithm.

Corollary 5.4.4 (Derived from Theorem 5.4.3)**.** Let $\mathcal{C} := (X, \mathcal{F})$ be an OCO instance such that X is closed and such that each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X . Let $R \colon (-\infty, +\infty] \to \mathbb{E}$ be a mirror map for X, let ENEMY be an enemy oracle for $\mathcal{C}$, and let $T \in \mathbb{N}$. Moreover, define

$$(\boldsymbol{x}, \boldsymbol{f}) := \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaOMD}^{X}_{\mathcal{R}}, \mathrm{ENEMY}, T),$$

Finally, let $g_t \in \partial f_t(x_t)$ be as in the definition of $\mathrm{EOMD}^X_R(\boldsymbol{f})$ on Algorithm 5.2 for each $t \in [T]$. If $\sigma \in \mathbb{R}_{++}$ is such that R is σ-strongly convex w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$, then,

$$\mathrm{Regret}(\mathrm{EOMD}^X_R, \boldsymbol{f}, u) \leq B_R(u, x_1) + \frac{1}{2\sigma}\sum_{t=1}^{T} \|g_t\|_*^2, \qquad \forall u \in X, \tag{5.16}$$

where $x_0 := x_1$. In particular, if every function in $\mathcal{F}$ is ρ-Lipschitz continuous w.r.t. $\|\cdot\|$ on a convex set $D \subseteq \mathbb{E}$ such that $X \subseteq \mathrm{int}(D)$, there is $\theta \in \mathbb{R}_{++}$ such that $\theta \geq \sup\{\, B_R(, yy) : x \in X, y \in X \cap \mathrm{int}(\mathrm{dom}\, R)\}$, and $R' := \big(\rho\sqrt{T}/(\sqrt{2\sigma\theta})\big)R$ is also a mirror map for X, then

$$\mathrm{Regret}(\mathrm{EOMD}^X_{R'}, \mathrm{ENEMY}, X) \leq \rho\sqrt{\frac{2\theta T}{\sigma}}.$$

Proof. Note that $\mathrm{EOMD}^X_R = \mathrm{AdaOMD}_{\mathcal{R}}$ where $\mathcal{R}$ is given by $\mathcal{R}(\boldsymbol{f}) := [\boldsymbol{f} = \langle\rangle]R$ for every $\boldsymbol{f} \in \mathrm{Seq}((-\infty, +\infty]^{\mathbb{E}})$. Moreover, since R is mirror map for X, $\mathcal{R}$ is a mirror map strategy for $\mathcal{C}$. Therefore, the first inequality is a direct application of Theorem 5.4.3 together with the fact the R is σ-strongly convex on X w.r.t. $\|\cdot\|$.

If each $f \in \mathcal{F}$ is ρ-Lipschitz continuous w.r.t. $\|\cdot\|$ on a convex set D such that $X \subseteq \mathrm{int}(D)$, then by Theorem 3.8.4 we have that $\partial f(x) \subseteq \{\, g \in \mathbb{E} : \|g\|_* \leq \rho\}$ for each $f \in \mathcal{F}$ and $x \in X$. Using this in (5.16) yields

$$\mathrm{Regret}_T(\mathrm{EOMD}^X_R, \mathrm{ENEMY}, u) \leq B_R(u, x_1) + \frac{T\rho^2}{2\sigma}.$$

Moreover, suppose there is $\theta \in \mathbb{R}_{++}$ such that $\theta \geq \sup\{\, R(x) - R(y) : x \in X, y \in X \cap \mathrm{dom}\, R\}$, and define

$$R' := \frac{\rho\sqrt{T}}{\sqrt{2\sigma\theta}} R.$$

Note that R' is a $(\rho\sqrt{\sigma T}/\sqrt{2\theta})$-strongly convex on X. Finally, suppose R' is also a mirror map. Then, plugging R' into the above inequality yields, for every $u \in X$,

$$\mathrm{Regret}_T(\mathrm{EOMD}^X_{R'}, \mathrm{ENEMY}, u) \leq \frac{\rho\sqrt{T}}{\sqrt{2\sigma\theta}} B_R(u, x_1) + \frac{\rho\sqrt{\theta T}}{\sqrt{2\sigma}} \leq \frac{\rho\sqrt{\theta T}}{\sqrt{2\sigma}} + \frac{\rho\sqrt{\theta T}}{\sqrt{2\sigma}} = \rho\sqrt{\frac{2\theta T}{\sigma}},$$

where in the second inequality we took the supremum over $u \in X$. □

5.5 Dual Averaging or Lazy Online Mirror Descent

In this section, we look at a variation of the Adaptive Online Mirror Descent oracle. In the original mirror descent algorithm, the step on direction of minus subgradient on round t is done from $\nabla R_t(x_{t-1})$, where R_t is the mirror map of round t and x_{t-1} is the point picked by the oracle on round $t-1$ as in Algorithm 5.1. Note that before making this subgradient step, we had just projected

y_{t-1} (as defined in Algorithm 5.1) back into the primal through the Bregman projection to get x_{t-1}. Thus, one may wonder what happens if we are lazy and make the subgradient step directly from y_t instead of computing the gradient of R_{t+1} at x_t to only then make a subgradient step from $\nabla R_{t+1}(x_t)$. Avoiding the computation of the gradient of the mirror map at every round (even though the algorithm still has to project the iterate from the dual to the primal space) may yield a drastic improvement in the time needed to compute each round in a practical implementation. This is exactly the idea of the **Adaptive Dual Averaging** (Adaptive DA or AdaDA) or **Adaptive Lazy Online Mirror Descent** algorithm. On Algorithm 5.3 we define the AdaDA oracle, which implements this algorithm. Moreover, on Figure 5.2 we present an schematic view of the computations done by AdaDA on round $t+1$ (one may find it useful to compare this figure with Figure 5.1). The name *Dual Averaging* comes originally from the static version of this algorithm for classic convex optimization [56] (though it is not originally presented in the same way as presented in Algorithm 5.3).

Algorithm 5.3 Definition of $\mathrm{AdaDA}_{\mathcal{R}}^{X}(\langle f_1, \dots, f_T \rangle)$

Input:

(i) A closed convex set $X \subseteq \mathbb{E}$,

(ii) Convex functions $f_1, \dots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and set of convex functions $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$ such that f_t is subdifferentiable on X for each $t \in [T]$,

(iii) $\mathcal{R}\colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ is a mirror map strategy for the OCO instance $(X, \mathcal{F})$ which is differentiable on the open convex set $D \subseteq \mathbb{E}$.

Output: $x_{T+1} \in D \cap X$

```
  $r_1 \leftarrow \mathcal{R}(\langle\rangle)$
  $\{x_1\} \leftarrow \arg\min_{x \in X} r_1(x)$
  $y_1 \leftarrow 0.$
  for $t = 1$ to $T$ do
      ▷ Computations for round $t + 1$
      Define $r_{t+1} \coloneqq \mathcal{R}(\langle f_1, \dots, f_t \rangle)$ and $R_{t+1} \coloneqq \sum_{i=1}^{t+1} r_i$
      $y_{t+1} \coloneqq y_t - g_t$, where $g_t \in \partial f_t(x_t)$
      $x_{t+1} \coloneqq \Pi_X^{R_{t+1}}(\nabla R_{t+1}^*(y_{t+1}))$
  return $x_{T+1}$
```

One may note that, as for the AdaOMD oracle, we pass the set $X \subseteq \mathbb{E}$ where the player is supposed to pick her points as a parameter for AdaDA. However, one may note that this is not necessary due to Proposition 5.4.1, which says that using a mirror map strategy plus the indicator function of X as a new mirror map renders the Bregman projection unnecessary. Still, we found that presenting the AdaDA oracle in the way most similar to the AdaOMD oracle is informative.

Finally, one may recall from the last section that making, at each round, a subgradient step from the gradient of R computed at the previous iterate was one of the sources of complications in writing the Adaptive OMD as an application of the Adaptive FTRL oracle. Thus, we may hope the AdaDA oracle to have a much cleaner connection with AdaFTRL if compared to AdaOMD. The following theorem shows that this is indeed the case.

Theorem 5.5.1. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that X is closed and such that each $f \in \mathcal{F}$ is proper and closed. Let $\mathcal{R}\colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ be a mirror map strategy for $\mathcal{C}$, let

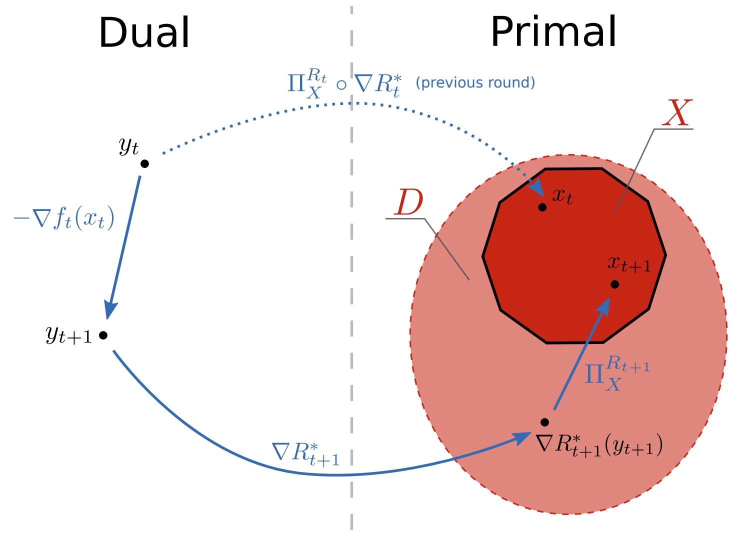

Figure 5.2: Graphic representation of the computations done by AdaDA on round $t+1$.

ENEMY be an enemy oracle for $\mathcal{C}$ and let $T \in \mathbb{N}$. Define

$$\begin{aligned} (\boldsymbol{x}, \boldsymbol{f}) &\coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaDA}_{\mathcal{R}}^{X}, \mathrm{ENEMY}, T), \\ x_{T+1} &\coloneqq \mathrm{AdaDA}_{\mathcal{R}}^{X}(\boldsymbol{f}), \\ \text{and } R_t &\coloneqq \sum_{i=1}^{t} \mathcal{R}(\langle f_1, \dots, f_{i-1} \rangle) \qquad\qquad \text{for each } t \in \{1, \dots, T+1\}. \end{aligned}$$

Moreover, let $g_t \in \partial f_t(x_t)$ be the same as in the definition of $\mathrm{AdaDA}_{\mathcal{R}}^{X}(\boldsymbol{f})$ on Algorithm 5.3 for each $t \in [T]$. If R_t is strongly convex[9] on X for each $t \in \{1, \dots, T+1\}$, then,

$$\{x_t\} = \operatorname*{arg\,min}_{x \in X} \Big(\sum_{i=1}^{t-1} \langle g_i, x \rangle + R_t(x) \Big) \tag{5.17}$$

for each $t \in \{1, \dots, T+1\}$. Additionally set $\mathcal{F}_g \coloneqq \{\, f \in \mathcal{F} : g \in \partial f(x) \text{ for some } x \in X \}$ for each $g \in \mathbb{E}$ and $\mathcal{L} \coloneqq \{\, \langle g, \cdot \rangle : g \in \mathbb{E} \text{ s.t. } \mathcal{F}_g \neq \varnothing \}$. Moreover, for every $h \in \mathcal{L}$ and for $g_h \coloneqq \nabla h(0)$ (that is, $h = \langle g_h, \cdot \rangle$), set

$$f_h \coloneqq \begin{cases} f_t & \text{if } g_h = g_t \text{ for some } t \in [T], \\ \text{some } f \in \mathcal{F}_{g_h} & \text{otherwise,} \end{cases} \qquad \forall h \in \mathcal{L}.$$

Finally, define

$$\mathcal{R}'(\boldsymbol{h}) \coloneqq \mathcal{R}(\langle f_{h_1}, f_{h_2}, \dots, f_{h_t} \rangle) + \delta(\cdot \,|\, X) \qquad \forall \boldsymbol{h} \in \mathcal{L}^t, \forall t \in \mathbb{N}.$$

In this case, $\mathcal{R}'$ is a FTRL regularizer strategy for $\mathcal{C}' \coloneqq (X, \mathcal{L})$ and

$$\mathrm{AdaDA}_{\mathcal{R}}^{X}(\boldsymbol{f}_{1:t-1}) = \mathrm{AdaFTRL}_{\mathcal{R}'}(\langle \langle g_1, \cdot \rangle, \dots, \langle g_{t-1}, \cdot \rangle \rangle) \qquad \forall t \in \{1, \dots, T+1\}.$$

Proof. Let $t \in [T]$ and let $y_t \in \mathbb{E}$ be as in the definition of $\mathrm{AdaDA}_{\mathcal{R}}^{X}(\boldsymbol{f})$ in Algorithm 5.3. Since $y_1 = 0$, by an easy induction one can see that $y_t = \sum_{i=1}^{t-1} g_i$. By the definition of AdaDA in

[9] We need this assumption in order to apply Proposition 5.4.1.

Algorithm 5.3 we have that $x_t = \Pi_X^{R_t}(\nabla R_t^*(y_t))$. Since R_t is strongly convex on X, by Lemma 3.11.4 we have $x_t = \nabla P_t^*(y_t)$, where $P_t \coloneqq R_t + \delta(\cdot \mid X)$. Since R_t and $\delta(\cdot \mid X)$ are closed (recall that X is closed), by Theorem 3.2.7 we know that P_t is closed. Therefore, by the properties of subgradients from Theorem 3.5.2 (namely items (ii) and (v)), and since $\{\nabla P_t^*(x_t)\} = \partial P_t^*(x_t)$ by Theorem 3.5.5, we have

$$x_t = \nabla P_t^*(y_t) \iff \{x_t\} = \arg\max_{x \in \mathbb{E}}(\langle y_t, x\rangle - P_t(x)) = \arg\min_{x \in X}\Big(\sum_{i=1}^{t-1}\langle g_i, x\rangle + R_t(x)\Big).$$

In particular, define $\mathcal{R}'$ as in the statement of the theorem. Let us first show that

$$\mathcal{R}' \text{ is a FTRL regularizer strategy for } \mathcal{C}'. \tag{5.18}$$

Let $T' \in \mathbb{N}$, let $\boldsymbol{h} \in \mathcal{L}^{T'}$, and set $R' \coloneqq \sum_{t=1}^{T'+1} \mathcal{R}'(\boldsymbol{h}_{1:t-1})$. Since $\mathcal{R}$ is a mirror map, since X is closed, and since the sum of closed and convex functions is also closed and convex by Theorem 3.2.7, we clearly have that $\mathcal{R}'(\boldsymbol{h})$ is a closed proper convex function, that is, $\mathcal{R}'$ satisfies condition (4.5.i) of a FTRL regularizer strategy for $\mathcal{C}'$. Thus, we only need to show that R is a classical FTRL regularizer for $\mathcal{C}'$. With the same arguments we have just used, it is easy to see that R is a proper closed convex function, that is, it satisfies property (4.4.i) of a FTRL regularizer for $\mathcal{C}$. Moreover, we clearly have $\operatorname{dom} R \subseteq X$, which is condition (4.4.ii) of a FTRL regularizer $\mathcal{C}'$. Let $T'' \in \mathbb{N}$ and $\boldsymbol{h}'' \in \mathcal{L}^{T''}$. Note that by assumption we have that each mirror map increment $\mathcal{R}$ is strongly convex on X. Thus, R is strongly convex on $\mathbb{E}$, which implies that $R + \sum_{t=1}^{T''} h_t''$ is strongly convex on $\mathbb{E}$ and closed by Theorem 3.2.7 since it is the sum of closed and convex functions. Therefore, by Lemma 3.9.14 we know that $\inf_{x \in \mathbb{E}}(R + \sum_{t=1}^{T''} h_t'')$ is attained, which finishes that proof of (5.18). Finally, note that for every $t \in \{1, \dots, T+1\}$ we have

$$\begin{aligned}
\{x_t\} &= \arg\min_{x \in X}\Big(\sum_{i=1}^{t-1}\langle g_i, x\rangle + R_t(x)\Big)\\
&= \arg\min_{x \in \mathbb{E}}\Big(\sum_{i=1}^{t-1}\langle g_i, x\rangle + \sum_{i=1}^{t}\Big(\big[\mathcal{R}(\langle f_1, \dots, f_{i-1}\rangle)\big](x) + \delta(x \mid X)\Big)\Big)\\
&= \arg\min_{x \in \mathbb{E}}\Big(\sum_{i=1}^{t-1}\langle g_i, x\rangle + \sum_{i=1}^{t}\big[\mathcal{R}'(\langle\langle g_1, \cdot\rangle, \dots, \langle g_{i-1}, \cdot\rangle\rangle)\big](x)\Big)\\
&= \text{AdaFTRL}_{R'}(\langle\langle g_1, \cdot\rangle, \dots, \langle g_{t-1}, \cdot\rangle\rangle).
\end{aligned}$$

□

The above theorem tell us something very interesting: Adaptive Dual Averaging with mirror map $\mathcal{R}$ is closely related (actually, almost equivalent) to the Adaptive FTRL algorithm with regularizer strategy $\mathcal{R}$ with $\delta(\cdot \mid X)$ added to each mirror map increment applied to the linearized versions of the functions played by the enemy. The name *Dual Averaging* stems exactly from the equation between AdaDA and AdaFTRL given by the above theorem. Indeed, on the application of AdaFTRL on the above theorem we are minimizing over the set X the linear function given by the average[10] of the subgradients of the past functions plus a regularizer function.

This simplification done by the Adaptive Dual Averaging algorithm when compared to the Adaptive Online Mirror Descent does not come without its costs. Note that, by the last theorem, AdaDA works like a general FTRL algorithm, while AdaOMD works as a *proximal* FTRL algorithm.

[10] Even though we are looking at the sum of the subgradients at the formula, recall that we can scale the regularizer to effectively normalize this sum.

As discussed on Section 4.7, this may influence the efficiency or the amount of previous information needed by the oracle in some cases. We will look more carefully at some of these cases on Chapter 6.

Given such a clean connection of the Adaptive Dual Averaging algorithm and the Adaptive FTRL algorithm, it is of no surprise that regret bounds for the AdaFTRL oracle directly imply regret bound for AdaDA, as we show in the next corollary.

Corollary 5.5.2 (Derived from Theorems 4.4.3 and 5.5.1)**.** Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that X is closed and such that each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X. Let $\mathcal{R} \colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ be a mirror map strategy for $\mathcal{C}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and let $T \in \mathbb{N}$. Moreover, define

$$\begin{aligned}
(\boldsymbol{x}, \boldsymbol{f}) &\coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaDA}_{\mathcal{R}}^{X}, \mathrm{ENEMY}, T), \\
r_t &\coloneqq \mathcal{R}(\langle f_1, \dots, f_{t-1} \rangle), && \text{for each } t \in [T]. \\
R_t &\coloneqq \sum_{i=1}^{t} r_i, && \text{for each } t \in [T].
\end{aligned}$$

Finally, let $g_t \in \partial f_t(x_t)$ be as in the definition of $\mathrm{AdaDA}_{\mathcal{R}}^{X}(\langle f_1, \dots, f_t \rangle)$ on Algorithm 5.3 for each $t \in [T]$, and suppose for each $t \in [T]$ there are $\sigma_t \in \mathbb{R}_{++}$ and a norm $\|\cdot\|_{(t)}$ on $\mathbb{E}$ such that R_t is σ_t-strongly convex w.r.t. $\|\cdot\|_{(t)}$ on $\mathbb{E}$. Then,

$$\mathrm{Regret}(\mathrm{AdaDA}_{\mathcal{R}}^{X}, \boldsymbol{f}, u) \leq \sum_{t=1}^{T} (r_t(u) - r_t(x_t)) + \frac{1}{2} \sum_{t=1}^{T} \frac{1}{\sigma_t} \|g_t\|_{(t),*}^2.$$

Proof. Define $h_t \coloneqq \langle g_t, \cdot \rangle$ for each $t \in [T]$, set $\mathcal{L} \coloneqq \{\, \langle g, \cdot \rangle : f \in \mathcal{F}, x \in X, g \in \partial f(x) \}$, and define the OCO instance $\mathcal{C}' \coloneqq (X, \mathcal{L})$. By Theorem 5.5.1, we know that there is a FTRL regularizer strategy $\mathcal{R}'$ for $\mathcal{C}'$ such that $x_t = \mathrm{AdaFTRL}_{\mathcal{R}'}(\langle h_1, \dots, h_{t-1} \rangle)$ for every $t \in [T]$. Therefore, by the subgradient inequality, for every $u \in X$ we have

$$\mathrm{Regret}(\mathrm{AdaDA}_{\mathcal{R}}^{X}, \boldsymbol{f}, u) = \sum_{t=1}^{T} (f_t(x_t) - f_t(u)) \leq \sum_{t=1}^{T} \langle g_t, x_t - u \rangle = \mathrm{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}'}, \boldsymbol{h}, u). \quad (5.19)$$

Moreover, by the definition of $\mathcal{R}'$ (see Theorem 5.5.1), we have

$$\sum_{i=1}^{t} \mathcal{R}'(\boldsymbol{h}_{1:i-1}) = \sum_{i=1}^{t} \mathcal{R}(\boldsymbol{f}_{1:i-1}) + \delta(\cdot \,|\, X) = R_t + \delta(\cdot \,|\, X), \qquad \forall t \in [T]$$

Since R_t is σ_t-strongly convex w.r.t. the norm $\|\cdot\|_{(t)}$ on $\mathbb{E}$ for every $t \in [T]$, we have that $\mathcal{R}'$ is $\boldsymbol{\sigma}$-strong[11] for $\boldsymbol{h}$ w.r.t. $\|\cdot\|_1, \dots, \|\cdot\|_T$, where $\boldsymbol{\sigma} \coloneqq \langle \sigma_1, \dots, \sigma_t \rangle$. Finally, since $\nabla h_t(x_t) = g_t$ for each $t \in [T]$, by the general AdaFTRL regret bound from Theorem 4.4.3 we have, for every $u \in X$,

$$\mathrm{Regret}(\mathrm{AdaFTRL}_{\mathcal{R}'}, \boldsymbol{h}, u) \leq \sum_{t=1}^{T} (r_t(u) - r_t(x_t)) + \frac{1}{2} \sum_{t=1}^{T} \frac{1}{\sigma_t} \|g_t\|_{(t),*}^2. \qquad \square$$

In a way similar to what we have done for the AdaFTRL and AdaOMD algorithms, let us look at a version of AdaDA with a static regularizer, which we call **(classical) Lazy Online Mirror Descent**. We define an oracle which implements this algorithm in Algorithm 5.4.

[11] Note that the condition on the relative interior of the regularizer and the functions in $\mathcal{L}$ is trivially satisfied since all functions in $\mathcal{L}$ are finite everywhere.

Algorithm 5.4 Definition of $\mathrm{LOMD}_R^X(\langle f_1, \dots, f_T \rangle)$

```
Input:
    (i) A closed convex set $X \subseteq \mathbb{E}$;
    (ii) Convex functions $f_1, \dots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$ such that $f_i$ is subdifferentiable on $X$ for each $i \in [T]$;
    (iii) A mirror map $R \colon \mathbb{E} \to (-\infty, +\infty]$ for $(X, \mathcal{F})$.
Output: $x_{T+1} \in \operatorname{int}(\operatorname{dom} R) \cap X$
  $\{x_1\} \leftarrow \arg\min_{x \in X} R(x)$
  $y_1 \leftarrow 0$.
  for $t = 1$ to $T$ do
      ▷ Computations for round $t+1$
    Compute $g_t \in \partial f_t(x_t)$
    $y_{t+1} \leftarrow y_t - g_t$
    $x_{t+1} \leftarrow \Pi_X^R(\nabla R^*(y_{t+1}))$
  return $x_{T+1}$
```

Corollary 5.5.3 (Derived from Corollary 5.5.2)**.** Let $\mathcal{C} := (X, \mathcal{F})$ be an OCO instance such that X is closed and such that each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X. Let $R \colon \mathbb{E} \to (-\infty, +\infty]$ be a mirror map for X, let ENEMY be an enemy oracle for $\mathcal{C}$, and let $T \in \mathbb{N}$. Moreover, define

$$(\boldsymbol{x}, \boldsymbol{f}) := \mathrm{OCO}_{\mathcal{C}}(\mathrm{LOMD}_R^X, \mathrm{ENEMY}, T).$$

Finally, let $g_t \in \partial f_t(x_t)$ be as in the definition of $\mathrm{LOMD}_R^X(\boldsymbol{f})$ on Algorithm 5.4 for each $t \in [T]$ and suppose there is $\sigma \in \mathbb{R}_{++}$ and a norm $\|\cdot\|$ on $\mathbb{E}$ such that R is σ-strongly convex w.r.t. $\|\cdot\|$ on X. Then,

$$\mathrm{Regret}(\mathrm{LOMD}_R^X, \boldsymbol{f}, u) \leq R(u) - R(x_1) + \frac{1}{2\sigma} \sum_{t=1}^{T} \|g_t\|_*^2, \qquad \forall u \in X, \tag{5.20}$$

In particular, if every function in $\mathcal{F}$ is ρ-Lipschitz continuous w.r.t. $\|\cdot\|$ on a convex set $D \subseteq \mathbb{E}$ such that $X \subseteq \operatorname{int} D$, if there is $\theta \in \mathbb{R}_{++}$ such that $\theta \geq \sup\{\, B(x, y) : x \in X, y \in X \cap \operatorname{dom} R\}$, and if $R' := \big(\rho\sqrt{T}/(\sqrt{2\sigma\theta})\big)R$ is also a mirror map for X, then

$$\mathrm{Regret}_T(\mathrm{LOMD}_{R'}, \mathrm{ENEMY}, X) \leq \rho\sqrt{\frac{2\theta T}{\sigma}}.$$

Proof. Note that $\mathrm{LOMD}_R = \mathrm{AdaDA}_{\mathcal{R}}$ where $\mathcal{R}$ is given by $\mathcal{R}(\boldsymbol{f}) := [\boldsymbol{f} = \langle\rangle]R$ for every $\boldsymbol{f} \in \mathrm{Seq}((-\infty, +\infty]^{\mathbb{E}})$. Moreover, since R is mirror map for X, $\mathcal{R}$ is a mirror map strategy for $\mathcal{C}$. Therefore, the first inequality is a direct application of Corollary 5.5.2 together with the fact the R is σ-strongly convex on X w.r.t. $\|\cdot\|$.

If each $f \in \mathcal{F}$ is ρ-Lipschitz continuous w.r.t. $\|\cdot\|$ on a convex set D such that $X \subseteq \operatorname{int} D$, then by Theorem 3.8.4 we have that $\partial f(x) \subseteq \{\, g \in \mathbb{E} : \|g\|_* \leq \rho\}$ for each $f \in \mathcal{F}$ and $x \in X$. Using this in (5.20) and the fact that $\min_{x \in \mathbb{E}} R(x) = R(x_1)$ yields

$$\mathrm{Regret}_T(\mathrm{LOMD}_R, \mathrm{ENEMY}, u) \leq R(u) - \min_{x \in X} R(x) + \frac{T\rho^2}{2\sigma}.$$

Moreover, suppose there is $\theta \in \mathbb{R}_{++}$ such that $\theta \geq \sup\{\, R(x) - R(y) : x \in X, y \in X \cap \operatorname{dom} R\}$, and define

$$R' := \frac{\rho\sqrt{T}}{\sqrt{2\sigma\theta}} R.$$

Note that R' is a $(\rho\sqrt{\sigma T}/\sqrt{2\theta})$-strongly convex on X. Suppose R' is also a mirror map. Then, plugging R' into the above inequality yields, for every $u \in X$,

$$\mathrm{Regret}_T(\mathrm{LOMD}_{R'}, \mathrm{ENEMY}, u) \leq \frac{\rho\sqrt{T}}{\sqrt{2\sigma\theta}}(R(u) - \min_{x \in X} R(x)) + \frac{\rho\sqrt{\theta T}}{\sqrt{2\sigma}} \leq \frac{\rho\sqrt{\theta T}}{\sqrt{2\sigma}} + \frac{\rho\sqrt{\theta T}}{\sqrt{2\sigma}} = \rho\sqrt{\frac{2\theta T}{\sigma}},$$

where in the second inequality we took the supremum over $u \in X$. □

5.6 When Lazy and Eager OMD are Equivalent

The eager and lazy versions of online mirror descent (see Algorithms 5.2 and 5.4) are very similar and the regret bounds we have computed for them on Corollaries 5.4.4 and 5.5.3 are practically the same. Thus, it is natural to as whether there are conditions under which they are equivalent, that is, conditions under which they compute the same iterates when applied to the same functions. If so, it would be interesting to understand exactly under which conditions this happens. They are indeed equivalent in some cases. For example, [3, Appendix A] shows that Eager OMD and FTRL applied to linear functions are equivalent when used with the regularizers/mirror maps and the OCO instance that they consider. As we have seen on Theorem 5.5.1, FTRL and LOMD are basically the same algorithm when applied to linear functions. Thus, the authors of [3] were indeed showing equivalence of EOMD and LOMD for the cases they were looking at. Namely, they consider OCO instances of the form $(\mathcal{S}_d, \mathcal{F})$, where $\mathcal{S}_d := \{\, A \in \mathbb{S}^d_+ : \mathrm{Tr}(A) = 1\}$ is known as the **spectraplex**, a matrix analogous of the simplex, and $\mathcal{F}$ is composed of functions of the form $A \in \mathbb{S}^d \mapsto \mathrm{Tr}(GA)$ for some matrices $G \in \mathbb{S}^d$. One mirror map that [3] considers, for example, is a matrix analogous of the negative entropy. Although interesting, their proof is slightly technical and does not reveal much about the general conditions under which LOMD and EOMD are equivalent. It would be more interesting to obtain more general conditions for equivalence between EOMD and LOMD, without the need of *ad hoc* proofs for specific cases.

At this point, writing both EOMD and LOMD as instances of FTRL is very enlightening. For the sake of simplicity, let us first look at the case of OCO instances with linear functions. That is, let $\mathcal{C} := (X, \mathcal{F})$ be an OCO instance such that X is closed and $\mathcal{F} := \{\, \langle g, \cdot \rangle : g \in \mathbb{E}\}$. Moreover, let $R \colon \mathbb{E} \to (-\infty, +\infty]$ be a mirror map for X. Finally, let $T \in \mathbb{N} \setminus \{0\}$ and let $\boldsymbol{f} := \langle \langle g_1, \cdot \rangle, \cdots, \langle g_T, \cdot \rangle \rangle \in \mathcal{F}^T$. By Theorem 5.4.2, by defining $x_t := \mathrm{EOMD}^X_R(\boldsymbol{f}_{1:t-1})$ for each $t \in [T]$, there are $p_1, \dots, p_T \in \mathbb{E}$ with $p_t \in N_X(x_t)$ for each $t \in [T]$ such that

$$\{\mathrm{EOMD}^X_R(\boldsymbol{f})\} = \operatorname*{arg\,min}_{x \in X} \Big(\sum_{t=1}^{T} \langle g_t, x \rangle + R(x) + \sum_{t=1}^{T} \langle p_t, x \rangle \Big).$$

At the same time, by Theorem 5.5.1 we have

$$\{\mathrm{LOMD}^X_R(\boldsymbol{f})\} = \operatorname*{arg\,min}_{x \in X} \Big(\sum_{t=1}^{T} \langle g_t, x \rangle + R(x) \Big).$$

This way of writing both algorithms makes thin differences between them pop out. Namely, the term which we need to look at to see if both oracles compute the same iterate is the one involving the vectors $p_1, \dots, p_T$, each lying in a different normal cone w.r.t. the set X. Note that, for any $t \in [T]$, if X had nonempty interior and $x_t \in \mathrm{int}(X)$, then $N_X(x_t) = \{0\}$, which would imply equivalence between both oracles. Thus, we should ask ourselves: under which conditions do the iterates of online mirror descent algorithms are guaranteed to lie in the interior of X? Recall that, by property (5.1.ii) of a mirror map, we know that the Bregman projections onto X w.r.t. R lie in $\operatorname{int}(\operatorname{dom} R) \cap X$. Therefore, if we have $\operatorname{int}(\operatorname{dom} R) \cap X \subseteq \operatorname{int} X$, equivalence between EOMD and LOMD holds for $\mathcal{C}$!

In fact, requiring the iterates to lie in the interior of X is excessively strong. For example, the spectraplex is not full-dimensional since it lives inside a hyperplane w.r.t. the trace inner product. Thus, the above argument does not cover the cases considered in [3]. Still, a very similar argument holds if we require the iterates to lie on $\operatorname{ri}(X)$. Indeed, the next theorem proves equivalence of the eager and lazy version of OMD under this assumption. One may note during the proof that a slightly technical issue arises: we need to ensure that both iterates pick the same subgradients whenever they have to choose one, that is, they have to pick a subgradient according to the same well-order on the subdifferentials. Usually the algorithms access the subgradients of a convex function $f\colon \mathbb{E} \to (-\infty, +\infty]$ through a function of the form $x \in X \mapsto g \in \partial f(x)$. Thus, assuming the both algorithms pick the same subgradient given the same function f and the same point x is not a strong assumption.

Theorem 5.6.1. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that X is a nonempty closed set and each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X. Let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a mirror map for X. If $\operatorname{int}(\operatorname{dom} R) \cap X \subseteq \operatorname{ri}(X)$ and both EOMD_R^X and LOMD_R^X use the same well-order on the sets $\partial f(x)$ for $f \in \mathcal{F}$ and $x \in X$, then $\mathrm{EOMD}_R^X = \mathrm{LOMD}_R^X$.

Proof. Suppose $\operatorname{int}(\operatorname{dom} R) \cap X \subseteq \operatorname{ri}(X)$. Let us prove, by induction on $T \in \mathbb{N}$, that

$$\mathrm{EOMD}_R^X(\boldsymbol{f}) = \mathrm{LOMD}_R^X(\boldsymbol{f}), \qquad \forall \boldsymbol{f} \in \mathcal{F}^T.$$

For $T = 0$, we have

$$\{\mathrm{EOMD}_R^X(\langle\rangle)\} = \operatorname*{arg\,min}_{x \in X} R(x) = \{\mathrm{LOMD}_R^X(\langle\rangle)\}.$$

Let $T \in \mathbb{N} \setminus \{0\}$ and $\boldsymbol{f} \in \mathcal{F}^T$. By induction, we have

$$x_t \coloneqq \mathrm{EOMD}_R^X(\boldsymbol{f}_{1:t-1}) = \mathrm{LOMD}_R^X(\boldsymbol{f}_{1:t-1}), \qquad \forall t \in [T].$$

Let us show that $\mathrm{EOMD}_R^X(\boldsymbol{f}) = \mathrm{LOMD}_R^X(\boldsymbol{f})$. Let $g_t \in \partial f_t(x_t)$ be as in the definition of $\mathrm{EOMD}_R^X(\boldsymbol{f})$ for each $t \in [T]$ (which are the same as g_t in the definition of $\mathrm{LOMD}_R^X(\boldsymbol{f})$ is both oracles use the same well-order on the subdifferentials to pick the subgradients). By Theorem 5.4.2, there are $p_1, \dots, p_T \in \mathbb{E}$ with $p_t \in N_X(x_t)$ for each $t \in [T]$ such that

$$\{\mathrm{EOMD}_R^X(\boldsymbol{f})\} = \operatorname*{arg\,min}_{x \in X} \Big(\sum_{t=1}^T \langle g_t, x\rangle + R(x) + \sum_{t=1}^T \langle p_t, x\rangle \Big)$$

By (5.1.ii) from the mirror map definition, $\Pi_X^R(z) \in \operatorname{int}(\operatorname{dom} R) \cap X \subseteq \operatorname{ri}(X)$ for any $z \in \operatorname{int}(\operatorname{dom} R)$. In particular, $x_t = \mathrm{EOMD}_R^X(\boldsymbol{f}_{1:t-1}) \in \operatorname{ri}(X)$ for every $t \in [T]$. Let us show that the terms $\langle p_t, x\rangle$ for $t \in [T]$ do not affect the point that attains the above minimum. More specifically, let us show

$$\langle p_t, x\rangle = \langle p_t, x_t\rangle, \qquad \forall x \in \operatorname{ri}(X), \forall t \in [T]. \tag{5.21}$$

Let $t \in [T]$. Since $p_t \in N_X(x_t)$, we have $\langle p_t, x - x_t\rangle \leq 0$ for every $x \in X$. Thus, suppose there is $\bar{x} \in X$ such that $\langle p_t, \bar{x} - x_t\rangle < 0$. Since $x_t \in \operatorname{ri}(X)$, by Theorem 3.2.2 there is $\mu > 1$ such that $x_\mu \coloneqq \mu x_t + (1 - \mu)\bar{x} \in X$. Since $1 - \mu \leq 0$,

$$\langle p_t, x_\mu - x_t\rangle = (1 - \mu)\langle p_t, \bar{x} - x_t\rangle > 0,$$

a contradiction to the fact that $p_t \in N_X(x_t)$. This proves (5.21). Thus, using that $g_t \in \partial f_t(x_t)$ are the same in the definitions of $\mathrm{EOMD}_R^X(\boldsymbol{f})$ and $\mathrm{LOMD}_R^X(\boldsymbol{f})$ for each $t \in [T]$, and by Theorem 5.5.1 we have

$$\begin{aligned}\{\mathrm{EOMD}_R^X(\boldsymbol{f})\} &= \operatorname*{arg\,min}_{x \in X} \Big(\sum_{t=1}^T \langle g_t, x \rangle + R(x) + \sum_{t=1}^T \langle p_t, x \rangle \Big) \\ &\overset{(5.21)}{=} \operatorname*{arg\,min}_{x \in X} \Big(\sum_{t=1}^T \langle g_t, x \rangle + R(x) \Big) \\ &\overset{\text{Thm. 5.5.1}}{=} \{\mathrm{LOMD}_R^X(\boldsymbol{f})\}. \end{aligned} \qquad \square$$

Chapter 6

Adaptive Regularization

Up to this point, all the algorithms we have seen for OCO problems required us to make a smart choice of regularizer/mirror map strategy based on the parameters of the instance at hand to guarantee good regret bounds. For example, most algorithms seen on Chapters 4 and 5 required previous knowledge of the Lipschitz constant of the functions played by the enemy to properly scale the regularizer and, in this way, guarantee low-regret bounds. Not only that, the strategies seen so far used little to no information from functions previously played by the enemy, usually only looking at most at the round number of the game to compute the new regularizer/mirror map increment. Moreover, the regret bounds from previous chapters depend on the Lipschitz constant of the functions played by the enemy, since by Theorem 3.8.4 the Lipschitz constant usually upper bounds the dual norms of the subgradients. The problem is that this only reflects a worst-case scenario, not giving much information in the case of enemies who play functions with subgradients whose dual norm is small. Thus, one may wonder if it might be possible to derive regret bounds for some OCO algorithms which still depend on the dual norms of the subgradients (in an informative way) instead of using a crude upper bound on such norms. Intuitively, when playing against "easy" enemies, the player oracle should perform better, and we should be able to have better regret guarantees in such cases.

In this chapter we describe algorithms which use information from the subgradients of the functions played by the enemy to better choose its regularizer increments during the game. In order to derive regret bounds we show that these algorithms are special cases of the AdaReg algorithm, first described in [33], which is a clever application of the Adaptive Online Mirror Descent algorithm from Chapter 5. This enables us to make a unified analysis, first presented in [33], of two known and similar algorithms from the OCO literature: the AdaGrad [31] and Online Newton Step [37] algorithms. As a warm-up, on Section 6.1 we describe an online mirror descent algorithm (and its lazy version) with step sizes which adapt based on the subgradient of the enemy's functions. On Section 6.2 we present the AdaReg algorithm, discuss its main ideas, and derive a general regret bound by writing it as an AdaOMD algorithm with a smart choice of mirror map strategy. On Section 6.3 we describe the AdaGrad algorithm and derive a regret bound for it from the regret bound we have for AdaReg. On Section 6.4 we show a more efficient version of AdaGrad which only uses diagonal instead of general matrices to skew the subgradient steps. On Section 6.5 we define exp-concave functions and present the Online Newton Step algorithm, which has a logarithmic regret bound against exp-concave functions. Again, we derive regret bounds for the Online Newton Step algorithm by writing it as an application of the AdaReg algorithm. Finally, on Section 6.6 we show a step size strategy for online gradient descent which attains logarithmic regret against strongly convex functions. Additionally, we show how it can be seen as a "scalar version" of the

Online Newton Step algorithm.

In this chapter we will extensively use norms induced by positive definite matrices. Thus, let us define some notation related to norms based on positive definite matrices. Let $A \in \mathbb{S}^d_{++}$. Abusing the notation of Bregman projection, for every closed convex set $X \subseteq \mathbb{R}^d$ define

$$\{\Pi^A_X(z)\} \coloneqq \operatorname*{arg\,min}_{x \in X} \|x - z\|_A, \qquad \forall z \in \mathbb{R}^d.$$

If $R \coloneqq \frac{1}{2}\|\cdot\|^2_A$, then one may verify that $B_R(x, y) = \frac{1}{2}\|x - y\|^2_A$. Thus, in this case, for any closed convex set $X \subseteq \mathbb{R}^d$ we have $\Pi^R_X = \Pi^A_X$. This equation of Bregman projections may be used without reference throughout this chapter. Moreover, throughout this chapter we consider that $\mathbb{S}^d$ is equipped with the *trace inner product* given by $\langle X, Y\rangle \coloneqq \operatorname{Tr}(XY)$ for every $X, Y \in \mathbb{S}^d$.

6.1 A First Example: Adaptive Online Gradient Descent

In Chapter 4, we have seen different FTRL regularizer strategies for some classes of OCO instances. Among all the regularizer strategies seen in that chapter, the regret bounds when using them in AdaFTRL depended mainly on whether the regularizer strategy was proximal or not, that is, whether the regularizer increment at round $t + 1$ was minimized by the iterate from round t (if $t > 1$) or not. In spite of the discussion made about the differences on the regret bounds we have for each of these regularizer strategy classes (see Theorem 4.4.3, 4.4.4, and the discussion which follows them), in all the cases studied in Chapter 4, proximal regularizer strategies did not yield any significant differences on the final regret bounds if compared to the ones given by non-proximal strategies. Thus, even though the regret bounds given by Theorems 4.4.3 and 4.4.4 hint at the possibility of proximal regularizer strategies being able to adapt better to the functions picked by the enemy, we have not yet exploited this difference in this text.

One aspect which also motivates the investigation of the advantages of proximal regularizer strategies are the connections of Online Mirror Descent algorithms to Adaptive FTRL algorithms seen in Chapter 5. Namely, let $\mathcal{R}$ be a mirror map strategy for some OCO instance $\mathcal{C} \coloneqq (X, \mathcal{F})$. In Section 5.4 we have seen that $\text{AdaOMD}^X_{\mathcal{R}}$ is equal to $\text{AdaFTRL}_{\mathcal{R}'}$ when applied to linear functions, where $\mathcal{R}'$ is a proximal regularizer strategy for $\mathcal{C}$ mostly based on Bregman divergences w.r.t. the regularizer increments given by $\mathcal{R}$. Additionally, in Section 5.5 we have seen that $\text{AdaDA}^X_{\mathcal{R}}$ is equivalent to $\text{AdaFTRL}_{\mathcal{R}''}$ when applied to linear functions, where $\mathcal{R}''$ is equal to $\mathcal{R}$ everywhere but on the empty sequence, where $\mathcal{R}''(\langle\rangle) \coloneqq \mathcal{R}(\langle\rangle) + \delta(\cdot \,|\, X)$. Thus, investigating when and how proximal FTRL regularizer strategies can be advantageous may shed light on the key differences between AdaOMD and AdaDA.

Our main goal in this chapter is to devise mirror map strategies for AdaOMD which take advantage of the adaptiveness present on its regret bound, which is inherited from the regret bound for proximal FTRL regularizer strategies from Theorem 4.4.4. One may build proximal FTRL regularizer strategies similar to the mirror maps seen in this chapter which yield similar regret bounds. Still, in this chapter our focus is on the Adaptive Online Mirror Descent algorithm since looking at adaptive regularizers for AdaOMD yields an unified analysis of two major OCO algorithms: AdaGrad [31] and Online Newton Step [37].

As a warm-up, let us devise versions of the Online Mirror Descent algorithm (with a static mirror map) with time-varying step sizes based on the subgradients of the enemy's functions. Our goal is to devise an algorithm with low-regret guarantees regardless of the number of rounds *and* without knowledge of a bound on the norms of the subgradients. Not only that, we want the regret bound to be better for an enemy which picks "easy" functions, that is, functions whose subgradients have small (dual) norm. For example, let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that each $f \in \mathcal{F}$ is subdifferentiable on $X \subseteq \mathbb{R}^d$ with[1] $\|g\|_2 \leq \rho$ for every $x \in X$ and $g \in \partial f(x)$ and that there is $\theta \in \mathbb{R}_{++}$ such that $\theta \geq \sup_{x,y \in X} \frac{1}{2}\|x - y\|_2^2$. On Section 4.7 we have seen how to build a *proximal* FTRL regularizer strategy which yields a regret bound that holds for any number of rounds. Roughly, Corollary 4.7.3 tells us that AdaFTRL with (cumulative) regularizer R_t at round $t \in \mathbb{N}$ for $t > 1$ given by

$$R_t(x) \coloneqq \sqrt{\frac{\theta}{(t-1)\rho^2}}\left(\frac{1}{2}\|x - x_{t-1}\|_2^2\right), \qquad \forall x \in \mathbb{R}^d,$$

where x_{t-1} is the point chosen by the player at round $t-1$, yields regret smaller than $2\rho\sqrt{\theta T}$ for a game with $T \in \mathbb{N}$ rounds. Let $g_1, \dots, g_T \in \mathbb{R}^d$ be such that, for every $t \in [T]$, we have $g_t \in \partial f_t(x_t)$ where $f_t \in \mathcal{F}$ is the function picked by the enemy on round t. Observe that $\sum_{j=1}^{t-1}\|g_j\|_2^2 \leq (t-1)\rho^2$. That is, the denominator of the multiplicative factor on the regularizer at round t is an upper-bound on the sum of the squared norms of the subgradients of the functions played by the enemy so far (i.e., on rounds 1 up to $t-1$). The idea to make this regularizer strategy adaptive is to, instead of using such an upper bound on the multiplicative factor of the regularizer, use the actual sum of the squared norms of the subgradients, without needing the knowledge of the constant ρ in advance. Interestingly, this implies that in the case where the enemy plays functions whose subgradients have small norm, then the step sizes[2] will be bigger, that is, the algorithm will be more aggressive. In the following theorem we describe a mirror map strategy with the same idea. Later we will see how this strategy need not work very well for non-proximal regularizer strategies.

Theorem 6.1.1. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that X is closed and such that each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X, and let $R \colon \mathbb{E} \to (-\infty, +\infty]$ be a mirror map for X which is σ-strongly convex on X w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$. Moreover, suppose there is $\theta \in \mathbb{R}_{++}$ such that $\theta \geq \sup\{\, B_R(u, x) : u \in X, x \in X \cap \operatorname{int}(\operatorname{dom} R)\}$ and define

$$\begin{aligned}
\eta(\boldsymbol{g}) &\coloneqq \begin{cases} \sqrt{\frac{\theta}{2\sum_{j=1}^t \|g_j\|_2^2}} & \text{if } \boldsymbol{g} \notin \{\mathbf{0}, \langle\rangle\}, \\ 1 & \text{otherwise,} \end{cases} && \text{for every } \boldsymbol{g} = \langle g_1, \dots, g_t\rangle \in \operatorname{Seq}(\mathbb{E}), \text{and} \\
\mathcal{R}(\boldsymbol{f}) &\coloneqq \left(\frac{1}{\eta(\boldsymbol{g}_{1:t})} - [t > 0]\frac{1}{\eta(\boldsymbol{g}_{1:t-1})}\right)\frac{1}{\sqrt{\sigma}}R, && \text{for every } \boldsymbol{f} = \langle f_1, \dots, f_t\rangle \in \operatorname{Seq}(\mathcal{F}), \text{where} \\
& && g_i \in \mathbb{E} \text{ is as in } \operatorname{AdaOMD}_{\mathcal{R}}(\boldsymbol{f}) \text{ for } i \in [t].
\end{aligned}$$

Let ENEMY be an enemy oracle for $\mathcal{C}$, let $T \in \mathbb{N}$, and define

$$(\boldsymbol{x}, \boldsymbol{f}) \coloneqq \operatorname{OCO}_{\mathcal{C}}(\operatorname{AdaOMD}_{\mathcal{R}}^X, \text{ENEMY}, T).$$

Finally, let $g_t \in \partial f_t(x_t)$ be the same as in the definition of $\operatorname{AdaOMD}_{\mathcal{R}}^X(\boldsymbol{f})$ on Algorithm 5.1 for each $t \in [T]$. Then

$$\operatorname{Regret}(\operatorname{AdaOMD}_{\mathcal{R}}^X, \boldsymbol{f}, X) \leq \sqrt{\frac{2\theta}{\sigma}\sum_{t=1}^T \|g_t\|_*^2}.$$

Moreover, if $R = \frac{1}{2}\|\cdot\|_2^2$, then, by setting $\eta_t \coloneqq \eta(\boldsymbol{g}_{1:t-1})$ for each $t \in [T]$ we have

$$x_t = \Pi_X^{\|\cdot\|_2}([t > 1](x_{t-1} - \eta_t g_{t-1})), \qquad \forall t \in [T]. \tag{6.1}$$

[1] We are using the squared ℓ_2-norm because for the sake of simplicity and since it yields the online gradient descent algorithm.

[2] The constant multiplying the regularizer.

Proof. Let us first prove the regret bound. First of all, by the definition of mirror map, it is clear that μR is a mirror map for X for any $\mu \in \mathbb{R}_{++}$ since R is a mirror map for X. Thus, $\mathcal{R}$ is a mirror map strategy for $\mathcal{C}$. For each $t \in \{1, \dots, T+1\}$ define $r_t \coloneqq \mathcal{R}(\boldsymbol{f}_{1:t-1})$. Note that, for every $t \in [T]$,

$$\sum_{i=1}^{t} r_i = \frac{1}{\sqrt{\sigma}\eta_t} R,$$

which is $(\sqrt{\sigma}/\eta_t)$-strongly convex w.r.t. $\|\cdot\|$. By setting $x_0 \coloneqq x_1$, Theorem 5.4.3 yields, for every $u \in X$,

$$\begin{aligned}
\operatorname{Regret}(\mathrm{AdaOMD}^X_{\mathcal{R}}, \boldsymbol{f}, u) &\leq \sum_{t=1}^{T+1} B_{r_t}(u, x_{t-1}) + \frac{1}{2}\sum_{t=1}^{T} \frac{\eta_{t+1}}{\sqrt{\sigma}} \|g_t\|_*^2 \\
&= \frac{1}{2}\sum_{t=1}^{T+1}\left(\frac{1}{\eta_t} - [t > 1]\frac{1}{\eta_{t-1}}\right)\frac{1}{\sqrt{\sigma}} B_R(u, x_{t-1}) + \frac{1}{2\sqrt{\sigma}}\sum_{t=1}^{T} \eta_{t+1}\|g_t\|_*^2 \\
&\leq \frac{\theta}{2\eta_{T+1}\sqrt{\sigma}} + \frac{1}{2\sqrt{\sigma}}\sum_{t=1}^{T}\eta_{t+1}\|g_t\|_*^2 \\
&\leq \frac{1}{2}\sqrt{\frac{2\theta}{\sigma}\sum_{t=1}^{T}\|g_t\|_*^2} + \frac{1}{2}\sqrt{\frac{\theta}{2\sigma}}\sum_{t=1}^{T}\frac{\|g_t\|_*^2}{\sqrt{\sum_{j=1}^{t}\|g_j\|_*^2}} \\
&\overset{\text{Le. 4.6.2}}{\leq} \frac{1}{2}\sqrt{\frac{2\theta}{\sigma}\sum_{t=1}^{T}\|g_t\|_*^2} + \frac{1}{2}\sqrt{\frac{2\theta}{\sigma}\sum_{t=1}^{T}\|g_t\|_*^2} = \sqrt{\frac{2\theta}{\sigma}\sum_{t=1}^{T}\|g_t\|_*^2}.
\end{aligned}$$

Let us show that (6.1) holds for $R \coloneqq \frac{1}{2}\|\cdot\|_2^2$. First, recall that, by Lemma 5.2.1, $\frac{1}{2}\|\cdot\|_2^2$ is a mirror map for X which is 1-strongly convex w.r.t. the ℓ_2-norm. Let us proceed with the proof by induction on $t \in [T]$. For $t = 1$, by the definition of $\mathrm{AdaOMD}^X_{\mathcal{R}}(\langle\rangle)$ we have $\{x_1\} = \arg\min_{x \in X}\|x\|_2 = \{\Pi_X^{\|\cdot\|_2}(0)\}$. Thus, let $t \in \{2, \dots, T\}$, and let R_t and y_t be defined as in the definition of $\mathrm{AdaOMD}^X_{\mathcal{R}}(\boldsymbol{f})$ in Algorithm 5.1, that is,

$$R_t \coloneqq \sum_{i=1}^{t} r_t = \frac{1}{2\eta_t}\|\cdot\|_2^2 \qquad \text{and} \qquad y_t \coloneqq \nabla R_t(x_{t-1}) - g_t = \frac{1}{\eta_t}x_{t-1} - g_t.$$

So, since the ℓ_2-norm is self-dual, by Theorem 3.8.2 we have $(\frac{1}{2}\|\cdot\|_2^2)^* = \frac{1}{2}\|\cdot\|_2^2$. Thus, by Theorem 3.4.3 we have $R_t^*(y) = \frac{1}{2\eta_t}\|\eta_t y\|_2^2 = \frac{\eta_t}{2}\|y\|_2^2$ for every $y \in \mathbb{E}$. Finally, the definition of x_t in $\mathrm{AdaOMD}^X_{\mathcal{R}}(\boldsymbol{f})$ yields

$$x_t = \Pi_X^{R_t}(\nabla R_t^*(y_t)) = \Pi_X^{\|\cdot\|_2}(x_{t-1} - \eta_t g_{t-1}). \qquad \square$$

Note that the above regret bound is at least as good as the one for EOMD from Corollary 5.4.4 (or as the one for the proximal FTRL example from Corollary 4.7.3). Indeed, let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance, R be a mirror map for X which is 1-strongly convex on X w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$, $T \in \mathbb{N}$, and let θ and $g_t \in \mathbb{R}^d$ be as in the above theorem for every $t \in [T]$. If we know $\rho \in \mathbb{R}$ such that $\|g_t\|_* \leq \rho$ for every $t \in \mathbb{N}$, then

$$\sqrt{2\theta\sum_{t=1}^{T}\|g_t\|_*^2} \leq \rho\sqrt{2\theta T},$$

and the latter exactly matches the bound from Corollary 5.4.4. However, the bound given by Theorem 6.1.1 can be much better since, if the enemy picks many functions whose subgradients norm are way smaller than ρ, then the algorithm will be able to exploit this fact (using bigger step sizes) and attain smaller regret.

One may be wondering if a mirror map strategy similar to the one from the above theorem may work for the Adaptive Dual Averaging algorithm. As we have discussed on Section 5.5, Dual Averaging is practically equivalent to the general FTRL algorithm applied to the subgradients of the functions picked by the enemy. Thus, not surprisingly, the regret bound for AdaDA from Corollary 5.5.2 is identical to the regret bound for AdaFTRL with general regularizer strategies given by Theorem 4.4.3. That is, when using AdaDA we do not have the additional adaptiveness present in the proximal FTRL regret bound from Theorem 4.4.4. Even though we can adjust the multiplicative factors of AdaDA with a mirror map similar to the one from the last theorem so that the final regret bound is similar as well, we still need in this case knowledge of the Lipschitz constant (or an upper-bound on the subgradients' norms) of the functions played by the enemy to properly adjust the multiplicative factor of the mirror map strategy. This is one case where the intuition that proximal regularizer strategies can "measure the subgradient from round t with the norm related to the regularizer increment from round $t+1$", as discussed on Section 4.4, yields a relevant difference between regret bounds.

Theorem 6.1.2. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that X is closed and such that each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X and let $R\colon \mathbb{E} \to (-\infty, +\infty]$ be a mirror map for X which is σ-strongly convex on X w.r.t. a norm $\|\cdot\|$ on $\mathbb{E}$. Suppose each $f \in \mathcal{F}$ is ρ-Lipschitz continuous w.r.t. $\|\cdot\|$ on a convex set $D \subseteq \mathbb{E}$ with nonempty interior such that[3] $\operatorname{int} D \supseteq X$, and suppose there is $\theta \in \mathbb{R}_{++}$ such that $\theta \geq \sup\{\, R(u) - R(x) : u, x \in X\}$. Define

$$\eta(\boldsymbol{g}) \coloneqq \sqrt{\frac{\theta}{2(\rho + \sum_{j=1}^{t} \|g_j\|_2^2)}} \qquad \text{for every } \boldsymbol{g} = \langle g_1, \dots, g_t \rangle \in \operatorname{Seq}(\mathbb{R}^d),$$

$$\mathcal{R}(\boldsymbol{f}) \coloneqq \left(\frac{1}{\eta(\boldsymbol{g}_{1:t})} - [t > 0] \frac{1}{\eta(\boldsymbol{g}_{1:t-1})} \right) \frac{1}{\sqrt{\sigma}} R, \qquad \text{for every } \boldsymbol{f} = \langle f_1, \dots, f_t \rangle \in \operatorname{Seq}(\mathcal{F}), \text{where}$$

$$g_i \in \mathbb{R}^d \text{ is as in } \operatorname{AdaDA}_{\mathcal{R}}(\boldsymbol{f}) \text{ for } i \in [t].$$

Let ENEMY be an enemy oracle for $\mathcal{C}$, let $T \in \mathbb{N}$, and define

$$(\boldsymbol{x}, \boldsymbol{f}) \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaDA}_{\mathcal{R}}^{X}, \mathrm{ENEMY}, T).$$

Moreover, let $g_t \in \partial f_t(x_t)$ be the same as in the definition of $\mathrm{AdaDA}_{\mathcal{R}}^{X}(\boldsymbol{f})$ on Algorithm 5.3 for each $t \in [T]$ and set $\eta_t \coloneqq \eta(\boldsymbol{g}_{1:t-1})$ for each $t \in [T]$. Then

$$\operatorname{Regret}(\mathrm{AdaDA}_{\mathcal{R}}^{X}, \boldsymbol{f}, X) \leq \sqrt{\frac{2\theta}{\sigma}(\rho + \sum_{t=1}^{T-1} \|g_t\|_*^2)}.$$

Proof. Since μR is a mirror map for X for any $\mu \in \mathbb{R}_{++}$, we have that $\mathcal{R}$ is a mirror map strategy for $\mathcal{C}$. For each $t \in \{1, \dots, T+1\}$ define $r_t \coloneqq \mathcal{R}(\boldsymbol{f}_{1:t-1})$. Note that, for every $t \in [T]$,

$$\sum_{i=1}^{t} r_i = \frac{1}{\eta_t \sqrt{\sigma}} R,$$

[3] Here we assume this so that, for any $f \in \mathcal{F}$, *any* subgradient of f at a point of $\mathcal{F}$ has small dual norm by Theorem 3.8.4. Still, if one can control the choice of subgradients of AdaDA, one may only require $X \subseteq D$.

which is $(\sqrt{\sigma}/\eta_t)$-strongly convex w.r.t. $\|\cdot\|$. By setting $x_0 \coloneqq x_1$, Corollary 5.5.2 yields, for every $u \in X$,

$$
\begin{aligned}
\operatorname{Regret}(\mathrm{AdaDA}^X_{\mathcal{R}}, \boldsymbol{f}, u) &\leq \sum_{t=1}^{T} (r_t(u) - r_t(x_t)) + \frac{1}{2}\sum_{t=1}^{T} \frac{\eta_t}{\sqrt{\sigma}} \|g_t\|_*^2 \\
&= \frac{1}{2}\sum_{t=1}^{T}\Big(\frac{1}{\eta_t} - [t > 1]\frac{1}{\eta_{t-1}}\Big)\frac{1}{\sqrt{\sigma}}(R(u) - R(x_t)) + \frac{1}{2\sqrt{\sigma}}\sum_{t=1}^{T} \eta_t \|g_t\|_*^2 \\
&\leq \frac{\theta}{2\eta_T\sqrt{\sigma}} + \frac{1}{2\sqrt{\sigma}}\sum_{t=1}^{T}\eta_t\|g_t\|_*^2 \\
&\leq \frac{1}{2}\sqrt{\frac{2\theta}{\sigma}\Big(\rho + \sum_{t=1}^{T-1}\|g_t\|_*^2\Big)} + \frac{1}{2}\sqrt{\frac{\theta}{2\sigma}}\sum_{t=1}^{T}\frac{\|g_t\|_*^2}{\sqrt{\rho + \sum_{j=1}^{t-1}\|g_j\|_*^2}} \\
&\leq \frac{1}{2}\sqrt{\frac{2\theta}{\sigma}\Big(\rho + \sum_{t=1}^{T-1}\|g_t\|_*^2\Big)} + \frac{1}{2}\sqrt{\frac{\theta}{2\sigma}}\sum_{t=1}^{T}\frac{\|g_t\|_*^2}{\sqrt{\sum_{j=1}^{t}\|g_j\|_*^2}} \\
&\overset{\text{Le. 4.6.2}}{\leq} \frac{1}{2}\sqrt{\frac{2\theta}{\sigma}\Big(\rho + \sum_{t=1}^{T-1}\|g_t\|_*^2\Big)} + \frac{1}{2}\sqrt{\frac{2\theta}{\sigma}\sum_{t=1}^{T}\|g_t\|_*^2} \\
&\leq \sqrt{\frac{2\theta}{\sigma}\Big(\rho + \sum_{t=1}^{T-1}\|g_t\|_*^2\Big)}. \qquad \square
\end{aligned}
$$

6.2 The AdaReg Algorithm

On the previous section, we have presented and analyzed a version of the Online Mirror Descent algorithm with a fixed mirror map but adaptive step sizes. We are still not using the full capabilities of the AdaOMD algorithm. Since the mirror map is fixed, the norm w.r.t. which the mirror map is strongly convex is also fixed, which is also restrictive. One may wonder if we can use a mirror map strategy which outputs, at each round, a mirror map which is strongly convex w.r.t. a different norm in order to get even better regret bounds.

Another question is: *how* to choose the norm(s) and, consequently, the mirror map(s) (and step sizes) to use in AdaOMD? In many interesting cases, such as in the prediction with expert advice problem, one has enough previous information to choose a properly scaled mirror map (the negative entropy function in the experts' case) which yields a good regret bound. However, in other cases it may not be clear which mirror map and step sizes yield good regret bounds, or it may be the case that one does not have enough previous information to make such a decision. Thus, an ideal scenario would be one in which the mirror map adapts itself to the problem at hand without the need of much prior information about the instance. For example, in the mirror map strategy from Theorem 6.1.1 from the previous section, the player does not need to know the Lipschitz constant of the functions played by the enemy to obtain the asymptotically best regret bounds given by the theorems we have proved.

One OCO algorithm which uses this idea of learning a regularizer during the game is the *Adaptive Gradient (AdaGrad)* algorithm [31], which is described formally in the next section. Its basic idea is to use, at round t, the norm induced by a positive definite matrix H_t, where H_t is constructed from rank-one matrices based mainly on the subgradients of the previous functions picked by the enemy. In this way, the update rule of the iterate of AdaGrad at round $t \in \mathbb{N} \setminus \{0\}$ is of the form

$$x_t = \Pi_X^{H_t^{-1}}([t > 1](x_{t-1} - H_t g_{t-1})), \tag{6.2}$$

where $X \subseteq \mathbb{R}^d$ is the set from where the player is allowed to pick his points, x_{t-1} and g_{t-1} are, respectively, the iterate and subgradient of enemy's choice at round $t-1$, and $\Pi_X^{H_t^{-1}}$ is the projection onto X w.r.t. the norm $\|\cdot\|_{H_t^{-1}}$. Thus, the matrix H_t intuitively skews the subgradient of the previous round in a desirable way, and adjusts the projection to balance the skewed subgradient step.

Another algorithm for Online Convex Optimization with a similar update rule is the *Online Newton Step* (ONS) algorithm [37], which is guaranteed to attain regret with a logarithmic dependence on the number of rounds if the functions played by the enemy are guaranteed to be differentiable and *exp-concave*, a generalization of strong convexity which will be formally described and discussed later. The algorithm's update rule is of the same form of (6.2), with only the choice of matrix H_t being different, even though it is still a function of the subgradients of the previous choices of the enemy.

In spite of their similarities, AdaGrad and ONS were discovered independently and each had non-related analyses. The authors of [33] proposed the *AdaReg* algorithm and showed that both AdaGrad and ONS are special cases of AdaReg. This sheds some light in the intuition behind these algorithms. Additionally, it leaves room for the creation of other similar and interesting OCO algorithms. We describe a player oracle which implements the AdaReg algorithm in Algorithm 6.1. The AdaReg algorithm is parameterized by a function $\Phi \colon \mathbb{S}^d \to (-\infty, +\infty]$, called *meta-regularizer*, which dictates which matrices to use in the update of (6.2).

Definition 6.2.1 (Meta-regularizer)**.** A function $\Phi \colon \mathbb{S}^d \to (-\infty, +\infty]$ is a **meta-regularizer** if, for any $G \in \mathbb{S}^d_{++}$,

(6.3.i) the infimum $\inf_{H \in \mathbb{S}^d_{++}} \big(\langle G, H\rangle + \Phi(H)\big)$ is attained,

(6.3.ii) for any $g \in \mathbb{E}$, if

$$H_T \in \operatorname*{arg\,min}_{H \in \mathbb{S}^d_{++}} \big(\langle G, H\rangle + \Phi(H)\big) \qquad \text{and} \qquad H_{T+1} \in \operatorname*{arg\,min}_{H \in \mathbb{S}^d_{++}} \big(\langle G + gg^{\mathsf{T}}, H\rangle + \Phi(H)\big),$$

then $H_T \succeq H_{T+1}$ (which implies $H_{T+1} \succeq H_T$ since H_T and H_{T+1} are positive definite).

Let us look a little bit closer at the definition of AdaReg on Algorithm 6.1 for a game with $T \in \mathbb{N}$ rounds and some $\varepsilon > 0$ and, during this discussion, we look at the reasons for the conditions imposed on meta-regularizers. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance, let $T \in \mathbb{N}$, and let $\boldsymbol{f} \in \mathcal{F}^T$. Moreover, let $t \in \{0, \ldots, T-1\}$. At round $t+1$, i.e. when the algorithm is computing x_{t+1}, the algorithm builds a positive definite matrix G_t, which is the sum of rank-one matrices (based on the subgradients of the enemy's functions) plus[4] εI. Then AdaReg performs its key step: the choice of the matrix H_{t+1} which it uses to perform the "skewed" gradient step as in (6.2). Namely, AdaReg with meta-regularizer Φ picks H_{t+1} that attains

$$\inf_{H \in \mathbb{S}^d_{++}} (\langle G_t, H\rangle + \Phi(H)), \tag{6.4}$$

[4] The main goal of this latter term is to ensure that G_t is invertible, but the value of $\varepsilon > 0$ may affect the guarantees of the algorithms we shall see later on.

Algorithm 6.1 Definition of $\mathrm{AdaReg}_{\Phi}^{X}(\langle f_1, \dots, f_T\rangle)$

```
Input:
    (i) A closed convex set $X \subseteq \mathbb{R}^d$,
    (ii) Convex functions $f_1, \dots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{R}^d}$ such that $f_t$ is subdifferentiable on $X$ for each $t \in [T]$,
    (iii) A meta-regularizer $\Phi \colon \mathbb{S}^d \to (-\infty, +\infty]$,
    (iv) A real number $\varepsilon > 0$ (usually clear from the context)
Output: $x_{T+1} \in X$
  $G_0 \leftarrow \varepsilon I$
  Let $H_1 \in \arg\min_{H \in \mathbb{S}^d_{++}} (\langle G_0, H\rangle + \Phi(H))$
  Let $\{x_1\} \leftarrow \arg\min_{x \in X} \|x\|_{H_1^{-1}} = \arg\min_{x \in X} x^{\mathsf{T}} H_1^{-1} x$
  for $t = 1$ to $T$ do
        ▷ Computations for round $t+1$
      Compute $g_t \in \partial f_t(x_t)$
      $G_t \leftarrow G_{t-1} + g_t g_t^{\mathsf{T}}$
      Compute $H_{t+1} \in \arg\min_{H \in \mathbb{S}^d_{++}} (\langle G_t, H\rangle + \Phi(H))$
      $x_{t+1} \leftarrow \Pi_X^{H_{t+1}^{-1}}(x_t - H_{t+1} g_t)$
  return $x_{T+1}$
```

where the above infimum is attained by property (6.3.i). Although the above expression can seem cryptic at first, it has a very elegant interpretation. By the definition of the AdaReg oracle, we have $G_t = \varepsilon I + \sum_{i=1}^{t} g_i g_i^{\mathsf{T}}$, where for each $t \in [T]$ the vector $g_t \in \mathbb{R}^d$ is a subgradient as defined in $\mathrm{AdaReg}_{\Phi}^{X}(\boldsymbol{f}_{1:t})$. Thus, for every $H \in \mathbb{S}^d_{++}$ we have

$$\begin{aligned}\langle G_t, H\rangle + \Phi(H) &= \sum_{i=1}^{t} \langle g_i g_i^{\mathsf{T}}, H\rangle + \varepsilon \operatorname{Tr}(H) + \Phi(H) = \sum_{i=1}^{t} \operatorname{Tr}(g_i g_i^{\mathsf{T}} H) + \varepsilon \operatorname{Tr}(H) + \Phi(H) \\ &= \sum_{i=1}^{t} g_i^{\mathsf{T}} H g_i + \varepsilon \operatorname{Tr}(H) + \Phi(H) = \sum_{i=1}^{t} \|g_i\|_H^2 + \varepsilon \operatorname{Tr}(H) + \Phi(H).\end{aligned} \tag{6.5}$$

That is, the matrix H_{t+1} is chosen so that the size of the subgradients measured by its induced norm are minimized while still not making $\Phi(H) + \varepsilon \operatorname{Tr}(H)$ too high[5]. Recall that the sum of the squared norms of the subgradients is directly connected to almost all the regret bounds seen on Chapters 4 and 5. Thus, H_{t+1} can be seen roughly as the best matrix with low complexity w.r.t. the meta-regularizer Φ through which to measure/see the subgradients of the functions played by the enemy so far. Another way to see the choice of H_{t+1}, which is the main idea the authors of [33] use in their analysis of AdaReg, is to note that H_{t+1} is the point picked by $\mathrm{FTRL}_{\Phi'}(\langle \psi_1, \dots, \psi_t\rangle)$, where $\psi_i(H) \coloneqq \langle g_i g_i^{\mathsf{T}}, H\rangle$ for each $i \in [t]$ and $\Phi' \coloneqq \Phi + \varepsilon \operatorname{Tr}(\cdot) + \delta(\cdot \,|\, \mathbb{S}^d_{++})$. That is, the problem of choosing a matrix norm through which to measure the subgradients played by the enemy is seen as a separate OCO instance! On the regret bounds which we prove later in this section it will be clear how well this strategy minimizes the norms of the subgradients.

The reader may still be confused about condition (6.3.ii) since, during the above discussion, this condition was never mentioned. Not only that, the AdaReg oracle from Algorithm 6.1 does not seem to need this condition for all of its operations to be well-defined. Indeed, condition (6.3.ii) from the

[5] The value of $\Phi(H) + \varepsilon \operatorname{Tr}(H)$ here can be interpreted as the "complexity" of the norm $\|\cdot\|_H$.

definition of meta-regularizers is not needed for the definition of AdaReg to make sense. However, as we shall soon see, this condition is fundamental for the regret bounds that we derive to hold. Interestingly, even though condition (6.3.ii) is not explicitly stated on [33], all the meta-regularizers the authors use satisfy this condition (which is used explicitly in their proofs).

As one may have noticed, the update on (6.2) resembles a lot the update from the Adaptive Online Mirror Descent. Indeed, to bound the regret of AdaReg we will write it as an Adaptive Online Mirror Descent algorithm with a carefully[6] crafted mirror map strategy, which we formally define in Algorithm 6.2.

Algorithm 6.2 Definition of $\big[\mathcal{M}(X, \Phi, \varepsilon)\big]\big(\langle f_1, \dots, f_T\rangle\big)$

```
Input:
    (i) A closed convex set $X \subseteq \mathbb{R}^d$,
    (ii) Convex functions $f_1, \dots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{R}^d}$ such that $f_t$ is
         subdifferentiable on $X$ for each $t \in [T]$,
    (iii) A meta-regularizer $\Phi\colon \mathbb{S}^d \to (-\infty, +\infty]$,
    (iv) A real number $\varepsilon > 0$
Output: A function $r_{T+1}\colon \mathbb{R}^d \to (-\infty, +\infty]$
  for $t = 1$ to $T$ do        ▷ Capture subgradients used on rounds $1, \dots, T$
    $x_t \leftarrow \mathrm{AdaOMD}^X_{\mathcal{M}(X,\Phi,\varepsilon)}(\langle f_1, \dots, f_{t-1}\rangle)$
      ▷ Equip the right well-order to match the subgradient choice of $\mathrm{AdaOMD}^X_{\mathcal{M}(X,\Phi,\varepsilon)}$
    Equip $\partial f_t(x_t)$ with the same well-order used by $\mathrm{AdaOMD}^X_{\mathcal{M}(X,\Phi,\varepsilon)}$
    Pick $g_t \in \partial f_t(x_t)$
   ▷ Compute the mirror map increment for round $T + 1$
  $G_{T-1} \leftarrow \varepsilon I + \sum_{t=1}^{T-1} g_t g_t^{\mathsf{T}}$
  $G_T \leftarrow G_{T-1} + g_T g_T^{\mathsf{T}}$
  Let $H_T \in \arg\min_{H \in \mathbb{S}^d_{++}} (\langle G_{T-1}, H\rangle + \Phi(H))$
  Let $H_{T+1} \in \arg\min_{H \in \mathbb{S}^d_{++}} (\langle G_T, H\rangle + \Phi(H))$
  $D_{T+1} \leftarrow H_{T+1}^{-1} - [T > 0] H_T^{-1}$
  return $x \in \mathbb{R}^d \mapsto \frac{1}{2} x^{\mathsf{T}} D_{T+1} x = \frac{1}{2}\|x\|^2_{D_{T+1}}$
```

Note that if Φ on the definition of $\mathcal{M}$ in Algorithm 6.2 is a meta-regularizer, then the matrices D_t on the definition of $\mathcal{M}$ are positive semidefinite by condition (6.3.ii). That is, the functions delivered by $\mathcal{M}$ are always convex in this case. This is important if we want to write AdaReg in the form of an AdaOMD algorithm, since in order for $\mathcal{M}$ to be a mirror map strategy (and for us to apply the regret bounds we have proved on Chapter 5), we need the functions it delivers at each round, i.e. the mirror map increments, to be convex. In the following lemma we prove that if we plug into $\mathcal{M}$ a meta-regularizer, then $\mathcal{M}$ is a mirror map strategy and the mirror map it builds at each round are scaled squared matrix norms.

Lemma 6.2.2. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that X is a closed set and such that each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X. Moreover, let $\varepsilon > 0$ and let $\Phi\colon \mathbb{S}^d \to (-\infty, +\infty]$ be a meta-regularizer. Let $T \in \mathbb{N}$, let $\boldsymbol{f} \in \mathcal{F}^T$, and let $H_t \in \mathbb{S}^d_{++}$ and $D_t \in \mathbb{S}^d$

[6] One may note that we need to ensure the subgradients used by the mirror map strategy matches the ones used by the AdaOMD oracle, and we do so by synchronizing the well-orders used on the subdifferentials by the AdaOMD oracle and by the mirror map strategy. See the discussion following Definition 4.2.1 to recall why we equip well-orders to the subdifferentials used.

be as defined in $\mathcal{M}(X, \Phi, \varepsilon)(\boldsymbol{f}_{1:t-1})$ for each $t \in \{1, \dots, T+1\}$. Finally, for every $t \in \{1, \dots, T+1\}$ define

$$r_t \coloneqq \mathcal{M}(X, \Phi, \varepsilon))(\boldsymbol{f}_{1:t-1}) \qquad \text{and} \qquad R_t \coloneqq \sum_{i=1}^{t} r_t.$$

Then $\mathcal{M}(X, \Phi, \varepsilon)$ is a mirror map strategy for $\mathcal{C}$ which is differentiable on $\mathbb{R}^d$. Moreover, for every $t \in \{1, \dots, T+1\}$ we have $D_t \succeq 0$, $r_t = \frac{1}{2}\|\cdot\|_{D_t}^2$, and $R_t = \frac{1}{2}\|\cdot\|_{H_t^{-1}}^2$. Moreover, R_t is 1-strongly convex w.r.t. $\|\cdot\|_{H_t^{-1}}$ on $\mathbb{R}^d$ for every $t \in \{1, \dots, T+1\}$.

Proof. Let $t \in \{1, \dots, T+1\}$. First, note that since Φ is a meta-regularizer, by condition (6.3.ii) we have that $D_t \succeq 0$. Let us now show that

$$r_t = \tfrac{1}{2}\|\cdot\|_{D_t}^2 \qquad \text{and} \qquad R_t = \tfrac{1}{2}\|\cdot\|_{H_t^{-1}}^2. \tag{6.6}$$

Note that the form of r_t as in (6.6) holds by the definition of $[\mathcal{M}(X, \Phi, \varepsilon)](\boldsymbol{f}_{1:t-1})$. Moreover, for every $x \in \mathbb{R}^d$ we have

$$R_t(x) = \sum_{i=1}^{t} r_i(x) = \sum_{i=1}^{t} \frac{1}{2} x^{\mathsf{T}} D_i x = \frac{1}{2} x^{\mathsf{T}} \Big(\sum_{i=1}^{t} (H_i^{-1} - [i > 1] H_{i-1}^{-1}) \Big) x = \frac{1}{2} x^{\mathsf{T}} H_t^{-1} x.$$

This proves (6.6). Let us now show that

$\mathcal{M}(X, \Phi, \varepsilon)$ is a mirror map strategy for $\mathcal{C}$ which is differentiable on $\mathbb{R}^d$ and such that R_t is 1-strongly convex w.r.t. $\|\cdot\|_{H_t^{-1}}$ on $\mathbb{E}$. (6.7)

First, note that r_t is two-times continuously differentiable (and, thus, closed) with $\nabla^2 r_t(x) = D_t$ for any $x \in \mathbb{R}^d$. Since $D_t \succeq 0$ by the conditions of a meta-regularizer, by Lemma 3.1.1 we conclude that r_t is convex. It only remains to show that R_t is a mirror for X. That is, we need to prove that

(i) R_t closed, proper, 1-strongly convex[7] on $\mathbb{R}^d$ w.r.t. $\|\cdot\|_{H_t^{-1}}$ and differentiable on $\mathbb{R}^d$,

(ii) $\mathbb{R}^d = \operatorname{int}(\operatorname{dom} R_t)$,

(iii) for any $y \in \mathbb{R}^d$, the infima $\inf_{x \in X} B_{R_t}(x, y)$ and $\inf_{x \in X} R_t(x)$ are attained, and

(iv) $\{\, \nabla R(x) : x \in \mathbb{R}^d \} = \mathbb{R}^d$.

First, note that (ii) clearly holds, and since $\nabla R_t(x) = H_t^{-1} x$ for any $x \in \mathbb{R}^d$, we conclude that (iv) holds since H_t^{-1} is invertible. Moreover, R_t is two-times continuously differentiable on $\mathbb{R}^d$, which implies that R_t is proper and closed (in fact, continuous), and since $\nabla^2 R_t(x) = H_t^{-1} \succ 0$ for any $x \in \mathbb{R}^d$, by Lemma 3.1.1 we conclude that R_t is convex. Note that if R_t is strongly convex, $B_{R_y}(\cdot, y)$ also is for any $y \in \mathbb{R}^d$, and then then the infima from (iii) would be attained by Lemma 3.9.14. Thus, it only remains to show that H_t is 1-strongly convex w.r.t. $\|\cdot\|_{H_t^{-1}}$. To see that[8], note that for every

[7] The definition of mirror map requires strict convexity, but recall that strong convexity implies strict convexity by definition.

[8] One easier way to prove strong convexity of R_t is to note that $\|\cdot\|_{H_t^{-1}}$ is a norm induced by the inner product $(x, y) \in \mathbb{R}^d \times \mathbb{R}^d \mapsto x^{\mathsf{T}} H_t^{-1} y$ and then use Lemma 3.9.5. However, using direct computations seems less cumbersome in this case

$x, y \in \mathbb{R}^d$ we have

$$\begin{aligned}\tfrac{1}{2}\|x-y\|_{H_t^{-1}}^2 &= \tfrac{1}{2}(x-y)^\mathsf{T} H_t^{-1}(x-y) = \tfrac{1}{2}x^\mathsf{T} H_t^{-1}x + \tfrac{1}{2}y^\mathsf{T} H_t^{-1} y - x^\mathsf{T} H_t^{-1} y \\ &= \tfrac{1}{2}x^\mathsf{T} H_t^{-1} x + -\tfrac{1}{2}y^\mathsf{T} H_t^{-1} y - (H_t^{-1}y)^\mathsf{T}(x-y) \\ &= \tfrac{1}{2}\|x\|_{H_t^{-1}}^2 + -\tfrac{1}{2}\|y\|_{H_t^{-1}}^2 - (H_t^{-1}y)^\mathsf{T}(x-y) \\ &= R_t(x) - R_t(y) - \nabla R_t(y)^\mathsf{T}(x-y).\end{aligned}$$

By Theorem 3.9.7 we conclude that R_t is 1-strongly convex w.r.t. $\|\cdot\|_{H_t^{-1}}$, which concludes the proof of (6.7). □

With the above lemma, we have the guarantee that $\mathcal{M}$ applied to a meta-regularizer and other properly chosen parameters is indeed a mirror map. In the next theorem we prove the main result of this section: if $\mathcal{C} := (X, \mathcal{F})$ is an OCO instance, $\varepsilon > 0$, and $\Phi\colon \mathbb{S}^d \to (-\infty, +\infty]$ is a meta-regularizer, then $\mathrm{AdaReg}_\Phi^X = \mathrm{AdaOMD}_{\mathcal{M}(X,\Phi,\varepsilon)}^X$. This theorem will allow us to derive regret bounds for AdaReg and the Online Newton Step algorithm from the regret bounds we have for AdaOMD.

Theorem 6.2.3. Let $\mathcal{C} := (X, \mathcal{F})$ be an OCO instance such that X is a nonempty closed set and such that each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X. Moreover, let $\varepsilon > 0$ and $\Phi\colon \mathbb{S}^d \to (-\infty, +\infty]$ be a meta-regularizer. Finally, suppose the same well-order[9] over the sets used in the definition of AdaReg_Φ^X are the same as in the definition of $\mathrm{AdaOMD}_{\mathcal{M}(X,\Phi,\varepsilon)}^X$. Then, for any $T \in \mathbb{N}$ and $\boldsymbol{f} \in \mathcal{F}^T$,

$$\mathrm{AdaReg}_\Phi^X(\boldsymbol{f}) = \mathrm{AdaOMD}_{\mathcal{M}(X,\Phi,\varepsilon)}^X(\boldsymbol{f}),$$

and the matrix $H_{T+1} \in \mathbb{S}_{++}^d$ as defined in $\mathrm{AdaReg}_\Phi^X(\boldsymbol{f})$ is equal to the matrix $H_{T+1} \in \mathbb{S}_{++}^d$ as defined in $\mathcal{M}(X, \Phi, \varepsilon)(\boldsymbol{f})$.

Proof. Let $T \in \mathbb{N}$ and $\boldsymbol{f} \in \mathcal{F}^T$. Moreover, for each $t \in \{1, \dots, T+1\}$ define

$$R_t := \sum_{i=1}^{t} [\mathcal{M}(X, \Phi, \varepsilon)](\boldsymbol{f}_{1:t-1}) \qquad \text{and} \qquad x_t := \mathrm{AdaOMD}_{\mathcal{M}(X,\Phi,\varepsilon)}^X(\boldsymbol{f}_{1:t-1}).$$

Finally, for each $t \in \{1, \dots, T+1\}$ let $H_t \in \mathbb{S}_{++}^d$ be defined as in $[\mathcal{M}(X, \Phi, \varepsilon)](\boldsymbol{f}_{1:t-1})$. Note that for every $t \in \{1, \dots, T+1\}$ we have that H_t is the same as the one in $\mathrm{AdaReg}_\Phi^X(\boldsymbol{f})$ by definition and since all the sets used in the definitions of the AdaReg and AdaOMD oracles are the same. Let us prove by induction on $t \in \{1, \dots, T+1\}$ that

$$x_t = \mathrm{AdaReg}_\Phi^X(\boldsymbol{f}_{1:t-1}), \qquad \forall t \in \{1, \dots, T+1\}.$$

For $t = 1$, we have $R_1(x) = \frac{1}{2}\|\cdot\|_{H_1^{-1}}^2$. Thus, the definition of $\mathrm{AdaOMD}_{\mathcal{M}(X,\Phi,\varepsilon)}^X$ yields

$$\{x_1\} = \operatorname*{arg\,min}_{x \in X} R_1(x) = \operatorname*{arg\,min}_{x \in X} \|x\|_{H_1^{-1}} = \{\mathrm{AdaReg}_\Phi^X(\langle\rangle)\}.$$

Let $t \in \{2, \dots, T+1\}$. By Lemma 6.2.2, we have that $R_t = \frac{1}{2}\|\cdot\|_{H_t^{-1}}^2$. With that, we have

$$y_t := \nabla R_t(x_{t-1}) - g_{t-1} = H_t^{-1} x_{t-1} - g_{t-1}.$$

[9] This is only a technical assumption to assure that, if we have in both algorithms a statement such as "let $g_t \in \partial f_t(x_t)$", then in both algorithms the element picked from the set $\partial f_t(x_t)$ is the same.

By Lemma 3.8.5, the dual norm of $\|\cdot\|_{H_t^{-1}}$ is $\|\cdot\|_{H_t}$, and by Theorem 3.8.2 we have $R_t^* = \frac{1}{2}\|\cdot\|_{H_t}^2$. This together with the definition of $\mathrm{AdaOMD}(\boldsymbol{f}_{1:t-1})$ yields

$$\begin{aligned} x_t = \Pi_X^{H_t^{-1}}(\nabla R_t^*(y_t)) &= \Pi_X^{H_t^{-1}}(H_t(H_t^{-1}x_{t-1} - g_{t-1})) \\ &= \Pi_X^{H_t^{-1}}(x_{t-1} - H_t g_{t-1}) = \mathrm{AdaReg}_\Phi^X(\boldsymbol{f}_{1:t-1}). \end{aligned}$$

□

Finally, let us now show a regret bound for the AdaReg algorithm. The proof of the next regret bound has two key steps. The first is almost obvious given the previous theorem: use the regret bound for AdaOMD we have proved previously (see Theorem 5.4.3). This together with the previous theorem yields a regret bound which is arguably not very useful. The second key step in the next proof is to use the FTL–BTL Lemma on the matrices H_t used by AdaReg to show the optimality, in some sense, of this choice of matrices with respect to the minimization of the norms of the subgradients. This yields a neat regret bound, whose intuition we discuss after proving the next theorem.

Theorem 6.2.4. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that $X \subseteq \mathbb{R}^d$ is a nonempty closed set and such that each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X. Let $\varepsilon > 0$ and let $\Phi\colon \mathbb{S}^d \to (-\infty, +\infty]$ be a meta-regularizer. Let $T \in \mathbb{N}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and define

$$(\boldsymbol{x}, \boldsymbol{f}) \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaReg}_\Phi^X, \mathrm{ENEMY}, T).$$

For each $t \in \{1, \dots, T+1\}$, let $H_t \in \mathbb{S}_{++}^d$ be as in the definition of $\mathrm{AdaReg}_\Phi^X(\boldsymbol{f}_{1:t-1})$ and define $D_t \coloneqq H_t^{-1} - [t > 1]H_{t-1}^{-1}$. Finally, let $G_T \in \mathbb{S}_{++}^d$ be as in the definition of $\mathrm{AdaReg}_\Phi^X(\boldsymbol{f})$. Then, for any $u \in X$ and for $x_0 \coloneqq x_1$,

$$\mathrm{Regret}(\mathrm{AdaReg}_\Phi^X, \boldsymbol{f}, u) \leq \frac{1}{2}\sum_{t=0}^{T}\|u - x_t\|_{D_{t+1}}^2 + \frac{1}{2}\min_{H \in \mathbb{S}_{++}^d}(\langle G_T, H\rangle + \Phi(H) - \Phi(H_1)).$$

Proof. By Theorem 6.2.3, we have $\mathrm{AdaReg}_\Phi^X = \mathrm{AdaOMD}_{\mathcal{M}(X,\Phi,\varepsilon)}^X$ and, in particular

$$\mathrm{Regret}(\mathrm{AdaReg}_\Phi^X, \boldsymbol{f}, u) = \mathrm{Regret}(\mathrm{AdaOMD}_{\mathcal{M}(X,\Phi,\varepsilon)}^X, \boldsymbol{f}, u), \qquad \forall u \in \mathbb{R}^d.$$

Thus, it suffices to bound the right hand side of the above equation. For every $t \in \{1, \dots, T+1\}$, define

$$r_t \coloneqq [\mathcal{M}(X, \Phi, \varepsilon)](\boldsymbol{f}_{1:t-1}) \qquad \text{and} \qquad R_t \coloneqq \sum_{i=1}^{t} r_i.$$

By Lemma 6.2.2 we know that $\mathcal{M}(X, \Phi, \varepsilon)$ is a mirror map strategy for $\mathcal{C}$ and that for every $t \in \{1, \dots, T+1\}$ we have $r_t = \frac{1}{2}\|\cdot\|_{D_t}^2$ and $R_t = \frac{1}{2}\|\cdot\|_{H_t^{-1}}^2$, the latter being 1-strongly convex w.r.t. $\|\cdot\|_{H_t^{-1}}$ on $\mathbb{R}^d$. By Lemma 3.8.5 we have that the dual norm of $\|\cdot\|_{H_t^{-1}}$ is $\|\cdot\|_{H_t}$. Finally, set $x_0 \coloneqq x_1$ and let $g_t \in \partial f_t(x_t)$ be as in the definition of $\mathrm{AdaOMD}_{\mathcal{M}(X,\Phi,\varepsilon)}(\boldsymbol{f})$ for every $t \in [T]$. Then, for every $u \in \mathbb{R}^d$ Theorem 5.4.3 yields

$$\begin{aligned} \mathrm{Regret}(\mathrm{AdaOMD}_{\mathcal{M}(X,\Phi,\varepsilon)}^X, \boldsymbol{f}, u) &\leq \sum_{t=1}^{T+1} B_{r_t}(u, x_{t-1}) + \frac{1}{2}\sum_{t=1}^{T}\|g_t\|_{H_{t+1}}^2 \\ &= \frac{1}{2}\sum_{t=1}^{T+1}\|u - x_{t-1}\|_{D_t}^2 + \frac{1}{2}\sum_{t=1}^{T}\|g_t\|_{H_{t+1}}^2. \end{aligned}$$

Thus, it only remains to show that

$$\sum_{t=1}^{T}\|g_t\|_{H_{t+1}}^2 \leq \min_{H\in\mathbb{S}^d_{++}} (\langle G_T, H\rangle + \Phi(H) - \Phi(H_1)). \tag{6.8}$$

Let $t \in [T]$. Note that

$$\|g_t\|_{H_{t+1}}^2 = g_t^{\mathsf{T}} H_{t+1} g_t = \operatorname{Tr}(g_t^{\mathsf{T}} H_{t+1} g_t) = \operatorname{Tr}(g_t g_t^{\mathsf{T}} H_{t+1}) = \langle g_t g_t^{\mathsf{T}}, H_{t+1}\rangle. \tag{6.9}$$

Define $\Phi' := \Phi + \varepsilon \operatorname{Tr}(\cdot) + \delta(\cdot \,|\, \mathbb{S}^d_{++})$. By definition of $\mathcal{M}(X, \Phi, \varepsilon)$, we have

$$H_t \in \operatorname*{arg\,min}_{H\in\mathbb{S}^d_{++}} \Big(\sum_{i=1}^{t-1}\langle g_i g_i^{\mathsf{T}}, H\rangle + \varepsilon\langle I, H\rangle + \Phi(H)\Big) = \operatorname*{arg\,min}_{H\in\mathbb{S}^d} \Big(\sum_{i=1}^{t-1}\langle g_i g_i^{\mathsf{T}}, H\rangle + \Phi'(H)\Big).$$

Thus, by setting $\psi_t(H) := \langle g_t g_t^{\mathsf{T}}, H\rangle$ for every $t \in [T]$ and $H \in \mathbb{S}^d$, we conclude that

$$H_t = \mathrm{FTRL}_{\Phi}(\langle \psi_1, \dots, \psi_{t-1}\rangle), \qquad \forall t \in \{1, \dots, T+1\}.$$

Therefore, the FTL–BTL Lemma (Lemma 4.9.1) together with (6.9) yields, for any $H \in \mathbb{S}^d_{++}$,

$$\begin{aligned}
\sum_{t=1}^{T}\|g_t\|_{H_{t+1}}^2 = \sum_{t=1}^{T}\langle g_t g_t^{\mathsf{T}}, H_{t+1}\rangle = \sum_{t=1}^{T}\psi_t(H_{t+1}) \qquad & \text{by (6.9)},\\
\leq \Phi'(H) - \Phi'(H_1) + \sum_{t=1}^{T}\psi_t(H) \qquad & \text{by Lemma 4.9.1},\\
= \Phi(H) - \Phi(H_1) - \varepsilon\operatorname{Tr}(H_1) + \varepsilon\operatorname{Tr}(H) + \sum_{t=1}^{T}\psi_t(H) \qquad & \text{by the definition of } \Phi',\\
= \Phi(H) - \Phi(H_1) - \varepsilon\operatorname{Tr}(H_1) + \Big\langle \varepsilon I + \sum_{t=1}^{T} g_t g_t^{\mathsf{T}}, H\Big\rangle \qquad & \text{by the definition of } \psi_t,\\
= \Phi(H) - \Phi(H_1) - \varepsilon\operatorname{Tr}(H_1) + \langle G_T, H\rangle \qquad & \text{by the definition of } G_T,\\
\leq \Phi(H) - \Phi(H_1) + \langle G_T, H\rangle \qquad & \text{by Cor. 1.1.2 since } H_1 \succeq 0.
\end{aligned}$$

Taking the infimum over $H \in \mathbb{S}^d_{++}$ on the last inequality above, which is attained since Φ is a meta-regularizer, completes the proof of (6.8). □

Let us try to understand the regret bound we have just proved for an OCO instance $\mathcal{C} := (X, \mathcal{F})$ in a game of $T \in \mathbb{N}$ rounds against an enemy oracle ENEMY for $\mathcal{C}$. Let Φ be a meta-regularizer, $\varepsilon > 0$, and set

$$(\boldsymbol{x}, \boldsymbol{f}) := \mathrm{OCO}_{\mathcal{C}}(\mathrm{PLAYER}, \mathrm{ENEMY}, T).$$

Finally, for each $t \in [T]$ let $g_t \in \partial f_t(x_t)$ be as in the definition of $\mathrm{AdaReg}_{\Phi}^{X}(\boldsymbol{f})$. The second term on the above regret bound, as we have already discussed (see the discussion regarding (6.4) and (6.5)), has a very nice meaning regarding the optimality of the norm which measure the sizes of the subgradients used. Namely, by setting $G_T := \varepsilon I + \sum_{t=1}^{T} g_t g_t^{\mathsf{T}}$ and letting H_1 be as in $\mathrm{AdaReg}_{\Phi}^{X}(\boldsymbol{f})$, we have

$$\min_{H\in\mathbb{S}^d_{++}} (\langle G_T, H\rangle + \Phi(H) - \Phi(H_1)) = \min_{H\in\mathbb{S}^d_{++}} (\sum_{t=1}^{T}\|g_t\|_H^2 + \varepsilon\operatorname{Tr}(H) + \Phi(H) - \Phi(H_1)). \tag{6.10}$$

That is, the norm in the regret bound which measures the size of the subgradients is, in some sense, optimal: it is the matrix norm which minimizes the sum of the squared norms of the subgradients plus a regularization term given by Φ and the trace of the matrix. That is, when choosing Φ there is a trade-off between minimizing the norms of the subgradients and minimizing the regularization term $\Phi(H) - \Phi(H_1)$ in (6.10). The terms in which Φ appears may clutter one's intuition, but when we look at specific choices of Φ, the intuition on (6.10) is usually stronger. For example, in the regret bound for the AdaGrad algorithm that we study in the next section, this term becomes $\min\{\sum_{t=1}^{T}\|g_t\|_H^2 : H \in \mathbb{S}^d_{++}, \operatorname{Tr}(H) \leq 1\}$.

Moreover, the first term of the regret bound from Theorem 6.2.4 can be seen as a measure of stability of the choices of the matrices $H_t \in \mathbb{S}^d_{++}$ by AdaReg (scaled by the diameter of X) for each $t \in \{1, \dots, T+1\}$. To see that, note that if $D_t \coloneqq H_t^{-1} - [t > 1]H_{t-1}^{-1}$ for each $t \in \{1, \dots, T+1\}$ and $x_0 \coloneqq x_1$, then, for any $u \in X$,

$$\sum_{t=0}^{T}\|u - x_t\|_{D_{t+1}}^2 = \sum_{t=0}^{T}(\|u - x_t\|_{H_{t+1}^{-1}}^2 - [t > 0]\|u - x_t\|_{H_t^{-1}}^2).$$

If the matrices H_t and H_{t+1} are similar for every $t \in [T]$, then the above terms are relatively small. We say "relatively" since this value invariably depends on the diameter $\theta \coloneqq \sup_{x,u \in X}\|x - u\|_2^2$ of X w.r.t. the ℓ_2-norm. Thus, when picking a meta-regularizer, one wants to avoid abrupt matrix transitions from one round to another. The next corollary shows a bound in the case where θ is finite, which makes the first term of the regret bound arguably clearer to interpret: it is the diameter of X times the sum of $\operatorname{Tr}(D_t)$ for $t \in \{1, \dots, T+1\}$. In this case, it is clearer how both the diameter of X and the stability of the choices of the matrices $H_1, \dots, H_{T+1}$ affect this term simultaneously.

Lemma 6.2.5. Let $A \in \mathbb{S}^d_+$ and let $v \in \mathbb{R}^d$. Then

$$v^{\mathsf{T}}Av \leq \|v\|_2^2 \operatorname{Tr}(A).$$

Proof. By the Cauchy-Schwarz inequality, we have

$$\begin{aligned} v^{\mathsf{T}}Av &= \operatorname{Tr}(v^{\mathsf{T}}Av) = \operatorname{Tr}(vv^{\mathsf{T}}A) = \langle vv^{\mathsf{T}}, A\rangle \\ &\leq \sqrt{\operatorname{Tr}(vv^{\mathsf{T}}vv^{\mathsf{T}})}\sqrt{\operatorname{Tr}(A^2)} = \|v\|_2^2\sqrt{\operatorname{Tr}(A^2)}. \end{aligned}$$

Thus, it only remains tho show that $\sqrt{\operatorname{Tr}(A^2)} \leq \operatorname{Tr}(A)$. Note that, for any $\alpha, \beta \in \mathbb{R}_+$, we have

$$(\sqrt{\alpha} + \sqrt{\beta})^2 = \alpha + 2\sqrt{\alpha\beta} + \beta \geq \alpha + \beta \implies \sqrt{\alpha} + \sqrt{\beta} \geq \sqrt{\alpha + \beta}.$$

Thus, by a simple induction we get

$$\Big(\sum_{i=1}^{d} u_i\Big)^{\frac{1}{2}} \leq \sum_{i=1}^{d}\sqrt{u_i}, \qquad \forall u \in \mathbb{R}^d_+.$$

Moreover, by Corollary 1.1.2 we have $\operatorname{Tr}(A^2) = \mathbb{1}^{\mathsf{T}}\lambda^{\uparrow}(A^2)$. Therefore,

$$\operatorname{Tr}(A^2)^{\frac{1}{2}} = \Big(\sum_{i=1}^{d}\lambda_i^{\uparrow}(A^2)\Big)^{\frac{1}{2}} = \Big(\sum_{i=1}^{d}\lambda_i^{\uparrow}(A)^2\Big)^{\frac{1}{2}} \leq \sum_{i=1}^{d}\lambda_i^{\uparrow}(A) = \operatorname{Tr}(A). \qquad \square$$

Corollary 6.2.6. Let $\mathcal{C} := (X, \mathcal{F})$ be an OCO instance such that X is a nonempty closed set and such that each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X. Let $\varepsilon > 0$ and let $\Phi \colon \mathbb{S}^d \to (-\infty, +\infty]$ be a meta-regularizer. Let $T \in \mathbb{N}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and define

$$(\boldsymbol{x}, \boldsymbol{f}) := \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaReg}_{\Phi}^{X}, \mathrm{ENEMY}, T).$$

Moreover, let $H_1, H_{T+1} \in \mathbb{S}^d_{++}$ be as in the definition of $\mathrm{AdaReg}_{\Phi}^{X}(\boldsymbol{f})$. Finally, let $G_T \in \mathbb{S}^d_{++}$ be as in the definition of $\mathrm{AdaReg}_{\Phi}^{X}(\boldsymbol{f})$ and suppose there is $\theta \in \mathbb{R}_{++}$ such that $\theta \geq \{\, \|x - u\|_2^2 : x, u \in X\}$. Then, for every $u \in X$ and for $x_0 := x_1$,

$$\mathrm{Regret}(\mathrm{AdaReg}_{\Phi}^{X}, \boldsymbol{f}, u) \leq \frac{\theta}{2} \mathrm{Tr}(H_{T+1}^{-1}) + \frac{1}{2}(\langle G_T, H_{T+1} \rangle + \Phi(H_{T+1}) - \Phi(H_1)).$$

Proof. Let $u \in X$, and, for every $t \in \{1, \dots, T+1\}$, let $H_t \in \mathbb{S}^d_{++}$ be as in the definition of $\mathrm{AdaReg}_{\Phi}^{X}(\boldsymbol{f})$ and define $D_t := H_t^{-1} - [t > 1]H_{t-1}^{-1}$. By Theorem 6.2.4, we have

$$\mathrm{Regret}(\mathrm{AdaReg}_{\Phi}^{X}, \boldsymbol{f}, u) \leq \frac{1}{2} \sum_{t=0}^{T} \|u - x_t\|_{D_{t+1}}^2 + \frac{1}{2} \min_{H \in \mathbb{S}^d_{++}} (\langle G_T, H \rangle + \Phi(H) - \Phi(H_1)).$$

By definition we have that H_{T+1} attains the above minimum. Thus, it only remains to bound the term $\sum_{t=0}^{T} \|u - x_t\|_{D_{t+1}}^2$. Note that

$$\begin{aligned}
\sum_{t=0}^{T} \|u - x_t\|_{D_{t+1}}^2 &= \sum_{t=0}^{T} (u - x_t)^{\mathsf{T}} D_{t+1} (u - x_T) \\
&\overset{\text{Le. 6.2.5}}{\leq} \sum_{t=0}^{T} \|u - x_t\|_2^2 \,\mathrm{Tr}(D_{t+1}) \\
&\leq \theta \sum_{t=0}^{T} \mathrm{Tr}(D_{t+1}) = \theta \,\mathrm{Tr}\Big(\sum_{t=0}^{T} D_{t+1}\Big) \\
&= \theta \,\mathrm{Tr}(H_{T+1}^{-1}).
\end{aligned}$$

□

6.3 The AdaGrad Algorithm

In this section we describe the AdaGrad algorithm from [31]. Its idea is a generalization of the Online Mirror Descent algorithm with adaptive step sizes (with the squared ℓ_2-norm as a mirror map) seen on Section 6.1. For $t \in \mathbb{N} \setminus \{0\}$, the algorithm from Section 6.1 performs at round $t+1$ a step in the direction of minus subgradient of size $O((\sum_{i=1}^{t} \|g_i\|^2)^{-1/2})$, where g_i is a subgradient of the enemy's functions at the player's iterate at round $i \in [t]$. Still, making the stepsize depend only on the norms of the subgradients might be sub-optimal.

For example, let $\mathcal{C} := (\Delta_d, \mathcal{F})$ be an instance of the randomized prediction with expert advice problem, where $d \in \mathbb{N} \setminus \{0\}$. Let $T \in \mathbb{N}$ be such that $T \geq d$ and let ENEMY be an enemy oracle for $\mathcal{C}$ which, for any $i \in [d]$, plays the function $x \in \Delta_d \mapsto e_i^{\mathsf{T}} x$ at any round $t \in [T]$ such that we have $t \equiv i - 1 \pmod{d}$. That is, at each interval of d rounds each expert is assigned a penalty of 1 in exactly one round, and no penalty at all at all the other rounds of the interval. Thus, after exactly nd rounds for any $n \in \mathbb{N}$ such that $nd \leq T$, a player will have the same amount of information about each one of the experts. Thus, one should expect the player's iterate at this round to be close to $\frac{1}{d}\mathbb{1}$, translating the intuition that all the experts are equal from the perspective of the player. However, note that with the strategy from Section 6.1 the step size at round $t \in [T]$ is $\Theta(\sqrt{t^{-1}})$. That is, even though the amount of information revealed by the enemy about each expert is the same after each interval of d rounds, the weights attributed to the experts at the end of each interval is not uniform, with their weights depending on the *order* in which they were penalized. Intuitively this may seem weird since the order of appearance should not matter much in a game against this enemy oracle.

The AdaGrad algorithm can be interpreted as trying to make the subgradients steps adaptive in a more nuanced fashion. Instead of only adapting the step size based on the norm of the subgradients, at round t the algorithm skews the subgradient with a matrix H_t built from rank-one updates based on the subgradients of previous rounds, and then performs the subgradient step. Finally, we define a player oracle which formally implements the Adaptive Gradient algorithm on Algorithm 6.3.

Algorithm 6.3 Definition of $\text{AdaGrad}^X(\langle f_1, \ldots, f_T \rangle)$

Input:

(i) A closed convex set $X \subseteq \mathbb{E}$,

(ii) Convex functions $f_1, \ldots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$ such that f_t is subdifferentiable on X for each $t \in [T]$,

(iii) Positive real numbers $\varepsilon > 0$ and $\eta > 0$ (usually clear from the context).

Output: $x_{T+1} \in X$

```
  $G_0 \leftarrow \varepsilon I$
  $\{x_1\} \leftarrow \arg\min_{x \in X} \|x\|_2$
  for $t = 1$ to $T$ do
        ▷ Computations for round $t + 1$
      Compute $g_t \in \partial f_t(x_t)$
      $G_t \leftarrow G_{t-1} + g_t g_t^{\mathsf{T}}$
      $x_{t+1} \leftarrow \Pi_X^{G_t^{1/2}}(x_t - \eta G_t^{-1/2} g_t)$
  return $x_{T+1}$
```

As we have discussed, the AdaGrad algorithm can be seen as a generalization of the algorithm from Theorem 6.1.1 with the squared ℓ_2-norm as the mirror map. In Theorem 6.1.1 the algorithm performs, at a round $t \in \mathbb{N} \setminus \{0\}$, a subgradient step with step size $O\big(\big(\sum_{i=1}^{t-1} \|g_i\|_2^2\big)^{-1/2}\big)$, where $g_1, \ldots, g_{t-1} \in \mathbb{R}^d$ are the subgradients (from the enemy's functions) used by the player oracle on past rounds. The AdaGrad algorithm, on the other hand, performs at round $t \in \mathbb{N} \setminus \{0\}$ a step in the direction of the subgradient *skewed* by a matrix $G_{t-1}^{-1/2}$, where $G_{t-1} \in \mathbb{S}_{++}^d$ is a matrix built from rank-one updates based on the subgradients $g_1, \ldots, g_{t-1} \in \mathbb{R}^d$ of the enemy's past functions (plus a small multiple of the identity to ensure that G_{t-1} is invertible). Additionally, the projection onto the set $X \subseteq \mathbb{R}^d$ from where the player can pick its points performed by AdaGrad is skewed by the matrix $G_t^{1/2}$.

To derive a regret bound for the above algorithm, we will show that the AdaGrad algorithm is equivalent to the AdaReg algorithm with a special (and simple) choice of meta-regularizer. An instructive way to define our meta-regularizer Φ is to define its behavior on the eigenvalues of the matrices[10] given to Φ as input, and then to see which known matrix operation this yields (if any). In this way, the analysis of the algorithm is greatly simplified since, for convex functions applied to eigenvalues, we have tools to compute subgradients and, thus, check optimality conditions, as we

[10] Namely, given a matrix $X \in \mathbb{S}^d$ we will apply a *symmetric* convex function $f : \mathbb{R}^d \to (-\infty, +\infty]$ on $\lambda^{\uparrow}(X)$ and then build a new matrix $X' \in \mathbb{S}^d$ from $f(\lambda^{\uparrow}(X))$. For details, see Section 3.7.

have seen on Section 3.7.

On the next lemma we look at the form and at some properties of the meta-regularizer which yields AdaGrad.

Lemma 6.3.1. Let $\eta > 0$, define $f \colon \mathbb{R}^d \to (-\infty, +\infty]$ by

$$f(x) \coloneqq \eta^2 \sum_{i=1}^{d} [x_i > 0] \frac{1}{x_i} + \delta(x \,|\, \mathbb{R}^d_{++}), \qquad \forall x \in \mathbb{R}^d,$$

and set $\Phi \coloneqq f_{\mathbb{S}}$. Then f is a proper closed convex function,

$$\Phi(H) = \eta^2 \operatorname{Tr}(H^{-1}), \qquad \text{and} \qquad \nabla\Phi(H) = -\eta^2 H^{-2} \qquad \forall H \in \mathbb{S}^d_{++}. \tag{6.11}$$

Additionally, for every $G \in \mathbb{S}^d_{++}$ the infimum $\inf_{H \in \mathbb{S}^d_{++}}(\langle G, H\rangle + \Phi(H))$ is attained by $\eta G^{-1/2}$. Moreover, Φ is a meta-regularizer and for every $\varepsilon > 0$ and for a certain well-order over the sets used by the oracles AdaGrad^X and AdaReg^X_Φ we have $\text{AdaGrad}^X = \text{AdaReg}^X_\Phi$ for every nonempty closed and convex set $X \subseteq \mathbb{R}^d$.

Proof. First, let us verify the f is a proper closed convex function. First of all, it is clear that f. Define $\phi(\alpha) \coloneqq [\alpha > 0]\alpha^{-1} + \delta(\alpha \,|\, \mathbb{R}_{++})$. Since $\phi(\alpha)'' = 2\alpha^{-3} > 0$ for every $\alpha \in \mathbb{R}_{++}$, by Lemma 3.1.1 we conclude that ϕ is convex. Since $f = \sum_{i=1}^d \phi(x_i)$, we conclude that f is convex. Finally, since $\lim_{\alpha \to 0} \phi(\alpha) = +\infty$ and ϕ is positive throughout $\mathbb{R}$, we conclude that $\liminf_{x \to \bar{x}} f(x) = +\infty = f(\bar{x})$ for any $\bar{x} \in \mathbb{R}^d_+ \setminus \mathbb{R}^d_{++}$. Therefore, f is closed.

Let $H \in \mathbb{S}^d_{++}$, set $\lambda \coloneqq \lambda^{\uparrow}(H)$, and set $\Lambda \coloneqq \operatorname{Diag}(\lambda)$. Let us first show that (6.11) holds. By the Spectral Decomposition Theorem (Theorem 1.1.1), there is an orthogonal matrix $Q \in \mathbb{R}^{d \times d}$ such that $H = Q\Lambda Q^{\mathsf{T}}$. Since $H \succ 0$, by Theorem 1.1.3 we know that $\lambda > 0$. Hence, Λ is invertible with $(\Lambda^{-1})_{i,j} = [i = j]\lambda_i^{-1}$ for every $i, j \in [d]$. Hence,

$$HQ\Lambda^{-1}Q^{\mathsf{T}} = Q\Lambda Q^{\mathsf{T}} Q\Lambda^{-1}Q^{\mathsf{T}} = Q\Lambda\Lambda^{-1}Q^{\mathsf{T}} = QQ^{\mathsf{T}} = I,$$

that is, $Q\Lambda^{-1}Q^{\mathsf{T}} = H^{-1}$. Finally, we have

$$\Phi(H) = f(\lambda) = \eta^2 \sum_{i=1}^{d} \lambda_i^{-1} = \eta^2 \operatorname{Tr}(\Lambda^{-1}) = \eta^2 \operatorname{Tr}(Q^{\mathsf{T}} Q \Lambda^{-1}) = \eta^2 \operatorname{Tr}(Q\Lambda^{-1}Q^{\mathsf{T}}) = \eta^2 \operatorname{Tr}(H^{-1}).$$

Moreover, note that $\nabla f(\lambda)_i = -\eta^2 \lambda_i^{-2}$ for every $i \in [d]$. Hence, $\operatorname{Diag}(\nabla f(\lambda)) = -\eta^2 \Lambda^{-2}$, and by Corollary 3.7.5 we have

$$\nabla\Phi(H) = Q \operatorname{Diag}(\nabla f(\lambda)) Q^{\mathsf{T}} = -\eta^2 Q\Lambda^{-2}Q^{\mathsf{T}} = -\eta^2 (Q\Lambda^{-1}Q^{\mathsf{T}})^2 = -\eta^2 H^{-2}.$$

This proves (6.11). Let $G \in \mathbb{S}^d_{++}$. Let us now show that

$$\{\eta G^{-1/2}\} = \operatorname*{arg\,min}_{H \in \mathbb{S}^d_{++}} (\langle G, H\rangle + \Phi(H)). \tag{6.12}$$

Let $\hat{H} \in \mathbb{S}^d_{++}$. Since $\operatorname{dom} f = \mathbb{R}^d$, we have $\operatorname{ri}(\operatorname{dom} \Phi) = \mathbb{S}^d$, and since $\mathbb{S}^d_{++}$ is an open set with nonempty interior, then $N_{\mathbb{S}^d_{++}}(\hat{H}) = \{0\}$. Therefore, $\mathbb{S}^d_{++} \cap \operatorname{ri}(\operatorname{dom}\Phi)$ is nonempty, and by Theorem 3.6.2 we have

$$\hat{H} \in \operatorname*{arg\,min}_{H \in \mathbb{S}^d_{++}} (\langle G, H\rangle + \Phi(H)) \iff G + \nabla\Phi(\hat{H}) = 0 \overset{(6.11)}{\iff} \eta^2 \hat{H}^{-2} = G \iff (\hat{H}^{-1})^2 = \frac{1}{\eta^2} G$$

$$\overset{\text{Prop. 1.1.4}}{\iff} \hat{H}^{-1} = \frac{1}{\eta} G^{1/2} \iff \hat{H} = \eta G^{-1/2}.$$

This finishes the proof of (6.12).

Now let us show that Φ is a meta-regularizer. Let $T \in \mathbb{N}$ and $\boldsymbol{g} \in (\mathbb{R}^d)^T$. Moreover, let $\varepsilon > 0$ and set $G_{T-1} \coloneqq \varepsilon I + \sum_{t=1}^{T-1} g_t g_t^{\mathsf{T}}$ and $G_T \coloneqq G_{T-1} + g_T g_T^{\mathsf{T}}$. Condition (6.3.i) of a meta-regularizer is satisfied by Φ since, by (6.12), we know that $\inf_{H \in \mathbb{S}^d_{++}} (\langle H, G_T \rangle + \Phi(H))$ is attained by $\eta G_T^{-1/2}$. Thus, set $H_{T+1} \coloneqq \eta G_T^{-1/2}$ and $H_T \coloneqq \eta G_{T-1}^{-1/2}$. Note that

$$H_{T+1}^{-1} - H_T^{-1} = \tfrac{1}{\eta}(G_T^{1/2} - G_{T-1}^{1/2}).$$

Since $\eta > 0$ and $G_T - G_{T-1} = g_T g_T^{\mathsf{T}} \succeq 0$, by Lemma 1.1.5 we have that $\frac{1}{\eta}(G_T^{1/2} - G_{T-1}^{1/2}) \succeq 0$. That is, Φ satisfies condition (6.3.ii), which completes the proof that Φ is a meta-regularizer.

Last but not least, let us show that $\mathrm{AdaGrad}^X = \mathrm{AdaReg}^X_\Phi$ for any nonempty closed and convex set $X \subseteq \mathbb{R}^d$ and any $\varepsilon > 0$ (recall that we already have $\eta > 0$ from the statement of the lemma). Let $X \subseteq X \subseteq \mathbb{R}^d$ be a nonempty closed and convex set and let $\varepsilon > 0$. Moreover, Let $\boldsymbol{f} \coloneqq \langle f_1, \dots, f_T \rangle \in \mathrm{Seq}((-\infty, +\infty]^{\mathbb{R}^d})$ be such that f_t is subdifferentiable on X for every $t \in [T]$. Let us show that $\mathrm{AdaGrad}^X(\boldsymbol{f}_{1:t-1}) = \mathrm{AdaReg}^X_\Phi(\boldsymbol{f}_{1:t-1})$ by induction on $t \in [T]$. Set $x_1 \coloneqq \mathrm{AdaReg}^X_\Phi(\langle\rangle)$ and let $H_1 \in \mathbb{S}^d_{++}$ be as in the definition of $\mathrm{AdaReg}^X_\Phi(\langle\rangle)$. By (6.12), we know that $H_1 = (\eta/\sqrt{\varepsilon}) I$. Thus,

$$x_1 \in \operatorname*{arg\,min}_{x \in X} \|x\|_{H_1^{-1}} = \operatorname*{arg\,min}_{x \in X} x^{\mathsf{T}} H_1^{-1} x = \operatorname*{arg\,min}_{x \in X} \tfrac{\sqrt{\varepsilon}}{\eta} x^{\mathsf{T}} x = \operatorname*{arg\,min}_{x \in X} \|x\|_2^2 = \operatorname*{arg\,min}_{x \in X} \|x\|_2.$$

Since the squared ℓ_2-norm is strongly convex (by Lemma 3.9.5), we have that x_1 is the unique point that attains the above minima. Thus, $x_1 = \mathrm{AdaGrad}^X(\langle\rangle)$. Let $t \in \{2, \dots, T+1\}$, and let $g_{t-1} \in \mathbb{R}^d$ and $G_{t-1} \in \mathbb{S}^d_{++}$ be as in the definition of $x_t \coloneqq \mathrm{AdaReg}^X_\Phi(\boldsymbol{f}_{1:t-1})$ (which are equal to g_{t-1} and G_{t-1} in the definition of $\mathrm{AdaGrad}^X_\Phi(\boldsymbol{f}_{1:t-1})$ with a proper choice of well-order on the subdifferentials used). Finally, let $H_t \in \mathbb{S}^d_{++}$ be as in the definition of $\mathrm{AdaReg}^X_\Phi(\boldsymbol{f}_{1:t-1})$ and set $x_{t-1} \coloneqq \mathrm{AdaReg}^X_\Phi(\boldsymbol{f}_{1:t-2}) = \mathrm{AdaGrad}^X(\boldsymbol{f}_{1:t-2})$. Then,

$$x_t = \Pi_X^{H_t^{-1}}(x_{t-1} - H_t g_{t-1}) \overset{(6.12)}{=} \Pi_X^{G_{t-1}^{1/2}}(x_{t-1} - \eta G_{t-1}^{-1/2} g_{t-1}) = \mathrm{AdaGrad}^X(\boldsymbol{f}_{1:t-1}). \qquad \square$$

Now that we know which meta-regularizer to use to write AdaGrad as AdaReg, we can apply the results from Section 6.2 to obtain regret bounds for AdaGrad.

Theorem 6.3.2. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that $X \subseteq \mathbb{R}^d$ is a nonempty closed set and such that each $f \in \mathcal{F}$ is a proper closed function which is subdifferentiable on X. Let[11] $\varepsilon > 0$, let $T \in \mathbb{N}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and define

$$(\boldsymbol{x}, \boldsymbol{f}) \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaGrad}^X, \mathrm{ENEMY}, T).$$

Moreover, let $G_T \in \mathbb{S}^d_{++}$ be as in the definition of $\mathrm{AdaGrad}^X(\boldsymbol{f})$, suppose there is $\theta \in \mathbb{R}_{++}$ such that $\theta \geq \sup\{\, \|u - x\|_2^2 : u, x \in X\}$, and set $\eta \coloneqq \sqrt{\theta/2}$. Then,

$$\mathrm{Regret}(\mathrm{AdaGrad}^X, \boldsymbol{f}, X) \leq \sqrt{2\theta}\, \mathrm{Tr}(G_T^{1/2}).$$

Proof. Define $f \colon \mathbb{R}^d \to (-\infty, +\infty]$ by $f(x) \coloneqq \eta^2 \sum_{i=1}^d [x_i \neq 0] \frac{1}{x_i}$ for each $x \in \mathbb{R}^d$, and set $\Phi \coloneqq f_{\mathbb{S}}$. By Lemma 6.3.1, we have $\Phi(H) = \eta^2 \mathrm{Tr}(H^{-1})$ for any $H \in \mathbb{S}^d_{++}$ and $\mathrm{AdaReg}^X_\Phi = \mathrm{AdaGrad}^X$.

[11] This ε is the needed to define AdaGrad, although it does not appear in the regret bound.

Thus, we only need to bound the regret of $\mathrm{AdaReg}_{\Phi}^{X}$. Let $H_1, H_{T+1} \in \mathbb{S}^d_{++}$ be as in the definition of $\mathrm{AdaReg}_{\Phi}^{X}(\boldsymbol{f})$. By Lemma 6.3.1 together with the definitions of H_1 and H_{T+1} we have

$$H_1 = \eta(\varepsilon I)^{-1/2} = \frac{\eta}{\sqrt{\varepsilon}} I \qquad \text{and} \qquad H_{T+1} = \eta G_T^{-1/2}.$$

Thus, by Corollary 6.2.6 we have, for every $u \in X$,

$$\begin{aligned}
\mathrm{Regret}(\mathrm{AdaReg}_{\Phi}^{X}, \boldsymbol{f}, u) &\leq \frac{\theta}{2}\operatorname{Tr}(H_{T+1}^{-1}) + \frac{1}{2}(\langle G_T, H_{T+1}\rangle + \Phi(H_{T+1}) - \Phi(H_1)) \\
&= \frac{\theta}{2\eta}\operatorname{Tr}(G_T^{1/2}) + \frac{1}{2}(\eta \operatorname{Tr}(G_T^{1/2}) + \Phi(\eta G_T^{-1/2}) - \Phi(\tfrac{\eta}{\sqrt{\varepsilon}} I)) \\
&= \frac{\theta}{2\eta}\operatorname{Tr}(G_T^{1/2}) + \frac{1}{2}(\eta \operatorname{Tr}(G_T^{1/2}) + \eta \operatorname{Tr}(G_T^{1/2}) - \eta\sqrt{\varepsilon}\operatorname{Tr}(I)) \\
&\leq \frac{\theta}{2\eta}\operatorname{Tr}(G_T^{1/2}) + \eta \operatorname{Tr}(G_T^{1/2}) \\
&= \sqrt{\frac{\theta}{2}}\operatorname{Tr}(G_T^{1/2}) + \sqrt{\frac{\theta}{2}}\operatorname{Tr}(G_T^{1/2}) = \sqrt{2\theta}\operatorname{Tr}(G_T^{1/2}).
\end{aligned}$$

□

One may have noticed that the value of ε is free to be as small as we want in the above result. This suggests that this parameter may not be needed after all. However, if $\varepsilon = 0$, then the matrices G_t in the definition of AdaGrad are not necessarily invertible anymore. To solve this one could use the *Moore-Penrose pseudo-inverse* instead of the inverse of the matrices. For the sake of brevity, we have chosen to describe only the case where all matrices are invertible (that is, the case for $\varepsilon > 0$).

One problem with the regret bound from Theorem 6.3.2 is that it is hard to interpret how good it is. The intuitive meaning of $\operatorname{Tr}(G_T^{1/2})$ is not clear, where $G_T \in \mathbb{S}^d_{++}$ is defined as in Theorem 6.3.2. We know that $\operatorname{Tr}(G_T)$ is the sum of the squared ℓ_2-norms of the subgradients (plus εd, where $d \in \mathbb{N}$ is the dimension of the problem), but interpreting $\operatorname{Tr}(G^{1/2})$ is way harder. The next proposition sheds some light about the meaning of the above regret bound and shows how it may be as good as the one from Section 6.1, for example.

Proposition 6.3.3 ([31, Lemma 15])**.** Let $A \in \mathbb{S}^d_{++}$. Then $\inf\{\operatorname{Tr}(X^{-1}A) : X \in \mathbb{S}^d_{++}, \operatorname{Tr}(X) = 1\}$ is attained by $A^{1/2}/\operatorname{Tr}(A^{1/2})$.

With the above proposition, we have the following corollary which makes the regret bound for AdaGrad way more palatable.

Corollary 6.3.4. Let $\varepsilon > 0$ and $\boldsymbol{g} \in (\mathbb{R}^d)^T$ for some $T \in \mathbb{N}$. Moreover, set $G_T \coloneqq \varepsilon I + \sum_{t=1}^{T} g_t g_t^{\mathsf{T}}$ and $\mathcal{S}_d \coloneqq \{X \in \mathbb{S}^d_{+} : \operatorname{Tr}(X) = 1\}$. Then

$$\operatorname{Tr}(G_T^{1/2}) = \sqrt{\min_{H \in \mathcal{S}_d \cap \mathbb{S}^d_{++}} \left(\varepsilon \operatorname{Tr}(H^{-1}) + \sum_{t=1}^{T} \|g_t\|_{H^{-1}}^2\right)}.$$

Proof. Set $\mathcal{S}_{++} \coloneqq \mathcal{S}_d \cap \mathbb{S}^d_{++}$. By Proposition 6.3.3, we have

$$\begin{aligned}
\sqrt{\min_{H \in \mathcal{S}_{++}} \left(\varepsilon \operatorname{Tr}(H^{-1}) + \sum_{t=1}^{T} \|g_t\|_{H^{-1}}^2\right)} &= \sqrt{\min_{H \in \mathcal{S}_{++}} \left(\varepsilon \langle I, H^{-1}\rangle + \sum_{t=1}^{T} \langle g_t g_t^{\mathsf{T}}, H^{-1}\rangle\right)} \\
&= \sqrt{\min_{H \in \mathcal{S}_{++}} \langle G_T, H^{-1}\rangle} = \sqrt{\left(\operatorname{Tr}(G_T^{1/2})\right)^2} \\
&= \operatorname{Tr}(G_T^{1/2}).
\end{aligned}$$

□

That is, the trace in the regret of AdaGrad has value close[12] to the square-root of sum of the norms of the gradients, where the norm is the best among all norms induced by matrices H^{-1} with $H \in \mathbb{S}^d_{++}$ in the *spectraplex* $\mathcal{S}_d$, that is, such that $\mathrm{Tr}(H) = 1$. Note that, for any $g \in \mathbb{R}^d$, by setting $\bar{H} \coloneqq d^{-1} H \in \mathcal{S}_d$ we have

$$\|g\|_{\bar{H}}^2 = \tfrac{1}{d} g^\mathsf{T} I g = \tfrac{1}{d}\|g\|_2^2 \qquad \text{and} \qquad \mathrm{Tr}(\bar{H}^{-1}) = d\,\mathrm{Tr}(I) = d^2. \tag{6.13}$$

Thus, for $\varepsilon > 0$ small enough (namely, smaller than d^{-2}), we conclude that the regret bound from Theorem 6.3.2 as good as the one from Theorem 6.1.1.

6.4 Diagonal AdaGrad Algorithm

Although the AdaGrad algorithm from the previous section has good regret guarantees, its implementation can be quite slow. One of the reasons why Online Convex Optimization algorithms are attractive is, besides their good regret bound guarantees, the usually low per-round computational cost of their implementations. However, one needs at least $\Omega(d^2)$ time at each round to compute the matrices from the definition of AdaGrad, where $d \in \mathbb{N}$ is the dimension of the problem. One solution, independently proposed by [31] and by [49], is to use a version of AdaGrad which maintains, at each round $t \in \mathbb{N} \setminus \{0\}$, *diagonal matrices* $\tilde{G}_t \in \mathbb{S}^d_{++}$ instead of full matrices $G_t \in \mathbb{S}^d$ as in the original definition of AdaGrad. We formally define a player oracle which implements this version of the AdaGrad algorithm on Algorithm 6.4.

Algorithm 6.4 Definition of $\mathrm{DiagAdaGrad}^X(\langle f_1, \dots, f_T \rangle)$

Input:

(i) A nonempty closed convex set $X \subseteq \mathbb{R}^d$,

(ii) Convex functions $f_1, \dots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{R}^d}$ such that f_t is subdifferentiable on X for each $t \in [T]$,

(iii) Real numbers $\varepsilon, \eta > 0$ (usually clear from the context).

Output: $x_{T+1} \in X$

```
Define $\tilde{G}_0 \coloneqq \varepsilon I$
$\{x_1\} \leftarrow \arg\min_{x \in X} \|x\|_2$
for $t = 1$ to $T$ do
        ▷ Computations for round $t + 1$
    Let $g_t \in \partial f_t(x_t)$
    $\tilde{G}_t \leftarrow \tilde{G}_{t-1} + \mathrm{Diag}(g_t g_t^\mathsf{T})$
    $x_{t+1} \leftarrow \Pi_X^{\tilde{G}_t^{1/2}}(x_t - \eta \tilde{G}_t^{-1/2} g_t)$
return $x_{T+1}$
```

By using only diagonal matrices, one may note that we can implement DiagAdaGrad with a cost of $O(d)$ per round (ignoring the cost of the projection). We need now to investigate if there are good regret guarantees for this algorithm. Hopefully, we may derive regret guarantees as good as the one for the AdaGrad algorithm from the previous section. At this point, writing this algorithm as a version of the AdaReg algorithm becomes very useful and informative. We will show that using the same meta-regularizer as the one used for AdaGrad restricted to diagonal matrices yields the

[12] We have the value of $\varepsilon > 0$ cluttering our intuition. Still, one can get rid of the term with ε by using pseudo-inverses. For details, see [36]

Diagonal Adaptive Gradient algorithm. Not only that, this allows us to use many of the results from the previous section as stepping stones to derive the results for this section.

Throughout the remainder of this section, we denote by $\mathbb{D}^d \coloneqq \{\operatorname{Diag}(x) \in \mathbb{S}^d : x \in \mathbb{R}^d\}$ the set of $d \times d$ diagonal matrices. Moreover, we overload the Diag operator and for every $A \in \mathbb{S}^d$ define $\operatorname{Diag}(A) \coloneqq \operatorname{Diag}(\operatorname{diag}(A))$. That is, $\operatorname{Diag}(A)$ is equal to the matrix A but with zeroes on its off-diagonal entries. Let us show the form and some properties of the meta-regularizer we will use to derive the DiagAdaGrad algorithm, proving first a simple lemma about the normal cone and the relative interior of $\mathbb{D}^d$.

Lemma 6.4.1. We have $\operatorname{ri}(\mathbb{D}^d) = \mathbb{D}^d$ and

$$N_{\mathbb{D}^d}(\tilde{A}) = \{\, A \in \mathbb{S}^d : \operatorname{diag}(A) = 0\}, \qquad \forall \tilde{A} \in \mathbb{D}.$$

Proof. First, note that for any $\tilde{A}, \tilde{B} \in \mathbb{D}^d$, we have $(1-\mu)\tilde{A} + \mu\tilde{B} \in \mathbb{D}$ for any $\mu \in \mathbb{R}$. Thus, $\mathbb{D}^d$ is an affine set and $\operatorname{ri}(\mathbb{D}^d) = \mathbb{D}^d$.

Let $\tilde{A} \in \mathbb{D}^d$, let $A \in \mathbb{S}^d$, and define $\bar{a} \coloneqq \operatorname{diag}(\tilde{A})$. Note that $\langle A, X - \tilde{A}\rangle \leq 0$ for any $X \in \mathbb{D}^d$ if and only if $\langle A, \operatorname{Diag}(x - \bar{a})\rangle \leq 0$ for any $x \in \mathbb{R}^d$. Moreover, $\langle A, \operatorname{Diag}(x - \bar{a})\rangle \leq 0$ for every $x \in \mathbb{R}^d$ if and only if $\operatorname{diag}(A)^{\mathsf{T}}(x - \bar{a}) \leq 0$ for every $x \in \mathbb{R}^d$. That is, $A \in N_{\mathbb{D}^d}(\tilde{A})$ if and only if $\operatorname{diag}(A) \in N_{\mathbb{R}^d}(\bar{a}) = \{0\}$. □

Lemma 6.4.2. Let $\eta > 0$, define $f \colon \mathbb{R}^d \to (-\infty, +\infty]$ by

$$f(x) \coloneqq \eta^2 \sum_{i=1}^{d} [x_i \neq 0] \frac{1}{x_i}, \qquad \forall x \in \mathbb{R}^d,$$

and set $\Phi \coloneqq f_{\mathbb{S}} + \delta(\cdot \,|\, \mathbb{D}^d)$. Then,

$$\Phi(H) = \eta^2 \operatorname{Tr}(H^{-1}) + \delta(H \,|\, \mathbb{D}^d), \qquad \forall H \in \mathbb{S}^d_{++}, \tag{6.14}$$

and for every $G \in \mathbb{S}^d_{++}$ the infimum $\inf_{H \in \mathbb{S}^d_{++}}(\langle G, H\rangle + \Phi(H))$ is attained by $\eta(\operatorname{Diag}(G))^{-1/2}$. In particular, for every $\varepsilon > 0$ we have $\mathrm{DiagAdaGrad}^X = \mathrm{AdaReg}^X_\Phi$ for every nonempty and closed set $X \subseteq \mathbb{R}^d$.

Proof. Note that (6.14) follows directly from Lemma 6.3.1. Let $G \in \mathbb{S}^d_{++}$. Let us show

$$\{\eta(\operatorname{Diag}(G))^{-1/2}\} = \operatorname*{arg\,min}_{H \in \mathbb{S}^d_{++}}(\langle G, H\rangle + \Phi(H)) \tag{6.15}$$

Since Φ is proper and infinite outside of $\mathbb{D}^d$, it is clear that the infimum can only be attained by a matrix in $\mathbb{D}^d \cap \mathbb{S}^d_{++}$. Let $\bar{H} \in \mathbb{D}^d \cap \mathbb{S}^d_{++}$ and define $\Phi_{\mathrm{AdaGrad}}(H) \coloneqq \eta^2 \operatorname{Tr}(H^{-1})$ for every $H \in \mathbb{S}^d_{++}$. Note that $\Phi = \Phi_{\mathrm{AdaGrad}} + \delta(\cdot \,|\, \mathbb{D}^d)$ in this case. By Lemma 6.4.1 we have $\operatorname{ri}(\mathbb{D}^d) = \mathbb{D}^d$. Thus, we have $(\operatorname{ri} \mathbb{S}^d_{++}) \cap (\operatorname{ri}(\mathbb{D}^d)) = \mathbb{S}^d_{++} \cap \mathbb{D}^d \neq \varnothing$. Thus, formula for the subdifferential of the sum of convex functions from Theorem 3.5.4 together with the differentiability of Φ_{AdaGrad} from Lemma 6.3.1,

$$\partial\Phi(\bar{H}) = \nabla\Phi_{\mathrm{AdaGrad}}(\bar{H}) + N_{\mathbb{D}^d}(\bar{H}) = -\eta^2 \bar{H}^{-2} + N_{\mathbb{D}^d}(\bar{H}).$$

Thus, by the optimality conditions from Theorem 3.6.2, $\bar{H}$ attains the infimum in (6.15) if and only if

$$-(G_T + \partial\Phi(\bar{H})) \cap N_{\mathbb{S}^d_{++}}(\bar{H}) \neq \varnothing \iff 0 \in (G_T + \partial\Phi(\bar{H})) = G_T - \eta^2\bar{H}^{-2} + N_{\mathbb{D}^d}(\bar{H}).$$

The above holds if and only if there is $A \in N_{\mathbb{D}^d}(\bar{H})$ such that $\eta^2 \bar{H}^{-2} = G + A$. Since $\bar{H} \in \mathbb{D}^d$, we have $\eta^2 \bar{H}^{-2} \in \mathbb{D}^d$ and, therefore, $G + A \in \mathbb{D}^d$. Hence, we have $G + A = \operatorname{Diag}(G + A)$ and $\bar{H}$ attains the infimum in (6.15) if and only if

$$\bar{H}^{-2} = \frac{1}{\eta^2}(G + A) = \frac{1}{\eta^2}\operatorname{Diag}(G + A) = \frac{1}{\eta^2}\operatorname{Diag}(G),$$

where in the last equation we have used that $\operatorname{diag}(A) = 0$ by Lemma 6.4.1. Thus, $\bar{H}$ attains the infimum from (6.15) if and only if $\bar{H} = \frac{1}{\eta}\operatorname{Diag}(G)^{-1/2}$, which proves (6.15).

Now let us show that Φ is a meta-regularizer. Let $T \in \mathbb{N} \setminus \{0\}$ and $\boldsymbol{g} \in (\mathbb{R}^d)^T$. Moreover, let $\varepsilon > 0$ and set $G_{T-1} \coloneqq \varepsilon I + \sum_{t=1}^{T-1} g_t g_t^{\mathsf{T}}$ and $G_T \coloneqq G_{T-1} + g_T g_T^{\mathsf{T}}$. Condition (6.3.i) is satisfied by Φ since by (6.12) we know that $\inf_{H \in \mathbb{S}^d_{++}}(\langle H, G_T\rangle + \Phi(H))$ is attained by $\eta \operatorname{Diag}(G_T)^{-1/2}$. Set $H_{T+1} \coloneqq \eta \operatorname{Diag}(G_T)^{-1/2}$ and $H_T \coloneqq \eta \operatorname{Diag}(G_{T-1})^{-1/2}$. By definition,

$$H_{T+1}^{-1} - H_T^{-1} = \tfrac{1}{\eta}(\operatorname{Diag}(G_T)^{1/2} - \operatorname{Diag}(G_{T-1})^{1/2}).$$

Since G_T and G_{T-1} are positive semidefinite, we have that $\operatorname{diag}(G_T)$ and $\operatorname{diag}(G_{T-1})$ are non-negative. Moreover, since $G_T - G_{T-1} = g_T g_T^{\mathsf{T}} \succeq 0$, for every $i \in [d]$ we

$$(G_T)_{i,i} = e_i^{\mathsf{T}} G_T e_i \geq e_i^{\mathsf{T}} G_{T-1} e_i = (G_{T-1})_{i,i} \implies (G_T)_{i,i} \geq (G_{T-1})_{i,i} \implies (G_T)_{i,i}^{1/2} \geq (G_{T-1})_{i,i}^{1/2}.$$

Since $\eta > 0$, we conclude that $\frac{1}{\eta}\operatorname{diag}(G_T)^{1/2} \geq \frac{1}{\eta}\operatorname{diag}(G_{T-1})^{1/2}$, which proves that $\frac{1}{\eta}(\operatorname{Diag}(G_T)^{1/2} - \operatorname{Diag}(G_{T-1})^{1/2})$ is positive semidefinite, that is, Φ satisfies condition (6.3.ii). This finishes the proof that Φ is a meta-regularizer.

Last but not least, let us show that $\text{DiagAdaGrad}^X = \text{AdaReg}^X_\Phi$ for any $\varepsilon > 0$ and any closed and convex set $\varnothing \neq X \subseteq \mathbb{R}^d$ (recall that $\eta > 0$ is already given by the statement). Let $X \subseteq \mathbb{R}^d$ be a closed convex and nonempty set, let $\varepsilon > 0$, and let $\boldsymbol{f} \coloneqq \langle f_1, \ldots, f_T\rangle \in \operatorname{Seq}((-\infty, +\infty]^{\mathbb{R}^d})$ be such that the function f_t is subdifferentiable on X for every $t \in [T]$. Let us show that $\text{DiagAdaGrad}^X(\boldsymbol{f}_{1:t-1}) = \text{AdaReg}^X_\Phi(\boldsymbol{f}_{1:t-1})$ by induction on $t \in [T]$. Set $x_1 \coloneqq \text{AdaReg}^X_\Phi(\langle\rangle)$ and let H_1 be as in the definition of $\text{AdaReg}^X_\Phi(\langle\rangle)$. By (6.15), we know that $H_1 = (\eta/\sqrt{\varepsilon})I$. Thus,

$$x_1 \in \operatorname*{arg\,min}_{x \in X} \|x\|_{H_1^{-1}} = \operatorname*{arg\,min}_{x \in X} x^{\mathsf{T}} H_1^{-1} x = \operatorname*{arg\,min}_{x \in X} \frac{\sqrt{\varepsilon}}{\eta} x^{\mathsf{T}} x = \operatorname*{arg\,min}_{x \in X} \|x\|_2.$$

Since $\|\cdot\|_2^2$ is strictly convex (see Lemma 3.9.5), the above minimizer is unique and, thus, we have $x_1 = \text{DiagAdaGrad}^X(\langle\rangle)$. Let $t \in \{2, \ldots, T+1\}$, and let $g_{t-1} \in \mathbb{R}^d$ and $G_{t-1} \in \mathbb{S}^d_{++}$ be as in the definition of $x_t \coloneqq \text{AdaReg}^X_\Phi(\boldsymbol{f}_{1:t-1})$. One may note that $g_1, \ldots, g_{t-1} \in \mathbb{R}^d$ as in the definition of $\text{DiagAdaGrad}^X_\Phi(\boldsymbol{f}_{1:t-1})$ matches $g_1, \ldots, g_{t-1}$ as in the definition of $\text{AdaReg}^X_\Phi(\boldsymbol{f}_{1:t-1})$ with a proper choice of well-order on the subdifferentials used by the oracles. In this case, by defining $\tilde{G}_{t-1} \coloneqq \varepsilon I + \sum_{i=1}^{t-1} g_i g_i^{\mathsf{T}}$ as in the definition of $\text{DiagAdaGrad}(\boldsymbol{f}_{1:t-1})$, we have $\tilde{G}_{t-1} = \operatorname{Diag}(G_{t-1})$. Finally, let H_t be as in the definition of $\text{AdaReg}^X_\Phi(\boldsymbol{f}_{1:t-1})$, set $x_{t-1} \coloneqq \text{AdaReg}^X_\Phi(\boldsymbol{f}_{1:t-2}) = \text{AdaGrad}^X(\boldsymbol{f}_{1:t-2})$ (where the equation holds by induction), and define $\tilde{G} \coloneqq \operatorname{Diag}(G_{t-1}) = \tilde{G}_{t-1}$. Then,

$$x_t = \Pi_X^{H_t^{-1}}(x_{t-1} - H_t g_{t-1}) \overset{(6.12)}{=} \Pi_X^{\tilde{G}^{-1/2}}(x_{t-1} - \eta \tilde{G}^{1/2} g_{t-1}) = \text{AdaGrad}^X(\boldsymbol{f}_{1:t-1}). \qquad \square$$

Finally, we are in place to prove a regret bound for the Diagonal AdaGrad algorithm.

Theorem 6.4.3. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that X is a nonempty closed set and such that each $f \in \mathcal{F}$ is a proper closed functions which is subdifferentiable on X. Let $\varepsilon > 0$, let $T \in \mathbb{N}$, and let ENEMY be an enemy oracle for $\mathcal{C}$. Suppose there is $\theta \in \mathbb{R}_{++}$ such that $\theta \geq \sup\{\, \|u - x\|_\infty^2 : u, x \in X\}$ and set $\eta \coloneqq \sqrt{\theta/2}$ for DiagAdaGrad. Finally, define

$$(\boldsymbol{x}, \boldsymbol{f}) \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{DiagAdaGrad}^X, \mathrm{ENEMY}, T)$$

and let $\tilde{G}_T \in \mathbb{S}^d_{++}$ be as in the definition of $\mathrm{DiagAdaGrad}^X(\boldsymbol{f})$. Then

$$\mathrm{Regret}(\mathrm{AdaReg}^X_\Phi, \boldsymbol{f}, X) \leq \sqrt{2\theta}\,\mathrm{Tr}(\tilde{G}_T^{1/2}).$$

Proof. Define $f\colon \mathbb{R}^d \to (-\infty, +\infty]$ by $f(x) \coloneqq \eta^2 \sum_{i=1}^d [x_i \neq 0] \frac{1}{x_i}$ for each $x \in \mathbb{R}^d$ and set $\Phi \coloneqq f_{\mathbb{S}} + \delta(\cdot \,|\, \mathbb{D}^d)$. By Lemma 6.4.2, we have $\Phi(H) = \eta^2\,\mathrm{Tr}(H^{-1}) + \delta(H \,|\, \mathbb{D}^d)$ for every $H \in \mathbb{S}^d_{++}$ and $\mathrm{AdaReg}^X_\Phi = \mathrm{AdaGrad}^X$ if a proper well-order is equipped to the sets used by AdaReg^X_Φ. In this case, we only to bound the regret of AdaReg^X_Φ. For each $t \in \{1, \dots, T+1\}$, let $H_t, G_{t-1} \in \mathbb{S}^d_{++}$ be as in the definition of $\mathrm{AdaReg}^X_\Phi(\boldsymbol{f})$, set $\tilde{G}_{t-1} \coloneqq \mathrm{Diag}(G_{t-1})$ (which matches the definition of $\tilde{G}_{t-1}$ on $\mathrm{DiagAdaGrad}^X$), define $D_t \coloneqq H_t^{-1} - [t > 1]H_{t-1}^{-1}$, and let $u \in \mathbb{R}^d$. Thus, by Theorem 6.2.4 with $x_0 \coloneqq x_1$ we have

$$\mathrm{Regret}(\mathrm{AdaReg}^X_\Phi, \boldsymbol{f}, u) \leq \frac{1}{2}\sum_{t=0}^T \|u - x_t\|^2_{D_{t+1}} + \frac{1}{2}\min_{H \in \mathbb{S}^d_{++}} (\langle G_T, H\rangle + \Phi(H) - \Phi(H_1)). \tag{6.16}$$

Let us bound each of the above terms separately. First, let us show that

$$\sum_{t=0}^T \|u - x_t\|^2_{D_{t+1}} \leq \sqrt{2\theta}\,\mathrm{Tr}(\tilde{G}_T^{1/2}). \tag{6.17}$$

By the definition of the matrices $H_1, \dots, H_{T+1}$ and by Lemma 6.4.2 we have $H_t = \eta\tilde{G}_{t-1}^{-1/2}$ for every $t \in \{1, \dots, T+1\}$. Thus, H_t and D_t are diagonal matrices with positive diagonal entries (the latter holds since $G_{t-1} \succ 0$). Thus, for every $t \in \{1, \dots, T+1\}$ and any $v \in \mathbb{R}^d$,

$$v^\mathsf{T} D_{t+1} v = \sum_{i=1}^d v_i^2 (D_{t+1})_{i,i} \leq \|v\|_\infty^2 \sum_{i=1}^d (D_{t+1})_{i,i} = \|v\|_\infty^2\,\mathrm{Tr}(D_{t+1}). \tag{6.18}$$

Therefore, using, among other facts, that $H_{T+1} = \eta\tilde{G}_T^{-11/2}$ by Lemma 6.4.2, we have

$$\begin{aligned}
\sum_{t=0}^T \|u - x_t\|^2_{D_{t+1}} &= \sum_{t=0}^T (u - x_t)^\mathsf{T} D_{t+1}(u - x_T) \\
&\overset{(6.18)}{\leq} \sum_{t=0}^T \|u - x_t\|_\infty^2\,\mathrm{Tr}(D_{t+1}) \\
&\leq \theta \sum_{t=0}^T \mathrm{Tr}(D_{t+1}) = \theta\,\mathrm{Tr}\Big(\sum_{t=0}^T D_{t+1}\Big) \\
&= \theta\,\mathrm{Tr}(H_{T+1}^{-1}) \overset{\text{Le. } 6.4.2}{=} \frac{\theta}{\eta}\,\mathrm{Tr}(\tilde{G}_T^{1/2}) \\
&= \sqrt{2\theta}\,\mathrm{Tr}(\tilde{G}_T^{1/2}).
\end{aligned}$$

This proves (6.17). Let us now show

$$\min_{H \in \mathbb{S}^d_{++}} (\langle \tilde{G}_T, H\rangle + \Phi(H) - \Phi(H_1)) \leq \sqrt{2\theta}\,\mathrm{Tr}(\tilde{G}_T^{1/2}). \tag{6.19}$$

By Lemma 6.4.2, we have that the above minimum is attained by $\eta \tilde{G}_T^{-1/2}$ and $H_1 = (\eta/\sqrt{\varepsilon})I$ since, by definition, $H_1 \in \arg\min_{H \in \mathbb{S}^d_{++}}(\langle \tilde{G}_0, H \rangle + \Phi(H))$ and $\tilde{G}_0 = \varepsilon I$. Therefore,

$$\begin{aligned}
\min_{H \in \mathbb{S}^d_{++}} (\langle \tilde{G}_T, H \rangle + \Phi(H) - \Phi(H_1)) &= \eta \operatorname{Tr}(\tilde{G}_T^{1/2}) + \Phi(\eta \tilde{G}_T^{-1/2}) - \Phi(\tfrac{\eta}{\sqrt{\varepsilon}} I) \\
&= \eta \operatorname{Tr}(\tilde{G}_T^{1/2}) + \eta \operatorname{Tr}(\tilde{G}_T^{1/2}) - \eta\sqrt{\varepsilon} \operatorname{Tr}(I) \\
&\leq 2\eta \operatorname{Tr}(\tilde{G}_T^{1/2}) = \sqrt{2\theta} \operatorname{Tr}(\tilde{G}_T^{1/2}),
\end{aligned}$$

which proves (6.19). Plugging (6.17) and (6.19) into the regret bound from (6.16) completes the proof of the statement. □

Again, one may find it hard to find any meaning on the trace in the regret bound in the above theorem. Since the matrices used by DiagAdaGrad are diagonal, there is a simpler formula for the trace. Namely, let $\varepsilon > 0$, let $\boldsymbol{g} \in (\mathbb{R}^d)^T$ for some $T \in \mathbb{N}$, and define $\tilde{G}_T \coloneqq \varepsilon I + \sum_{t=1}^T \operatorname{diag}(g_t g_t^{\mathsf{T}})$. Then, one may verify that

$$\operatorname{Tr}(G_T^{1/2}) = \sum_{i=1}^d \sqrt{\varepsilon + \sum_{t=1}^T g_t(i)^2}.$$

Still, the above formula may not be very informative. Let us prove a proposition similar to Proposition 6.4.4 which sheds some light into the meaning of the above trace.

Proposition 6.4.4. Let $A \in \mathbb{S}^d_{++} \cap \mathbb{D}^d$. Then $\inf\{\operatorname{Tr}(X^{-1}A) : X \in \mathbb{S}^d_{++}, \operatorname{Tr}(X) = 1\}$ is attained by $\operatorname{Tr}(A^{1/2})^{-1}A^{1/2}$.

Proof. Define $a \coloneqq \operatorname{diag}(A)$. Since $A \in \mathbb{D}^d$, we have $A = \operatorname{Diag}(a)$. Additionally, note that

$$\begin{aligned}
\inf\{\langle X^{-1}, A \rangle : X \in \mathbb{S}^d_{++} \cap \mathbb{D}^d, \operatorname{Tr}(X) = 1\} &= \inf\{\langle \operatorname{Diag}(x)^{-1}, \operatorname{Diag}(a) \rangle : x \in \mathbb{R}^d_{++} \cap \Delta_d\} \\
&= \inf\left\{ \sum_{i=1}^d \frac{a_i}{x_i} : x \in \mathbb{R}^d_{++} \cap \Delta_d \right\}.
\end{aligned}$$

Not only that, we also have that $\bar{x} \in \mathbb{R}^d_{++} \cap \Delta_d$ attains the last infimum above if and only if $X \coloneqq \operatorname{Diag}(x)^{-1}$ attains the first infimum above. Define $\bar{x} \in \Delta_d$ by

$$\bar{x}_i \coloneqq \frac{a_i^{1/2}}{\sum_{i=1}^d a_i^{1/2}}, \qquad \forall i \in [d].$$

Note that

$$\operatorname{Diag}(\bar{x}) = \frac{1}{\sum_{i=1}^d a_i^{1/2}} \operatorname{Diag}(a)^{1/2} = \frac{1}{\operatorname{Tr}(\operatorname{Diag}(a)^{1/2})} \operatorname{Diag}(a)^{1/2} = \frac{1}{\operatorname{Tr}(A^{1/2})} A^{1/2}.$$

Thus, to prove the statement, it suffices to show that

$$\bar{x} \in \arg\min\left\{ \sum_{i=1}^d \frac{a_i}{x_i} : x \in \mathbb{R}^d_{++} \cap \Delta_d \right\}. \tag{6.20}$$

Define the convex function $c \colon \mathbb{R}^d \to (-\infty, +\infty]$ by

$$c(x) = \sum_{i=1}^d [x_i > 0] \frac{a_i}{x_i} + \delta(x \,|\, \mathbb{R}^d_{++}), \qquad \forall x \in \mathbb{R}^d.$$

First of all, note that c is closed. Indeed, c is continuous on $\mathbb{R}^d_{++}$ and, for every $\bar{x} \in \mathbb{R}^d_+ \setminus \mathbb{R}^d_{++}$,

$$\liminf_{x \to \bar{x}} c(x) = +\infty = c(\bar{x}).$$

Moreover, note that

$$(\nabla c(x))_i = -\frac{a_i}{x_i^2}, \qquad \forall i \in [d], \forall x \in \mathbb{R}^d_{++}.$$

Thus, for every $x \in \Delta_d$,

$$-\nabla c(\bar{x})^\mathsf{T}(x - \bar{x}) = -(\sum_{i=1}^d a_i^{1/2})^2 \mathbb{1}^\mathsf{T}(x - \bar{x}) = 0.$$

That is, $-\nabla c(x) \in N_{\Delta_d}$. By the optimality conditions for minima of convex functions (see Theorem 3.6.2), this implies that $\bar{x} \in \arg\min_{x \in \Delta_d} c(x)$, which is equivalent to (6.20). □

Corollary 6.4.5. Let $\varepsilon > 0$ and $\boldsymbol{g} \in (\mathbb{R}^d)^T$ for some $T \in \mathbb{N}$. Moreover, set $\tilde{G}_T := \varepsilon I + \sum_{t=1}^T \operatorname{diag}(g_t g_t^\mathsf{T})$ and $\mathcal{S}_d := \{\, X \in \mathbb{S}^n_{++} : \operatorname{Tr}(X) = 1 \}$. Then

$$\operatorname{Tr}(\tilde{G}_T^{1/2}) = \sqrt{\min_{H \in \mathcal{S}_d \cap \mathbb{D}^d} \left(\varepsilon \operatorname{Tr}(H^{-1}) + \sum_{t=1}^T \|g_t\|^2_{H^{-1}} \right)}.$$

Proof. Set $\mathcal{H} := \mathcal{S}_d \cap \mathbb{D}^d$. Note that if $g \in \mathbb{R}^d$ and $H \in \mathbb{D}^d$, then $\|g\|_H^2 = \operatorname{Tr}(\operatorname{Diag}(g) H \operatorname{Diag}(g))$. Using this fact and Proposition 6.4.4, we have

$$\begin{aligned}
\sqrt{\min_{H \in \mathcal{H}} \left(\varepsilon \operatorname{Tr}(H^{-1}) + \sum_{t=1}^T \|g_t\|^2_{H^{-1}} \right)} &= \sqrt{\min_{H \in \mathcal{H}} \left(\varepsilon \langle H^{-1}, I \rangle + \sum_{t=1}^T \operatorname{Tr}(\operatorname{Diag}(g_t) H^{-1} \operatorname{Diag}(g_t)) \right)} \\
&= \sqrt{\min_{H \in \mathcal{H}} \left(\varepsilon \langle H^{-1}, I \rangle + \sum_{t=1}^T \langle H^{-1}, \operatorname{Diag}(g_t g_t^\mathsf{T}) \rangle \right)} \\
&= \sqrt{\min_{H \in \mathcal{H}} \langle H^{-1}, \tilde{G}_T \rangle} = \sqrt{\operatorname{Tr}^2(\tilde{G}_T^{1/2})} \\
&= \operatorname{Tr}(\tilde{G}_T^{1/2}).
\end{aligned}$$

□

With the above corollary, we can compare the regret bound for the Diagonal AdaGrad from Theorem 6.4.3 with the regret bounds for the classic AdaGrad algorithm (Theorem 6.3.2) and with the regret for the Online Mirror Descent algorithm with adaptive step size (Theorem 6.1.1). As expected, the regret bound for AdaGrad seems to be better than the one for its diagonal version. We can see this by comparing Corollaries 6.3.4 and 6.4.5, which show more palatable ways of writing the traces that appear on the bounds of both algorithms. On Corollary 6.3.4, the minimum is taken over all positive definite matrices in the spectraplex, while in the minima in above corollary the search space is restricted to diagonal matrices. Still, the regret bound for the Diagonal AdaGrad seems to be as good as the one for the OMD algorithm with adaptive step sizes from Theorem 6.1.1. To see this, recall from (6.13) that we know the (scaled) ℓ_2-norm can be written as a norm induced by $d^{-1}I$, where $d \in \mathbb{N} \setminus \{0\}$ is the dimension of the problem. Thus, for a value of $\varepsilon > 0$ small enough in the above corollary, we conclude that the norm chosen by the above minimum is as good as the ℓ_2-norm if the goal is to minimize the sum of the norms of the subgradients. However, one problem appears when trying to compare the regret bound for DiagAdaGrad with the bounds on previous sections: the diameter $\theta \in \mathbb{R}_{++}$ in Theorem 6.4.3 is w.r.t. the ℓ_∞-norm, while in previous sections the ℓ_2-norm was used. Thus, more informative comparisons of the these regret bounds need more information on the set $X \subseteq \mathbb{R}^d$ from where the player picks her points.

6.5 The Online Newton Step Algorithm

Let us look now at an OCO player oracle which attains logarithmic regret with respect to the number of rounds in problems with some nice properties for the player. As we have discussed on Chapter 4, if we devise a player oracle to play games where the functions played by the enemy only need to be closed, convex, bounded, and Lipschitz continuous, then its worst-case regret is no better than $\Omega(\sqrt{T})$, where $T \in \mathbb{N}$ is the number of rounds (see [2] for details). Thus, we need some additional assumptions on the functions played by the enemy if we want to devise a player oracle which attains logarithmic regret. As we have seen in Section 4.8, if the functions played by the enemy are all strongly convex, for example, we may attain logarithmic regret by simply using the FTL algorithm (that is, the FTRL algorithm with no regularizer). At this point, a natural question is: are there other assumptions on the functions played by the enemy (hopefully weaker than strong convexity) which make it possible for a player oracle to attain logarithmic regret w.r.t. the number of rounds? As we are going to see in this section, if the functions picked by the enemy are guaranteed to be *exp-concave*, an assumption similar but slightly weaker than strong convexity, we may devise player oracles which attain logarithmic worst-case regret.

Definition 6.5.1 (α-exp-concave functions)**.** Let $\alpha \in \mathbb{R}_{++}$. A function $f \colon \mathbb{R}^d \to (-\infty, +\infty]$ is α-**exp-concave** on a set $X \subseteq \mathbb{R}^d$ if the function[13]

$$x \in \mathbb{R}^d \mapsto e^{-\alpha f(x)} - \delta(x \,|\, X)$$

is concave. If the set X is not explicitly stated, assume $X = \mathbb{R}^d$.

The definition of exp-concavity is hardly interpretable by itself. Thus, both for the sake of our understanding of this property and as tools for later use, let us prove some properties about exp-concave functions. The first lemma we will prove shows a characterization of exp-concave function which can be better interpreted.

To better understand the idea behind the next lemma, recall that by Lemma 3.9.5 a function two-times continuously differentiable function[14] $f \colon \mathbb{R}^d \to \mathbb{R}$ is α-strongly convex on $\mathbb{R}^d$ w.r.t. the ℓ_2-norm if and only if $f - \frac{\alpha}{2}\|\cdot\|_2^2$ is convex. By Lemma 3.1.1, the latter function is convex if and only if $\nabla^2 f(x) - \alpha I \succ 0$ for each $x \in \mathbb{R}^d$, that is, $\nabla^2 f(x) \succ \alpha I$ for every $x \in \mathbb{R}^d$. Intuitively, this mean that the at every $x \in \mathbb{R}^d$ function is curved in all directions. The next lemma shows a similar characterization of exp-concavity for two-times continuously differentiable functions. Namely, it shows that a two-times continuously differentiable function $f \colon \mathbb{R}^d \to \mathbb{R}$ is α-exp-concave if and only if its hessian is positive definite "in the direction of its gradient". In some sense, this means that f is α-strongly convex w.r.t. the ℓ_2-norm (or curved) on the direction of its gradient.

Lemma 6.5.2. Let $f \colon \mathbb{R}^d \to (-\infty, +\infty]$ and let $X \subseteq \operatorname{dom} f$ be a nonempty convex set such that $f + \delta(\cdot \,|\, X)$ is two-times continuously differentiable on X. Moreover, let $\alpha > 0$. Then f is α-exp-concave on X if and only if

$$\nabla^2 f(x) \succeq \alpha \nabla f(x) \nabla f(x)^{\mathsf{T}}, \qquad \forall x \in X.$$

Proof. Define $h \coloneqq e^{-\alpha f(\cdot)} - \delta(\cdot \,|\, X)$. Then, for any $x \in X$ we have

$$\nabla h(x) = -\alpha e^{-\alpha f(x)} \nabla f(x) \qquad \text{and} \qquad \nabla^2 h(x) = \alpha^2 e^{-\alpha f(x)} \nabla f(x) \nabla f(x)^{\mathsf{T}} - \alpha e^{-\alpha f(x)} \nabla^2 f(x).$$

[13] One may worry that we are using a function which is valued $-\infty$ outside of X. Recall, however, that a function f is concave if and only if $-f$ is convex. Thus, the natural way to indicate points outside of the domain of a concave function is to attribute $-\infty$ to them.

[14] We restrict our discussion to functions which are finite everywhere only to avoid technicalities while we build intuition.

By Lemma 3.1.1, h is concave if and only if $0 \succeq \nabla^2 h(x)$ for every $x \in \operatorname{dom} h = X$. The latter holds if and only if, for every $x \in X$,

$$\alpha^2 e^{-\alpha f(x)} \nabla f(x) \nabla f(x)^{\mathsf{T}} \preceq \alpha e^{-\alpha f(x)} \nabla^2 f(x) \iff \alpha \nabla f(x) \nabla f(x)^{\mathsf{T}} \preceq \nabla^2 f(x). \qquad \square$$

Let us now look at some examples of exp-concave functions for the sake of concreteness. First, let us show that the functions from the sequential investment problem defined on Section 2.2.4 are all exp-concave.

Proposition 6.5.3. Let $r \in \mathbb{R}^d_{++}$, define the convex set $X := \{\, x \in \mathbb{R}^d : \mathbb{1}^{\mathsf{T}} x > 0 \}$, and define the function $f(x) := -\ln(r^{\mathsf{T}} x) + \delta(x \,|\, X)$ for every $x \in \mathbb{R}^d$. Then f is 1-exp-concave on X.

Proof. Since $x \in \mathbb{R}^d \mapsto r^{\mathsf{T}} x$ is two-times continuously differentiable on $\mathbb{R}^d$, since $\alpha \in \mathbb{R}_{++} \mapsto \ln(\alpha)$ is two-times continuously differentiable on $\mathbb{R}_{++}$, and since $r^{\mathsf{T}} x > 0$ for every $x \in X$, we conclude that f is two-times continuously differentiable on X. Moreover, note that

$$\nabla f(x) = -\frac{1}{r^{\mathsf{T}} x} r, \qquad \text{and} \qquad \nabla^2 f(x) = \frac{1}{(r^{\mathsf{T}} x)^2} r r^{\mathsf{T}} \qquad \forall x \in X.$$

Therefore, for every $x \in X$,

$$\nabla^2 f(x) - \nabla f(x) \nabla f(x)^{\mathsf{T}} = \frac{1}{(r^{\mathsf{T}} x)^2} r r^{\mathsf{T}} - \frac{1}{(r^{\mathsf{T}} x)^2} r r^{\mathsf{T}} = 0.$$

Thus, by Lemma 6.5.2 we conclude that f is 1-exp-concave on X. □

The next theorem, which we will use later as a tool to prove regret bounds, shows an inequality for exp-concave functions which is similar to the inequality for strongly convex functions given by Theorem 3.9.7. Namely, the latter theorem states that if a closed convex function $f\colon \mathbb{R}^d \to (-\infty, +\infty]$ is α-strongly convex and f is subdifferentiable at $x \in X$, then

$$f(x) \geq f(y) + g^{\mathsf{T}}(x - y) + \frac{\alpha}{2}\|x - y\|_2^2, \qquad \forall y \in X, \forall g \in \partial f(x).$$

The inequality we prove in the next theorem for exp-concave functions is similar to the one above. The main difference is that instead of the squared norm, the inequality from the next theorem uses a “local norm”[15] based on the gradient of the function. Before jumping into the lemma, we need a simple result which we prove next.

Lemma 6.5.4. For every $\alpha \in [-1/4, 1/4]$, we have

$$-\ln(1 - \alpha) \geq \alpha + \frac{1}{4}\alpha^2.$$

Proof. For every $\alpha \in \mathbb{R}$ define

$$f(\alpha) := -\ln(1 - \alpha) \qquad \text{and} \qquad h(\alpha) := \alpha + \frac{1}{4}\alpha^2.$$

Since $f(0) = 0 = h(0)$ and since both f and h are differentiable on $[-1/4, 1/4]$, to prove $f(\alpha) \geq h(\alpha)$ for every $\alpha \in [-1/4, 1/4]$ it suffices to prove $f'(\alpha) \geq h'(\alpha)$ for every $\alpha \in [0, 1/4]$ and $f'(\alpha) \leq h'(\alpha)$ for every $\alpha \in [-1/4, 0)$. Note that, for every $\alpha \in [-1/4, 1/4]$, since $1 - \alpha > 0$ we have

$$\begin{aligned} f'(\alpha) \geq h'(\alpha) &\iff \frac{1}{1 - \alpha} \geq 1 + \frac{1}{2}\alpha \iff 2 \geq (2 + \alpha)(1 - \alpha) \\ &\iff 2 \geq 2 - \alpha - \alpha^2 \iff \alpha^2 + \alpha \geq 0. \end{aligned}$$

Since $\alpha^2 + \alpha \geq 0$ for every $\alpha \in [0, 1/4]$ and $\alpha^2 + \alpha \leq 0$ for every $\alpha \in [-1/4, 0)$, we are done. □

[15] Note that it is not a norm since the matrix in the inequality is a rank-one matrix.

Theorem 6.5.5. Let $X \subseteq \mathbb{R}^d$, let $\|\cdot\|$ be a norm on $\mathbb{R}^d$, and let $\alpha \in \mathbb{R}_{++}$. Let $f \colon \mathbb{R}^d \to (-\infty, +\infty]$ be a closed convex function which is α-exp-concave on X, ρ-Lipschitz continuous w.r.t. $\|\cdot\|$ on X, and differentiable on X. Moreover, suppose there is $\theta \in \mathbb{R}_{++}$ such that $\sup_{x,y \in X} \|x - y\|^2 \leq \theta$. Finally, let $\beta \in \mathbb{R}_{++}$ be such that $\beta \leq \frac{1}{2} \min\{(4\rho\sqrt{\theta})^{-1}, \alpha\}$. Then, for any $x, y \in X$ and $g \in \partial f(x)$ we have

$$f(y) \geq f(x) + \nabla f(x)^{\mathsf{T}}(y - x) + \frac{\beta}{2}(x - y)\nabla f(x)\nabla f(x)^{\mathsf{T}}(x - y).$$

Proof. Since $2\beta \leq \alpha$, we have that f is 2β-exp-concave on X. Thus, $h \coloneqq e^{-2\beta f(\cdot)} + \delta(\cdot \,|\, X)$ is a concave function. Let $x, y \in X$. Then, by the subgradient inequality,

$$\begin{aligned} h(y) \leq h(x) + \nabla h(x)^{\mathsf{T}}(y - x) &\implies e^{-2\beta f(y)} \leq e^{-2\beta f(x)} - 2\beta e^{-2\beta f(x)} \nabla f(x)^{\mathsf{T}}(y - x) \\ &\implies e^{-2\beta f(y)} \leq e^{-2\beta f(x)}(1 - 2\beta \nabla f(x)^{\mathsf{T}}(y - x)) \\ &\implies -2\beta f(y) \leq -2\beta f(x) + \ln(1 - 2\beta \nabla f(x)^{\mathsf{T}}(y - x)) \\ &\implies f(y) \geq f(x) - \frac{1}{2\beta} \ln(1 - 2\beta \nabla f(x)^{\mathsf{T}}(y - x)). \end{aligned} \tag{6.21}$$

By Theorems 3.5.5 and 3.8.4, we have $\|\nabla f(x)\|_* \leq \rho$. Hence,

$$2\beta \nabla f(x)^{\mathsf{T}}(y - x) \leq 2\beta \|\nabla f(x)\|_* \|x - y\| \leq 2\beta\rho\sqrt{\theta} \leq \frac{1}{4}.$$

Thus, we can use Lemma 6.5.4 on (6.21), which yields

$$\begin{aligned} f(y) &\geq f(x) - \frac{1}{2\beta} \ln(1 - 2\beta \nabla f(x)^{\mathsf{T}}(y - x)) \\ &\geq f(x) + \frac{1}{2\beta}\left(2\beta \nabla f(x)^{\mathsf{T}}(y - x) + \frac{1}{4}(2\beta \nabla f(x)^{\mathsf{T}} y - x)^2\right) \\ &= f(x) + \nabla f(x)^{\mathsf{T}}(y - x) + \frac{\beta}{2}(\nabla f(x)^{\mathsf{T}} y - x)^2. \end{aligned}$$

□

Finally, let us describe the *Online Newton Step (ONS) algorithm*, which was first presented in [37]. We will show that, if the functions played by the enemy are guaranteed to be differentiable and exp-concave on the set from where the player picks her choices, then the ONS algorithm's worst-case regret bound is logarithmic w.r.t. the number of rounds of the game. A player oracle which formally implements the ONS algorithm is defined in Algorithm 6.5.

One may have noticed some similarities between the above algorithm and the AdaGrad algorithm we have presented earlier in this chapter. The algorithm still maintains, at each round $t \in \mathbb{N} \setminus \{0\}$, a matrix constructed from rank-one updates based on the gradients of previous functions. On the other hand, how these matrices are used on the iterate updates are slightly different.

As before, let us see how to write ONS as an instance of the AdaReg algorithm. Again, in order to do so, we will pick a convex function on $\mathbb{R}^d$ and transform it into a function on symmetric matrices by applying the function only to the eigenvalues (see Section 3.7 for details). Surprisingly, the meta-regularizer we will use is a multiple of $X \in \mathbb{S}^d_{++} \mapsto -\ln \det X$, a barrier function deeply connected with interior-point methods [58]. It is very interesting to see this connection, since the ONS algorithm did not seem to be based on any of the main concepts from interior-point methods when first proposed on [37].

Lemma 6.5.6. Let $\eta > 0$, define $f \colon \mathbb{R}^d \to (-\infty, +\infty]$ by

$$f(x) \coloneqq -\eta \sum_{i=1}^{d} [x_i > 0] \ln x_i + \delta(x \,|\, \mathbb{R}^d_+), \qquad \forall x \in \mathbb{R}^d,$$

Algorithm 6.5 Definition of $\mathrm{ONS}^X(\langle f_1, \dots, f_T \rangle)$

```
Input:
    (i) A closed and convex set $\varnothing \neq X \subseteq \mathbb{R}^d$,
    (ii) Convex functions $f_1, \dots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{R}^d}$ such that $f_t$ is
         differentiable on $X$ for each $t \in [T]$,
    (iii) Real numbers $\eta, \varepsilon > 0$ (usually clear from the context).
Output: $x_{T+1} \in X$
  Define $G_0 \leftarrow \varepsilon I$
  Let $\{x_1\} \leftarrow \arg\min_{x \in X} \|x\|_2$
  for $t = 1$ to $T$ do
        ▷ Computations for round $t + 1$
     Define $G_t \leftarrow G_{t-1} + \nabla f_t(x_t) \nabla f_t(x_t)^\mathsf{T}$
     $x_{t+1} \leftarrow \Pi_X^{G_t}(x_t - \eta G_t^{-1} \nabla f_t(x_t))$
  return $x_{T+1}$
```

and set $\Phi := f_{\mathbb{S}}$. Then,

$$\Phi(H) = -\eta \ln \det(H) \qquad \text{and} \qquad \nabla \Phi(H) = -\eta H^{-1}, \qquad \forall H \in \mathbb{S}^d_{++}, \tag{6.22}$$

and for every $G \in \mathbb{S}^d_{++}$ the infimum $\inf_{H \in \mathbb{S}^d_{++}}(\langle G, H \rangle + \Phi(H))$ is attained by ηG^{-1}. Moreover, the function Φ is a meta-regularizer and we have $\mathrm{ONS}^X = \mathrm{AdaReg}^X_\Phi$ for every nonempty closed convex set $X \subseteq \mathbb{R}^d$ and for every $\varepsilon > 0$, where the value of η in ONS is the same as in the definition of f.

Proof. Let $H \in \mathbb{S}^d_{++}$ and set $\lambda := \lambda^{\uparrow}(H)$. First, let us show that (6.22) holds for H. By the definition of $f_{\mathbb{S}}$ we have

$$f_{\mathbb{S}}(H) = f(\lambda) = -\eta \sum_{i=1}^d \ln \lambda_i = -\eta \ln \prod_{i=1}^d \lambda_i = -\eta \ln \det(H),$$

where in the last equation we used Corollary 1.1.2. Let us now check that Φ is differentiable at H. Define $\Lambda := \mathrm{Diag}(\lambda)$. By the Spectral Decomposition Theorem (Theorem 1.1.1), there is an orthogonal matrix $Q \in \mathbb{R}^{d \times d}$ such that $H = Q \Lambda Q^\mathsf{T}$. Since f is differentiable on $\mathbb{R}^d_{++}$, by Corollary 3.7.5 we have that Φ is differentiable on $\mathbb{S}^d_{++}$ and that

$$\nabla \Phi(H) = Q \,\mathrm{Diag}(\nabla f(\lambda)) Q^\mathsf{T} = -\eta Q \Lambda^{-1} Q^\mathsf{T} = -\eta H^{-1}.$$

This ends the proof of (6.22). Let $G \in \mathbb{S}^d_{++}$. Let us now show that

$$\{\eta G^{-1}\} = \arg\min_{H \in \mathbb{S}^d_{++}} (\langle H, G \rangle + \Phi(H)). \tag{6.23}$$

Let $\hat{H} \in \mathbb{S}^d_{++}$. By Theorem 3.6.2, $\hat{H}$ attains the above infimum if and only if

$$0 = G + \nabla \Phi(\hat{H}) = G - \eta \hat{H}^{-1} \iff \hat{H} = \eta G^{-1}.$$

This proves (6.23).

Let us now show that

$$\Phi \text{ is a meta-regularizer.} \tag{6.24}$$

Let $T \in \mathbb{N}$ and $\boldsymbol{g} \in (\mathbb{R}^d)^T$. Moreover, let $\varepsilon > 0$ and set $G_{T-1} \coloneqq \varepsilon I + \sum_{t=1}^{T-1} g_t g_t^{\mathsf{T}}$ and $G_T \coloneqq G_{T-1} + g_T g_T^{\mathsf{T}}$. Condition (6.3.i) is satisfied by Φ since, by (6.23), we know that $\inf_{H \in \mathbb{S}^d_{++}}(H \bullet G_T + \Phi(H))$ is attained by ηG_T^{-1}. Thus, set $H_{T+1} \coloneqq \eta G_T^{-1}$ and $H_T \coloneqq \eta G_{T-1}^{-1}$. Note that

$$H_{T+1}^{-1} - H_T^{-1} = \tfrac{1}{\eta}(G_T - G_{T-1}) = \tfrac{1}{\eta} g_T g_T^{\mathsf{T}} \succeq 0,$$

that is, Φ satisfies condition (6.3.ii), which finishes the proof of (6.24).

Last but not least, let us show that, $\mathrm{AdaGrad}^X = \mathrm{AdaReg}^X_\Phi$ for any $\varepsilon > 0$ and any closed and convex set $\varnothing \neq X \subseteq \mathbb{R}^d$ (recall that $\eta > 0$ is already given by the statement). Let $\boldsymbol{f} \coloneqq \langle f_1, \dots, f_T \rangle \in \mathrm{Seq}((-\infty, +\infty]^{\mathbb{R}^d})$ be such that f_t is closed, convex, and subdifferentiable on X for every $t \in [T]$. Let us show that $\mathrm{AdaGrad}^X(\boldsymbol{f}_{1:t-1}) = \mathrm{AdaReg}^X_\Phi(\boldsymbol{f}_{1:t-1})$ by induction on $t \in [T]$. Set $x_1 \coloneqq \mathrm{AdaReg}^X_\Phi(\langle\rangle)$ and let H_1 be as in the definition of $\mathrm{AdaReg}^X_\Phi(\langle\rangle)$. By (6.23), we know that $H_1 = (\eta/\varepsilon)I$. Thus,

$$x_1 \in \operatorname*{arg\,min}_{x \in X} \|x\|_{H_1^{-1}} = \operatorname*{arg\,min}_{x \in X} x^{\mathsf{T}} H_1^{-1} x = \operatorname*{arg\,min}_{x \in X} \tfrac{\varepsilon}{\eta} x^{\mathsf{T}} x = \operatorname*{arg\,min}_{x \in X} \|x\|_2.$$

Since the squared ℓ_2-norm is strictly convex (see Lemma 3.9.5), x_1 is the unique minimizer of the above minima and, thus, $x_1 = \mathrm{ONS}^X(\langle\rangle)$. Let $t \in \{2, \dots, T+1\}$ and define $x_i \coloneqq \mathrm{AdaReg}^X_\Phi(\boldsymbol{f}_{1:i-1}) = \mathrm{ONS}^X(\boldsymbol{f}_{1:i-1})$ for every $i \in \{1, \dots, t-1\}$ (where the equation holds by induction). Moreover, for every $i \in \{1, \dots, t-1\}$ let $g_i \in \mathbb{R}^d$ and $G_i \in \mathbb{S}^d_{++}$ be as in the definition of $x_t \coloneqq \mathrm{AdaReg}^X_\Phi(\boldsymbol{f}_{1:t-1})$. For every $i \in \{1, \dots, t-1\}$, since f_i is differentiable on X, by Theorem 3.5.5 we conclude that $g_{t-1} = \nabla f_{t-1}(x_{t-1})$. Thus, G_{t-1} is the same as the one in the definition of $\mathrm{ONS}^X(\boldsymbol{f}_{1:t-1})$. Finally, let H_t be as in the definition of $\mathrm{AdaReg}^X_\Phi(\boldsymbol{f}_{1:t-1})$. Then,

$$x_t = \Pi_X^{H_t^{-1}}(x_{t-1} - H_t g_{t-1}) \overset{(6.23)}{=} \Pi_X^{G_{t-1}}(x_{t-1} - \eta G_{t-1}^{-1} g_{t-1}) = \mathrm{ONS}^X(\boldsymbol{f}_{1:t-1}). \qquad \square$$

Finally, let us show that ONS attains logarithmic regret (w.r.t. the number of rounds) when playing against an enemy who plays only differentiable and exp-concave functions. Before proving the regret itself on Theorem 6.5.8, we need to prove a simple lemma to bound the eigenvalues of the matrices $G_t \in \mathbb{S}^n_{++}$ which the ONS oracle builds through its iterations.

Lemma 6.5.7. Let $T \in \mathbb{R}_+$ and $g_1, \dots, g_T \in \mathbb{R}^d$ be such that $\|g_t\|_2 \leq \rho$ for every $t \in [T]$. Moreover, let $\varepsilon > 0$ and set $G \coloneqq \varepsilon I + \sum_{t=1}^T g_t g_t^{\mathsf{T}}$. Then, for every $i \in [d]$,

$$\lambda_i^{\uparrow}(G) \leq \rho^2 T + \varepsilon \qquad \text{and} \qquad \det(G) \leq (\rho^2 T + \varepsilon)^d.$$

Proof. Let $i \in [d]$ and let $v \in \mathbb{R}^d$ be an eigenvector of G associated with $\lambda_i^{\uparrow}(G)$. Then,

$$\lambda_i^{\uparrow}(G) v = Gv = \varepsilon v + \sum_{t=1}^T g_t g_t^{\mathsf{T}} v.$$

Therefore,

$$\lambda_i^{\uparrow}(G)\|v\|_2^2 = \varepsilon\|v\|_2^2 + \sum_{t=1}^T (g_t^{\mathsf{T}} v)^2 \leq \varepsilon\|v\|_2^2 + \sum_{t=1}^T \|g_t\|_2^2 \|v\|_2^2 \leq \varepsilon\|v\|_2^2 + T\rho^2\|v\|_2^2.$$

Dividing the above inequality by $\|v\|_2^2$ (which is nonzero since v is an eigenvector) yields the first bound from the statement. The bound on the determinant follows directly from Corollary 1.1.2, which shows that $\det(G) = \prod_{i=1}^d \lambda_i^{\uparrow}(G)$. $\square$

Theorem 6.5.8. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that $X \subseteq \mathbb{R}^d$ is a nonempty closed set and such that each $f \in \mathcal{F}$ is a proper closed convex function. Moreover, suppose that there is a convex set $D \supseteq X$ with nonempty interior such that every $f \in \mathcal{F}$ is differentiable on $\operatorname{int}(D)$, α-exp-concave on D, and ρ-Lipschitz continuous on D. Suppose there is $\theta \in \mathbb{R}_{++}$ such that $\theta \geq \sup\{\, \|x - u\|_2^2 : x, u \in X\}$ is finite. Define

$$\beta \coloneqq \frac{1}{2}\min\left\{\alpha, \frac{1}{4\rho\sqrt{\theta}}\right\}, \qquad \eta \coloneqq \frac{1}{\beta}, \qquad \text{and} \qquad \varepsilon \coloneqq \frac{d}{\eta^2\theta}.$$

Finally, let $T \in \mathbb{N}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and define

$$(\boldsymbol{x}, \boldsymbol{f}) \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{ONS}^X, \mathrm{ENEMY}, T).$$

Then,

$$\mathrm{Regret}(\mathrm{ONS}^X, \boldsymbol{f}, X) \leq \left(\frac{1}{\alpha} + \rho\sqrt{\theta}\right) 4d(1 + d^{-1} + \ln T).$$

Proof. Define $h \colon \mathbb{R}^d \to (-\infty, +\infty]$ by

$$h(x) \coloneqq -\eta \sum_{i=1}^{d} [x_i > 0] \ln x_i, \qquad \forall x \in \mathbb{R}^d,$$

and set $\Phi \coloneqq h_{\mathbb{S}}$. By Lemma 6.5.6, we have $\Phi(H) = -\eta \ln\det(H)$ for every $H \in \mathbb{S}^d_{++}$ and $\mathrm{AdaReg}^X_\Phi = \mathrm{ONS}^X$. Thus, we only need to bound the regret of AdaReg^X_Φ. By Theorem 3.5.5, for every $t \in [T]$ we have $g_t = \nabla f_t(x_t)$, where $g_t \in \partial f_t(x_t)$ is as in in the definition of $\mathrm{AdaReg}^X_\Phi(\boldsymbol{f})$. For each $t \in [T]$ define $\tilde{f}_t \colon \mathbb{R}^d \to (-\infty, +\infty]$ by

$$\tilde{f}_t(x) \coloneqq \nabla f_t(x_t)^{\mathsf{T}} x, \qquad \forall x \in \mathbb{R}^d.$$

Let $u \in X$. Since f_t is α-exp-concave for each $t \in [T]$, by Theorem 6.5.5 we have

$$\begin{aligned}
\mathrm{Regret}(\mathrm{AdaReg}^X_\Phi, \boldsymbol{f}, u) &= \sum_{t=1}^{T}(f_t(x_t) - f_t(u)) \leq \sum_{t=1}^{T} \nabla f_t(x_t)^{\mathsf{T}}(x_t - u) - \frac{\beta}{2}\sum_{t=1}^{T}(\nabla f_t(x_t)^{\mathsf{T}}(x_t - u))^2 \\
&= \sum_{t=1}^{T} \tilde{f}_t(x_t) - \tilde{f}_t(u)) - \frac{\beta}{2}\sum_{t=1}^{T}(\nabla f_t(x_t)^{\mathsf{T}}(x_t - u))^2 \\
&= \mathrm{Regret}(\mathrm{AdaReg}^X_\Phi, \tilde{\boldsymbol{f}}, u) - \frac{\beta}{2}\sum_{t=1}^{T}(\nabla f_t(x_t)^{\mathsf{T}}(x_t - u))^2,
\end{aligned}$$

where in the last inequality we have used the fact that, for every $t \in [T]$, we have $\mathrm{AdaReg}^X_\Phi(\boldsymbol{f}_{1:t-1}) = \mathrm{AdaReg}^X_\Phi(\tilde{\boldsymbol{f}}_{1:t-1})$ since $\nabla \tilde{f}_t(x_t) = \nabla f_t(x_t)$. For each $t \in \{1, \ldots, T+1\}$, let $H_t, G_{t-1} \in \mathbb{S}^d_{++}$ be as in the definition of $\mathrm{AdaReg}^X_\Phi(\tilde{\boldsymbol{f}})$, and define $D_t \coloneqq H_t^{-1} - [t > 1]H_{t-1}^{-1}$. Thus, by Theorem 6.2.4 with $x_0 \coloneqq x_1$ we have

$$\mathrm{Regret}(\mathrm{AdaReg}^X_\Phi, \tilde{\boldsymbol{f}}, u) \leq \frac{1}{2}\sum_{t=0}^{T}\|u - x_t\|^2_{D_{t+1}} + \frac{1}{2}\min_{H \in \mathbb{S}^d_{++}}(\langle G_T, H\rangle + \Phi(H) - \Phi(H_1)).$$

Therefore,

$$\begin{aligned}\operatorname{Regret}(\mathrm{AdaReg}^X_\Phi, \boldsymbol{f}, u) \leq \frac{1}{2}\Big(\sum_{t=0}^{T}\|u - x_t\|^2_{D_{t+1}} - \eta\sum_{t=1}^{T}(\nabla f_t(x_t)^{\mathsf{T}}(x_t - u))^2\Big) \\ + \frac{1}{2}\min_{H \in \mathbb{S}^d_{++}}(\langle G_T, H\rangle + \Phi(H) - \Phi(H_1)).\end{aligned} \tag{6.25}$$

Let us bound each of the above terms separately. By Lemma 6.5.6, we have

$$H_t = \eta G_{t-1}^{-1} = \tfrac{1}{\beta}G_{t-1}^{-1}, \qquad \forall t \in \{1, \dots, T+1\}. \tag{6.26}$$

Hence, $D_1 = \beta\varepsilon I$ (since $G_0 = \varepsilon I$ and, thus, $H_1 = q(\beta\varepsilon)^{-1}I$) and $D_{t+1} = \beta(G_t - G_{t-1}) = \beta\nabla f_t(x_t)\nabla f_{t-1}(x_t)^{\mathsf{T}}$ for each $t \in [T]$. Thus,

$$\begin{aligned}\sum_{t=0}^{T}\|x_t - u\|^2_{D_{t+1}} = \beta\varepsilon\|x_0 - u\|_2^2 + \beta\sum_{t=1}^{T}(\nabla f_t(x_t)^{\mathsf{T}}(x_t - u))^2 \leq \beta\varepsilon\theta + \beta\sum_{t=1}^{T}(\nabla f_t(x_t)^{\mathsf{T}}(x_t - u))^2 \\ \implies \sum_{t=0}^{T}\|x_t - u\|^2_{D_{t+1}} - \beta\sum_{t=1}^{T}(\nabla f_t(x_t)^{\mathsf{T}}(x_t - u))^2 \leq \beta\varepsilon\theta.\end{aligned}$$

Moreover,

$$\begin{aligned}&\min_{H \in \mathbb{S}^d_{++}}(\langle G_T, H\rangle + \Phi(H) - \Phi(H_1)) \\ &\quad= \langle G_T, H_{T+1}\rangle + \Phi(H_{T+1}) - \Phi(H_1) \\ &\quad= \frac{1}{\beta}(\operatorname{Tr}(I) + \Phi(\beta^{-1}G_T^{-1}) - \Phi((\beta\varepsilon)^{-1}I)) && \text{by (6.26),} \\ &\quad= \frac{1}{\beta}(d - \ln\det(\beta^{-1}G_T^{-1}) + \ln\det((\beta\varepsilon)^{-1}I)) \\ &\quad= \frac{1}{\beta}(d + \ln\det(\beta G_T) - \ln(\beta\varepsilon)^d) && \text{since } \det(A^{-1}) = \det(A)^{-1}, \\ &\quad= \frac{1}{\beta}\left(d + \ln\frac{\det(\beta G_T)}{(\beta\varepsilon)^d}\right) \\ &\quad\leq \frac{1}{\beta}\left(d + \ln\frac{\beta^d(\rho^2 T + \varepsilon)^d}{\beta^d\varepsilon^d}\right) && \text{by Lemma 6.5.7,} \\ &\quad= \frac{d}{\beta}\left(1 + \ln\left(\frac{\rho^2 T}{\varepsilon} + 1\right)\right).\end{aligned}$$

By the definition of β, we have

$$\frac{1}{\beta} \leq 2\left(\frac{1}{\alpha} + 4\rho\sqrt{\theta}\right) \leq 8\left(\frac{1}{\alpha} + \rho\sqrt{\theta}\right) \implies \frac{1}{2\beta} \leq 4\left(\frac{1}{\alpha} + \rho\sqrt{\theta}\right).$$

Plugging these inequalities into (6.25) and using the definitions of ε and η yield

$$\begin{aligned}\mathrm{Regret}(\mathrm{ONS}^X, \boldsymbol{f}, u) &\leq \frac{1}{2}\left(\beta\varepsilon\theta + \frac{d}{\beta}\left(1 + \ln\left(\frac{T\rho^2}{\varepsilon} + 1\right)\right)\right)\\ &= \frac{1}{2}\left(\frac{1}{\beta} + \frac{d}{\beta}(1 + \ln(T\beta^2\rho^2\theta + 1))\right)\\ &= \frac{1}{2\beta}\left(1 + d\left(1 + \ln\left(\frac{T}{64} + 1\right)\right)\right)\\ &\leq \frac{1}{2\beta}(1 + d(1 + \ln T))\\ &= \frac{1}{2\beta}(1 + d + d\ln T)\\ &\leq \left(\frac{1}{\alpha} + \rho\sqrt{\theta}\right)4d(d^{-1} + 1 + \ln T). \qquad \square\end{aligned}$$

Different from the case of the AdaGrad algorithm, the player needs a lot of prior information about the problem, such as the Lipschitz and exp-concavity constants, to use the right parameters so that the above regret bound holds. In spite of this, the above regret is still impressive, since it is an exponential improvement (w.r.t. the number of rounds) if compared to the regret of AdaGrad.

6.6 Online Gradient Descent for Strongly Convex Functions

On [33], the authors show a diagonal version of the ONS algorithm using ideas similar to the ones used on Section 6.4 to derive the diagonal version of AdaGrad. However, in the case for diagonal matrices, to maintain the logarithmic regret bound the functions picked by the enemy need to satisfy an adapted form of exp-concavity. For the sake of brevity, we will not show how to derive this algorithm. Instead, we show a result from [33] which uses a similar idea to the one used on Section 6.4: we will constrain the meta-regularizer from the previous section to be finite only on a specific set of matrices. However, now we will restrict the meta-regularizer to the set of multiples of the identity matrix, i.e., the set $\mathbb{I}^d := \{\, \alpha I \in \mathbb{S}^d : \alpha \in \mathbb{R}\}$. Interestingly, this yields a strategy to choose step sizes on the projected gradient descent algorithm which, if the functions played by the enemy are all strongly convex, is guaranteed to attain logarithmic regret in the worst case. This is preferable than the FTL algorithm from Section 4.8 since it gives an efficient algorithm, while in the case of the FTL algorithm it is not clear how to compute it efficiently in general. Even though one could make the regret analysis of the algorithm we will see in this section using the theorems from Chapter 5 if one knew beforehand which step sizes to use, it is interesting to see that, in some sense, the Online Newton Step algorithm is indeed a generalization of the Online Gradient Descent algorithm for strongly convex functions.

The meta-regularizer which are going to use is the same as the one used in the previous section restricted to be finite only on $\mathbb{I}^d$. Thus, as we have done in the previous sections, let us prove the form and some properties of the meta-regularizer we are going to use.

Lemma 6.6.1. Let $\eta > 0$, define $f\colon \mathbb{R}^d \to \mathbb{R}$ by

$$f(x) := -\eta \sum_{i=1}^{d} [x_i > 0] \ln x_i, \qquad \forall x \in \mathbb{R}^d,$$

and set $\Phi := f_{\mathbb{S}} + \delta(\cdot \,|\, \mathbb{I}^d)$. Then,

$$\Phi(\alpha I) = -\eta d \ln \alpha, \qquad \forall \alpha \in \mathbb{R}_{++}, \tag{6.27}$$

and for every $G \in \mathbb{S}^d_{++}$ the infimum $\inf_{H \in \mathbb{S}^d_{++}}(\langle G, H\rangle + \Phi(H))$ is attained by

$$\frac{d\eta}{\mathrm{Tr}(G)} I.$$

Additionally, Φ is a meta-regularizer.

Proof. By Lemma 6.5.6, we know that $f_{\mathbb{S}}(H) = -\eta \ln \det(H)$ for every $H \in \mathbb{S}^d_{++}$. Therefore, Φ is infinite outside of $\mathbb{I}^d$ and $\Phi(\alpha I) = -\eta \ln \alpha^d = -\eta d \ln \alpha$ for $\alpha \in \mathbb{R}_{++}$, which proves (6.27). Let $G \in \mathbb{S}^d_{++}$. Let us show that

$$\left\{ \frac{d\eta}{\mathrm{Tr}(G)} I \right\} = \operatorname*{arg\,min}_{H \in \mathbb{S}^d_{++}}(\langle G, H\rangle + \Phi(H)). \tag{6.28}$$

Since Φ is proper and infinite outside of $\mathbb{I}^d$, the above minimum may be attained only by a matrix in $\mathbb{I}^d \cap \mathbb{S}^d_{++}$. Let $\bar{\alpha} \in \mathbb{R}_{++}$. Since $\mathbb{I}^d$ is an affine set, we have $\mathrm{ri}(\mathbb{I}^d) = \mathbb{I}^d$ and, thus, we have $(\mathrm{ri}(\mathbb{I}^d)) \cap \mathrm{ri}(\mathbb{S}^d_{++}) = \mathbb{I}^d \cap \mathbb{S}^d_{++} \neq \varnothing$. Hence, by the optimality conditions from Theorem 3.6.2 and since $N_{\mathbb{S}^d_{++}}(\bar{\alpha} I) = \{0\}$, we have that $\bar{\alpha} I$ attains the infimum in (6.28) if and only if $0 \in G + \partial \Phi(\bar{\alpha} I)$ is nonempty. Since $\mathrm{ri}(\mathrm{dom}\, f_{\mathbb{S}}) = \mathbb{S}^d$, by Theorem 3.5.4 we have $\partial\Phi(\bar{\alpha} I) = \nabla(f_{\mathbb{S}})(\bar{\alpha} I) + N_{\mathbb{I}^d}(\bar{\alpha} I)$. Note that

$$\begin{aligned} N_{\mathbb{I}^d}(\bar{\alpha} I) &= \{\, A \in \mathbb{S}^d : \langle A, (\alpha - \bar{\alpha}) I\rangle \leq 0, \forall \alpha \in \mathbb{R}\} \\ &= \{\, A \in \mathbb{S}^d : (\alpha - \bar{\alpha}) \,\mathrm{Tr}(A) \leq 0, \forall \alpha \in \mathbb{R}\} \\ &= \{\, A \in \mathbb{S}^d : \mathrm{Tr}(A) = 0\}. \end{aligned}$$

Moreover, by Lemma 6.5.6 we have $\nabla(f_{\mathbb{S}})(\bar{\alpha} I) = -\frac{\eta}{\bar{\alpha}} I$. Therefore, $\bar{\alpha} I$ attains the infimum in (6.28) if and only if there is $A \in \mathbb{S}^d$ with $\mathrm{Tr}(A) = 0$ such that

$$0 = G - \frac{\eta}{\bar{\alpha}} I + A \iff \frac{\eta}{\bar{\alpha}} I = G + A.$$

Note that such a matrix $A \in N_{\mathbb{I}^d}(\bar{\alpha} I)$ exists if and only if $\frac{\eta}{\bar{\alpha}} \mathrm{Tr}(I) = \mathrm{Tr}(G + A) = \mathrm{Tr}(G)$. Indeed, note that the sufficiency is clear. To see the necessity, suppose $\frac{\eta}{\bar{\alpha}} \mathrm{Tr}(I) = \mathrm{Tr}(G)$. Then, by setting $A \coloneqq -G + \frac{\eta}{\bar{\alpha}} I$, we have $\mathrm{Tr}(A) = \mathrm{Tr}(G) - \frac{\eta}{\bar{\alpha}} \mathrm{Tr}(I) = 0$ by assumption and $G + A = \frac{\eta}{\bar{\alpha}} I$. Finally, we have

$$\frac{\eta d}{\bar{\alpha}} = \mathrm{Tr}(G) \iff \bar{\alpha} = \frac{\eta d}{\mathrm{Tr}(G)}.$$

This finishes the proof of (6.28).

Let us now show that Φ is a meta-regularizer. Let $T \in \mathbb{N} \setminus \{0\}$ and $\boldsymbol{g} \in (\mathbb{R}^d)^T$. Moreover, let $\varepsilon > 0$, set $G_{T-1} \coloneqq \varepsilon I + \sum_{t=1}^{T-1} g_t g_t^{\mathsf{T}}$, and $G_T \coloneqq G_{T-1} + g_T g_T^{\mathsf{T}}$. Condition (6.3.i) is satisfied by Φ since by (6.12) we know that $\inf_{H \in \mathbb{S}^d_{++}}(\langle H, G_T\rangle + \Phi(H))$ is attained by $\bar{\alpha}_T I$, where $\bar{\alpha}_T \coloneqq \eta d \,\mathrm{Tr}(G_T)^{-1}$. Set $H_{T+1} \coloneqq \bar{\alpha}_T I$ and $H_T \coloneqq \bar{\alpha}_{T-1} I$, where $\bar{\alpha}_{T-1} \coloneqq \eta d \,\mathrm{Tr}(G_{T-1})^{-1}$. Note that

$$H_{T+1}^{-1} - H_T^{-1} = I\left(\frac{1}{\bar{\alpha}_T} - \frac{1}{\bar{\alpha}_{T-1}}\right) = I\left(\frac{\mathrm{Tr}(G_T - G_{T-1})}{\eta d}\right).$$

Since $\mathrm{Tr}(G_T - G_{T-1}) = \mathrm{Tr}(g_T g_T^{\mathsf{T}}) = \|g_T\|_2^2 \geq 0$, we conclude that the above matrix is positive semidefinite. Thus, Φ satisfies condition (6.3.ii). This finishes the proof that Φ is a meta-regularizer. □

We are in place to prove a regret bound for AdaReg using as a meta-regularizer the function Φ from the previous lemma. Additionally, we will show that the form of its update is the same as the one of online gradient descent, but with a special choice of step sizes.

Theorem 6.6.2. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that $X \subseteq \mathbb{R}^d$ is a nonempty closed convex set and such that each $f \in \mathcal{F}$ is a proper closed convex function such that f is σ-strongly convex on X w.r.t. $\|\cdot\|_2$ and subdifferentiable on X. Moreover, suppose that there is a convex set D with $X \subseteq \operatorname{int} D$ such that every function in $\mathcal{F}$ is ρ-Lipschitz continuous on D. Let $T \in \mathbb{N}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and define

$$\begin{aligned} \eta &\coloneqq \frac{\rho^2}{\sigma d},\ \varepsilon \coloneqq \frac{\rho^2}{d}, \\ h(x) &\coloneqq -\eta \sum_{i=1}^{d} [x_i > 0] \ln x_i, && \forall x \in \mathbb{R}^d, \\ \Phi &\coloneqq h_{\mathbb{S}} + \delta(\cdot \,|\, \mathbb{I}^d), \\ (\boldsymbol{x}, \boldsymbol{f}) &\coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{AdaReg}_{\Phi}^{X}, \mathrm{ENEMY}, T). \end{aligned}$$

Then,

$$\mathrm{Regret}(\mathrm{AdaReg}_{\Phi}^{X}, \boldsymbol{f}, X) \leq \frac{\rho^2}{\sigma}(\ln(T+1) + 9).$$

Additionally, let $g_t \in \partial f_t(x_t)$ be as in the definition of $\mathrm{AdaReg}_{\Phi}^{X}(\boldsymbol{f})$ for each $t \in [T]$. Then,

$$x_t = \Pi_X^{\|\cdot\|_2}([t > 1](x_{t-1} - \alpha_t g_{t-1})), \qquad \forall t \in [T], \tag{6.29}$$

where

$$\alpha_t \coloneqq \frac{\rho^2}{\sigma(\rho^2 + \sum_{i=1}^{t-1} \|g_i\|_2^2)}, \qquad \forall t \in [T].$$

Proof. Let us first prove the regret bound from the statement. For each $t \in [T]$, define $\tilde{f}_t \colon \mathbb{R}^d \to \mathbb{R}$ by

$$\tilde{f}_t(x) \coloneqq g_t^{\mathsf{T}} x, \qquad \forall x \in \mathbb{R}^d,$$

and let $u \in X$. By Theorem 3.9.7, we have

$$\begin{aligned} \mathrm{Regret}(\mathrm{AdaReg}_{\Phi}^{X}, \boldsymbol{f}, u) &= \sum_{t=1}^{T} (f_t(x_t) - f_t(u)) \leq \sum_{t=1}^{T} g_t^{\mathsf{T}}(x_t - u) - \frac{\sigma}{2} \sum_{t=1}^{T} \|x_t - u\|_2^2 \\ &= \sum_{t=1}^{T} \tilde{f}_t(x_t) - \tilde{f}_t(u)) - \frac{\sigma}{2} \sum_{t=1}^{T} \|x_t - u\|_2^2 \\ &= \mathrm{Regret}(\mathrm{AdaReg}_{\Phi}^{X}, \tilde{\boldsymbol{f}}, u) - \frac{\sigma}{2} \sum_{t=1}^{T} \|x_t - u\|_2^2, \end{aligned}$$

where in the last inequality we have used the fact that, for every $t \in [T]$, we have $\mathrm{AdaReg}_{\Phi}^{X}(\boldsymbol{f}_{1:t-1}) = \mathrm{AdaReg}_{\Phi}^{X}(\tilde{\boldsymbol{f}}_{1:t-1})$ since $\nabla \tilde{f}_i(x_i) = g_i$ for every $i \in [T]$. For each $t \in \{1, \ldots, T+1\}$ let $H_t, G_{t-1} \in \mathbb{S}_{++}^d$ be as in the definition of $\mathrm{AdaReg}_{\Phi}^{X}(\tilde{\boldsymbol{f}})$ and define $D_t \coloneqq H_t^{-1} - [t > 1]H_{t-1}^{-1}$. By Theorem 6.2.4 with

$x_0 \coloneqq x_1$ we have

$$\begin{aligned}\mathrm{Regret}(\mathrm{AdaReg}_\Phi^X, \tilde{\boldsymbol{f}}, u) &\leq \frac{1}{2}\sum_{t=0}^T \|u - x_t\|_{D_{t+1}}^2 + \frac{1}{2}\min_{H \in \mathbb{S}_{++}^d}(\langle G_T, H\rangle + \Phi(H) - \Phi(H_1))\\ &= \frac{1}{2}\sum_{t=0}^T \|u - x_t\|_{D_{t+1}}^2 + \frac{1}{2}(\langle G_T, H_{T+1}\rangle + \Phi(H_{T+1}) - \Phi(H_1)),\end{aligned}$$

where in the last equation we have used the definition of H_{T+1}. Therefore,

$$\begin{aligned}\mathrm{Regret}(\mathrm{AdaReg}_\Phi^X, \boldsymbol{f}, u) \leq \frac{1}{2}\Big(\sum_{t=0}^T (\|u - x_t\|_{D_{t+1}}^2 - \sigma \sum_{t=1}^T \|u - x_t\|_2^2\\ + \langle G_T, H_{T+1}\rangle + \Phi(H_{T+1}) - \Phi(H_1)\Big).\end{aligned} \tag{6.30}$$

Let us prove this bound in two parts. First, let us show that

$$\sum_{t=0}^T \|u - x_t\|_{D_{t+1}}^2 - \sigma \sum_{t=1}^T \|u - x_t\|_2^2 \leq \frac{16\varepsilon d}{\sigma}. \tag{6.31}$$

Let $t \in \{1, \dots, T+1\}$. Note that

$$\mathrm{Tr}(G_{t-1}) = \mathrm{Tr}\Big(\varepsilon I + \sum_{i=1}^{t-1} g_i g_i^\mathsf{T}\Big) = \mathrm{Tr}(\varepsilon I) + \sum_{i=1}^{t-1}\mathrm{Tr}(g_i g_i^\mathsf{T}) = \varepsilon d + \sum_{i=1}^{t-1}\|g_i\|_2^2 = \rho^2 + \sum_{i=1}^{t-1}\|g_i\|_2^2. \tag{6.32}$$

The above equation with Lemma 6.6.1 yields

$$H_t = \frac{\eta d}{\mathrm{Tr}(G_{t-1})} I = \frac{\rho^2}{\sigma\,\mathrm{Tr}(G_{t-1})} I = \frac{\rho^2}{\sigma(\rho^2 + \sum_{i=1}^{t-1}\|g_i\|_2^2)} I = \alpha_t I. \tag{6.33}$$

Moreover, since the functions in $\mathcal{F}$ are all ρ-Lipschitz continuous on D and $X \subseteq \mathrm{int}(D)$, by Theorem 3.8.4 we have

$$\|g_t\|_2 \leq \rho, \qquad \forall t \in [T]. \tag{6.34}$$

Therefore,

$$\begin{aligned}\sum_{t=0}^T \|x_t - u\|_{D_{t+1}}^2 &= \alpha_1^{-1}\|x_0 - u\|_2^2 + \sum_{t=1}^T (\alpha_{t+1}^{-1} - \alpha_t^{-1})\|x_t - u\|_2^2\\ &= \frac{\sigma\varepsilon d}{\rho^2}\|x_0 - u\|_2^2 + \sum_{t=1}^T \frac{\sigma\|g_t\|_2^2}{\rho^2}\|x_t - u\|_2^2\\ &\leq \frac{\sigma\varepsilon d}{\rho^2}\|x_0 - u\|_2^2 + \sum_{t=1}^T \sigma\|x_t - u\|_2^2.\end{aligned}$$

Thus, it only remains to show that $\|x_0 - u\| \leq 4\rho/\sigma$. Since $X \subseteq \mathrm{dom}\, f_1$ due to the subdifferentiablity of f_1 on X, we have that $\mathrm{dom}\, f \cap X$ is nonempty. Since f_1 and X are closed and since f_1 is strongly convex on X, by Lemma 3.9.14 there is $\bar{x} \in X$ which attains $\inf_{x \in X} f_1(x)$. Let $\bar{g} \in \partial f_1(\bar{x})$, which by Theorem 3.8.4 satisfies $\|\bar{g}\|_2 \leq \rho$ since f_1 is Lipschitz continuous on $\mathrm{int}(D) \supseteq X$. By Theorem 3.9.7 together with the minimality of $\bar{x}$ and with the fact that $\|\bar{g}\|_2 \leq \rho$, for every $x \in X$ we have

$$0 \leq f_1(\bar{x}) - f_1(x) \leq \bar{g}^\mathsf{T}(\bar{x} - x) - \frac{\sigma}{2}\|\bar{x} - x\|_2^2 \leq \rho\|\bar{x} - x\|_2 - \frac{\sigma}{2}\|\bar{x} - x\|_2^2,$$

which implies that $\|\bar{x} - x\| \leq 2\rho/\sigma$ for every $x \in X$. Thus, by the triangle inequality,

$$\|x_0 - u\|_2 \leq \|x_0 - \bar{x}\|_2 + \|\bar{x} - u\|_2 \leq \frac{4\rho}{\sigma}.$$

This ends the proof of (6.31). Now let us show that

$$\langle G_T, H_{T+1} \rangle + \Phi(H_{T+1}) - \Phi(H_1) \leq \frac{\rho^2}{\sigma}\left(1 + \ln\left(\frac{T\rho^2}{n\varepsilon} + 1\right)\right). \tag{6.35}$$

Note that

$$\operatorname{Tr}(G_T) \overset{(6.35)}{=} \varepsilon d + \sum_{t=1}^{T} \|g_t\|_2^2 \overset{(6.34)}{\leq} \varepsilon d + T\rho^2 \qquad \text{and} \qquad \operatorname{Tr}(G_0) = \operatorname{Tr}(\varepsilon I) = \varepsilon d.$$

Therefore,

$$\begin{aligned}
\langle G_T, H_{T+1} \rangle + \Phi(H_{T+1}) - \Phi(H_1) &= \alpha_{T+1} \operatorname{Tr}(G_T) - \eta d \ln \alpha_{T+1} + \eta d \ln \alpha_1 \\
&= \alpha_{T+1} \operatorname{Tr}(G_T) + \eta d \ln\left(\frac{\alpha_1}{\alpha_{T+1}}\right) \\
&\overset{(6.33)}{=} \eta d \left(1 + \ln\left(\frac{\operatorname{Tr}(G_T)}{\operatorname{Tr}(G_0)}\right)\right) \\
&= \frac{\rho^2}{\sigma}\left(1 + \ln\left(\frac{\varepsilon d + T\rho^2}{\varepsilon d}\right)\right) \\
&\leq \frac{\rho^2}{\sigma}\left(1 + \ln\left(1 + \frac{T\rho^2}{\varepsilon d}\right)\right).
\end{aligned}$$

This proves (6.35). Plugging into (6.30) both (6.31) and (6.35) yields

$$\begin{aligned}
\operatorname{Regret}(\mathrm{AdaReg}_{\Phi}^{X}, \boldsymbol{f}, u) &\leq \frac{1}{2}\left(\frac{16\varepsilon d}{\sigma} + \frac{\rho^2}{\sigma}\left(1 + \ln\left(1 + \frac{T\rho^2}{\varepsilon d}\right)\right)\right) \\
&= \frac{8\varepsilon d}{\sigma} + \frac{\rho^2}{2\sigma}\left(1 + \ln\left(1 + \frac{T\rho^2}{\varepsilon d}\right)\right) \\
&= \frac{\rho^2}{\sigma}\left(8 + \frac{1}{2}(1 + \ln(1 + T))\right) \\
&\leq \frac{\rho^2}{\sigma}(9 + \ln(1 + T)).
\end{aligned}$$

This ends the proof of the regret bound. Finally, let us see that (6.29) holds. Let $t \in [T]$. By (6.33), we have $H_t = \alpha_t I$, where α_t is positive. Thus, $\Pi^{H_t^{-1}} = \Pi^{\|\cdot\|_2}$ for every $t \in [T]$. Moreover, using (6.33) together with the definition of $\mathrm{AdaReg}_{\Phi}^{X}(\boldsymbol{f})$ we have $x_1 \in \arg\min_{x \in X} \|x\|_{H_1^{-1}} = \Pi_X^{H_1^{-1}}(0) = \Pi_X^{\|\cdot\|_2}(0)$ and

$$x_t = \Pi_X^{H_t^{-1}}(x_{t-1} - H_t g_{t-1})) = \Pi_X^{\|\cdot\|_2}(x_{t-1} - \alpha_t g_{t-1}), \qquad \forall t \in \{2, \ldots, T\}. \qquad \square$$

Chapter 7

A Genealogy of Algorithms

In the previous chapters we have presented and analyzed many algorithms for online convex optimization. One may have noticed that, in our presentation, we often derived regret bounds for an algorithm by showing that it is a special case of another, more general one. This technique of analysis is not necessarily the simplest one for all the cases and is well-known, with most of the proofs presented here based on [33, 48]. Still, it shows interesting connections among the algorithms, revealing a kind of "genealogy" of online convex optimization algorithms. Such connections may shed light on the reasons behind the effectiveness (or the lack thereof) of certain algorithms in specific cases. Not only that, it may reveal interesting branches of the genealogy which were not yet properly investigated. In this chapter, we derive and analyze classical algorithms for online convex optimization, comment on previously derived algorithms, and discuss the connections made throughout the text, summarizing them in a hopefully insightful way. Some algorithms that shall be presented in this chapter may have been derived before in other portions of the text, even if the algorithm itself was not explicitly stated. In such cases we still prove any statements we make about the algorithm for the sake of completeness.

On Section 7.1 we discuss the online gradient descent method, first presented in [72], derive a regret bound for it using the regret bounds from Chapter 5, and discuss the techniques used. On Section 7.2 we derive the Exponentiated Gradient algorithm, discuss its application to the experts' problem (a case in which the algorithm is better known as Hedge [32] or Exponentiated Multiplicative Weights Update Method [6]) and discuss some confusion regarding the update rule of the iterates of the Multiplicative Weights Update Method (MWUM). On Section 7.3 we derive the slightly more general version of the Matrix Multiplicative Weights Update Method from [43] and prove regret bounds for this algorithm. On Section 7.4 we derive a convergence bound for the Mirror Descent method for traditional convex optimization from the regret bounds for Online Mirror Descent algorithms, and discuss the limitations of this technique. Finally, on Section 7.5 we take a bird's-eye view of the connections made throughout the text among algorithms for online convex optimization.

7.1 Online Gradient Descent

The Online Gradient Descent (OGD) method, first proposed in [72], is one of the most well-known methods for online convex optimization. One major reason for its fame is its inspiration on the gradient descent method from classic convex optimization (for more information on gradient descent methods, see [15, 55]). The reader may recall that we have already derived online gradient descent methods with constant (Proposition 5.2.2) and adaptive (w.r.t. the subgradients) time-varying step sizes (see Section 6.1 and Section 6.6). On this section we will look at the Online Gradient Descent method with arbitrary non-increasing step sizes. The addition of time-varying step sizes makes the proofs of regret bounds a bit more tedious, but it is informative to describe at least one algorithm on this chapter with general step sizes. Still, for the sake of simplicity and conciseness, the next algorithms will be presented with fixed step sizes. On Algorithm 7.1 we define an oracle which implements the online gradient descent method with non-increasing time-varying step sizes.

Algorithm 7.1 Definition of $\mathrm{OGD}_{\eta}^{X}(\langle f_1, \dots, f_T \rangle)$

```
Input:
    (i) A closed convex set $X \subseteq \mathbb{E}$,
   (ii) Convex functions $f_1, \dots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{E}}$ such that $f_t$ is
        subdifferentiable on $X$ for each $t \in [T]$,
  (iii) A non-increasing function $\eta \colon \mathbb{N} \setminus \{0\} \to \mathbb{R}_{++}$
Output: $x_{T+1} \in X$
  Let $\{x_1\} \leftarrow \arg\min_{x \in X} \|x\|_2$
  $y_1 \leftarrow 0$.
  for $t = 1$ to $T$ do
      Let $g_t \in \partial f_t(x_t)$
      $y_{t+1} \leftarrow x_t - \eta_{t+1} g_t$
      $x_{t+1} \leftarrow \Pi_X^{\|\cdot\|_2}(y_{t+1})$
  return $x_{T+1}$
```

Let us show that the above algorithm is a special case of the Adaptive Online Mirror Descent algorithm from Section 5.1, and with that we derive a regret bound for OGD.

Theorem 7.1.1. Let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance such that $X \subseteq \mathbb{E}$ is closed and such that each $f \in \mathcal{F}$ is proper and closed, and let $\eta \colon \mathbb{N} \setminus \{0\} \to \mathbb{R}_{++}$ be non-increasing. Then, there is a mirror map strategy $\mathcal{R} \colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ for $\mathcal{C}$ such that $\mathrm{AdaOMD}_{\mathcal{R}}^{X} = \mathrm{OGD}_{\eta}^{X}$ in the case where both oracles use the same well-order on the sets they use in their definitions. In particular, suppose there is a nonempty open convex set $D \supseteq X$ such that $f \in \mathcal{F}$ is ρ-Lipschitz continuous on D w.r.t. $\|\cdot\|_2$ and suppose there is $\theta \in \mathbb{R}_{++}$ such that $\sup\{\, \|x - y\|_2^2 : x, y \in X\} \leq \theta$. Then, if $\mu \colon \mathbb{N} \setminus \{0\} \to \mathbb{R}_{++}$ is given by $\mu_1 \coloneqq 1$ and

$$\mu_t \coloneqq \frac{1}{\rho}\sqrt{\frac{\theta}{2(t-1)}}, \qquad \forall t \in \mathbb{N} \setminus \{0, 1\}, \tag{7.1}$$

then, for any $T \in \mathbb{N}$ and any enemy oracle ENEMY for $\mathcal{C}$ we have

$$\mathrm{Regret}_T(\mathrm{OGD}_{\mu}^{X}, \mathrm{ENEMY}, X) \leq \rho\sqrt{2\theta T}.$$

Proof. Set $R \coloneqq \frac{1}{2}\|\cdot\|_2^2$ and define $\mathcal{R} \colon \mathrm{Seq}(\mathcal{F}) \to (-\infty, +\infty]^{\mathbb{E}}$ by

$$\mathcal{R}(\boldsymbol{f}) \coloneqq \left(\frac{1}{\eta_t} - [t > 1]\frac{1}{\eta_{t-1}}\right) R, \qquad \forall \boldsymbol{f} \in \mathcal{F}^t, \forall t \in \mathbb{N}.$$

Let us show that

$$\mathcal{R} \text{ is a mirror map for } \mathcal{C}. \tag{7.2}$$

Let $t \in \mathbb{N}$ and $\boldsymbol{f} \in \mathcal{F}^t$. Since η is non-increasing, we have $\gamma_t \coloneqq \eta_t^{-1} - [t > 1]\eta_{t-1}^{-1} \geq 0$. Thus, since R is a proper closed convex function which is differentiable on $\mathbb{E}$, we have that $\mathcal{R}(\boldsymbol{f}) = \gamma_t R$

is also a proper closed convex function which is differentiable on $\mathbb{E}$. Moreover, note that $R_t \coloneqq \sum_{i=1}^{t} \mathcal{R}(\boldsymbol{f}_{1:t-1}) = \eta_t^{-1} R$. Since $\eta_t > 0$, by Lemma 5.2.1 (which shows properties of the scaled squared ℓ_2-norm when used as a mirror map), we have that R_t is a $(1/\eta_t)$-strongly convex (w.r.t the ℓ_2-norm) mirror map for X. This proves (7.2). Now, suppose OGD_{η}^{X} and $\mathrm{AdaOMD}_{\mathcal{R}}^{X}$ use the same well-order on the sets in their definitions. Let us show that

$$\mathrm{OGD}_{\eta}^{X} = \mathrm{AdaOMD}_{\mathcal{R}}^{X}. \tag{7.3}$$

Let $T \in \mathbb{N}$ and $\boldsymbol{f} \in \mathcal{F}^T$. If $T = 0$, then

$$\{\mathrm{OGD}_{\eta}^{X}(\langle\rangle)\} = \operatorname*{arg\,min}_{x \in X} \|x\|_2 = \operatorname*{arg\,min}_{x \in X} \tfrac{1}{2\eta_1}\|x\|_2^2 = \operatorname*{arg\,min}_{x \in X} \eta_1^{-1} R(x) = \{\mathrm{AdaOMD}_{\mathcal{R}}^{X}(\langle\rangle)\}.$$

Suppose now $T > 0$, and set $x_T \coloneqq \mathrm{OGD}_{\eta}^{X}(\boldsymbol{f}_{1:T-1}) = \mathrm{AdaOMD}_{\mathcal{R}}^{X}(\boldsymbol{f}_{1:T-1})$, where the equation holds by induction. Set $R_{T+1} \coloneqq \sum_{t=1}^{T+1} \mathcal{R}(\boldsymbol{f}_{1:t-1})$ as in the definition of $\mathrm{AdaOMD}_{\mathcal{R}}^{X}(\boldsymbol{f}_{1:T-1})$, and let the points $y_{T+1}, x_{T+1} \in \mathbb{E}$ and $g_T \in \partial f_T(x_T)$ be as in the definition of $\mathrm{AdaOMD}_{\mathcal{R}}^{X}(\boldsymbol{f})$ (which matches $g_T \in \partial f_T(x_T)$ as in $\mathrm{OGD}_{\eta}^{X}(\boldsymbol{f})$ by the well-order assumption). By the definition of $\mathcal{R}$, we know that $R_{T+1} = \eta_{T+1}^{-1} R$. Thus,

$$y_{T+1} = \nabla R_{T+1}(x_T) - g_T = \frac{1}{\eta_{T+1}} \nabla R(x_T) - g_T = \frac{1}{\eta_{T+1}} x_T - g_T.$$

Finally, by Lemma 5.2.1 about the squared ℓ_2-norm mirror map, we have $\nabla R_{T+1}^*(y) = \eta_{T+1} y$ for any $y \in \mathbb{E}$. Therefore,

$$x_{T+1} = \Pi_X^{R_{T+1}}(\nabla R_{T+1}^*(y_{T+1})) = \Pi_X^{R}(\eta_{T+1} y_{T+1}) = \Pi_X^{\|\cdot\|_2}(x_T - \eta_{T+1} g_T) = \mathrm{OGD}_{\eta}^{X}(\boldsymbol{f}).$$

This proves (7.3).

For the regret bound, suppose there is a convex set $D \supseteq X$ with nonempty interior such that each $f \in \mathcal{F}$ is ρ-Lipschitz continuous on D w.r.t. $\|\cdot\|_2$, suppose there is $\theta \in \mathbb{R}_{++}$ such that $\sup\{\, \|x - y\|_2^2 : x, y \in X\} \leq \theta$, and let $\mu \colon \mathbb{N} \setminus \{0\} \to (-\infty, +\infty]$ be defined as in (7.1). Let $T \in \mathbb{N}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and define

$$(\boldsymbol{x}, \boldsymbol{f}) \coloneqq \mathrm{OCO}_{\mathcal{C}}(\mathrm{OGD}_{\mu}^{X}, \mathrm{ENEMY}, T).$$

Finally, let $g_t \in \partial f_t(x_t)$ be as in the definition of $\mathrm{OGD}_{\mu}^{X}(\boldsymbol{f})$ for each $t \in [T]$ (which match $g_t \in \partial f_t(x_t)$ as in $\mathrm{AdaOMD}_{\mathcal{R}}^{X}(\boldsymbol{f})$ for each $t \in [T]$ in the case where both oracles use the same well-order on the subdifferentials they use). By Theorem 3.8.4, $\|g_t\|_2 \leq \rho$ for each $t \in [T]$. Additionally, for any $t \in [T]$ we have that $\sum_{i=1}^{t} \mathcal{R}(\boldsymbol{f}_{1:i-1}) = \mu_t^{-1} R$ is (μ_t^{-1})-strongly convex w.r.t. the ℓ_2-norm since R is 1-strongly convex w.r.t. the ℓ_2-norm. Therefore, by (7.3) together with the regret bound from

Theorem 5.4.3 yields, for any $u \in X$ and $x_0 \coloneqq x_1$,

$$\begin{aligned}
\mathrm{Regret}_T(\mathrm{OGD}_\mu^X, \mathrm{ENEMY}, u) &= \mathrm{Regret}_T(\mathrm{AdaOMD}_{\mathcal{R}}^X, \mathrm{ENEMY}, u)\\
&\leq \sum_{t=1}^{T+1} \Big(\frac{1}{\mu_t} - [t > 1] \frac{1}{\mu_{t-1}} \Big) B_R(u, x_{t-1}) + \frac{1}{2} \sum_{t=1}^{T} \mu_{T+1} \|g_t\|_2^2\\
&\leq \frac{1}{2} \sum_{t=1}^{T+1} \Big(\frac{1}{\mu_t} - [t > 1] \frac{1}{\mu_{t-1}} \Big) \|u - x_{t-1}\|_2^2 + \frac{1}{2} \sum_{t=1}^{T} \mu_{t+1} \|g_t\|_2^2\\
&\leq \frac{\theta}{2\mu_{T+1}} + \frac{\rho^2}{2} \sum_{t=1}^{T} \mu_{t+1}\\
&= \frac{\rho\sqrt{2\theta T}}{2} + \frac{\rho\sqrt{\theta}}{2\sqrt{2}} \sum_{t=1}^{T} \frac{1}{\sqrt{t}}\\
&\overset{\text{Le. 4.6.2}}{\leq} \frac{\rho\sqrt{2\theta T}}{2} + \frac{\rho\sqrt{\theta T}}{\sqrt{2}} = \rho\sqrt{2\theta T}. \qquad \square
\end{aligned}$$

The above regret bound is $\Omega(\sqrt{T})$, where $T \in \mathbb{N}$ is the number of rounds of the game. This was expected since all the bounds derived in Chapter 5 for the case of Lipschitz continuous functions are of same order, and since this is the best possible regret bound in the worst case [2] when considering only Lipschitz continuous functions. Actually, it is interesting to note that the OGD oracle is practically oblivious to any additional properties of the functions it receives as input.

For example, let $\mathcal{C} \coloneqq (X, \mathcal{F})$ be an OCO instance, let $T \in \mathbb{N}$, and let $\boldsymbol{f} \in \mathcal{F}^T$ be such that f_t is differentiable on X for each $t \in [T]$. Moreover, let $\eta \colon \mathbb{N} \to \mathbb{R}_{++}$ be non-increasing and define

$$x_t \coloneqq \mathrm{OGD}_\eta(\boldsymbol{f}_{1:t-1}) \qquad \forall t \in [T].$$

Finally, for each $t \in [T]$ define $\tilde{f}_t(x) \coloneqq \nabla f_t(x_t)^\mathsf{T} x$. Since $\nabla \tilde{f}_t(x_t) = \nabla f_t(x_t)$ for every $t \in [T]$, we have

$$\mathrm{OGD}_\eta(\boldsymbol{f}_{1:t-1}) = \mathrm{OGD}_\eta(\langle \tilde{f}_1, \dots, \tilde{f}_{t-1} \rangle), \qquad \forall t \in [T].$$

With the above equation, one may see that regardless of the functions on the sequence $\boldsymbol{f}$ being strongly convex or not (for example), the iterates from OGD will be the same. This behavior is distinct from the AdaFTRL oracle, for example, on which the properties of the functions (such as strong convexity) given as input may drastically affect the iterates. In spite of the above discussion, we are still able to prove regret bounds for online gradient descent which break the $\Omega(T)$ barrier for some special cases, such as in Theorem 6.6.2 which shows a special case where AdaReg is an instance of OGD for strongly convex functions which attains logarithmic regret. In such cases, one usually needs to upper-bound the regret of the functions $f_1, \dots, f_T$ by the regret against their linearized counterparts minus some factor yielded by the additional properties of the functions $f_1, \dots, f_T$. One may see that this is exactly what happens on Theorem 6.6.2.

7.2 Exponentiated Online Gradient Descent and Hedge

Let us now derive another classic algorithm from the online learning and online convex optimization literature (and which we have already derived previously in the text) known as *Exponentiated Gradient* algorithm [44], designed for OCO instances in which the player has to pick points in the simplex, a common problem in online learning. An oracle which formally defines it is given on Algorithm 7.2.

Algorithm 7.2 Definition of $\mathrm{ExpOGD}_\eta(\langle f_1, \dots, f_T\rangle)$

```
Input:
    (i) Convex functions $f_1, \dots, f_T \in \mathcal{F}$ for some $T \in \mathbb{N}$ and $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{R}^d}$ such that $f_t$ is
        subdifferentiable on $\Delta_d$ for each $t \in [T]$,
    (ii) A scalar $\eta \in \mathbb{R}_{++}$
Output: $x_{T+1} \in \Delta_d$
  Let $x_1 \leftarrow d^{-1}\mathbb{1}$ and set
  for $t = 1$ to $T$ do
      Let $g_t \in \partial f_t(x_t)$
      for $i = 1$ to $d$ do
          $y_{t+1}(i) \leftarrow y_t(i)\exp(-\eta g_t(i))$
      $x_{t+1} \leftarrow \frac{1}{\|y_{t+1}\|_1} y_{t+1}$
  return $x_{T+1}$
```

Let us now show that the exponentiated gradient algorithm is a special case of the Lazy Online Mirror Descent algorithm from Section 5.5 with negative entropy as a mirror map. Interestingly, in this case, eager and lazy online mirror descent are equivalent, and this is easily shown by using the result from Section 5.6.

Theorem 7.2.1. Let $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{R}^d}$ be a set of proper closed convex functions such that each $f \in \mathcal{F}$ is subdifferentiable on Δ_d, define the OCO instance $\mathcal{C} \coloneqq (\Delta_d, \mathcal{F})$, and let $\eta \in \mathbb{R}_{++}$. Finally, define

$$R(x) \coloneqq \frac{1}{\eta}\sum_{i=1}^{d}[x_i > 0]x_i \ln x_i + \delta(x \,|\, \mathbb{R}^d_+), \qquad \forall x \in \mathbb{R}^d$$

and suppose the oracles EOMD_R^X, LOMD_R^X, and ExpOGD_η use the same well-orders on the sets they use in their definitions. Then $\mathrm{EOMD}_R^X = \mathrm{LOMD}_R^X = \mathrm{ExpOGD}_\eta$. In particular, suppose there is a nonempty open convex set $D \supseteq X$ such that each $f \in \mathcal{F}$ is ρ-Lipschitz continuous on D w.r.t. $\|\cdot\|_1$. In this case, for any $T \in \mathbb{N}$ if we set

$$\mu \coloneqq \frac{1}{\rho}\sqrt{\frac{2\ln d}{T}},$$

then, for any enemy oracle ENEMY for $\mathcal{C}$ we have

$$\mathrm{Regret}_T(\mathrm{ExpOGD}_\mu, \mathrm{ENEMY}, \Delta_d) \leq \rho\sqrt{2T\ln d}.$$

Proof. Let us show that

$$R \text{ is a mirror map for } \Delta_d \text{ which is differentiable on } \mathbb{R}^d_{++} \text{ and such that } \mathrm{EOMD}_R^{\Delta_d} = \mathrm{LOMD}_R^{\Delta_d}. \tag{7.4}$$

By Lemma 5.2.3, we know that R is a 1-strongly convex w.r.t $\|\cdot\|_1$ on Δ_d mirror map for Δ_d. Moreover, by Corollary 3.2.3 we have $\operatorname{ri}\Delta_d = \{\, x \in \mathbb{R}^d : \|x\|_1 = 1, x_i > 0 \text{ for } i \in [d]\} = \Delta_d \cap \mathbb{R}^d_{++} = \Delta_d \cap \operatorname{int}(\operatorname{dom} R)$. Thus, by Theorem 5.6.1 together with the well-order assumption we have $\mathrm{EOMD}_R^X = \mathrm{LOMD}_R^X$. This ends the proof of (7.4). Let us now show that

$$\mathrm{LOMD}_R^X(\boldsymbol{f}) = \mathrm{ExpOGD}_\eta(\boldsymbol{f}), \qquad \forall \boldsymbol{f} \in \mathcal{F}^T, \forall T \in \mathbb{N}.$$

Let us prove the above claim by induction on $T \in \mathbb{N}$. So, suppose $T = 0$. Note that $\nabla R(d^{-1}\mathbb{1}) = \eta^{-1}(\mathbb{1} + \sum_{i=1}^d e_i \ln d^{-1}) = \eta^{-1}(1 - \ln d)\mathbb{1}$. Thus, for any $x \in \Delta_d$ we have $\nabla R(d^{-1}\mathbb{1})^\mathsf{T}(x - d^{-1}\mathbb{1}) = 0$, that is, $\nabla R(d^{-1}\mathbb{1}) \in N_{\Delta_d}(d^{-1}\mathbb{1})$. By the optimality conditions from Theorem 3.6.2 together with the strict convexity of R, we conclude that

$$\{\mathrm{LOMD}_R^X(\langle\rangle)\} = \operatorname*{arg\,min}_{x \in \Delta_d} R(x) = \left\{\tfrac{1}{d}\mathbb{1}\right\} = \mathrm{ExpOGD}_\eta(\langle\rangle).$$

Now let $T \in \mathbb{N} \setminus \{0\}$ and $\boldsymbol{f} \in \mathcal{F}^T$. Define $x_T \coloneqq \mathrm{LOMD}_R^X(\boldsymbol{f}_{1:T-1}) = \mathrm{ExpOGD}_\eta(\boldsymbol{f}_{1:T-1})$, where the equation holds by induction, and let $y_{T+1} \in \mathbb{R}^d$ be as in the definition of LOMD_R^X. By an easy induction, we have that $y_{T+1} = -\sum_{t=1}^T g_t$. Moreover, define $y'_{T+1} \in \mathbb{R}^d_{++}$ by

$$y'_{T+1}(i) \coloneqq \exp\Big(-\eta \sum_{t=1}^T g_t(i)\Big), \qquad \forall i \in [d].$$

Again by an easy induction, one may check that y'_{T+1} is equal to $y_{T+1} \in \mathbb{R}^d$ as defined in $\mathrm{ExpOGD}_\eta(\boldsymbol{f})$. Note that

$$\nabla R^*(y_{T+1}) = \nabla R^*\Big(-\sum_{t=1}^T g_t\Big) = \sum_{i=1}^d e_i \exp\Big(-\eta \sum_{t=1}^T g_t(i) - 1\Big)$$
$$= \sum_{i=1}^d e_i y'_{T+1} \exp(-1) = \frac{1}{e} y'_{T+1}.$$

Since $\Pi_{\Delta_d}^R(y) = \|y\|_1^{-1} y$ for any $y \in \mathbb{R}^d_{++}$ by Proposition 3.11.5, we have

$$\mathrm{LOMD}_R^X(\boldsymbol{f}) = \Pi_X^R(\nabla R^*(y_{T+1})) = \Pi_X^R\Big(\frac{1}{e} y'_{T+1}\Big) = \frac{1}{\|y'_{T+1}\|_1} y'_{T+1} = \mathrm{ExpOGD}_\eta(\boldsymbol{f}).$$

This ends the proof of (7.2).

For the regret bound, suppose there is a nonempty open convex set $D \supseteq X$ such that each $f \in \mathcal{F}$ is ρ-Lipschitz continuous on D w.r.t. $\|\cdot\|_1$. Moreover, let $T \in \mathbb{N}$, let ENEMY be an enemy oracle for $\mathcal{C}$, and define μ as in (7.2.1). Since $\mathrm{EOMD}_R^X = \mathrm{LOMD}_R^X$, the regret bound from Corollary 5.4.4 for EOMD_R^X directly yields

$$\mathrm{Regret}_T(\mathrm{ExpOGD}_\eta, \mathrm{ENEMY}, X) = \mathrm{Regret}_T(\mathrm{EOMD}_R^X, \mathrm{ENEMY}, X) \leq \rho\sqrt{2T \ln d}. \qquad \square.$$

When the ExpOGD algorithm is used on the randomized experts' problem, that is, against linear functions in the class

$$\mathcal{F} \coloneqq \{\, x \in \mathbb{R}^d \mapsto g^\mathsf{T} x : g \in [-1,1]^d \},$$

the algorithm is better known as *Hedge* [32] or *(Exponentiated) Multiplicative Weights Update Method (MWUM)* [6]. In fact, algorithms with different kinds of per-round updates of the iterates are sometimes denoted simply as "Multiplicative Weights Update Method". Thus, it is worth understanding which of these different versions of the MWUM algorithm fit our genealogy.

Let $\eta_1 \in \mathbb{R}_{++}$, let $\eta_2, \eta_3 \in (1, 1/2]$, let $g_t \in [-1,1]^d$, and let $x_t, x_{t+1} \in \Delta_d$. There are three major kinds of iterate updates which are said to be the iterate update rules of the Multiplicative Weights Update Method[1]:

$$x_{t+1}(i) \coloneqq x_t(i)\exp(-\eta_1 g_t(i)), \qquad \forall i \in [d], \tag{7.5}$$

$$x_{t+1}(i) \coloneqq \begin{cases} x_t(i)(1-\eta_2)^{g_t(i)}, & \text{if } g_t(i) \geq 0 \\ x_t(i)(1+\eta_2)^{-g_t(i)}, & \text{if } g_t(i) < 0 \end{cases} \qquad \forall i \in [d], \tag{7.6}$$

$$x_{t+1}(i) \coloneqq x_t(i)(1-\eta_3 g_t(i)), \qquad \forall i \in [d]. \tag{7.7}$$

[1] We are omitting the normalization factor present on all of them for the sake of simplicity.

The update rule derived in this section matches the one from (7.5). On [43], Satyen Kale calls by MWUM the algorithm with iterate updates as in (7.6), while in [6] MWUM is the algorithm with updates as in (7.7). It is important to note that on a remark after [43, Theorem 2] Kale already says that the update on (7.7) yields (with almost the same proof) the same bounds as the updates from (7.6) and are "easier to implement". This comes from the facts that $(1-\eta_2)^{\alpha} \leq 1-\eta_2\alpha$ for any $\alpha \in [0,1]$ and that $(1+\eta_2)^{-\alpha} \leq 1-\eta_2\alpha$ for any $\alpha \in [-1,0)$, which are used in the proofs of regret bounds on [6, 43]. Moreover, one may argue that the update rules from (7.5) and (7.7) are usually not that different in the experts case since $1-\alpha \approx e^{-\alpha}$ for small values of $\alpha \in [0,1]$. However, it is not clear if it is possible to obtain updates rules as in (7.6) or (7.7) from any of the more general algorithms from the text (AdaFTRL, AdaOMD, and AdaDA). That is, these update rules seem to be isolated in our genealogy. Not only that, on [6] the regret bounds[2] they obtain with (7.5) and (7.7) are slightly different, and in some of the applications they look at, using (7.7) leads to sharper results. Thus, it would be very interesting to discover a way (if any) to obtain the update rule (7.7) from AdaFTRL, AdaOMD, or AdaDA.

7.3 Matrix Multiplicative Weights Update Method

In this section, let us derive the *Matrix Multiplicative Weights Update Method* (MMWUM), an algorithm proposed[3] by Kale [43] with many application such as solving some kinds of semidefinite programs [7, 43] and graph sparsification [3, 20, 43]. The MMWUM algorithm relies on exponential updates similar to the ones from the previous section. Not surprisingly, we will need to define matrix exponentials. Thus, let us first define and describe some properties of the matrix exponential (and of the matrix logarithm).

Define the **matrix exponential** function $\exp\colon \mathbb{S}^d \to \mathbb{S}^d$ by

$$\exp(X) \coloneqq \sum_{k=0}^{\infty} \frac{1}{k!} X^k, \qquad \forall X \in \mathbb{S}^d.$$

A natural question is whether the above series converges for any symmetric matrix since, otherwise, the above definition would not make much sense as it is. Fortunately, this is exactly the case.

Proposition 7.3.1 ([34, Proposition 2.1])**.** For every $X \in \mathbb{S}^d$ the series

$$\sum_{k=0}^{\infty} \frac{1}{k!} X^k$$

converges and $\exp(\cdot)$ is a continuous function on $\mathbb{S}^d$.

It is easy to see that the exponential of diagonal matrix $A \in \mathbb{S}^d$ is simply a diagonal matrix with the diagonal entries exponentiated. However, for general symmetric matrices, matters get complicated. Known rules which hold for the function $\alpha \in \mathbb{R} \mapsto e^{\alpha}$, such as $e^{\alpha+\beta} = e^{\alpha}e^{\beta}$ for any $\alpha, \beta \in \mathbb{R}$, do not always hold in the case for matrices. The following proposition summarizes the main properties of the matrix exponential.

[2] The regret bound for the Hedge algorithm from [6, Theorem 2.3] is slightly different and sharper than ours. This regret bound can be seen as using "local norms" to measure the subgradients instead of fixed norms. For details and for a proof of a regret bound for Hedge using local norms, see [67, Section 2.8].

[3] Other versions of the same algorithm had already been proposed previously. Thus, it is not accurate to say that Kale was the first to propose this method. For historical details, see [43].

Proposition 7.3.2. [34, Proposition 2.3] Let $X, Y \in \mathbb{S}^d$. Then,

(i) $\exp(0) = I$,

(ii) $\exp(X)$ is invertible and $\exp(X)^{-1} = \exp(-X)$,

(iii) for any $\alpha, \beta \in \mathbb{R}$, we have $\exp((\alpha + \beta)X) = \exp(\alpha X)\exp(\beta X)$,

(iv) if $XY = YX$ (i.e., X and Y commute), then $\exp(X + Y) = \exp(X)\exp(Y) = \exp(Y)\exp(X)$,

(v) if $U \in \mathbb{R}^{d \times d}$ is invertible, then $\exp(UXU^{-1}) = U\exp(X)U^{-1}$.

Now that we have the notion of matrix exponential, we may properly define the notion of *matrix logarithm*.

Theorem 7.3.3 ([34, Theorem 2.17])**.** For every $X \in \mathbb{S}^d_{++}$ there is an unique symmetric matrix $\ln X \in \mathbb{S}^d$ such that $X = \exp(\ln X)$. Conversely, for every $X \in \mathbb{S}^d$ we have $\exp(X) \succ 0$.

For every $X \in \mathbb{S}^d_{++}$, the **logarithm** of X is the unique matrix $\ln X \in \mathbb{S}^d$ such that $\exp(\ln X) = X$. The above theorem guarantees the existence of the logarithm of positive definite matrices. However, we have not yet seen ways to write the exponential and the logarithm in ways which are easier to handle and manipulate in proofs. The following corollary gives us a way to look at the exponential and the logarithm of a matrix as functions which simply act on the eigenvalues of the matrix.

Corollary 7.3.4. Let $X \in \mathbb{S}^d$ and let $Q \in \mathbb{R}^{d \times d}$ be an orthogonal matrix such that $X = Q\operatorname{Diag}(\lambda^{\uparrow}(X))Q^{\mathsf{T}}$. Then $\exp(X) = Q\operatorname{Diag}(\mu)Q^{\mathsf{T}}$ where $\mu \in \mathbb{R}^d_{++}$ is given by

$$\mu_i := e^{\lambda_i^{\uparrow}(X)}, \qquad \forall i \in [d].$$

Moreover, if $X \succ 0$, then $\ln X = Q\operatorname{Diag}(\omega)Q^{\mathsf{T}}$, where $\omega \in \mathbb{R}^d$ is given by

$$\omega_i := \ln \lambda_i^{\uparrow}(X), \qquad \forall i \in [d].$$

Proof. Define $\lambda := \lambda^{\uparrow}(X)$. Since $\mu_i = \exp(\lambda_i)$ for each $i \in [d]$, we have $\exp(\operatorname{Diag}(\lambda)) = \operatorname{Diag}(\mu)$ by the definition of $\exp(\cdot)$. Moreover, by Proposition 7.3.2 item (iv) we have

$$\exp(Q\operatorname{Diag}(\lambda)Q^{\mathsf{T}}) = Q\exp(\operatorname{Diag}(\lambda))Q^{\mathsf{T}} = Q\operatorname{Diag}(\mu)Q^{\mathsf{T}}.$$

Suppose $X \succ 0$ and define $Y := Q\operatorname{Diag}(\omega)Q^{\mathsf{T}}$. By $\ln X = Y$. Note that $\exp(\omega_i) = \exp(\ln \lambda_i) = \lambda_i$. Therefore, $\exp(\operatorname{Diag}(\omega)) = \operatorname{Diag}(\lambda)$. Again by Proposition 7.3.2 item (iv), we have

$$\exp(Q\operatorname{Diag}(\omega)Q^{\mathsf{T}}) = Q\exp(\operatorname{Diag}(\omega))Q^{\mathsf{T}} = Q\operatorname{Diag}(\lambda)Q^{\mathsf{T}} = X.$$

That is, $\exp(Y) = X$ and, thus, $\ln X = Y$. □

Last but not least, let us show a property of matrix logarithms which shall be useful later on. We skip the statement/proof of more properties about matrix logarithms for the sake of conciseness.

Corollary 7.3.5. Let $X \in \mathbb{S}^d_{++}$ and $\alpha \in \mathbb{R}_{++}$. Then

$$\ln(\alpha X) = \ln X + (\ln \alpha)I.$$

Proof. Define $\lambda \coloneqq \lambda^{\uparrow}(X)$. By the Spectral Decomposition Theorem (Theorem 1.1.1), there is an orthogonal matrix $Q \in \mathbb{R}^{d \times d}$ such that $X = Q\,\mathrm{Diag}(\lambda)Q^{\mathsf{T}}$. Define $\omega \in \mathbb{R}^d$ by $\omega_i \coloneqq \ln(\lambda_i)$ for each $i \in [d]$. Since $\ln(\alpha\lambda_i) = \ln(\alpha) + \ln(\lambda_i)$ for each $i \in [d]$, by Corollary 7.3.4 we have $\ln(\mathrm{Diag}(\alpha\lambda)) = \ln(\alpha)I + \mathrm{Diag}(\omega)$. Therefore,

$$\ln(\alpha X) = Q\ln(\mathrm{Diag}(\alpha\lambda))Q^{\mathsf{T}} = Q\big((\ln\alpha)I + \mathrm{Diag}(\omega)\big)Q^{\mathsf{T}} = (\ln\alpha)I + \mathrm{Diag}(\omega). \qquad \square$$

Finally, we are able to define the Matrix Multiplicative Weights Update Method formally. We define a player oracle which formally implements this algorithm on Algorithm 7.3.

Algorithm 7.3 Definition of $\mathrm{MMWUM}_\eta\big(\langle f_1, \dots, f_T\rangle\big)$

```
Input:
    (i) Convex functions $f_1, \dots, f_T \colon \mathbb{S}^d \to (-\infty, +\infty]$ for some $T \in \mathbb{N}$ such that $f_t$ is subdifferentiable
        on $\mathcal{S}_d$ for each $t \in [T]$,
   (ii) A scalar $\eta \in \mathbb{R}_{++}$
Output: $X_{T+1} \in \mathcal{S}_d$
  Set $X_1 \leftarrow \frac{1}{d}I \in \mathbb{S}^d_{++}$
  for $t = 1$ to $T$ do
      Let $G_t \in \partial f_t(X_t)$
      $Y_{t+1} \leftarrow \exp(-\eta \sum_{j=1}^{t} G_j)$
      $X_{t+1} \leftarrow \frac{1}{\mathrm{Tr}(Y_{t+1})} Y_{t+1}$
  return $X_{T+1}$
```

Interestingly, we can derive Algorithm 7.3 from the Lazy Online Mirror Descent algorithm from Section 5.5 using as a mirror map a matrix analogous of the negative entropy used to derive the Exponentiated Gradient algorithm. Interestingly, in the matricial case, the negative entropy is strongly convex with a kind of ℓ_1-norm for symmetric matrices: the Shatten ℓ_1-norm.

Formally, the norm $\|\cdot\|_{S(1)} \colon \mathbb{S}^d \to (-\infty, +\infty]$ given by

$$\|X\|_{S(1)} \coloneqq \|\lambda^{\uparrow}(X)\|_1, \qquad \forall X \in \mathbb{S}^d,$$

is known as the **Shatten ℓ_1-norm**. As expected, the results we shall derive will depend on the norm dual to the Shatten ℓ_1-norm. The next lemma shows that such a dual norm is the operator norm induced by the ℓ_2-norm on $\mathbb{R}^d$.

Lemma 7.3.6. The dual norm of $\|\cdot\|_{S(1)}$ on $\mathbb{S}^d$ is the operator norm $\|\cdot\|_2$.

Proof. Let $\|\cdot\|_{S(1),*}$ be the dual norm of $\|\cdot\|_{S(1)}$. By Theorem 3.8.2, we have that $(\frac{1}{2}\|\cdot\|_{S(1)}^2)^* = \frac{1}{2}\|\cdot\|_{S(1),*}^2$. Note that[4] $\|\cdot\|_{S(1)} = (\|\cdot\|_1)_{\mathbb{S}}$, where $\|\cdot\|_1$ is the ℓ_1-norm on $\mathbb{R}^d$. Moreover, by Theorem 3.7.2 we have $(f_{\mathbb{S}})^* = (f^*)_{\mathbb{S}}$ for any proper and symmetric function $f \colon \mathbb{R}^d \to (-\infty, +\infty]$. Thus,

$$(\tfrac{1}{2}\|\cdot\|_{S(1)}^2)^* = ((\tfrac{1}{2}\|\cdot\|_1^2)^*)_{\mathbb{S}} = (\tfrac{1}{2}\|\cdot\|_\infty^2)_{\mathbb{S}} = \tfrac{1}{2}\|\cdot\|_2^2,$$

where the second equation holds by Lemma 3.8.3 and in the last equation we have used Lemma 3.8.6, which says that $\|A\|_2 = \|\lambda^{\uparrow}(A)\|_\infty$. $\square$

The next theorem, due to Ben-Tal and Nemirovski [14], shows that the matrix negative entropy is strongly convex on the spectraplex $\mathcal{S}_d \coloneqq \{\, X \in \mathbb{S}^d_+ : \mathrm{Tr}(X) = 1\}$ w.r.t. to the Shatten ℓ_1-norm.

[4] Recall from Section 3.7 that, for any symmetric function $f \colon \mathbb{R}^d \to (-\infty, +\infty]$ we define $f_{\mathbb{S}}(X) \coloneqq f(\lambda^{\uparrow}(X))$ for every $X \in \mathbb{S}^d$.

Theorem 7.3.7 ([14, Section 6.2])**.** Define $R\colon \mathbb{S}^d \to (-\infty, +\infty]$ by

$$R(X) \coloneqq \sum_{i=1}^{d} [\lambda_i^{\uparrow}(X) > 0]\lambda_i^{\uparrow}(X) \ln \lambda_i^{\uparrow}(X) + \delta(X \,|\, \mathbb{S}^d_+), \qquad \forall X \in \mathbb{S}^d.$$

Then R is $(1/2)$-strongly convex on $\mathcal{S}_d$ w.r.t. $\|\cdot\|_{S(1)}$.

Let us now show some properties of the negative matrix entropy which will be useful when deriving regret bounds for the MMWUM algorithm.

Lemma 7.3.8. Let $\eta \in \mathbb{R}_{++}$ and define $R\colon \mathbb{S}^d \to (-\infty, +\infty]$ by

$$R(X) \coloneqq \frac{1}{\eta}\sum_{i=1}^{d} [\lambda_i^{\uparrow}(X) > 0]\lambda_i^{\uparrow}(X) \ln \lambda_i^{\uparrow}(X) + \delta(X \,|\, \mathbb{S}^d_+), \qquad \forall X \in \mathbb{S}^d.$$

Then,

(i) R is a proper closed strictly convex function,

(ii) $\nabla R(X) = \eta^{-1}(I + \ln X)$ for every $X \in \mathbb{S}^d_{++}$,

(iii) $\Pi^R_{\mathcal{S}_d}(X) = \frac{1}{\mathrm{Tr}(X)} X$ for every $X \in \mathbb{S}^d_{++}$,

(iv) $R^*(X^*) = \frac{1}{\eta}\sum_{i=1}^d \exp(\eta\lambda_i^{\uparrow}(X^*) - 1) = \mathrm{Tr}(\exp(\eta X^* - I))$ for every $X^* \in \mathbb{S}^d$,

(v) $\nabla R^*(X^*) = \exp(\eta X^* - I)$ for every $X^* \in \mathbb{S}^d_{++}$.

Proof. Let us prove (i). Define $r(x) \coloneqq \sum_{i=1}^d [x_i > 0] x_i \ln x_i + \delta(\cdot \,|\, \mathbb{R}^d_+)$ for every $x \in \mathbb{R}^d$ and set $r' \coloneqq \eta^{-1} r$. Note that $R = (r')_{\mathbb{S}}$. By Lemma 5.2.3, we know that r' is a mirror map for Δ_d. In particular, r' is a proper closed strictly convex function. Thus, by Corollary 3.7.3 we have that $(r')_{\mathbb{S}} = R$ is a proper closed convex function. Additionally, strict convexity of R follows directly from the strict convexity of r', finishing the proof of (i).

For (ii), note that $\nabla r'(x) = \eta^{-1}(\mathbb{1} + \sum_{i=1}^d e_i \ln x_i)$ for every $x \in \mathbb{R}^d_{++}$. Thus, (ii) follows from Corollary 3.7.5.

For (iii), we need only apply the optimality conditions for Bregman projections from Lemma 3.11.4. Namely, let $X \in \mathbb{S}^d_{++}$ and define $\bar{X} \coloneqq \mathrm{Tr}(X)^{-1}X$. Lemma 3.11.4, $\bar{X} = \Pi^R_X(X)$ if and only if $\nabla R(\bar{X}) - \nabla R(X) \in N_{\mathcal{S}_d}(\bar{X})$. Note that, for every $Y \in \mathcal{S}_d$,

$$\begin{aligned}
\langle \nabla R(\bar{X}) - \nabla R(X), Y - \bar{X}\rangle &= \tfrac{1}{\eta}\langle \ln \bar{X} - \ln X, Y - \bar{X}\rangle \\
&\overset{\text{Cor. 7.3.5}}{=} \tfrac{1}{\eta}\langle \ln X - \ln(\mathrm{Tr}(X))I - \ln X, Y - \bar{X}\rangle \\
&= -\tfrac{1}{\eta}\ln(\mathrm{Tr}(X))\,\mathrm{Tr}(Y - \bar{X}) \\
&= -\tfrac{1}{\eta}\ln(\mathrm{Tr}(X))(1 - 1) = 0.
\end{aligned}$$

That is, $\nabla R(\bar{X}) - \nabla R(X) \in N_{\mathcal{S}_d}(\bar{X})$.

Let us now prove (iv). Let $X^* \in \mathbb{S}^d$. Recall that $R = (r')_{\mathbb{S}} = (r')_{\mathbb{S}}$. By Theorem 3.7.2, $R^* = ((r')^*)_{\mathbb{S}^d}$. By Proposition 3.4.4 together with, we have $(r')^*(x^*) = \frac{1}{\eta}\sum_{i=1}^d \exp(\eta x_i^* - 1)$, which proves (iv).

Finally, (v) follows from Corollary 3.7.5 together with the fact that $\nabla r'(x^*)_i = e^{\eta x_i^* - 1}$ for every $i \in [d]$. □

Before jumping to the regret bound, let us show that the matrix negative entropy is indeed a mirror map for the spectraplex.

Proposition 7.3.9. Let $\eta \in \mathbb{R}_{++}$ and define $R \colon \mathbb{S}^d \to (-\infty, +\infty]$ by

$$R(X) \coloneqq \frac{1}{\eta}\sum_{i=1}^{d}[\lambda_i^{\uparrow}(X) > 0]\lambda_i^{\uparrow}(X)\ln\lambda_i^{\uparrow}(X) + \delta(X \,|\, \mathbb{S}^d_{+}), \qquad \forall X \in \mathbb{S}^d,$$

Then R is a mirror map for $\mathcal{S}_d$ which is differentiable on $\mathbb{S}^d_{++}$.

Proof. From Lemma 7.3.8, we already know that

- R is closed, convex, and strictly convex from (i),
- R is differentiable on $\mathbb{S}^d_{++}$ from (ii),
- for any $Y \in \mathbb{S}^d_{++}$, the infimum $\inf_{X \in \mathbb{S}^d} B_R(X, Y)$ is attained by a matrix in $\mathbb{S}^d_{++}$ from (iii).

Thus, it only remains to shows that

$$\{\, \nabla R(X) : X \in \mathbb{S}^d_{++}\} = \mathbb{S}^d.$$

Let $X^* \in \mathbb{S}^d$. By Lemma 7.3.8, $\nabla R^*(X^*) = \exp(\eta X^* - I)$, which is positive definite by Corollary 7.3.4. Thus, R is differentiable at $\nabla R^*(X^*)$. Moreover, by Corollary 3.5.6 we have $\nabla R(\nabla R^*(X^*)) = X^*$. That is, for every $X^* \in \mathbb{S}^d_{++}$ there is $Y \coloneqq \nabla R^*(X^*) \in \mathbb{S}^d_{++}$ such that $\nabla R(Y) = X^*$. □

Finally, we are in place to prove a regret bound for the MMWUM algorithm. We first show that MMWUM is equivalent to the LOMD algorithm with mirror map the matrix negative entropy (with a scaling factor). The regret bound for the MMWUM method follows easily from the regret bound for LOMD from Corollary 5.5.3.

Theorem 7.3.10. Let $\mathcal{F} \subseteq (-\infty, +\infty]^{\mathbb{S}^d}$ be such that each $f \in \mathcal{F}$ is subdifferentiable on $\mathcal{S}_d$, define the OCO instance $\mathcal{C} \coloneqq (\mathcal{S}_d, \mathcal{F})$, and let $\eta \in \mathbb{R}_{++}$. Moreover, define

$$R(X) \coloneqq \sum_{i=1}^{d}[\lambda_i^{\uparrow}(X) > 0]\lambda_i^{\uparrow}(X)\ln\lambda_i^{\uparrow}(X) + \delta(X \,|\, \mathbb{S}^d_{+}), \qquad \forall X \in \mathbb{S}^d,$$

set $R' \coloneqq \frac{1}{\eta}R$, and suppose $\text{LOMD}^{\mathcal{S}_d}_{R'}$ and MMWUM_{η} use the same well-orders on the subdifferentials in their definitions. Then $\text{LOMD}^{\mathcal{S}_d}_{R'} = \text{MMWUM}_{\eta}$. In particular, suppose there is a nonempty open convex set $D \supseteq \mathcal{S}_d$ such that each $f \in \mathcal{F}$ is ρ-Lipschitz continuous on D w.r.t. $\|\cdot\|_{S(1)}$. In this case, for any $T \in \mathbb{N} \setminus \{0\}$, if we define

$$\mu \coloneqq \frac{\sqrt{\ln d}}{\rho\sqrt{T}} \qquad \text{and} \qquad R'' \coloneqq \frac{1}{\mu}R, \tag{7.8}$$

then, for any enemy oracle ENEMY for $\mathcal{C}$ we have

$$\text{Regret}_T(\text{MMWUM}_{\mu}, \text{ENEMY}, \Delta_d) \leq 2\rho\sqrt{T\ln d}$$

Proof. Define for every $X \in \mathbb{S}^d_{++}$. By Proposition 7.3.9, R is a mirror map for $\mathcal{S}_d$. Let us now show that $\mathrm{LOMD}^{\mathcal{S}_d}_{R'} = \mathrm{MMWUM}_\eta$. Using Lemma 7.3.8 we have, for any $Y \in \mathcal{S}_d$,

$$-\langle \nabla R'(\tfrac{1}{d}I), Y - \tfrac{1}{d}I\rangle = -\frac{1}{\eta d}\langle I + \ln I - (\ln d)I, Y - \tfrac{1}{d}I\rangle = -\frac{1-\ln d}{\eta d}\langle I, Y - \tfrac{1}{d}I\rangle = 0.$$

That is, $-\nabla R'(\frac{1}{d}I) \in N_{\mathcal{S}_d}(\frac{1}{d}I)$. Thus, by the optimality conditions from Theorem 3.6.2 together with the strict convexity of R' we have

$$\{\mathrm{LOMD}^X_{R'}(\langle\rangle)\} = \operatorname*{arg\,min}_{X \in \mathcal{S}_d} R'(X) = \{\tfrac{1}{d}I\} = \{\mathrm{MMWUM}_\eta(\langle\rangle)\}. \tag{7.9}$$

Let $T \in \mathbb{N}$ be such that $T > 0$ and let $\boldsymbol{f} \in \mathcal{F}^T$. Moreover, for each $t \in [T]$ define $X_t := \mathrm{LOMD}^X_{R'}(\boldsymbol{f}_{1:t-1}) = \mathrm{MMWUM}_\eta(\boldsymbol{f}_{1:t-1})$ (where the equation holds by induction), and let $G_t \in \partial f_t(X_t)$ be as g_t in the definition of $X_{T+1} := \mathrm{LOMD}^X_{R'}(\boldsymbol{f})$ (which matches G_t as in the definition of $\mathrm{MMWUM}_\eta(\boldsymbol{f})$ due to the well-order assumption). Finally, let $Y_{T+1} := -\sum_{t=1}^T G_t$ (which matches the definition of y_{T+1} in $\mathrm{LOMD}^X_{R'}(\boldsymbol{f})$). Note that

$$\nabla R'^*(Y_{T+1}) = \exp\Big(-\eta\sum_{t=1}^T G_t - I\Big) = \exp\Big(-\eta\sum_{t=1}^T G_t\Big)\exp(-I) = \frac{1}{e}\exp\Big(-\eta\sum_{t=1}^T G_t\Big),$$

where the second equation holds by Proposition 7.3.2 since $-I$ commutes with $-\eta\sum_{t=1}^T G_t$. Thus,

$$X_{T+1} = \Pi^{R'}_X(\nabla R'^*(Y_{T+1})) = \mathrm{Tr}\Big(\exp\Big(-\eta\sum_{t=1}^T G_t\Big)\Big)^{-1}\exp\Big(-\eta\sum_{t=1}^T G_t\Big) = \mathrm{MMWUM}_\eta(\boldsymbol{f}).$$

This completes the proof that $\mathrm{LOMD}^X_{R'} = \mathrm{MMWUM}_\eta$.

For the regret bound, suppose there is a convex set $D \supseteq \mathcal{S}_d$ with nonempty interior such that each $f \in \mathcal{F}$ is ρ-Lipschitz continuous on D w.r.t. $\|\cdot\|_{S(1)}$. Moreover, let $T \in \mathbb{N} \setminus \{0\}$, define μ and R'' as in (7.8), and let ENEMY be an enemy oracle for $\mathcal{C}$. By Theorem 7.3.7, R is $(1/2)$-strongly convex on $\mathcal{S}_d$ w.r.t. $\|\cdot\|_{S(1)}$. Moreover, for every $U \in \mathcal{S}_d$, since $\lambda^\uparrow(U) \geq 0$ and $\mathbb{1}^\mathsf{T}\lambda^\uparrow(U) = \mathrm{Tr}(U) = 1$ we have $\ln \lambda^\uparrow_i(U) \leq 0$ for each $i \in [d]$, which implies $R(U) \leq 0$. Moreover, from (7.9) we have that

$$\inf_{X \in \mathcal{S}_d} R(X) = R(\tfrac{1}{d}I) = \frac{1}{d}\sum_{i=1}^d \ln\Big(\frac{1}{d}\Big) = -\ln d.$$

Therefore, $\sup\{\, R(U) - R(X) : U, X \in \mathcal{S}_d\} \leq \ln d$. Finally, by setting $\theta := \ln d$ and $\sigma := 1/2$ we have

$$R'' = \frac{1}{\mu}R = \frac{\rho\sqrt{T}}{\sqrt{\ln d}}R = \frac{\rho\sqrt{T}}{\sqrt{2\sigma\ln d}}R = \frac{\rho\sqrt{T}}{\sqrt{2\sigma\theta}}R$$

Hence, by the regret bound for $\mathrm{LOMD}^X_{R''}$ from Corollary 5.5.3 we have

$$\begin{aligned}\mathrm{Regret}_T(\mathrm{MMWUM}_\eta, \mathrm{ENEMY}, \mathcal{S}_d) &= \mathrm{Regret}_T(\mathrm{LOMD}^X_{R''}, \mathrm{ENEMY}, \mathcal{S}_d) \\ &\leq \rho\sqrt{\frac{2\theta T}{\sigma}} = 2\rho\sqrt{(\ln d)T}.\end{aligned}$$

□

7.4 Offline Algorithms

As one may have already noticed, there are major similarities between some OCO algorithms and some classic optimization methods such as gradient or mirror descent methods. Thus, one may wonder whether we can use OCO algorithms in classic (convex) optimization. In methods for classic optimization which rely only on first-order information (values and subgradients of the function we are looking at), we are usually interested in finding a point with approximates the infimum with a fixed precision[5]. That is, given a single proper convex function $f \colon \mathbb{E} \to (-\infty, +\infty]$, a nonempty closed convex set $X \subseteq \mathbb{E}$, and a precision $\varepsilon > 0$, we want to find a point $\bar{x} \in X$ such that

$$f(\bar{x}) - \inf_{x \in X} f(x) \leq \varepsilon.$$

In such cases, we are interested in the number of iterations/rounds needed to obtain a precision of $\varepsilon > 0$.

It turns out that we can indeed recover the classic Mirror Descent algorithm from classic convex optimization (see [4, 13, 18] for details about the classic mirror descent algorithm and historical references). Moreover, we can obtain convergence guarantees almost directly from the sub-linear regret bounds we have for Online Mirror Descent algorithms. On Algorithm 7.4 we define a function/oracle which implements the Mirror Descent algorithm for convex optimization.

Algorithm 7.4 Definition of $\mathrm{MD}_R^X(f, T)$

```
Input:
   (i)  Number of steps $T > 0$,
   (ii)  A nonempty closed convex set $X \subseteq \mathbb{E}$,
   (iii)  A closed convex function $f \colon \mathbb{E} \to (-\infty, +\infty]$ such that $f$ is subdifferentiable on $X$,
   (iv)  A mirror map $R \colon \mathbb{E} \to (-\infty, +\infty]$ for $X$.
Output: $x_T \in X$
  $\{x_1\} \leftarrow \arg\min_{x \in X} R(x)$.
  for $t = 1$ to $T - 1$ do
      Let $g_t \in \partial f(x_t)$
      $y_{t+1} \leftarrow \nabla R(x_t) - g_t$
      $x_{t+1} \leftarrow \Pi_X^R(\nabla R^*(y_{t+1}))$
  return $\frac{1}{T} \sum_{t=1}^{T} x_t$
```

Given a proper convex function $f \colon \mathbb{E} \to (-\infty, +\infty]$, a nonempty closed convex set $X \subseteq \mathbb{E}$, a mirror map R for X, and a number of iterations $T \in \mathbb{N}$, the above algorithm performs T rounds of the (online) mirror descent algorithm, and returns the *average* of its iterates. In the next theorem we show a convergence bound for the algorithm from Algorithm 7.4. Interestingly, the proof is almost a direct application of the convexity of f together with the regret bound for the EOMD algorithm from Corollary 5.4.4.

Theorem 7.4.1. Let $X \subseteq \mathbb{E}$ be convex and closed, let $f \colon \mathbb{E} \to (-\infty, +\infty]$ be a proper closed convex function which is subdifferentiable on X, and let R be a mirror map for X. Moreover, let $\|\cdot\|$ be a norm on $\mathbb{E}$ such that

[5] It is worth noting that this is not the only goal optimization methods focus on. For example, some methods on finding a point whose euclidean distance with a global minimizer is as small as we want.

- f is ρ-Lipschitz continuous w.r.t. $\|\cdot\|$ on a nonempty open convex set $D \supseteq X$, and
- R is σ-strongly convex w.r.t. $\|\cdot\|$ on X.

Finally, suppose there is $\theta \in \mathbb{R}_{++}$ such that $\sup\{\, R(u) - R(x) : u \in X, x \in X \cap \operatorname{dom} R\} \leq \theta$, and suppose $\inf_{x \in X} f(x)$ is attained. Then, by setting

$$\eta \coloneqq \frac{\sqrt{2\sigma\theta}}{\rho\sqrt{T}} \qquad \text{and} \qquad R' \coloneqq \frac{1}{\eta} R,$$

for any $T \in \mathbb{N}$ with $T \geq \frac{\rho^2 2\theta}{\sigma\varepsilon^2}$ we have

$$f(\mathrm{MD}^X_{R'}(f, T)) - \min_{x \in X} f(x) \leq \varepsilon.$$

Proof. Let $T \in \mathbb{N} \setminus \{0\}$ and for every $t \in \mathbb{N}$ define $\boldsymbol{f}^{(t)} \coloneqq \langle f, \dots, f\rangle \in \{f\}^t$. For each $t \in [T]$ equip to the subdifferentials used on the definition of $\mathrm{EOMD}^X_{R'}(\boldsymbol{f}^{(t-1)})$ the same well-order used on the subdifferentials on the definition of $\mathrm{EOMD}^X_{R'}(\boldsymbol{f}^{(t-1)})$. Moreover, define $x_t \coloneqq \mathrm{EOMD}^X_{R'}(\boldsymbol{f}^{(t-1)})$. In this way, we have

$$\frac{1}{T}\sum_{t=1} x_t = \mathrm{MD}^X_{R'}(f, T).$$

Finally, let $\bar{x} \in X$ attain $\inf_{x \in X} f(x)$. Using the regret bound from Corollary 5.4.4 for $\mathrm{EOMD}^X_{R'}$ and the convexity of f, we have

$$\begin{aligned} f(\mathrm{MD}^X_{R'}(f, T)) - f(\bar{x}) &= f\Big(\frac{1}{T}\sum_{t=1}^{T} x_t\Big) - \frac{T}{T} f(\bar{x}) \leq \frac{1}{T}\big(\sum_{t=1}^{T}(f(x_t) - f(x^*))\big) \\ &= \frac{1}{T}\,\mathrm{Regret}(\mathrm{EOMD}^X_{R'}, \boldsymbol{f}^{(T)}, x^*) \leq \rho\sqrt{\frac{2\theta}{\sigma T}}. \end{aligned}$$

Thus, if $\varepsilon > 0$ and $T \geq \frac{\rho^2 2\theta}{\varepsilon^2}$, then we have that $f(\mathrm{MD}^X_{R'}(f, T)) - f(\bar{x}) \leq \varepsilon$. □

The above result recovers the traditional convergence bounds for mirror descent (see, for example, [18, Theorem 4.2]). Indeed, from the above proof one may see that from any OCO algorithm with a $O(\sqrt{T})$ regret bound (where T is the number of rounds of the game) we may derive an algorithm for classic convex optimization (by computing the average of the iterates) which converges to a solution with a precision of $\varepsilon > 0$ in roughly $O(\frac{1}{\varepsilon^2})$ iterations. As we have already seen, using the squared ℓ_2-norm as a mirror map yields the (sub)gradient descent method, and the convergence rate of $O(\frac{1}{\varepsilon^2})$ matches the known convergence rates for the subgradient descent method [18, Theorem 3.2]. However, it is known that gradient descent for strongly smooth functions, for example, converges (with properly chosen step sizes) to a solution of precision $\varepsilon > 0$ in $O(\frac{1}{\varepsilon})$ rounds [18, Theorem 3.3]. If we were to derive such a convergence rate using a proof method similar to the one from Theorem 7.4.1, it seems we would need to have a regret bound of $O(1)$ w.r.t. the number of rounds since the convergence rate we get is $O(R_T/T)$, where R_T is the regret of the algorithm on a game with $T \in \mathbb{N} \setminus \{0\}$ rounds. Thus, it seems that trying to obtain convergence rates for convex optimization methods directly from regret bounds for their online counter parts has major limitations.

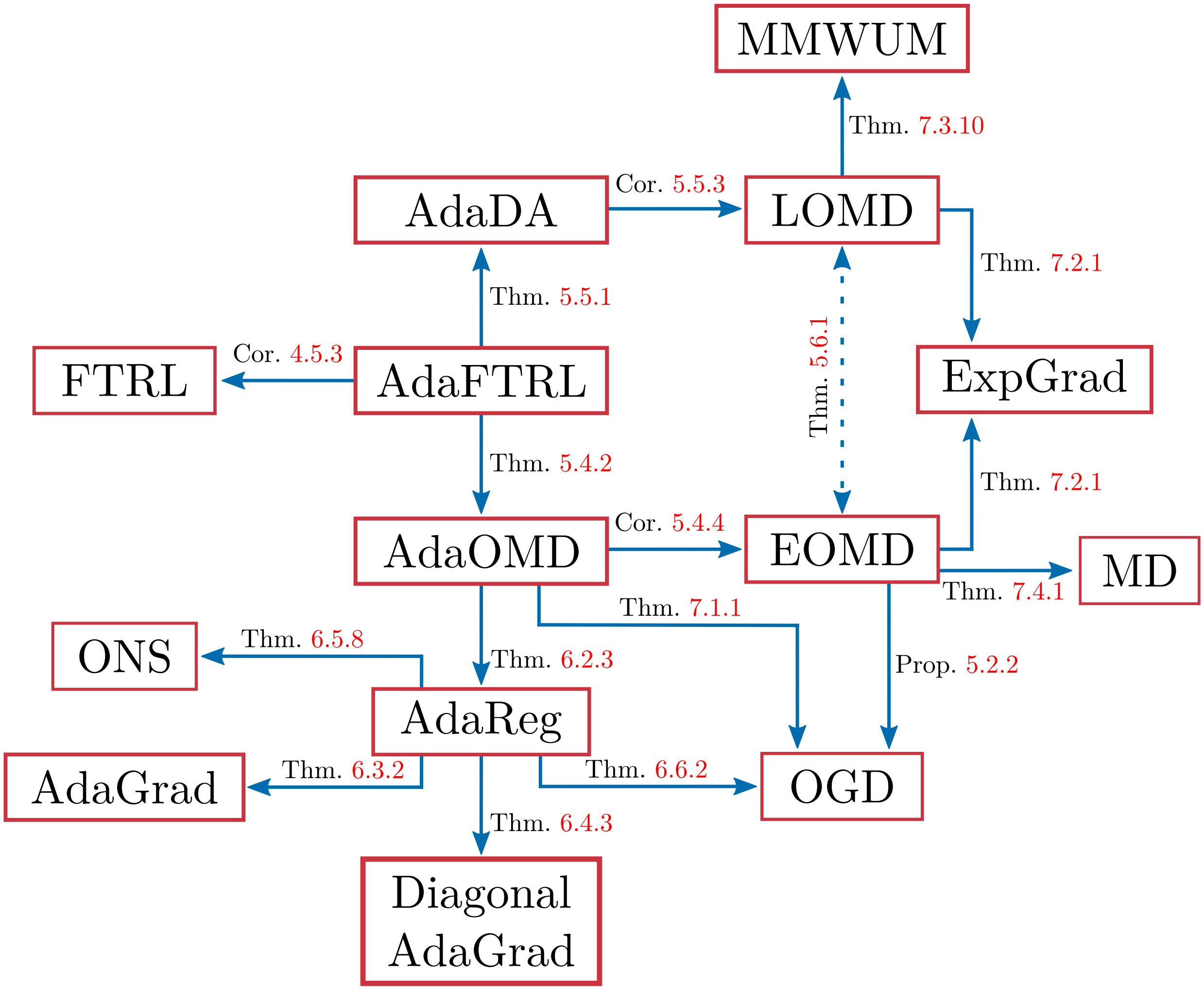

Figure 7.1: A graphic representation of the connections among algorithms described throughout the text.

7.5 A Genealogy

On Figure 7.1 we present a schematic representation of the connections drawn among algorithms throughout the text. On the figure, an arrow from an algorithm A to B means that B can be written as a special case of A.

From Figure 7.1 it is clear that we can trace back all the algorithms present on the figure to the AdaFTRL algorithm. This fact is also reflected on the regret proofs we have done throughout the text, which had as their fundamental building blocks the lemmas from Section 4.3. As we have commented previously, proving regret bounds mainly based on these lemmas and on equivalences between player oracles, as we have done throughout the text, is not always the simplest and cleanest proof technique. For example, the connections between AdaOMD and AdaFTRL and the regret bounds yielded by these connections shown in Section 5.4 had quite technical and tiresome proofs.

On the other hand, this way of analysis revealed interesting intuition and facts about OCO algorithms. Maybe the result which most benefited from writing algorithms as special cases of others was the proof of equivalence between the EOMD and LOMD algorithms for some special kinds of mirror maps. As we have seen on Section 5.6, writing both algorithms as special cases of FTRL makes almost obvious some simple sufficient conditions for them to be equivalent.

Another interesting connection revealed by this type of analysis is to that AdaGrad and ONS are special cases of the AdaReg algorithm and that their proofs of convergence can be unified in a very interesting fashion, a fact which was not clear at first in spite of the similarities between the AdaGrad and ONS algorithms. Not only that, we have seen that the ONS algorithm is a special case of the AdaReg algorithm with the function $X \in \mathbb{S}^d_{++} \mapsto -\ln\det(X)$ as a meta-regularizer, where the latter is a well-known function used in interior-point methods [58]. Finally, we have seen on Section 6.6 how, in some sense, ONS can be seen as a generalization[6] of the online gradient descent for strongly convex functions.

Moreover, even though the cornerstone algorithm of the genealogy is the AdaFTRL algorithm, we can see that all algorithms are closer to Online Mirror Descent methods. This is probably due to the simpler (and usually easier to implement) iterate update rule of the Online Mirror Descent algorithms. Still, we have seen in the text that writing these algorithms in the format of the AdaFTRL algorithm can be extremely useful for deriving regret bounds and relationships among algorithms.

Finally, this bird's-eye view of the connections among OCO algorithms helps us see some interesting paths for future investigation. For example, throughout the text we relied on strong convexity of the functions played by the enemy or of the regularizers/mirror maps used to derive interesting regret bounds. One interesting alternative, first proposed by [1], is to use self-concordant barriers from the theory of interior-point methods [58] as regularizers on the FTRL oracle. It would be interesting if we could come up with a lemma analogous to Lemma 4.3.2 for the case of self-concordant barriers instead of strongly-convex functions. Moreover, their focus is on the bandit setting (that is, the case with limited feedback) against linear functions. Thus, it would also be interesting to see the effects of self-concordant barriers in other cases.

Another interesting path of investigation are second-order methods for OCO. In spite of its name, the Online Newton Step algorithm does not use the Hessians of the functions to skew the gradient steps, using instead a matrix built purely from rank-one updates based on the gradients used in the past. Thus, there is no real online counterpart for the Newton's method algorithm for convex optimization (see [15, Section 6.2] for a brief and introductory description of the Newton's method). Thus, investigating algorithms which use the Hessians of the enemy's functions may yield interesting algorithms, or discovering that such second-order algorithms would be no better than first-order ones might also shed some light on the limits of the OCO framework.

[6] Although there is no arrow from ONS to OGD in Figure 7.1, the meta-regularizers used on Theorem 6.5.8 for ONS and on Theorem 6.6.2 for OGD are almost the same, with the distinction that the meta-regularizer for OGD is restricted to multiples of the identity matrix.

References

[1] J. D. Abernethy, E. Hazan, and A. Rakhlin. “Interior-point methods for full-information and bandit online learning”. In: *IEEE Trans. Inform. Theory* 58.7 (2012), pages 4164–4175. URL: https://doi.org/10.1109/TIT.2012.2192096 (cited on page 186).

[2] J. D. Abernethy, P. Bartlett, A. Rakhlin, and A. Tewari. “Optimal Strategies and Minimax Lower Bounds for Online Convex Games”. In: UCB/EECS-2008-19 (February 2008). URL: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-19.pdf (cited on pages 96, 105, 159, 174).

[3] Z. Allen-Zhu, Z. Liao, and L. Orecchia. “Spectral sparsification and regret minimization beyond matrix multiplicative updates [extended abstract]”. In: *STOC’15—Proceedings of the 2015 ACM Symposium on Theory of Computing*. ACM, New York, 2015, pages 237–245 (cited on pages 4, 131, 132, 177).

[4] Z. Allen-Zhu and L. Orecchia. “Linear Coupling: An Ultimate Unification of Gradient and Mirror Descent”. Version 5. In: (November 2016). arXiv: 1407.1537 [cs.DS]. URL: https://arxiv.org/abs/1407.1537 (cited on page 183).

[5] R. Arora, O. Dekel, and A. Tewari. “Online Bandit Learning against an Adaptive Adversary: from Regret to Policy Regret”. In: *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. 2012. URL: http://icml.cc/2012/papers/749.pdf (cited on pages 38–40).

[6] S. Arora, E. Hazan, and S. Kale. “The multiplicative weights update method: a meta-algorithm and applications”. In: *Theory Comput.* 8 (2012), pages 121–164 (cited on pages 16, 114, 171, 176, 177).

[7] S. Arora and S. Kale. “A combinatorial, primal-dual approach to semidefinite programs [extended abstract]”. In: *STOC’07—Proceedings of the 39th Annual ACM Symposium on Theory of Computing*. ACM, New York, 2007, pages 227–236. URL: https://doi.org/10.1145/1250790.1250823 (cited on pages 4, 177).

[8] R. B. Ash. *Probability and measure theory*. Second. With contributions by Catherine Doléans-Dade. Harcourt/Academic Press, Burlington, MA, 2000, pages xii+516 (cited on page 7).

[9] M. F. Atiyah. *Duality in Mathematics and Physics*. 2007. URL: http://www.iecl.univ-lorraine.fr/~Wolfgang.Bertram/Atiyah-Duality.pdf (cited on pages 48, 50).

[10] J.-Y. Audibert and S. Bubeck. “Regret bounds and minimax policies under partial monitoring”. In: *J. Mach. Learn. Res.* 11 (2010), pages 2785–2836 (cited on page 37).

[11] N. Bansal and A. Gupta. “Potential-Function Proofs for First-Order Methods”. In: (December 2017). arXiv: 1712.04581 [cs.LG]. URL: http://arxiv.org/abs/1712.04581 (cited on page 123).

[12] B. Barak, M. Hardt, and S. Kale. “The uniform hardcore lemma via approximate Bregman projections”. In: *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, Philadelphia, PA, 2009, pages 1193–1200 (cited on page 4).

[13] A. Beck and M. Teboulle. “Mirror descent and nonlinear projected subgradient methods for convex optimization”. In: *Oper. Res. Lett.* 31.3 (2003), pages 167–175. URL: https://doi.org/10.1016/S0167-6377(02)00231-6 (cited on pages 108, 112, 115, 183).

[14] A. Ben-Tal and A. Nemirovski. “Non-Euclidean restricted memory level method for large-scale convex optimization”. In: *Math. Program.* 102.3, Ser. A (2005), pages 407–456. URL: https://doi.org/10.1007/s10107-004-0553-4 (cited on pages 179, 180).

[15] A. Ben-Tal and A. Nemirovski. *Optimization III*. 2013. URL: http://www2.isye.gatech.edu/~nemirovs/OPTIII_LectureNotes2015.pdf (cited on pages 41, 43, 171, 186).

[16] D. P. Bertsekas. *Convex analysis and optimization*. With Angelia Nedić and Asuman E. Ozdaglar. Athena Scientific, Belmont, MA, 2003, pages xvi+534 (cited on page 64).

[17] J. M. Borwein and A. S. Lewis. *Convex analysis and nonlinear optimization*. Second. Volume 3. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Theory and examples. Springer, New York, 2006, pages xii+310. URL: https://doi.org/10.1007/978-0-387-31256-9 (cited on pages 41, 58, 59).

[18] S. Bubeck. “Convex Optimization: Algorithms and Complexity”. In: *Foundations and Trends in Machine Learning* 8.3-4 (2015), pages 231–357. URL: https://doi.org/10.1561/2200000050 (cited on pages 4, 41, 108, 109, 183, 184).

[19] S. Bubeck. *Introduction to Online Optimization*. Princeton University, December 14, 2011. URL: http://www.cse.iitd.ac.in/~naveen/courses/CSL866/BubeckLectureNotes.pdf (cited on pages 3, 4, 10, 12, 18, 75, 108, 112).

[20] M. K. de Carli Silva, N. J. A. Harvey, and C. M. Sato. *Sparse Sums of Positive Semidefinite Matrices*. October 2011. arXiv: 1107.0088 [cs.DM]. URL: http://arxiv.org/abs/1107.0088 (cited on pages 4, 177).

[21] N. Cesa-Bianchi, A. Conconi, and C. Gentile. “On the generalization ability of on-line learning algorithms”. In: *IEEE Trans. Inform. Theory* 50.9 (2004), pages 2050–2057. URL: https://doi.org/10.1109/TIT.2004.833339 (cited on pages 25, 26).

[22] N. Cesa-Bianchi, O. Dekel, and O. Shamir. “Online Learning with Switching Costs and Other Adaptive Adversaries”. In: *Advances in Neural Information Processing Systems 26*. Edited by C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger. Curran Associates, Inc., 2013, pages 1160–1168. URL: http://papers.nips.cc/paper/5151-online-learning-with-switching-costs-and-other-adaptive-adversaries.pdf (cited on page 40).

[23] N. Cesa-Bianchi and C. Gentile. “Improved risk tail bounds for on-line algorithms”. In: *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*. 2005, pages 195–202. URL: http://papers.nips.cc/paper/2839-improved-risk-tail-bounds-for-on-line-algorithms (cited on page 26).

[24] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006 (cited on pages 10, 16, 20, 22, 37, 38, 40, 75).

[25] P. Christiano, J. A. Kelner, A. Mądry, D. A. Spielman, and S.-H. Teng. "Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs". In: *STOC'11—Proceedings of the 43rd ACM Symposium on Theory of Computing*. ACM, 2011, pages 273–281. STOC Best Paper Award (cited on page 4).

[26] T. M. Cover. "Behavior of sequential predictors of binary sequences". In: *Trans. Fourth Prague Conf. on Information Theory, Statistical Decision Functions, Random Processes (Prague, 1965)*. Academia, Prague, 1967, pages 263–272 (cited on pages 3, 22).

[27] T. M. Cover. "Universal portfolios". In: *Math. Finance* 1.1 (1991), pages 1–29. URL: https://doi.org/10.1111/j.1467-9965.1991.tb00002.x (cited on page 19).

[28] V. Dani and T. P. Hayes. "Robbing the bandit: less regret in online geometric optimization against an adaptive adversary". In: *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM, New York, 2006, pages 937–943. URL: https://doi.org/10.1145/1109557.1109660 (cited on page 37).

[29] O. Dekel. "From Online to Batch Learning with Cutoff-Averaging". In: *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*. 2008, pages 377–384. URL: http://papers.nips.cc/paper/3514-from-online-to-batch-learning-with-cutoff-averaging (cited on page 26).

[30] O. Dekel and Y. Singer. "Data-Driven Online to Batch Conversions". In: *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*. 2005, pages 267–274. URL: http://papers.nips.cc/paper/2775-data-driven-online-to-batch-conversions (cited on page 26).

[31] J. Duchi, E. Hazan, and Y. Singer. "Adaptive subgradient methods for online learning and stochastic optimization". In: *J. Mach. Learn. Res.* 12 (2011), pages 2121–2159 (cited on pages 134, 135, 139, 148, 152, 153).

[32] Y. Freund and R. E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". In: *J. Comput. System Sci.* 55.1, part 2 (1997). Second Annual European Conference on Computational Learning Theory (EuroCOLT '95) (Barcelona, 1995), pages 119–139. URL: https://doi.org/10.1006/jcss.1997.1504 (cited on pages 114, 171, 176).

[33] V. Gupta, T. Koren, and Y. Singer. *A Unified Approach to Adaptive Regularization in Online and Stochastic Optimization*. June 2017. arXiv: 1706.06569 [cs.LG]. URL: http://arxiv.org/abs/1706.06569 (cited on pages 4, 134, 140–142, 166, 171).

[34] B. Hall. *Lie groups, Lie algebras, and representations*. Second. Volume 222. Graduate Texts in Mathematics. An elementary introduction. Springer, Cham, 2015, pages xiv+449. URL: https://doi.org/10.1007/978-3-319-13467-3 (cited on pages 177, 178).

[35] P. R. Halmos. *Measure Theory*. D. Van Nostrand Company, Inc., New York, N. Y., 1950, pages xi+304 (cited on page 7).

[36] E. Hazan. "Introduction to online convex optimization". In: *Foundations and Trends® in Optimization* 2.3-4 (2016), pages 157–325. URL: http://ocobook.cs.princeton.edu/OCObook.pdf (cited on pages 3, 4, 10, 12, 38, 112, 153).

[37] E. Hazan, A. Kalai, S. Kale, and A. Agarwal. “Logarithmic regret algorithms for online convex optimization”. In: *Learning theory*. Volume 4005. Lecture Notes in Comput. Sci. Springer, Berlin, 2006, pages 499–513. URL: https://doi.org/10.1007/11776420_37 (cited on pages 134, 135, 140, 161).

[38] S. C. H. Hoi, D. Sahoo, J. Lu, and P. Zhao. *Online Learning: A Comprehensive Survey*. February 2018. arXiv: 1802.02871 [cs.LG]. URL: http://arxiv.org/abs/1802.02871 (cited on page 12).

[39] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, 1990. xiv+561 (cited on pages 9, 63).

[40] R. Jain, Z. Ji, S. Upadhyay, and J. Watrous. “QIP = PSPACE”. In: *STOC’10—Proceedings of the 2010 ACM International Symposium on Theory of Computing*. ACM, New York, 2010, pages 573–581 (cited on page 4).

[41] S. M. Kakade, S. Shalev-Shwartz, and A. Tewari. “Regularization techniques for learning with matrices”. In: *J. Mach. Learn. Res.* 13 (2012), pages 1865–1890 (cited on page 73).

[42] A. Kalai and S. Vempala. “Efficient algorithms for online decision problems”. In: *J. Comput. System Sci.* 71.3 (2005), pages 291–307. URL: http://dx.doi.org/10.1016/j.jcss.2004.10.016 (cited on pages 78, 84, 106).

[43] S. Kale. “Efficient Algorithms Using The Multiplicative Weights Update Method”. PhD thesis. Princeton University, 2007 (cited on pages 171, 177).

[44] J. Kivinen and M. K. Warmuth. “Exponentiated gradient versus gradient descent for linear predictors”. In: *Inform. and Comput.* 132.1 (1997), pages 1–63. URL: https://doi.org/10.1006/inco.1996.2612 (cited on page 174).

[45] A. S. Lewis. “Convex analysis on the Hermitian matrices”. In: *SIAM J. Optim.* 6.1 (1996), pages 164–177. URL: https://doi.org/10.1137/0806009 (cited on page 57).

[46] N. Littlestone. “From on-line to batch learning”. In: *Proceedings of the Second Annual Workshop on Computational Learning Theory (Santa Cruz, CA, 1989)*. Morgan Kaufmann, San Mateo, CA, 1989, pages 269–284 (cited on pages 25, 26).

[47] N. Littlestone. “Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm (Extended Abstract)”. In: *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*. 1987, pages 68–77. URL: https://doi.org/10.1109/SFCS.1987.37 (cited on page 12).

[48] H. B. McMahan. “A survey of algorithms and analysis for adaptive online learning”. In: *J. Mach. Learn. Res.* 18 (2017), Paper No. 90, 50 (cited on pages 4, 78, 84, 86, 88, 108, 123, 171).

[49] H. B. McMahan and M. J. Streeter. “Adaptive Bound Optimization for Online Convex Optimization”. In: (February 2010). arXiv: 1002.4908 [cs.LG]. URL: https://arxiv.org/abs/1002.4908 (cited on page 153).

[50] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2012, pages xii+412 (cited on pages 3, 12, 13, 22, 24).

[51] J.-J. Moreau. “Fonctions convexes duales et points proximaux dans un espace hilbertien”. In: *C. R. Acad. Sci. Paris* 255 (1962), pages 2897–2899 (cited on page 115).

[52] J.-J. Moreau. “Inf-convolution des fonctions numériques sur un espace vectoriel”. In: *C. R. Acad. Sci. Paris* 256 (1963), pages 5047–5049 (cited on page 115).

[53] J.-J. Moreau. “Propriétés des applications “prox””. In: *C. R. Acad. Sci. Paris* 256 (1963), pages 1069–1071 (cited on page 115).

[54] A. S. Nemirovsky and D. B. a. Yudin. *Problem complexity and method efficiency in optimization.* A Wiley-Interscience Publication. Translated from the Russian and with a preface by E. R. Dawson, Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, Inc., New York, 1983, pages xv+388 (cited on pages 108, 109).

[55] Y. Nesterov. *Introductory lectures on convex optimization. A basic course.* Volume 87. Applied Optimization. Kluwer Academic Publishers, Boston, MA, 2004, pages xviii+236. URL: http://dx.doi.org/10.1007/978-1-4419-8853-9 (cited on pages 41, 72, 171).

[56] Y. Nesterov. “Primal-dual subgradient methods for convex problems”. In: *Math. Program.* 120.1, Ser. B (2009), pages 221–259. URL: https://doi.org/10.1007/s10107-007-0149-x (cited on page 126).

[57] N. Parikh and S. Boyd. “Proximal Algorithms”. In: *Foundations and Trends® in Optimization* 1.3 (2014), pages 127–239. URL: http://dx.doi.org/10.1561/2400000003 (cited on pages 108, 115, 116).

[58] J. Renegar. *A mathematical view of interior-point methods in convex optimization.* MPS/SIAM Series on Optimization. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Programming Society (MPS), Philadelphia, PA, 2001, pages viii+117. URL: https://doi.org/10.1137/1.9780898718812 (cited on pages 161, 186).

[59] R. T. Rockafellar. *Convex analysis.* Princeton Landmarks in Mathematics. Reprint of the 1970 original, Princeton Paperbacks. Princeton, NJ: Princeton University Press, 1997, pages xviii+451 (cited on pages 41–48, 50, 52–55, 57, 71).

[60] R. T. Rockafellar and R. J.-B. Wets. *Variational analysis.* Volume 317. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Springer-Verlag, Berlin, 1998, pages xiv+733. URL: https://doi.org/10.1007/978-3-642-02431-3 (cited on pages 68, 115).

[61] R. T. Rockafellar. “Convex Functions and Dual Eextremum Problems”. PhD thesis. Harvard University, Cambridge, Massachusets, 1963 (cited on page 115).

[62] F. Rosenblatt. *The perceptron: A theory of statistical separability in cognitive systems.* Cornell Aeronautical Laboratory, Inc., Rep. No. VG-1196-G-1. U.S. Department of Commerce, Office of Technical Services, PB 151247, 1958, pages xii+262 (cited on page 12).

[63] F. Rosenblatt. *Two theorems of statistical separability in the perceptron.* Cornell Aeronautical Laboratory, Inc., Rep. No. VG-1196-G-2. U.S. Department of Commerce, Office of Technical Services, PB 151247 S, 1958, pages iii+42 (cited on page 12).

[64] W. Rudin. *Principles of mathematical analysis.* Third. International Series in Pure and Applied Mathematics. McGraw-Hill Book Co., New York-Auckland-Düsseldorf, 1976, pages x+342 (cited on page 71).

[65] W. Rudin. *Real and complex analysis.* Third. McGraw-Hill Book Co., New York, 1987, pages xiv+416 (cited on page 6).

[66] R. E. Schapire. “The Strength of Weak Learnability (Extended Abstract)”. In: *30th Annual Symposium on Foundations of Computer Science, Research Triangle Park, North Carolina, USA, 30 October - 1 November 1989.* IEEE Computer Society, 1989, pages 28–33. URL: https://doi.org/10.1109/SFCS.1989.63451 (cited on page 17).

[67] S. Shalev-Shwartz. "Online Learning and Online Convex Optimization". In: *Foundations and Trends® in Machine Learning* 4.2 (2011), pages 107–194. URL: http://dx.doi.org/10.1561/2200000018 (cited on pages 3, 4, 10, 12, 13, 15, 38, 62, 78, 98, 108, 177).

[68] S. Shalev-Shwartz. "Online Learning: Theory, Algorithms, and Applications". PhD thesis. The Hebrew University of Jerusalem, 2007 (cited on pages 3, 10, 69, 78).

[69] S. Shalev-Shwartz and Y. Singer. "A primal-dual perspective of online learning algorithms". In: *Machine Learning* 69.2-3 (2007), pages 115–142. URL: http://dx.doi.org/10.1007/s10994-007-5014-x (cited on pages 4, 31, 78).

[70] L. G. Valiant. "A Theory of the Learnable". In: *Commun. ACM* 27.11 (1984), pages 1134–1142. URL: http://doi.acm.org/10.1145/1968.1972 (cited on page 24).

[71] V. Vapnik. "An overview of statistical learning theory". In: *IEEE Trans. Neural Networks* 10.5 (1999), pages 988–999. URL: https://doi.org/10.1109/72.788640 (cited on pages 3, 10, 22).

[72] M. Zinkevich. "Online Convex Programming and Generalized Infinitesimal Gradient Ascent". In: *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*. 2003, pages 928–936. URL: http://www.aaai.org/Library/ICML/2003/icml03-120.php (cited on pages 112, 171).