Sequential Decision Making in Non-stochastic Environments

Jacob Abernethy

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2012-25
http://www.eecs.berkeley.edu/Pubs/TechRpts/2012/EECS-2012-25.html

February 10, 2012

Sequential Decision Making in Non-stochastic Environments

by

Jacob Duncan Abernethy

A dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Sciences

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:

Professor Peter Bartlett, Chair
Professor David Ahn
Professor Christos Papadimitriou
Professor Satish Rao

Fall 2011

Sequential Decision Making in Non-stochastic Environments

Jacob Duncan Abernethy

Abstract

Sequential Decision Making in Non-stochastic Environments

by

Jacob Duncan Abernethy

Doctor of Philosophy in Computer Sciences

University of California, Berkeley

Professor Peter Bartlett, Chair

Decision making is challenging, of course, because the world presents itself with a generous portion of uncertainty. In order to have a model for any decision problem we must characterize this uncertainty. The typical approach is a statistical one, to imagine that the events in the world are generated randomly according to an as-of-yet-unknown probability distribution. One can then formulate the decision problem through the lens of estimating the parameters of this distribution or by learning its properties.

This dissertation shall focus on a rather different model, one in which the world is assumed to be *non-stochastic*. Rather than aim for a guarantee that holds "in expectation" or "with high probability," requiring obvious stochasticity assumptions, we instead strive for guarantees that hold "no matter what happens," even under the worst conditions. This approach, while seemingly pessimistic, leads to surprisingly optimistic guarantees when we appropriately tailor the goal of the decision maker. The main objective we consider is that of *regret* which, roughly speaking, describes the difference between the decision maker's cost minus the cost of the best decision *with the benefit of hindsight*.

We give an overview of the aforementioned non-stochastic framework for decision making. We present a generic problem, known as *Online Linear Optimization* (OLO), and prove upper and lower bounds. We consider the *bandit* version of the problem as well, and give the first known efficient algorithm achieving an optimal regret rate. We then show a strong connection between a classic result from the 1950s, known as Blackwell's Approachability Theorem, and the OLO problem. We also look at the non-stochastic decision problem as a repeated game between a Player and Nature, and we show in two cases that the Player's optimal minimax strategy is both easy to describe and efficiently computable.

Contents

Curriculum Vitæ

Jacob Duncan Abernethy

Education

2002	Massachusetts Institute of Technology B.S., Mathematics
2006	Toyota Technological Institute at Chicago M.S., Computer Science
2011	University of California, Berkeley Ph.D., Electrical Engineering and Computer Science

Personal

Born	July 14, 1981 Jackson, Missippi, USA

Chapter 1

Introduction

> Regrets, I've had a few; but then again, too few to mention.
> I did what I had to do and saw it through without exemption.
> I planned each charted course, each careful step along the byway.
> And more, much more than this, I did it my way.
>
> Frank Sinatra

When presented with a problem, how ought we decide what to do? When we have encountered this problem previously, how should our experience guide us? After one or many decisions, what is the best way to measure our performance, to determine success or failure?

In short, the present dissertation is about sequential decision making. More specifically, we shall focus on decision-making models in which the actor in question wants to avoid making statistical assumptions on nature of his observations. Before diving into details we shall tell a story. This story is about a man named Al.

1.1 Al's Dilemma

Let us begin by imagining the following very simple world.

> *A man, who we shall call Al, one day wakes up to find himself in a room. The room contains a single locked door but is completely empty, save for a collection of buttons on the wall with the following configuration.*
>
A	B	C	D	E	F	G
●	●	●	●	●	●	●
>
> *Above the buttons, there is a sign that reads "YOU MUST PRESS ONE BUTTON BEFORE EXITING." Al stares at the buttons, pondering his situation.*

What should Al do?

Putting aside the peculiar scenario in which Al now finds himself, the dilemma is entirely familiar: "I'm in a world that I don't entirely understand, yet I have to make a decision with a high level of uncertainty." Who has not encountered such a scenario thousands of times throughout one's life?

> *Not entirely sure what to do, Al selects one of the buttons, the one labeled* ***A****, and pushes it. Al pauses a moment, but the room remains silent.*

We would now like to address the following question: did Al make "the right" decision? Put more broadly, we want to know, what is a "good" approach to decision-making in uncertain environments? This question is, of course, unanswerable without some amount of *feedback* associated with the chosen action. As far as Al is concerned, the decision problem is entirely opaque without more information.

> *While Al stares at the locked door, a loud BEEP occurs from somewhere in the room, and then, under each button, a number appears.*

A	*B*	*C*	*D*	*E*	*F*	*G*
●	●	●	●	●	●	●
5	*0*	*2*	*20*	*3*	*0*	*8*

> *Immediately after the BEEP the locked door opens and Al, relieved that he is no longer stuck in the empty room, proceeds to exit. As he leaves, he is handed an envelope with the letter* ***A*** *written on the front. Somewhat puzzled, Al opens the envelope to find that it contains exactly $5.*

We can imagine that Al is rather content at this moment. He has found himself in a totally unfamiliar situation, been presented with an array of options and, knowing nothing of what mechanism lies behind any of these buttons, has managed to receive five bucks for his choice. Heck, it appears that if he had simply chosen one button to the right, he wouldn't have received anything! There's just this one thing...

> *Satisfied with his accomplishment, Al pockets the money and proceeds to exit the building. But he hesitates, thinking to himself, "But wait, if only I had pressed button* ***D****, I would have gotten twenty bucks!" So Al turns around to see if he can sneak in another button press or two, but the door is shut. Only mildly disappointed, Al departs and goes about his day as if nothing had happened.*

Al is now in a much more familiar situation: after making his decision, at a moment with very little certainty about the nature of the problem, he observes the outcome and now considers, with the benefit of hindsight, the quality of his decision relative to the alternatives. Unfortunately this analysis is totally hypothetical, as his decision is irrevocable. (Indeed, we might call this the *detriment* of hindsight, for it can only cause us to *regret* our decisions.)

Al's experience in the room with the buttons has hardly ended. The following morning, Al again awakes inside the mysterious room, and is presented with the same decision problem: press one of the buttons to exit the room. After selecting one, he observes the payoffs associated with each button and, as he leaves, receives the payoff for his selection. The same occurs on the third morning. And the fourth. And fifth. By this point, he has carefully recorded the values displayed below each button on every day. This is what he has observed:

	A	*B*	*C*	*D*	*E*	*F*	*G*
day1	*5*	*0*	*2*	*20*	*3*	*0*	*8*
day2	*1*	*0*	*2*	*0*	*8*	*1*	*1*
day3	*9*	*0*	*2*	*0*	*9*	*1*	*3*
day4	*5*	*0*	*2*	*0*	*9*	*0*	*2*
day5	*5*	*0*	*2*	*10*	*0*	*1*	*10*

Al looks at the data and ponders, again, which button to press.

The problem has suddenly become a lot more interesting. On the first day, Al had an entirely arbitrary set of options available to him. As far as Al was concerned, button **A** was indistinguishable from button **C** but for the label. Yet we now realize that the same decision problem is being presented to our hero on each of a potentially long sequence of days. Before making a decision on any given day, we expect that Al will have *learned* from the dataset he has been collecting.

1.2 Abstract Decision Making and Al's Dilemma

The world in which Al finds himself is plainly a fantasy yet, as an abstraction, the problem he faces could not be more realistic. On each of a sequence of time periods he is presented with a decision from a fixed menu. He must select one such decision and, upon making this decision, is presented with some feedback regarding the outcome of this decision. On each round the decision maker can utilize information gleaned from past decisions.

When does this type of sequential decision problem arise? It does not take long to find pertinent examples, such as:

- an investor, making a sequence of trading decision, while observing price fluctuations;
- a gambler, choosing which horse to bet on, while observing each race;
- a caching algorithm, choosing which memory elements to evict, while observing memory requests;
- a weather forecaster, predicting rain or shine, while observing the outcome the following day;

- a driver, choosing which route to work each day, while observing the traffic on the roads.

Let us formalize this abstract problem at hand. We will imagine that Al's decisions (actions, strategies, etc.) come from some fixed decision set D. For now we will imagine that D is arbitrary, but typically we will assume that D is either finite or comes from some linear space in $\mathbb{R}^n$ or, at the very least, some Hilbert space. Al then observes some "outcome" that is revealed by Nature, and we shall assume this outcome is an element of a fixed outcome set Z. We will imagine that this sequential decision problem occurs on some number of rounds, and that on round t Al can choose d_t as a function of $z_1, \ldots, z_{t-1}$—naturally, as these outcomes are the only ones thus revealed by Nature.

Al, of course, has some *objective* in mind when making his decisions. If this is a purely financial problem, Al may simply want to maximize his total wealth. If the problem involves some uncertainty (which it inevitably will), Al may also want to encode a notion of risk aversion into the objective. In any case, we shall assume that Al has a pre-determined real-valued function $\text{Objective}(d_1, z_1, \ldots, d_T, z_T)$ which allows him to measure his decisions $d_1, d_2, \ldots$ given the sequence of observed actions $z_1, z_2, \ldots$.

In perhaps the simplest case, we can imagine that the objective function is *cumulative*, in the sense that it can be written as

$$\text{Objective}(d_1, z_1, \ldots, d_T, z_T) = \sum_{t=1}^{T} \ell(d_t, z_t)$$

where $\ell : D \times Z \to \mathbb{R}$ is some arbitrary cost function. Let us note that an objective that can be written this way encodes several very strong assumptions about the problem at hand. First, the decision maker's cost can be separated into a sum of costs over the sequence of days, and a given day's cost depends only on the events d_t, z_t associated with that day. Moreover, the per-day cost function $\ell(\cdot, \cdot)$ is constant throughout.

It is worth noting that cumulative objectives have historically been very popular in the literature. This is perhaps less surprising when we look at the list of examples above. Our earnings from betting on the stock market, for example, are cumulative[1]. Or when a forecaster predicts the weather she is judged, in aggregate, according to her prediction on the day relative to the outcome on that day—it would be meaningless, of course, to compare her prediction today to yesterday's weather.

Now that we have specified what Al wants to achieve, we must also specify *how* he shall achieve it. We say that $\mathcal{A}$ is a *deterministic decision-making algorithm* (or, more frequently, we will simply use the term *algorithm*) if it provides a set of maps $\{\mathcal{A}_1, \mathcal{A}_2, \ldots\}$, where $\mathcal{A}_t$ has the form

$$\mathcal{A}_t : \overbrace{Z \times Z \times \cdots \times Z}^{t-1 \text{ copies}} \to D$$

[1]Depending on the setting, it may be better to think about cumulative performance of an investing strategy as *multiplicative* rather than *additive*, since we can reinvest prior returns. On the other hand, a common technique to convert the former to the latter is to take the logarithmic of the multiplicative returns.

for all nonnegative integers t. A *randomized* decision-making algorithm is one that provides maps of the form $Z \times Z \times \cdots \times Z \to \Delta(D)$, where $\Delta(D)$ is the set of distributions over D.

1.3 Who is Nature? A Statistical Approach

Let us return our attention to the dilemma that Al faces on each day, namely, "Given what I know up until today, what button shall I press?" Let us imagine that Al comes up with a given strategy, and let us imagine that this strategy instructs Al to select button X. It is natural to ask, is button X the best choice available? Indeed, what do we even mean by "the best" in this situation? This could be the button that has the lowest risk. Alternatively, we could look for the button with the highest payoff in expectation. But what does "in expectation" mean when we have not settled on a notion of randomness?

This brings us to a key philosophical question that one must address in any decision problem. Let us use the term *Nature* to refer to the entity which produces the uncertain outcome associated with each decision. In Al's case, we shall assume that the rewards associated with each button are chosen by Nature. More generally, we may think of any source of data as being produced by Nature. We must ask ourselves, what or who is Nature, and how shall we model its behavior?

Let us discuss two *statistical* settings. It is not surprising that we would look to the statistics to look for advice in solving sequential decisions problems, as a central question of the field is how to interpret data in a meaningful way. The dilemma Al faces is precisely the problem of how to leverage past observations on future decisions.

1.3.1 Bayesian Assumptions

We shall now work under the assumption that the data is *independently and identically distributed* (i.i.d.); that is, where Nature selects the outcomes z_t by independently drawing each according to a single (fixed) distribution P_θ. We use the symbol θ to refer to the *parameters* of the distribution, and we imagine that θ is a member of some parameter space Θ. We can go a step further and include the assumption that θ was chosen from some *prior distribution* $\pi \in \Delta(\Theta)$. This is often called the Bayesian setting as it allows us to apply Bayes Rule in order to reason about the conditional probability of a hypothesis given some data.

Let us consider a cumulative objective function, so that Al wants to minimize the sum $\sum_t \ell(d_t, z_t)$ for some given loss function $\ell(\cdot, \cdot)$. Under this Bayesian setting, we can write down explicitly what Al can ultimately achieve:

$$\text{Optimal Cost} \quad = \quad \inf_{\text{algs. } \mathcal{A}} \mathop{\mathbb{E}}_{\theta \sim \pi} \left[\sum_{t=1}^{T} \mathop{\mathbb{E}}_{z_t \sim P_\theta} [\ell(d_t, z_t)] \right], \tag{1.1}$$

where each decision d_t is selected as the output of $\mathcal{A}_t(z_1, \ldots, z_{t-1})$.

Let us spend a moment to consider an intuitive interpretation of the above expression. We imagine that, before any actions are taken by either Al or Nature, Al must commit to some algorithm $\mathcal{A}$ amongst a large class of such algorithms. In particular Al's goal is to attain the small cost possible, hence the search for an infimal value. Once this algorithm is selected, it is Nature's turn to select the parameter θ according to the fixed and known prior distribution π. With the data distribution P_θ now chosen the sequential game proceeds. Al queries $\mathcal{A}_1(\emptyset)$ to obtain d_1, Nature samples an outcome $z_1 \sim P_\theta$, and Al suffers $\ell(d_1, z_1)$. On the following round Al selects $d_2 \leftarrow \mathcal{A}_2(z_1)$, Nature samples $z_2 \sim P_\theta$, Al suffers $\ell(d_2, z_2)$. And so on.

There is a fundamental philosophical principle expressed in (1.1): the decision maker is concerned entirely with the *expected* cost of the objective, with respect to the set of random choices made by Nature. This principle is pervasive throughout much of the literature on decision theory, but it is worth stepping back to ask "Why?" After all, if Al will have but one chance to play this game then what good is the average case? Here is a potential answer a decision theorist could give:

> Assume our actions will be measured as a function of some uncertain variable X, and that to the best of our knowledge X assumes one of a range of states, each having an associated known probability. Without further information, we may as well imagine that X assumes *every* such state, weighted by the probability value. Thus, in order to make a utility calculation over this range of states, our only option is to perform a weighted average and integrate over the distribution.

This is, of course, a philosophical explanation to a philosophical question, although it is a reasonably compelling answer. We shall proceed now under the principle of "maximize expected utility" but we shall return to this question soon.

We now return our attention to the objective in (1.1). By using the linearity of expectation, and the fact that $\mathcal{A}$ is defined by a set of independent algorithms $\mathcal{A}_1, \mathcal{A}_2, \ldots$ we can rewrite

$$\text{Optimal Cost} \quad = \quad \sum_{t=1}^{T} \inf_{\mathcal{A}_t} \mathop{\mathbb{E}}_{\theta\sim\pi} \mathop{\mathbb{E}}_{z_1,\ldots,z_t\sim P_\theta} \left[\ell(\mathcal{A}_t(z_1,\ldots,z_{t-1}), z_t)\right]. \tag{1.2}$$

For the time being, let P^π be the marginal distribution of the joint outcomes $z_1, \ldots, z_{t-1}$ and let $\pi|z_{1:t-1}$ be the *posterior distribution*, i.e. the distribution π over Θ conditioned on having observed the samples $z_1, \ldots, z_{t-1}$. Using Bayes rule we can see that

$$\inf_{\mathcal{A}_t} \mathop{\mathbb{E}}_{\theta\sim\pi} \mathop{\mathbb{E}}_{z_1,\ldots,z_t\sim P_\theta} \left[\ell(\mathcal{A}_t(z_1,\ldots,z_{t-1}), z_t)\right] = \mathop{\mathbb{E}}_{(z_1,\ldots,z_{t-1})\sim P^\pi} \inf_{d_t\in D} \mathop{\mathbb{E}}_{\theta\sim\pi|z_{1:t-1}} \mathop{\mathbb{E}}_{z_t\sim P_\theta} [\ell(d_t, z_t)].$$

The right hand side of this final equation brings us to our main point: the optimal algorithm in the Bayesian setting is to (a) compute the posterior distribution on Θ given the observed data $z_1, \ldots, z_{t-1}$ and (b) to minimize the expected cost according to this distribution. This statement will come as no surprise to those familiar with Bayesian statistics as this technique already has a well-known name, *maximum a posteriori* (MAP) estimation.

Summarizing the above discussion, we have considered Al's dilemma under the condition that he is willing to make the i.i.d. assumption on Nature and, furthermore, is willing put faith in a given prior π. Al, an unabashed Bayesian, can now consider his decision-making task solved, for he has in front of him a simple prescription: select decision $d_t \in D$ with the goal of maximizing *a posteriori* utility. Indeed, Al can sleep well at night.

1.3.2 A Frequentist Approach

We finished the previous section by noting that, as long as the decision maker can commit to a prior distribution π over the parameter space Θ, the decision-making problem can be reduced to (a) computing the a posterior distribution and then (b) maximizing expected utility with respect to this distribution. But what if we are unwilling to commit to any particular prior? Or worse, what if we commit to a prior distribution yet discover that it is far from accurate, only to have paid a large cost in the process?

In this "frequentist" paradigm, we still imagine that our data is selected according to a fixed probability distribution P_θ, and that each z_t is drawn (independently) from this distribution, but we stop short of attempting to answer the question "Where did this magical P_θ come from?" This is a key distinction with the Bayesian approach, since without a prior we can not meaningfully have a conversation about the "probabilities over states θ of the world." Indeed, within a frequentist setting we can even go so far as to imagine that θ was selected by an adversary.

This is not quite as pessimistic as it sounds. Let us consider a very simple example: mean estimation. Al's decision set, as well as Nature's outcome set, will be the unit ball $B \subset \mathbb{R}^n$ for some positive integer n. The loss will be the squared error, $\ell(\mathbf{x}, \mathbf{z}) := \|\mathbf{x} - \mathbf{z}\|^2$. Under this loss function, we are effectively solving the problem of "estimating the mean" of P_θ because the minimization problem is solved as $\arg\min_{\mathbf{x} \in B} \mathbb{E}_{\mathbf{z} \sim P_\theta} \|\mathbf{x} - \mathbf{z}\|^2 = \mathbb{E}_{P_\theta}[\mathbf{z}]$. Let us consider a simple decision-making algorithm, simply predicting the empirical mean thus far,

$$\mathbf{x}_t \leftarrow \mathcal{A}_t(\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}) = \frac{1}{t-1} \sum_{s=1}^{t-1} \mathbf{z}_s.$$

How does this algorithm perform? It is easy to show the following bound

$$\sum_{t=1}^{T} \ell(\mathbf{x}_t, \mathbf{z}_t) \leq T \mathrm{var}(P_\theta) + c \log T \tag{1.3}$$

for some universal constant $c > 0$. Also, it is worth noting that one can also establish a matching upper bound.

What does this statement tell us? Put another way, this bound says that there exists some algorithm (empirical averaging, in particular) such that for *any* distribution $P_\theta \in \Delta(B)$ our objective will be no more than the RHS of (1.3). This can be stated in terms of a minimax

bound:

$$\inf_{\mathcal{A}} \sup_{\theta \in \Theta} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{z}_t \sim P_\theta} \ell(\mathcal{A}_t(\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}), \mathbf{z}_t) \leq T \sup_{\theta \in \Theta} \mathrm{var}(P_\theta) + c \log T = T + c \log T. \tag{1.4}$$

We have expressed the above bound as a sum of two quantities for good reason. The first term, $T\mathrm{var}(P_\theta)$, is exactly the loss suffered by the decision maker *even with knowledge of* θ. In other words, even under optimal play one can not avoid paying at least some cost, since $\min_{\mathbf{x}} \mathbb{E}_{P_\theta} \|\mathbf{x} - \mathbf{z}\|^2 = \mathrm{var}(P_\theta)$ which must be positive for any non-degenerate distribution. With this in mind it is worth asking whether we should even count this value towards Al's objective—should his performance even factor in this minimum cost of playing the game? Would it not be more reasonable to consider his cost *compared to* some hypothetical omniscient player? This suggests a modified objective function:

$$\text{Objective}(\mathcal{A}|\theta) := \sum_{t=1}^{T} \mathop{\mathbb{E}}_{\mathbf{z}_t \sim P_\theta} \ell(\mathcal{A}_t(\mathbf{z}_1, \ldots, \mathbf{z}_{t-1}), \mathbf{z}_t) - T \min_{\mathbf{x} \in B} \mathop{\mathbb{E}}_{\mathbf{z} \sim P_\theta} \ell(\mathbf{x}, \mathbf{z}). \tag{1.5}$$

In a statistics problem, where $\ell(\cdot, \cdot)$ is some estimator loss, this objective is often referred to as the *estimation error*. We have proven a *relative loss bound* if we can show how to control this objective.

1.4 The Competitive Decision Framework

In the previous section, we discussed a range of statistical approaches to sequential decision making. We showed that, under a Bayesian setting where the decision maker has faith in a prior distribution, the "optimal" algorithm, while not necessarily computationally feasible, is always simple to declare in terms of posterior probabilities. In a frequentist setting, we do not have such a canonical algorithm, but we can certainly propose algorithms (estimators) and provide robust guarantees.

However, by taking a statistical view we are adhering to a particular principle, laid out implicitly in the i.i.d. assumption, about the nature of our observations and our expectations. We may state this explicitly:

> **Uncertainty-as-Randomness (UAR) Principle:** Each observation we receive is the result of a draw from a probability distribution; this distribution has a fixed set of parameters; hence, previous observations ought to inform us about future outcomes; finally, we ought to view the decision problem through the lens of estimation.

The UAR Principle is quite broad and is implicit in a vast majority of the literature in not only statistics but economics, information theory, computer science, etc., as well. Indeed, it's crucial in our everyday lives, as it gives us the ability to *generalize*, to make statements

of the form "I can't see the future, I have a finite dataset, but I still strongly believe X to be true." The UAR principle is what leads us to believe the weatherman when we're told that a hurricane is approaching—even when we know he is not clairvoyant.

In the present section, and indeed in the rest of this document, we shall be dispensing with the Uncertainty-as-Randomness principle.

Why would we want to do that, when it has proven to be so powerful? The problem is that the UAR principle collides with what you might call the Real World Principle: *in a large number of scenarios of interest, our data (observations) are often not generated randomly but are instead created from the actions of other agents, each optimizing for their own objectives and making their own informed decisions.* We can consider the most extreme example of this: a two-player zero-sum game. Against a savvy opponent, why should we expect previous actions to predict future behavior? How do we know we're not being tricked?

This is not intended to be a full-throated critique of the UAR principle, nor do we suggest that one should never take a statistical viewpoint. On the contrary, interpreting uncertainty through the lens of randomness and estimation can be very powerful in vast number of problems—weather prediction being a prime example. But we emphasize that these approaches are only robust up to the validity of their assumptions. When we want to know if it will rain tomorrow, treating weather conditions as a finitely parameterized random process would seem perfectly reasonable. When we want to predict if a given email is legitimate or spam, it seems less useful to treat the email's creator as a random process.

We will spend the rest of this section laying out a model for sequential decision making in the absence of statistical assumptions. We shall begin in §1.4.1 with an illustrative example, returning back to Al's simple dilemma. We will follow that up in §1.4.2 with a discussion of the history of this framework, and we mention some of the key results in context. We will spend the entirety of §1.5 surveying the various works laid out in the remainder of this dissertation.

1.4.1 A Warmup Example

Let us look back at Al's dilemma which we discussed at the outset of the present Chapter. On each of a sequence of days, Al is presented with a fixed set of buttons, and he may push only one. After having pressed one such button, he receives a hidden reward associated with this particular button. In addition Al also learns the rewards associated with all of the alternatives, noting what he might have earned, for better or worse.

We shall now switch terminology for the remainder of this chapter and use the term *expert* rather than button. This should seem an odd replacement, since typically we do not typically associate the action of "pushing" with the concept of "expert", nor do experts typically return "rewards". Nevertheless this term has become quite standard in the literature, and indeed it is common to use the phrase *expert setting* to describe the abstract problem Al faces; the historical precedent will be described soon (§1.4.2).

In the expert setting, we imagine an algorithm $\mathcal{A}$, playing the role of Al, that must make

a sequence of predictions from the set $[N] := \{1, 2, \ldots, N\}$, where each index i corresponds to an expert. It is more generally assumed that the algorithm shall make randomized predictions, choosing on every round t a distribution $\mathbf{w}_t \in \Delta_N$. We shall refer to $\mathbf{w}_t[i]$ as a *weight* for expert i. After $\mathcal{A}$ commits to $\mathbf{w}_t$, we imagine that Nature assigns a *loss* to each expert, represented by a vector $\boldsymbol{\ell}_t$, where the loss for expert i is some bounded real value $\boldsymbol{\ell}_t[i]$. Whereas Al's problem involved a *gain* on each round, one generally works with losses in the expert setting, and it is usually assumed that the loss values lie in $[0, 1]$. On day t, $\mathcal{A}$ will sample an expert according to the distribution $\mathbf{w}_t$, and will suffer the expected loss according to $\mathbf{w}_t$; that is, $\mathbb{E}_{i \sim \mathbf{w}_t} \boldsymbol{\ell}_t[i] = \mathbf{w}_t \cdot \boldsymbol{\ell}_t$.

Now we can formally recast Al's dilemma: how ought he select $\mathbf{w}_t$ having observed $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_{t-1}$? This question, of course, depends heavily on the objective function that he would like to optimize. But it also depends very much on his view of Nature: is Nature an oblivious random process, or is Nature an adversary? We shall imagine that Al is a skeptical individual and that Al would not want to put faith in any statistical assumptions (bayesian, i.i.d., etc.). How should our skeptic proceed?

The simplest objective he might like to minimize is the cumulative loss, $\sum_{t=1}^{T} \mathbf{w}_t \cdot \boldsymbol{\ell}_t$, yet unfortunately this objective is not well suited for such pessimistic assumptions. For one, if Nature is indeed an Adversary, then Nature can simply choose $\boldsymbol{\ell}_t = \langle 1, 1, \ldots, 1 \rangle$ on every round, thus ensuring that $\sum_{t=1}^{T} \mathbf{w}_t \cdot \boldsymbol{\ell}_t = T$ independent of Al's choices. And even considering this sequence of outcomes, Al would presumably think to himself, "Well this actually isn't so bad, I couldn't possibly have done any better no matter how I acted!" Indeed, even under a hypothetical i.i.d. assumption on the sequence $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T$, Al could only conclude that the true distribution P_θ on the loss vectors was really lousy — Even if he'd known this hypothetical θ he could not have improved.

The previous discussion brings out a key point: we should always control for the "best case scenario comparison"—the decision maker should never be able to say "yeah but even if...". We have already alluded to this issue in equation (1.5) and the discussion preceding it. In a frequentist paradigm it is quite natural to pose the following question, "what is our performance relative to what it would have been if we knew exactly the true parameters of the distribution?" Taking this viewpoint, we could propose that Al focuses on the following objective:

$$\mathop{\mathbb{E}}_{\{\boldsymbol{\ell}_t\} \sim P_\theta} \left[\sum_{t=1}^{T} \mathbf{w}_t \cdot \boldsymbol{\ell}_t \right] - T \min_{i \in [N]} \mathop{\mathbb{E}}_{\boldsymbol{\ell} \sim P_\theta} \boldsymbol{\ell}[i].$$

In other words, an option is to recommend that Al consider his expected performance over the course of the sequential problem *minus* the performance of the optimal strategy that knows the distribution P_θ (represented by $\min_i \mathbb{E}\, \boldsymbol{\ell}[i]$). This is a perfectly reasonable proposal, except for one flaw: this makes no sense to Al, who is a skeptic and does not believe in the a notion of a "true distribution" P_θ. Al believes only what he sees, and after T rounds all Al has seen is T vectors, $\boldsymbol{\ell}_1$ up through $\boldsymbol{\ell}_T$.

Our goal is clear: we must find an objective that aims at minimizing the cumulative loss, that controls for the "best in hindsight" comparison, but avoids appealing to "the true distribution" or similar such notions. This brings us to the objective which is now commonly

known as *regret*:

$$\text{Regret}_T(\mathcal{A}) := \sum_{t=1}^{T} \mathbf{w}_t \cdot \boldsymbol{\ell}_t - \min_{\mathbf{w} \in \Delta_N} \sum_{t=1}^{T} \mathbf{w} \cdot \boldsymbol{\ell}_t. \tag{1.6}$$

Notice that this quantity does not require any assumptions on the process that generated the vectors $\boldsymbol{\ell}_t$, it is an entirely empirical measurement of performance. At the end of any sequence of decisions, Al can ask himself "What's my regret so far?" and can compute this exactly. This is not true for the statistical estimation error, which requires considering a hypothetical data distribution. When we needn't consider the data generating process, we can go a step further and discuss the *worst case regret* of an algorithm, that is the largest value of (1.6) over all possible sequences $\{\boldsymbol{\ell}_t\}_{t=1}^T$. Indeed, at this point we may as well imagine that an adversary chooses the sequence with the single goal of inflicting maximal regret on the Al.

A first glance at this might appear to be a major burden; need we really be robust to an adversarially chosen sequence of data? Of course one would like to take a pessimistic view of Nature as we would have incredibly robust guarantees ("no matter what happens..."). The surprising fact is that *we can actually prove such robust guarantees*. Precisely, we shall be able to prove statements of the followin form:

$$\exists \mathcal{A} : \qquad \lim_{T \to \infty} \frac{1}{T} \text{Regret}_T(\mathcal{A}) \to 0$$

This statement is quite strong, and it is worth summarizing in words:

> There is some algorithm whose average performance on any sequence of data is eventually no worse than the performance of the best fixed decision in hindsight.

1.4.2 History

We shall begin this section with a review of the history of these universal/adversarial prediction models, starting with Hannan and Blackwell in the 1950s and leading to the present. We will then spend some time summarizing the contributions of the present document and how they fit into what has been known.

It was most likely James Hannan who originally proposed a distribution-free framework for sequential decision making. In his seminal paper "Approximation to Bayes Risk in Repeated Play" (1957, [43]), he makes a clear case for the proposed approach.

> The present paper is concerned with a sequence of N decision problems, which are formally alike except for the fact that the state of nature may vary arbitrarily from problem to problem. Decisions are required to be made successively and it is assumed that they may be allowed to depend on the e.d. (empirical distribution) of the states of nature across the previous problems in the sequence. This total lack of assumptions regarding the behavior of the state sequence is a feature distinguishing the present structure from many considerations of multistage processes...

> The most important conclusion of this paper is that the knowledge of the successive e.d. of the past states makes it constructively possible to do almost as well at reducing the average inutility across problems as in the case where N and the distribution of the N states are known in advance.

The sequential decision game Hannan proposed is identical to Al's dilemma, and the goal of "reducing the average inutility across problems" relative to when the distribution of the "states are known in advance" is exactly that of minimizing average regret.

In 1956, around the same period, David Blackwell proved a result now known as the Approachability Theorem. We shall address this theorem in detail in Chapter 5, but we give a quick summary here. Blackwell's theorem concerned the problem of playing a repeated game with a vector-valued payoff function. The goal proposed by Blackwell was to determine under what circumstances we can design an adaptive strategy for player 1 so that the average payoff vector will "approach" a given convex set; hence the term "approachability". Blackwell gave a precise necessary and sufficient condition for when a set is approachable. Soon thereafter he showed that this approachability result can be used to construct, as a special case, Hannan's regret minimization strategy.

Between 1960 and 1988 not a great deal was published on the subject of sequential prediction and decision making under worst-case assumptions. But interest in the topic surged in the late 80s and throughout the 90s. There is the early work by Littlestone [52] from 1988 on learning linear threshold functions, Vovk's work on so-called "aggregating strategies" [73] which appeared in 1990, work by Foster [36] from 1991, and Cover's 1991 work on universal portfolios [25]. The notion "learning from experts" (similarly, "combining expert advice") was introduced by Littlestore and Warmuth in 1994, and they proposed the "weighted majority algorithm" although variants of this algorithm had been published previously. Following these early papers, there as been a large amount of work in this area over the past 15 years and the list is much too long to give here. Among the major contributors to this area are Peter Auer, Peter Bartlett, Avrim Blum, Nicolo Cesa-Bianchi, Dean Foster, Yoav Freund, Claudio Gentile, Sergiou Hart, Elad Hazan, David Helmbold, Adam Kalai, Satyen Kale, Nick Littlestone, Phil Long, Gabor Lugosi, Andreu Mas-Colell, Alexander Rakhlin, Robert Schapire, Yoram Singer, Valdimir Vovk, and Manfred Warmuth. An excellent summary of this area can be found in the book of Cesa-Bianchi and Lugosi [24]. It can not be emphasized enough that the latter book provided the launching point for a great deal of the work presented herein.

1.5 Overview of Results

We will now give a short overview of the results in the remainder of this dissertation.

In Chapter 2, we introduce a very generic sequential decision problem known as *Online Convex Optimization* (OCO). This problem has been relatively well-studied, and we mostly review the literature. We discuss a key algorithm, known as *Follow the Regularized Leader*,

which provides a key tool for later analysis. We prove a very generic bound on this algorithm, setting the stage for a much more interesting analysis in Chapter 4.

In Chapter 3, we focus on lower bounds for the OCO setting. Lower bounds for online learning problems had existed previously, but we prove a number of very precise bounds. In particular, for the particular problems we address, we can provide the precise minimax strategies for both the decision maker and Nature.

In Chapter 4, we consider the *bandit version* of the Online Linear Optimization problem. When we refer to a bandit problem, we mean where the decision maker receives very limited feedback in his decision process. In the full-information version, the decision maker may observe the entire cost function at the end of each round, whereas in a bandit problem the decision maker may only observe the cost of the chosen action. For some time the optimal regret rate was unknown for this problem, as there remained a gap between $O(\sqrt{T})$ and $O(T^{2/3})$. We provide the first *efficient* algorithm to close this gap. A surprising aspect of this result is the technique used: the algorithm requires the use of *self-concordant barrier functions* that have been studied in the optimization community.

In Chapter 5, we look back at a famous result of David Blackwell known as the Approachability Theorem. Within a year of publishing this result, Blackwell observed that his result was connected to Hannan's result in the nascent field of Online Learning. We take this observation much further and show that Blackwell's result is *algorithmically equivalent to* the problem of Online Linear Optimization. We provide a specific reduction from the approachability problem to an OLO problem, and vice versa. We apply this reduction to the problem of achieving "asymptotic calibration" for sequential binary forecasting, and we exhibit the first efficient algorithm that attains the desired asymptotic guarantee.

We switch gears in Chapter 6 and consider the expert setting as a zero-sum multistage game between a gambler and a casino. At each round t, the gambler must choose an action $\mathbf{w}_t$ inside of the N-simplex, the casino must then select a cost vector $\boldsymbol{\ell}_t \in \{0,1\}^N$, and the gambler suffers $\mathbf{w}_t \cdot \boldsymbol{\ell}_t$. We assume that the "loss of the best expert" has a fixed a priori bound, say an integer k, and as soon as the loss of the best expert is greater than k the game must end. Although this game has an exponentially-sized state space, and might appear to be computationally challenging to solve, we can exhibit the minimax strategy for both gambler and casino. The gambler's optimal strategy can be described simply in terms of a random walk.

We build on the latter work in Chapter 7 and consider a more general scenario where a player takes part in a repeated game against an adversary that has a *budget*. In the Gambler/Casino game, the Casino's budget was that "no expert shall suffer more than k losses." But what if the budget is arbitrary, i.e. we have some given function which determines, as a function of the opponent's actions, when the game ends? We show that an the optimal strategy for this game can also be run efficiently, up to an approximation. We also give several examples where this result applies.

Thanks to Co-Authors

Over the last several years I have had the great fortune of working with several excellent collaborators, and I owe a big debt to all of them. All the results presented in this dissertation were previously published across different venues and, for the most part, the presentation herein draws verbatim from these previous works. I will take a moment to reference these works and to thank the various coauthors.

- The content of Chapter 3 was originally published in 2008 in the Proceedings of the Nineteenth Annual Conference on Learning Theory (COLT), with Peter Bartlett, Alexander Rakhlin, and Ambuj Tewari [3].
- Chapter 4 is based on the results published in COLT 2008 with Alexander Rakhlin and Elad Hazan [8]. The version presented here is from the significantly-modified journal version of the COLT paper, where the former is currently under review for IEEE Transactions on Information Theory. A large portion of this journal writeup is concerned with the "full information" Online Linear Optimization problem, and this section makes up the entirety of Chapter 2.
- Chapter 5 is based on work published in COLT 2011 with Peter Bartlett and Elad Hazan [2].
- Chapter 6 draws from work with Manfred Warmuth and Joel Yellin published at COLT 2008 [6].
- Chapter 7 draws from work with Manfred Warmuth published in Advances in Neural Information Processing Systems 2010 [7].

Chapter 2

First Steps: Online Linear Optimization

2.1 Introduction

Should we expect the past to predict the future? And if no relationship between past and future can be assumed, may we still obtain some guarantee on our performance? In essence, this is the problem of *universal prediction*. In the words of Merhav and Feder [58] in 1998, a universal prediction strategy is one that "does not depend on the unknown underlying model and yet performs essentially as well as if the model were known in advance." Since those words were written, the universal framework has received much attention in a range of communities – learning theory, information theory, game theory, optimization – and we refer the reader to the survey by Merhav and Feder [58] as well as to the excellent book of Cesa-Bianchi and Lugosi [24] for a thorough exposition.

Let us now present a general model of prediction. A "learner" observes a sequence of data $f_1, f_2, \ldots$, and must select a sequence of "strategies" $x_1, x_2, \ldots$ chosen from a class $\mathcal{K}$, where x_t may depend on $f_1, \ldots, f_{t-1}$. For every pair (x_t, f_t), there is a real-valued loss $\ell(x_t, f_t)$ charged to the learner for playing x_t when the data was f_t. In the universal prediction framework, the learner hopes to perform nearly as well as the best strategy in hindsight; after T observations, this is precisely $\inf_{x \in \mathcal{K}} \frac{1}{T} \sum_{t=1}^{T} \ell(x, f_t)$. This general framework is identical to that presented by Merhav and Feder [57], and is the foundation of the results in Cesa-Bianchi et al. [21] and Haussler et al. [45].

The results in this chapter shall focus on the universal prediction framework when (a) the set of strategies $\mathcal{K}$ forms a *convex set*, (b) the loss $\ell(\cdot, \cdot)$ is *linear*, and (c) the learner only receives *limited feedback* – that is, partial information – about the data. Let us now give a precise definition of the model. Imagine a sequential game $\mathcal{G}(\mathcal{K}, \mathcal{F})$ between the learner (algorithm) and the environment (adversary) where, for each $t = 1$ to T,

- Player chooses $\mathbf{x}_t \in \mathcal{K} \subset \mathbb{R}^n$
- Adversary independently chooses $\mathbf{f}_t \in \mathcal{F} \subseteq \mathbb{R}^n$

- Player suffers loss $\mathbf{f}_t^\top \mathbf{x}_t$ and observes feedback $\mathscr{F}_t$.

To produce a "good" algorithm for this particular problem, we have to fix a yardstick by which to measure an algorithm. In the decision-theoretic framework, it is common to measure the performance of the prediction method through the notion of *regret*. That is, the goal of the Player is to minimize her total loss relative to the best fixed action in $\mathcal{K}$:

$$\text{Regret}_T := \sum_{t=1}^{T} \mathbf{f}_t^\top \mathbf{x}_t - \min_{\mathbf{x}^* \in \mathcal{K}} \sum_{t=1}^{T} \mathbf{f}_t^\top \mathbf{x}^*. \tag{2.1}$$

In particular, we are interested in rates of increase of R_T in terms of T. Without diving into the details just yet, we mention that $\Theta(T^{1/2})$ is the minimax optimal rate in many instances of sequential prediction with linear loss. In the present chapter, we provide an associated upper bound of the form $O(T^{1/2})$ and in Chapter 3 we prove associated lower bounds.

While the adversary chooses $\mathbf{f}_t$ based on $\{\mathbf{x}_s\}_{s=1}^{t-1}$, we assume that the feedback sequence $\mathscr{F}_t$ is the only information provided to the player about the adversary's moves. We distinguish two types of feedback. The *Full Information* setting corresponds to $\mathscr{F}_t = \mathbf{f}_t$, while the *Bandit* setting corresponds to $\mathscr{F}_t = \mathbf{f}_t^\top \mathbf{x}_t$. The latter type of feedback owes its name to the famous *multi-armed bandit* problem, which can be seen as an instance of $\mathcal{G}(\mathcal{K}, \mathcal{F})$ with $\mathcal{K}$ being the n-simplex. The Bandit feedback relays the cost of the chosen decision $\mathbf{x}_t$ to the player, while the Full Information supplies the full cost vector.

In the present chapter we shall focus exclusively on the Full Information version of this problem, but we return to the Bandit setting in Chapter 4. Indeed, the algorithm we give for the Bandit setting is actually via a reduction to the Full Information setting. For the case of Full Information, we now present a class of algorithms known as *Follow the Regularized Leader (FTRL)*, in which the learner attempts to minimize a regularized objective at every step of the game.

2.2 Low-Regret Learning with Full-Information

The *online linear optimization* problem is defined as the following repeated game between the learner (player) and the environment (adversary).

At each time step $t = 1$ to T,

- Player chooses $\mathbf{x}_t \in \mathcal{K}$
- Adversary independently chooses $\mathbf{f}_t \in \mathbb{R}^n$
- Player suffers loss $\mathbf{f}_t^\top \mathbf{x}_t$ and observes feedback $\mathscr{F}_t$

The goal of the Player is not simply to minimize his total loss $\sum_{t=1}^{T} \mathbf{f}_t^\top \mathbf{x}_t$, for an adversary could simply choose $\mathbf{f}_t$ to be as large as possible at every point in $\mathcal{K}$. Rather, the Player's goal

is to minimize his *regret*. If the player uses some algorithm $\mathcal{A}$ that chooses the predictions $\mathbf{x}_1, \mathbf{x}_2, \ldots$ and is presented with a sequence of functions $\mathbf{f}_{1:T} := (\mathbf{f}_1, \ldots, \mathbf{f}_T)$, then we define

$$\text{Regret}(\mathcal{A}; \mathbf{f}_{1:T}) := \sum_{t=1}^{T} \mathbf{f}_t^\top \mathbf{x}_t - \min_{\mathbf{x}^* \in \mathcal{K}} \sum_{t=1}^{T} \mathbf{f}_t^\top \mathbf{x}^*.$$

At times, we may refer to the regret with respect to a particular *comparator* $\mathbf{u}$, namely

$$\text{Regret}^{\mathbf{u}}(\mathcal{A}; \mathbf{f}_{1:T}) := \sum_{t=1}^{T} \mathbf{f}_t^\top \mathbf{x}_t - \sum_{t=1}^{T} \mathbf{f}_t^\top \mathbf{u}.$$

It is generally assumed that the linear costs $\mathbf{f}_t$ are chosen from some bounded set $\mathcal{L} \subset \mathbb{R}^n$. With this in mind, we also define the worst case regret $\text{Regret}_T(\mathcal{A}) := \sup_{\mathbf{f}_{1:T} \in \mathcal{L}^T} \text{Regret}(\mathcal{A}; \mathbf{f}_{1:T})$ with respect to $\mathcal{L}$.

2.2.1 Follow the Regularized Leader and Associated Bounds

Follow The Leader (FTL) is perhaps the simplest online learning strategy one might arrive at: the Player simply uses the heuristic "select the best choice thus far". In game theory, this strategy is known as fictitious play, and was introduced by G.W. Brown in 1951. For the online optimization task we study, this can be written as

$$\mathbf{x}_{t+1} := \arg\min_{\mathbf{x} \in \mathcal{K}} \sum_{s=1}^{t} \mathbf{f}_s^\top \mathbf{x}. \tag{2.2}$$

For certain types of problems, applying FTL does guarantee low regret. Unfortunately, when the loss functions $\mathbf{f}_t$ are linear on the input space it can be shown that FTL will suffer regret that grows linearly in T.

A natural approach[1], and more well-known within statistical learning, is to *regularize* the optimization problem (2.2) with an appropriate regularization function $\mathcal{R}(\mathbf{x})$, which is generally considered to be smooth and convex. The decision strategy is described in the following algorithm, which we refer to as Follow the Regularized Leader (FTRL).

Algorithm 1 FTRL$(\mathcal{R}, \eta)$: Follow the Regularized Leader

Input: $\eta > 0$, regularization $\mathcal{R}$.
On round $t + 1$, play

$$\mathbf{x}_{t+1} := \arg\min_{\mathbf{x} \in \mathcal{K}} \left[\eta \sum_{s=1}^{t} \mathbf{f}_s^\top \mathbf{x} + \mathcal{R}(\mathbf{x}) \right]. \tag{2.3}$$

We recall that this algorithm can only applied in the full-information setting. That is, the choice of $\mathbf{x}_{t+1}$ requires observing $\mathbf{f}_1, \ldots, \mathbf{f}_t$ to solve the objective in (2.3).

[1] In the context of classification, this approach has been formulated and analyzed by Shalev-Shwartz and Singer [70].

We now prove a simple bound on the regret of FTRL for a given regularization function $\mathcal{R}$ and parameter η. This bounds is not particularly useful in and of itself, yet it shall serve as a launching point for several results we give in the remainder of this chapter.

Proposition 1. *Given any sequence of cost vectors $\mathbf{f}_1, \ldots, \mathbf{f}_T$ and for any point $\mathbf{u} \in \mathcal{K}$, Algorithm 1 (FTRL) enjoys the guarantee*

$$\begin{aligned}&\text{Regret}^{\mathbf{u}}(\mathit{FTRL}(\mathcal{R}, \eta); \mathbf{f}_{1:T}) \\ &\qquad \leq \sum_{t=1}^{T} \mathbf{f}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) + \frac{\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)}{\eta}.\end{aligned}$$

Proof. Towards bounding the regret of FTRL$(\mathcal{R}, \eta)$, let us first imagine a slightly modified algorithm, BTRL$(\mathcal{R}, \eta)$ for Be The Regularized Leader: instead of playing the point $\mathbf{x}_t$ on round t, the algorithm BTRL$(\mathcal{R}, \eta)$ plays the point $\mathbf{x}_{t+1}$, that is, the point that would be played by FTRL$(\mathcal{R}, \eta)$ with knowledge of *one additional round*. This algorithm is, of course, entirely fictitious as we are assuming it has access to the yet-to-be-observed $\mathbf{f}_t$, but it will be a useful hypothetical in our analysis.

Let us now bound the regret of BTRL$(\mathcal{R}, \eta)$. Precisely, we shall show the bound for the "worst-case" comparator $\mathbf{u} \in \mathcal{K}$, that is

$$\sum_{s=1}^{t} \mathbf{f}_s^\top \mathbf{x}_{s+1} \leq \min_{\mathbf{u} \in \mathcal{K}} \sum_{s=1}^{t} \mathbf{f}_s^\top \mathbf{u} + \frac{\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)}{\eta}. \tag{2.4}$$

Notice that, with the latter established, the proof is completed easily. The total loss of BTRL$(\mathcal{R}, \eta)$ is $\sum_{t=1}^{T} \mathbf{f}_t^\top \mathbf{x}_{t+1}$, whereas the total loss of FTRL$(\mathcal{R}, \eta)$ is $\sum_{t=1}^{T} \mathbf{f}_t^\top \mathbf{x}_t$. It follows that the difference in loss, and hence the difference in regret, for any $\mathbf{u} \in \mathcal{K}$, is identically

$$\begin{aligned}&\text{Regret}^{\mathbf{u}}(\text{FTRL}(\mathcal{R}, \eta); \mathbf{f}_{1:T}) \\ &\qquad = \text{Regret}^{\mathbf{u}}(\text{BTRL}(\mathcal{R}, \eta); \mathbf{f}_{1:T}) + \sum_{t=1}^{T} \mathbf{f}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}).\end{aligned}$$

Combining this with (2.4) gives the proof.

We now proceed prove (2.4) by induction. The base case, for $t = 0$, holds trivially. Now assume the above bound holds for $t - 1$. The crucial observation is that the point $\mathbf{x}_{t+1}$ is chosen as the minimizer of the right-hand side of (2.4).

$$\begin{aligned}\sum_{s=1}^{t} \mathbf{f}_s^\top \mathbf{x}_{s+1} &= \mathbf{f}_t^\top \mathbf{x}_{t+1} + \sum_{s=1}^{t-1} \mathbf{f}_s^\top \mathbf{x}_{s+1} \\ \text{(induction)} &\leq \mathbf{f}_t^\top \mathbf{x}_{t+1} + \min_{\mathbf{u} \in \mathcal{K}} \sum_{s=1}^{t} \mathbf{f}_s^\top \mathbf{u} + \frac{\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)}{\eta} \\ (\mathbf{u} \leftarrow \mathbf{x}_{t+1}) &\leq \mathbf{f}_t^\top \mathbf{x}_{t+1} + \sum_{s=1}^{t} \mathbf{f}_s^\top \mathbf{x}_{t+1} + \frac{\mathcal{R}(\mathbf{x}_{t+1}) - \mathcal{R}(\mathbf{x}_1)}{\eta} \\ &= \min_{\mathbf{u} \in \mathcal{K}} \sum_{s=1}^{t} \mathbf{f}_s^\top \mathbf{u} + \frac{\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)}{\eta},\end{aligned}$$

which completes the proof. □

2.2.2 Upper Bounds for Strongly Convex $\mathcal{R}$

The bound stated in Proposition 1 is difficult to interpret for, at present, it tells us that the regret is bounded by the size of successive steps between $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$. Notice that the point $\mathbf{x}_{t+1}$ depends on both $\mathbf{f}_t$ and η as well as on the behavior of $\mathcal{R}$. Ultimately, we want a bound independent of the $\mathbf{x}_t$'s since these points are not under our control once we have fixed $\mathcal{R}$.

We arrive at a much more useful set of bounds if we require certain conditions on the regularizer $\mathcal{R}$. Indeed, the purpose of including the regularizer was to ensure stability of the solutions $\mathbf{x}_t$, which will help control $\mathbf{f}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1})$. Via Hölder's Inequality, we always have

$$\mathbf{f}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}) \leq \|\mathbf{f}_t\|^* \|\mathbf{x}_t - \mathbf{x}_{t+1}\| \tag{2.5}$$

for any dual norm pair $\|\cdot\|, \|\cdot\|^*$. Typically, it is assumed that $\mathbf{f}_t$ is explicitly bounded, and hence our remaining work is to bound $\|\mathbf{x}_t - \mathbf{x}_{t+1}\|$. The usual approach is to require that $\mathcal{R}$ be *suitably curved*. To discuss curvature, it is helpful to define the notion of a *Bregman divergance*.

Definition 2. *Given any strictly convex function $\mathcal{R}$, define the* Bregman divergence *with respect to $\mathcal{R}$ as*

$$D_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) = \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{y}) - \langle \nabla\mathcal{R}(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle.$$

A Bregman divergence measures the "distance" between points $\mathbf{x}$ and $\mathbf{y}$ in terms of the "gap" in Jensen's Inequality, that is by how much the function $\mathcal{R}$ deviates at $\mathbf{y}$ from it's linear approximation at $\mathbf{x}$. It is natural to see that the Bregman divergence is larger for functions $\mathcal{R}$ with greater curvature, which leads us to the following definition.

Definition 3. *A function $\mathcal{R}(\mathbf{x})$ is* strongly convex *with respect to some norm $\|\cdot\|$ whenever the associated Bregman divergence satisifes $D_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \geq \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$ for all $\mathbf{x}, \mathbf{y}$.*

While it might not be immediately obvious, the strong convexity of the regularization function in the FTRL algorithm is directly connected to the bound in Proposition 1. Specifically, the term $\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)$ increases with larger curvature of $\mathcal{R}$, whereas the terms $\mathbf{f}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1})$ shrink. Towards making the latter more precise, we give two lemmas regarding the "distance" between the pairs $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$.

Lemma 4. *For the sequence $\{\mathbf{x}_t\}$ chosen according to FTRL$(\mathcal{R}, \eta)$, we have that for any t:*

$$\begin{aligned} D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) &\leq \langle \nabla\mathcal{R}(\mathbf{x}_t) - \nabla\mathcal{R}(\mathbf{x}_{t+1}), \mathbf{x}_t - \mathbf{x}_{t+1} \rangle \\ &\leq \langle \eta\mathbf{f}_t, \mathbf{x}_t - \mathbf{x}_{t+1} \rangle. \end{aligned}$$

Proof. Recalling that the divergence is always nonnegative, we obtain the first inequality by noting that for any $\mathbf{x}, \mathbf{y} \in \mathcal{K}$, $D_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \leq D_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) + D_{\mathcal{R}}(\mathbf{y}, \mathbf{x}) = \langle \nabla\mathcal{R}(\mathbf{x}) - \nabla\mathcal{R}(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$.

For the second inequality, we observe that $\mathbf{x}_{t+1}$ is obtained in the optimization (2.3), and hence we have the first-order optimality condition

$$\left\langle \nabla \mathcal{R}(\mathbf{x}_{t+1}) + \eta \sum_{s=1}^{t} \mathbf{f}_s, \mathbf{y} - \mathbf{x}_{t+1} \right\rangle \geq 0 \ \ \forall \mathbf{y} \in \mathcal{K}. \tag{2.6}$$

We now apply this inequality twice: for rounds t and $t+1$ set $\mathbf{y} = \mathbf{x}_{t+1}$ and $\mathbf{y} = \mathbf{x}_t$, respectively. Adding the inequalities together gives

$$\langle \nabla \mathcal{R}(\mathbf{x}_t) - \nabla \mathcal{R}(\mathbf{x}_{t+1}), \mathbf{x}_t - \mathbf{x}_{t+1} \rangle \leq \langle \eta \mathbf{f}_t, \mathbf{x}_t - \mathbf{x}_{t+1} \rangle,$$

concluding the proof. □

Lemma 5. *For the sequence $\{\mathbf{x}_t\}$ chosen according to FTRL$(\mathcal{R}, \eta)$, we have that for any t:*

$$\|\mathbf{x}_t - \mathbf{x}_{t+1}\| \leq \eta \|\mathbf{f}_t\|^*,$$

where $\|\cdot\|^*$ is the associated dual norm.

Proof. Using the definition of strong convexity, we have

$$\begin{aligned} \|\mathbf{x}_t - \mathbf{x}_{t+1}\|^2 &\leq D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) + D_{\mathcal{R}}(\mathbf{x}_{t+1}, \mathbf{x}_t) \\ &= \langle \nabla \mathcal{R}(\mathbf{x}_t) - \nabla \mathcal{R}(\mathbf{x}_{t+1}), \mathbf{x}_t - \mathbf{x}_{t+1} \rangle \\ \text{(Lemma 4)} \quad &\leq \langle \eta \mathbf{f}_t, \mathbf{x}_t - \mathbf{x}_{t+1} \rangle \\ \text{(Hölder's Ineq.)} \quad &\leq \eta \|\mathbf{f}_t\|^* \|\mathbf{x}_t - \mathbf{x}_{t+1}\|. \end{aligned}$$

Dividing both sides by $\|\mathbf{x}_t - \mathbf{x}_{t+1}\|$ gives the result. □

Applying (2.5) and Lemma 5 to Proposition 1, we arrive at the following.

Proposition 6. *When $\mathcal{R}$ is strongly convex with respect to the norm $\|\cdot\|$, then for any $\mathbf{u} \in \mathcal{K}$*

$$\text{Regret}^{\mathbf{u}}(FTRL(\mathcal{R}, \eta); \mathbf{f}_{1:T}) \leq \eta \sum_{t=1}^{T} \|\mathbf{f}_t\|^{*2} + \eta^{-1} \mathcal{R}(\mathbf{u}).$$

What have we done here? By including the additional strong-convexity assumption on $\mathcal{R}$, we can now measure the algorithm's regret without concerning ourselves with the specific points $\mathbf{x}_t$ chosen in the optimization. Instead, we have a bound which depends solely on the sequence of inputs $\{\mathbf{f}_t\}$ and the choice of regularization $\mathcal{R}$. We can take this one step further and obtain a worst-case bound on the regret explicitly in terms of T, the maximum value of $\mathcal{R}$, and the size of the $\mathbf{f}_t$'s.

Corollary 7. *When $\mathcal{R}$ is strongly convex with respect to the norm $\|\cdot\|$, and for constants $G, R > 0$ we have $\|\mathbf{f}_t\|^* \leq G$ for every t and $\mathcal{R}(\mathbf{x}) \leq R$ for every $\mathbf{x} \in \mathcal{K}$, then by setting $\eta = \sqrt{\frac{R}{GT}}$ we have*

$$\text{Regret}_T(FTRL(\mathcal{R}, \eta)) \leq 2\sqrt{TGR}.$$

Chapter 3

Lower Bounds

3.1 Introduction

The decision maker's greatest fear is *regret*: knowing, with the benefit of hindsight, that a better alternative existed. Yet, given only hindsight and not the gift of foresight, imperfect decisions can not be avoided. It is thus the decision maker's ultimate goal to suffer as little regret as possible.

In the present chapter, we consider the notion of "regret minimization" for a particular class of decision problems. Assume we are given a set X and some set of functions $\mathcal{F}$ on X. On each round $t = 1, \ldots, T$, we must choose some $\mathbf{x}_t$ from a set X. After we have made this choice, the *environment* chooses a function $f_t \in \mathcal{F}$. We incur a cost (loss) $f_t(\mathbf{x}_t)$, and the game proceeds to the next round. Of course, had we the fortune of perfect foresight and had access to the sum $f_1 + \ldots + f_T$, we would know the optimal choice $\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{t=1}^{T} f_t(\mathbf{x})$. Instead, at time t, we will have only seen $f_1, \ldots, f_{t-1}$, and we must make the decision $\mathbf{x}_t$ with only historical knowledge. Thus, a natural long-term goal is to minimize the *regret*, which here we define as

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \inf_{\mathbf{x} \in X} \sum_{t=1}^{T} f_t(\mathbf{x}).$$

A special case of this setting is when the decision space X is a convex set and $\mathcal{F}$ is some set of convex functions on X. In the literature, this framework has been referred to as Online Convex Optimization (OCO), since our goal is to minimize a global function, i.e. $f_1 + f_2 + \cdots + f_T$, while this objective is revealed to us but one function at a time. Online Convex Optimization has attracted much interest in recent years [48, 78, 69, 15], as it provides a general analysis for a number of standard online learning problems including, among others, online classification and regression, prediction with expert advice, the portfolio selection problem, and online density estimation.

While instances of OCO have been studied over the past two decades, the general problem was first analyzed by Zinkevich [78], who showed that a very simple and natural algorithm, online gradient descent, elicits a bound on the regret that is on the order of $\sqrt{T}$. Online gradient descent can be described simply by the update $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \nabla f_t(\mathbf{x}_t)$, where η

is some parameter of the algorithm. This regret bound only required that f_t be smooth, convex, and with bounded derivative.

A regret bound of order $O(\sqrt{T})$ is not surprising: a number of online learning problems give rise to similar bounds. More recently, however, Hazan et al. [48] showed that when $\mathcal{F}$ consists of *curved* functions, i.e. f_t is strongly convex, then we get a bound of the form $O(\log T)$. It is quite surprising that curvature gives such a great advantage to the player. Curved loss functions, such as square loss or logarithmic loss, are very natural in a number of settings.

Finding algorithms that can guarantee low regret is, however, only half of the story; indeed, it is natural to ask "can we obtain even lower regret?" or "do better algorithms exist?" The goal of the present chapter is to address these questions, in some detail, for several classes of such online optimization problems. We answer both in the negative: the algorithms of Zinkevich and Hazan et al. are tight even up to their multiplicative constants.

This is achieved by a game-theoretic analysis: if we pose the above online optimization problem as a game between a Player who chooses $\mathbf{x}_t$ and an Adversary who chooses f_t, we may consider the regret achieved when each player is playing optimally. This is typically referred to as the *value* V_T of the game. In general, computing the value of zero-sum games is difficult, as we may have to consider exponentially many, or even uncountably many, strategies of the Player and the Adversary. Ultimately we will show that this value, as well as the optimal strategies of both the player and the adversary, can be computed *exactly and efficiently* for certain classes of online optimization games.

The central results of this chapter are as follows:

- When the adversary plays *linear* loss functions, we use a known randomized argument to lower bound the value V_T. We include this mainly for completeness.
- We show that indeed this same linear game can be solved *exactly* for the case when the input space X is a ball, and we provide the optimal strategies for the player and adversary.
- We perform a similar analysis for the *quadratic game*, that is where the adversary must play quadratic functions. We describe the adversary's strategy, and we prove that the well-known Follow the Leader strategy is optimal for the player.
- We show that the above results apply to a much wider class of games, where the adversary can play either convex or strongly convex functions, suggesting that indeed the linear and quadratic games are the "hard cases".

3.2 Online Convex Games

The general optimization game we consider is as follows. We have two agents, a player and an adversary, and the game proceeds for T rounds with T known in advance to both agents. The player's choices will come from some convex set $X \subset \mathbb{R}^n$, and the adversary

will choose functions from the class $\mathcal{F}$. For the remainder of the chapter, n denotes the dimension of the space X. To consider the game in full generality, we assume that the adversary's "allowed" functions may change on each round, and thus we imagine there is a sequence of allowed sets $L_1, L_2, \ldots, L_T \subset \mathcal{F}$.

Online Convex Game

$\mathcal{G}(X, \{L_t\})$:

1: **for** $t = 1$ to T **do**
2: Player chooses (predicts) $\mathbf{x}_t \in X$.
3: Adversary chooses a function $f_t \in L_t$.
4: **end for**
5: Player suffers regret

$$R_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \inf_{\mathbf{x} \in X} \sum_{t=1}^{T} f_t(\mathbf{x}).$$

From this general game, we obtain each of the examples above with appropriate choice of $X, \mathcal{F}$ and the sets $\{L_t\}$. We define a number of particular games in the definitions below.

It is useful to prove regret bounds within this model as they apply to any problem that can be cast as an Online Convex Game. The known general upper bounds are as follows:

- **Zinkevich [78]:** If $L_1 = \ldots = L_T = \mathcal{F}$ consist of continuous twice differentiable functions f, where $\|\nabla f\| \leq G$ and $\nabla^2 f \succeq \mathbf{0}$, then[1]

  $$R_T \leq \frac{1}{2} DG\sqrt{T}.$$

  where $D := \max_{\mathbf{x},\mathbf{y} \in X} \|\mathbf{x} - \mathbf{y}\|$ and G is some positive constant.

- **Hazan et al. [48]:** If $L_1 = \ldots = L_T = \mathcal{F}$ consist of continuous twice differentiable functions f, where $\|\nabla f\| \leq G$ and $\nabla^2 f \succeq \sigma I$, then

  $$R_T \leq \frac{1}{2} \frac{G^2}{\sigma} \log T,$$

  where G and σ are positive constants.

- **Bartlett et al. [15]:** If L_t consists of continuous twice differentiable functions f, where $\|\nabla f\| \leq G_t$ and $\nabla^2 f \succeq \sigma_t I$, then

  $$R_T \leq \frac{1}{2} \sum_{t=1}^{T} \frac{G_t^2}{\sum_{s=1}^{t} \sigma_s},$$

  where G_t and σ_t are positive constants. Moreover, the algorithm does not need to know G_t, σ_t on round t.

[1] This bound can be obtained by a slight modification of the analysis in [78].

All three of these games posit an upper bound on $\|\nabla f\|$ which is required to make the game nontrivial (and is natural in most circumstances). However, the first requires only that the second derivative be nonnegative, while the second and third game has a strict positive lower bound on the eigenvalues of the Hessian $\nabla^2 f$. Note that the bound of Bartlett et al recovers the logarithmic regret of Hazan et al whenever G_t and σ_t do not vary with time.

In the present chapter, we analyze each of these games with the goal of obtaining the exact minimax value of the game, defined as:

$$V_T(\mathcal{G}(X, \{L_t\})) = \inf_{\mathbf{x}_1 \in X} \sup_{f_1 \in L_1} \dots \inf_{\mathbf{x}_T \in X} \sup_{f_T \in L_T} \left(\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \inf_{\mathbf{x} \in X} \sum_{t=1}^{T} f_t(\mathbf{x}) \right).$$

The quantity $V_T(\mathcal{G})$ tells us the worst case regret of an *optimal* strategy in this game.

First, in the spirit of [15], we consider V_T for the games where constants G and σ, which respectively bound the first and second derivatives of f_t, can change throughout the game. That is, the Adversary is given two sequences before the game begins, $\langle G_1, \dots, G_T \rangle$ and $\langle \sigma_1, \dots, \sigma_T \rangle$. We also require only that the gradient of f_t is bounded *at the point* $\mathbf{x}_t$, i.e. $\|\nabla f_t(\mathbf{x}_t)\| \leq G_t$, as opposed to the global constraint $\|\nabla f_t(\mathbf{x})\| \leq G_t$ for all $\mathbf{x} \in X$. We may impose both of the above constraints by carefully choosing the sets $L_t \subseteq \mathcal{F}$, and we note that these sets will depend on the choices $\mathbf{x}_t$ made by the Player.

We first define the Linear and Quadratic Games, which are the central objects of this chapter.

Definition 8. *The Linear Game* $\mathcal{G}_{lin}(X, \langle G_t \rangle)$ *is the game* $\mathcal{G}(X, \{L_t\})$ *where*

$$L_t = \{f : f(\mathbf{x}) = v^\top(\mathbf{x} - \mathbf{x}_t) + c, v \in \mathcal{R}^n, c \in \mathcal{R}; \|v\| \leq G_t\}.$$

Definition 9. *The Quadratic Game* $\mathcal{G}_{quad}(X, \langle G_t \rangle, \langle \sigma_t \rangle)$ *is the game* $\mathcal{G}(X, \{L_t\})$ *where*

$$L_t = \{f : f(\mathbf{x}) = v^\top(\mathbf{x} - \mathbf{x}_t) + \frac{\sigma_t}{2}\|\mathbf{x} - \mathbf{x}_t\|^2 + c, \\ v \in \mathcal{R}^n, c \in \mathcal{R}; \|v\| \leq G_t\}.$$

The functions in these definitions are parametrized through $\mathbf{x}_t$ to simplify proofs of the last section. In Section 3.4, however, we will just consider the standard parametrization $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$.

We also introduce more general games: the Convex Game and the Strongly Convex Game. While being defined with respect to a much richer class of loss functions, we show that these games are indeed no harder than the Linear and the Quadratic Games defined above.

Definition 10. *The Convex Game* $\mathcal{G}_{conv}(X, \langle G_t \rangle)$ *is the game* $\mathcal{G}(X, \{L_t\})$ *where*

$$L_t = \{f : \|\nabla f(\mathbf{x}_t)\| \leq G_t, \nabla^2 f \succeq 0\}.$$

Definition 11. *The Strongly Convex Game* $\mathcal{G}_{st\text{-}conv}(X, \langle G_t \rangle, \langle \sigma_t \rangle)$ *is the game* $\mathcal{G}(X, \{L_t\})$ *where*

$$L_t = \{f : \|\nabla f(\mathbf{x}_t)\| \leq G_t, \nabla^2 f - \sigma_t I \succeq 0\}.$$

We write $\mathcal{G}(G)$ instead of $\mathcal{G}(\langle G_t \rangle)$ when all values $G_t = G$ for some fixed G. This holds similarly for $\mathcal{G}(\sigma)$ instead of $\mathcal{G}(\langle \sigma_t \rangle)$. Furthermore, we suppose that $\sigma_1 > 0$ throughout the chapter.

3.3 Previous Work

Several lower bounds for various online settings are available in the literature. Here we review a number of such results relevant to the present chapter and highlight our primary contributions.

The first result that we mention is the lower bound of Vovk in the online linear regression setting [74]. It is shown that there exists a randomized strategy of the Adversary such that the expected regret is at least $[(n - \varepsilon)G^2 \ln T - C_\varepsilon]$ for any $\varepsilon > 0$ and C_ε a constant. One crucial difference between this particular setting and ours is that the loss functions of the form $(y_t - \mathbf{x}_t \cdot \mathbf{w}_t)^2$ used in linear regression are curved in only one direction and linear in all other, thus this setting does not quite fit into any of the games we analyze. The lower bound of Vovk scales roughly as $n \log T$, which is quite interesting given that n does not enter into the lower bound of the Strongly Convex Game we analyze.

The lower bound for the log-loss functions of Ordentlich and Cover [65] in the setting of Universal Portfolios is also logarithmic in T and linear in n. Log-loss functions are parameterized as $f_t(\mathbf{x}) = -\log(\mathbf{w} \cdot \mathbf{x})$ for $\mathbf{x}$ in the simplex, and these fit more generally within the class of "exp-concave" functions. Upper bounds on the class of log-loss functions were originally presented by Cover [25] whereas Hazan et al. [48] present an efficient method for competing against the more general exp-concave functions. The log-loss lower bound of [65] is quite elegant yet, contrary to the minimax results we present, the optimal play is not efficiently computable.

The work of Takimoto and Warmuth [72] is most closely related to our results for the Quadratic Game. The authors consider functions $f(\mathbf{x}) = \frac{1}{2}||\mathbf{x} - \mathbf{y}||^2$ corresponding to the log-likelihood of the datapoint $\mathbf{y}$ for a unit-variance Gaussian with mean $\mathbf{x}$. The lower bound of $\frac{1}{2}D^2(\ln T - \ln \ln T + O(\ln \ln T / \ln T))$ is obtained, where D is the bound on the norm of adversary's choices $\mathbf{y}$. Furthermore, they exhibit the minimax strategy which, in the end, corresponds to a biased maximum-likelihood solution. We emphasize that these results differ from ours in several ways. First, we enforce a constraint on the size of the gradient of f_t whereas [72] constrain the location of the point $\mathbf{y}$ when $f_t(\mathbf{x}) = \frac{1}{2}||\mathbf{x} - \mathbf{y}||^2$. With our slightly weaker constraint, we can achieve a regret bound of the order $\log T$ instead of the $\log T - \log \log T$ of Takimoto and Warmuth. Interestingly, the authors describe the "$-\log \log T$" term of their lower bound as "surprising" because many known games "were shown to have $O(\log T)$ upper bounds". They conjecture that the apparent slack is due to the learner being unaware of the time horizon T. In the present chapter, we resolve this issue

by noting that our slightly weaker assumption erases the additional term; it is thus the limit on the adversary, and not knowledge of the horizon, that gives rise to the slack. Furthermore, the minimax strategy of Takimoto and Warmuth, a biased maximum likelihood estimate on each round, is also an artifact of their assumption on the boundedness of adversary's choices. With our weaker assumption, the minimax strategy is *exactly* maximum likelihood (generally called "Follow The Leader").

All previous work mentioned above deals with "curved" functions. We now discuss known lower bounds for the Linear Game. It is well-known that in the expert setting, it is impossible to do better than $O(\sqrt{T})$. The lower bound in Cesa-Bianchi and Lugosi [24], Theorem 3.7, proves an asymptotic bound: in the limit of $T \to \infty$, the value of the game behaves as $\sqrt{(\ln N)T/2}$, where N is the number of experts. We provide a similar randomized argument, which has been sketched in the literature (e.g. Hazan et al [48]), but our additional minimax analysis indeed gives the tightest bound possible for any T.

Finally, we provide reductions between Quadratic and Strongly Convex as well as Linear and Convex Games. While apparent that the Adversary does better by playing linear approximations instead of convex functions, it requires a careful analysis to show that this holds for the minimax setting.

3.4 The Linear Game

In this section we begin by providing a relatively standard proof of the $O(\sqrt{T})$ lower bound on regret when competing against linear loss functions. The more interesting result is our *minimax* analysis which is given in Section 3.4.2.

3.4.1 The Randomized Lower Bound

Lower bounds for games with linear loss functions have appeared in the literature though often not in detail. The rough idea is to imagine a randomized Adversary and to compute the Player's expected regret. This generally produces an $O(\sqrt{T})$ lower bound yet it is not fully satisfying since the analysis is not tight. In the following section we provide a much improved analysis with minimax strategies for both the Player and Adversary.

Theorem 12. *Suppose $X = [-D/(2\sqrt{n}), D/(2\sqrt{n})]^n$, so that the diameter of X is D. Then*

$$V_T(\mathcal{G}_{lin}(X, \langle G_t \rangle)) \geq \frac{D}{2\sqrt{2}} \sqrt{\sum_{t=1}^{T} G_t^2}$$

Proof. Define the scaled cube

$$\mathcal{C}_t = \{-G_t/\sqrt{n}, G_t/\sqrt{n}\}^n.$$

Suppose the Adversary chooses functions from

$$\hat{L}_t = \{f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} : \mathbf{w} \in \mathcal{C}_t\}.$$

Note that $\|\nabla f\| = \|\mathbf{w}_t\| = G_t$ for any $f \in \hat{L}_t$.

Since we are restricting the Adversary to play linear functions with restricted $\mathbf{w}$,

$$\begin{aligned}
&V_T(\mathcal{G}_{\text{lin}}(X, \langle G_t \rangle)) \geq V_T(\mathcal{G}(X, \hat{L}_1, \ldots, \hat{L}_T)) \\
&= \inf_{\mathbf{x}_1 \in X} \sup_{f_1 \in \hat{L}_1} \ldots \inf_{\mathbf{x}_T \in X} \sup_{f_T \in \hat{L}_T} \left[\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \inf_{\mathbf{x} \in X} \sum_{t=1}^{T} f_t(\mathbf{x}) \right] \\
&= \inf_{\mathbf{x}_1 \in X} \sup_{\mathbf{w}_1 \in \mathcal{C}_1} \ldots \inf_{\mathbf{x}_T \in X} \sup_{\mathbf{w}_T \in \mathcal{C}_T} \left[\sum_{t=1}^{T} \mathbf{w}_t \cdot \mathbf{x}_t - \inf_{\mathbf{x} \in X} \mathbf{x} \cdot \sum_{t=1}^{T} \mathbf{w}_t \right] \\
&\geq \inf_{\mathbf{x}_1 \in X} \mathop{\mathbb{E}}_{\mathbf{w}_1} \ldots \inf_{\mathbf{x}_T \in X} \mathop{\mathbb{E}}_{\mathbf{w}_T} \left[\sum_{t=1}^{T} \mathbf{w}_t \cdot \mathbf{x}_t - \inf_{\mathbf{x} \in X} \mathbf{x} \cdot \sum_{t=1}^{T} \mathbf{w}_t \right],
\end{aligned}$$

where $\mathbb{E}_{\mathbf{w}_t}$ denotes expectation with respect to any distribution over the set $\mathcal{C}_t$. In particular, it holds for the uniform distribution, i.e. when the coordinates of $\mathbf{w}_t$ are $\pm G_t/\sqrt{n}$ with probability $1/2$. Since in this case $\mathbb{E}_{\mathbf{w}_T} \mathbf{w}_T \cdot \mathbf{x}_T = 0$ for any $\mathbf{x}_T$, we obtain

$$\begin{aligned}
&V_T(\mathcal{G}_{\text{lin}}(X, \langle G_t \rangle)) \\
&\geq \inf_{\mathbf{x}_1 \in X} \mathop{\mathbb{E}}_{\mathbf{w}_1} \ldots \inf_{\mathbf{x}_{T-1} \in X} \mathop{\mathbb{E}}_{\mathbf{w}_{T-1}} \inf_{\mathbf{x}_T \in X} \\
&\qquad \mathop{\mathbb{E}}_{\mathbf{w}_T} \left[\sum_{t=1}^{T} \mathbf{w}_t \cdot \mathbf{x}_t - \inf_{\mathbf{x} \in X} \mathbf{x} \cdot \sum_{t=1}^{T} \mathbf{w}_t \right] \\
&= \inf_{\mathbf{x}_1 \in X} \mathop{\mathbb{E}}_{\mathbf{w}_1} \ldots \inf_{\mathbf{x}_{T-1} \in X} \mathop{\mathbb{E}}_{\mathbf{w}_{T-1}} \inf_{\mathbf{x}_T \in X} \\
&\qquad \left[\sum_{t=1}^{T-1} \mathbf{w}_t \cdot \mathbf{x}_t - \mathop{\mathbb{E}}_{\mathbf{w}_T} \inf_{\mathbf{x} \in X} \mathbf{x} \cdot \sum_{t=1}^{T} \mathbf{w}_t \right] \\
&= \inf_{\mathbf{x}_1 \in X} \mathop{\mathbb{E}}_{\mathbf{w}_1} \ldots \inf_{\mathbf{x}_{T-1} \in X} \\
&\qquad \mathop{\mathbb{E}}_{\mathbf{w}_{T-1}} \left[\sum_{t=1}^{T-1} \mathbf{w}_t \cdot \mathbf{x}_t - \mathop{\mathbb{E}}_{\mathbf{w}_T} \inf_{\mathbf{x} \in X} \mathbf{x} \cdot \sum_{t=1}^{T} \mathbf{w}_t \right],
\end{aligned}$$

where the last equality holds because the expression no longer depends on $\mathbf{x}_T$. Repeating the process, we obtain

$$\begin{aligned}
V_T(\mathcal{G}_{\text{lin}}(X, \langle G_t \rangle)) &\geq - \mathop{\mathbb{E}}_{\mathbf{w}_1, \ldots, \mathbf{w}_T} \inf_{\mathbf{x} \in X} \mathbf{x} \cdot \sum_{t=1}^{T} \mathbf{w}_t \\
&= - \mathop{\mathbb{E}}_{\{\varepsilon_{i,t}\}} \min_{\mathbf{x} \in \left\{ -\frac{D}{2\sqrt{n}}, \frac{D}{2\sqrt{n}} \right\}^n} \left(\mathbf{x} \cdot \sum_{t=1}^{T} \mathbf{w}_t \right),
\end{aligned}$$

where $\mathbf{w}_t(i) = \varepsilon_{i,t} G_t/\sqrt{n}$, with i.i.d. Rademacher variables $\varepsilon_{i,t} = \pm 1$ with probability $1/2$. The last equality is due to the fact that a linear function is minimized at the vertices of the

cube. In fact, the dot product is minimized by matching the sign of $\mathbf{x}(i)$ with that of the ith coordinate of $\sum_{t=1}^{T} \mathbf{w}_t$. Hence,

$$V_T(\mathcal{G}_{\text{lin}}(X, \langle G_t \rangle)) \geq - \mathop{\mathbb{E}}_{\{\varepsilon_{i,t}\}} \sum_{i=1}^{n} -\frac{D}{2\sqrt{n}} \left| \sum_{t=1}^{T} \varepsilon_{i,t} \frac{G_t}{\sqrt{n}} \right|$$
$$= \frac{D}{2} \mathop{\mathbb{E}}_{\{\varepsilon_{i,t}\}} \left| \sum_{t=1}^{T} \varepsilon_{i,t} G_t \right| \geq \frac{D}{2\sqrt{2}} \sqrt{\sum_{t=1}^{T} G_t^2},$$

where the last inequality follows from the Khinchine's inequality [24]. □

3.4.2 The Minimax Analysis

While in the previous section we found a particular lower bound on $V_T(\mathcal{G}_{\text{lin}})$, here we present a complete minimax analysis for the case when X is a ball in $\mathcal{R}^n$ (of dimension n at least 3). We are indeed able to compute exactly the value

$$V_T(\mathcal{G}_{\text{lin}}(X, \langle G_t \rangle))$$

and we provide the simple minimax strategies for both the Player and the Adversary. The unit ball, while a special case, is a very natural choice for X as it is the *largest* convex set of diameter 2.

For the remainder of this section, let $f_t(\mathbf{x}) := \mathbf{w}_t \cdot \mathbf{x}$ where $\mathbf{w}_t \in \mathbb{R}^n$ with $\|\mathbf{w}_t\| \leq G_t$. Also, we define $\mathbf{W}_t = \sum_{s=1}^{t} \mathbf{w}_s$, the cumulative functions chosen by the Adversary.

Theorem 13. *Let $X = \{\mathbf{x} : \|\mathbf{x}\|_2 \leq D/2\}$ and suppose the Adversary chooses functions from*

$$L_t = \{f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} : \|\mathbf{w}\|_2 \leq G_t\}.$$

Then the value of the game

$$V_T(\mathcal{G}_{lin}(X, \langle G_t \rangle)) = \frac{D}{2} \sqrt{\sum_{t=1}^{T} G_t^2}.$$

Furthermore, the optimal strategy for the player is to choose

$$\mathbf{x}_{t+1} = \left(\frac{-D}{2\sqrt{\|\mathbf{W}_t\|^2 + \sum_{s=t+1}^{T} G_s}} \right) \mathbf{W}_t.$$

To prove the theorem, we will need a series of short lemmas.

Lemma 14. *When X is the unit ball $B = \{\mathbf{x} : \|\mathbf{x}\| \leq 1\}$, the value V_T can be written as*

$$\inf_{\mathbf{x}_1 \in B} \sup_{\mathbf{w}_1 \in L_1} \dots \inf_{\mathbf{x}_T \in B} \sup_{\mathbf{w}_T \in L_T} \left[\sum_{t=1}^{T} \mathbf{w}_t \cdot \mathbf{x}_t + \|\mathbf{W}_T\| \right] \tag{3.1}$$

In addition, if we choose a larger radius D, the value of the game will scale linearly with this radius and thus it is enough to assume $X = B$.

Proof. The last term in the regret

$$\inf_{\mathbf{x}\in B} \sum_t f_t(\mathbf{x}) = \inf_{\mathbf{x}\in B} \mathbf{W}_T \cdot \mathbf{x} = -\|\mathbf{W}_T\|$$

since the infimum is obtained when $\mathbf{x} = \frac{\mathbf{W}_T}{\|\mathbf{W}_T\|}$. This implies equation (3.1). The fact that the bound scales linearly with $D/2$ follows from the fact that both the norm $\|\mathbf{W}_T\|$ will scale with $D/2$ as well as the terms $\mathbf{w}_t \cdot \mathbf{x}_t$. □

For the remainder of this section, we simply assume that $X = B$, the unit ball with diameter $D = 2$.

Lemma 15. *Regardless of the Player's choices, the Adversary can always obtain regret at least*

$$\sqrt{\sum_{t=1}^{T} G_t^2} \tag{3.2}$$

whenever the dimension n is at least 3.

Proof. Consider the following adversarial strategy and assume $X = B$. On round t, after the Player has chosen $\mathbf{x}_t$, the adversary chooses $\mathbf{w}_t$ such that $\|\mathbf{w}_t\| = G_t$, $\mathbf{w}_t \cdot \mathbf{x}_t = 0$ and $\mathbf{w}_t \cdot \mathbf{W}_{t-1} = 0$. Finding a vector of length G_t that is perpendicular to two arbitrary vectors can always be done when the dimension is at least 3. With this strategy, it is guaranteed that $\sum_t \mathbf{w}_t \cdot \mathbf{x}_t = 0$ and we claim also that

$$\|\mathbf{W}_T\| = \sqrt{\sum_{t=1}^{T} G_t^2}.$$

This follows from a simple induction. Assuming $\|\mathbf{W}_{t-1}\| = \sqrt{\sum_{s=1}^{t-1} G_s^2}$, then

$$\|\mathbf{W}_t\| = \|\mathbf{W}_{t-1} + \mathbf{w}_t\| = \sqrt{\|\mathbf{W}_{t-1}\|^2 + \|\mathbf{w}_t\|^2},$$

implying the desired conclusion. □

The result of the last lemma is quite surprising: the adversary need only play some vector with length G_t which is perpendicular to both $\mathbf{x}_t$ and $\mathbf{W}_{t-1}$. Indeed, this lower bound has a very different flavor from the randomized argument of the previous section. To obtain a full minimax result, all that remains is to show that the Adversary can *do no better!*

Lemma 16. *Let $\mathbf{w}_0 = \mathbf{0}$. If the player always plays the point*

$$\mathbf{x}_t = \frac{-\mathbf{W}_{t-1}}{\sqrt{\|\mathbf{W}_{t-1}\|^2 + \sum_{s=t}^{T} G_s^2}} \tag{3.3}$$

then

$$\sup_{\mathbf{w}_1}\sup_{\mathbf{w}_2}\ldots\sup_{\mathbf{w}_T}\left[\sum_{t=1}^{T}\mathbf{w}_t\cdot\mathbf{x}_t+\|\mathbf{W}_T\|\right]\leq\sqrt{\sum_{t=1}^{T}G_t^2}$$

i.e., the regret can be no greater than the value in (3.2)*.*

Proof. As before, $\mathbf{W}_t=\sum_{s=1}^{t}\mathbf{w}_s$. Define $\Gamma_t^2=\sum_{s=t}^{T}G_s^2$, the forward sum, with $\Gamma_{T+1}=0$. Define

$$\Phi_t(\mathbf{w}_1,\ldots,\mathbf{w}_{t-1})=\sum_{s=1}^{t-1}\mathbf{x}_s\cdot\mathbf{w}_s+\sqrt{\|\mathbf{W}_{t-1}\|^2+\Gamma_t^2}$$

where $\mathbf{x}_t$ is as defined in (3.3) and Φ_1 is $\sqrt{\sum_{t=1}^{T}G_t^2}$. Let

$$V_t(\mathbf{w}_1,\ldots,\mathbf{w}_{t-1})=\sup_{\mathbf{w}_t}\ldots\sup_{\mathbf{w}_T}\left[\sum_{t=1}^{T}\mathbf{w}_t\cdot\mathbf{x}_t+\|\mathbf{W}_T\|\right]$$

be the optimum payoff to the adversary given that he plays $\mathbf{w}_1,\ldots,\mathbf{w}_{t-1}$ in the beginning and then plays optimally. The player plays according to (3.3) throughout. Note that the value of the game is V_1.

We prove by backward induction that, for all $t\in\{1,\ldots,T\}$,

$$V_t(\mathbf{w}_1,\ldots,\mathbf{w}_{t-1})\leq\Phi_t(\mathbf{w}_1,\ldots,\mathbf{w}_{t-1})$$

The base case, $t=T+1$ is obvious. Now assume it holds for $t+1$ and we will prove it for t. We have

$$\begin{aligned}
& V_t(\mathbf{w}_1,\ldots,\mathbf{w}_{t-1}) \\
&= \sup_{\mathbf{w}_t} V_{t+1}(\mathbf{w}_1,\ldots,\mathbf{w}_t) \\
\text{(induc.)}\quad &\leq \sup_{\mathbf{w}_t}\Phi_{t+1}(\mathbf{w}_1,\ldots,\mathbf{w}_t) \\
&= \sum_{s=1}^{t-1}\mathbf{x}_s\cdot\mathbf{w}_s+ \\
(*)\qquad & \qquad \sup_{\mathbf{w}_t}\left[\mathbf{x}_t\cdot\mathbf{w}_t+\sqrt{\|\mathbf{W}_{t-1}+\mathbf{w}_t\|^2+\Gamma_{t+1}^2}\right]
\end{aligned}$$

Let us consider the final supremum term above. If we can show that it is no more than

$$\sqrt{\|\mathbf{W}_{t-1}\|^2+\Gamma_t^2} \tag{3.4}$$

then we will have proved $V_t\leq\Phi_t$ thus completing the induction. This is the objective of the remainder of this proof.

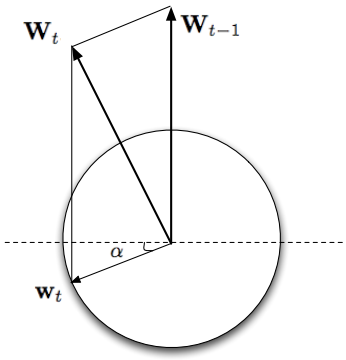

Figure 3.1. Illustration for the proof of the minimax strategy for the ball. We suppose that $\mathbf{x}_t$ is aligned with $\mathbf{W}_{t-1}$ and depict the plane spanned by $\mathbf{W}_{t-1}$ and $\mathbf{w}_t$. We assume that $\mathbf{w}_t$ has angle α with the line perpendicular to $\mathbf{W}_{t-1}$ and show that $\alpha = 0$ is optimal.

We begin by noting two important facts about the expression (*). First, the supremum is taken over a convex function of $\mathbf{w}_t$ and thus the maximum occurs at the boundary, i.e. where $\|\mathbf{w}_t\| = G_t$ exactly. This is easily checked by computing the Hessian with respect to $\mathbf{w}_t$. Second, since $\mathbf{x}_t$ is chosen parallel to $\mathbf{W}_{t-1}$, the only two vectors of interest are $\mathbf{w}_t$ and $\mathbf{W}_{t-1}$. Without loss of generality, we can assume that $\mathbf{W}_{t-1}$ is the 2-dim vector $\langle F, 0\rangle$, where $F = \|\mathbf{W}_{t-1}\|$, and that $\mathbf{w}_t = \langle -G_t \sin\alpha, G_t \cos\alpha\rangle$ for any α. Plugging in the choice of $\mathbf{x}_t$ in (3.3), we may now rewrite (*) as

$$\sup_{\alpha} \underbrace{\frac{FG_t \sin\alpha}{\sqrt{F^2 + G_t^2 + \Gamma_{t+1}^2}} + \sqrt{F^2 + G_t^2 + \Gamma_{t+1}^2 - 2FG_t \sin\alpha}}_{\phi(\alpha)}$$

We illustrate this problem in Figure 3.1. Bounding the above expression requires some care, and thus we prove it in Lemma 24 found in the appendix. The result of Lemma 24 gives us that, indeed,

$$\phi(\alpha) \leq \sqrt{F^2 + G_t^2 + \Gamma_{t+1}^2} = \sqrt{\|\mathbf{W}_{t-1}\|^2 + \Gamma_t^2}.$$

Since (*) is exactly $\sup_\alpha \phi(\alpha)$, which is no greater than

$$\sqrt{F^2 + \Gamma_t^2},$$

we are done.

□

We observe that the minimax strategy for the ball is exactly the Online Gradient Descent strategy of Zinkevich [78]. The value of the game for the ball is exactly the upper bound for the proof of Online Gradient Descent if the initial point is the center of the ball. The lower bound of the randomized argument in the previous section differs from the upper bound for Online Gradient Descent by $\sqrt{2}$.

3.5 The Quadratic Game

As in the last section, we now give a minimax analysis of the game $\mathcal{G}_{\text{quad}}$. Ultimately we will be able to compute the exact value of $V_T(\mathcal{G}_{\text{quad}}(X, \langle G_t \rangle, \langle \sigma_t \rangle))$ and provide the optimal strategy of both the Player and the Adversary. What is perhaps most interesting is that the optimal Player strategy is the well-known Follow The Leader approach. This general strategy can be defined simply as

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x} \in X} \sum_{s=1}^{t} f_s(\mathbf{x});$$

that is, we choose the best $\mathbf{x}$ "in hindsight". As has been pointed out by several authors, this strategy can incur $\Omega(T)$ regret when the loss functions are linear. It is thus quite surprising that this strategy is optimal when instead we are competing against quadratic loss functions.

For this section, define $F_t(\mathbf{x}) := \sum_{s=1}^{t} f_s(\mathbf{x})$ and $\mathbf{x}_t^* := \arg\min_{\mathbf{x}} F_t(\mathbf{x})$. Define $\boldsymbol{\sigma}_{1:t} = \sum_{s=1}^{t} \sigma_s$. We assume from the outset that $\sigma_1 > 0$. We also set $\boldsymbol{\sigma}_{1:0} = 0$.

3.5.1 A Necessary Restriction

Recall that the upper bound in Hazan et al. [48] is

$$R_T \leq \frac{1}{2} \frac{G^2}{\sigma} \log T$$

and note that this expression has no dependence on the size of X. We would thus ideally like to consider the case when $X = \mathcal{R}^n$, for this would seem to be the "hardest" case for the Player. The unbounded assumption is problematic, however, not because the game is too difficult for the Player, but the game is *too difficult for the Adversary!*. This ought to come as quite a surprise, but arises from the particular restrictions we place on the Adversary.

Proposition 17. *For $G, \sigma > 0$, if $\max_{\mathbf{x},\mathbf{y} \in X} \|\mathbf{x} - \mathbf{y}\| = D > 4G/\sigma$, there is an $\alpha > 0$ such that $V_T(\mathcal{G}_{quad}(X, G, \sigma)) \leq -\alpha T$.*

Proof. Fix $\mathbf{x}_o, \mathbf{x}_e \in X$ with $\|\mathbf{x}_o - \mathbf{x}_e\| > 4G/\sigma$. Consider a player that plays $\mathbf{x}_{2k-1} = \mathbf{x}_o$, $\mathbf{x}_{2k} = \mathbf{x}_e$. Then for any $\mathbf{x} \in X$,

$$f_{2k-1}(\mathbf{x}) \geq f_{2k-1}(\mathbf{x}_o) - G\|\mathbf{x} - \mathbf{x}_o\| + \frac{\sigma}{2}\|\mathbf{x} - \mathbf{x}_o\|^2,$$

And similarly for f_{2k} and $\mathbf{x}_e$. Summing over t (assuming that T is even) shows that $V_t(\mathcal{G}_{\text{quad}}(X, G, \sigma))$ is no more than

$$\sum_{t=1}^{T} f_t(\mathbf{x}_t) - \sum_{t=1}^{T} f_t(\mathbf{x}) \leq \frac{T}{2} \Big(G\|\mathbf{x} - \mathbf{x}_o\| - \frac{\sigma}{2}\|\mathbf{x} - \mathbf{x}_o\|^2 + G\|\mathbf{x} - \mathbf{x}_e\| - \frac{\sigma}{2}\|\mathbf{x} - \mathbf{x}_e\|^2 \Big).$$

But by the triangle inequality, any $\mathbf{x} \in X$ has $\|\mathbf{x} - \mathbf{x}_o\| + \|\mathbf{x} - \mathbf{x}_e\| \geq D$. Subject to this constraint, plus the constraints $0 \leq \|\mathbf{x} - \mathbf{x}_o\| \leq D, 0 \leq \|\mathbf{x} - \mathbf{x}_e\| \leq D$ shows that $V_t(\mathcal{G}_{\text{quad}}(X, G, \sigma)) \leq T(GD - \sigma D^2/4)/2 \leq -\alpha T$ for some $\alpha > 0$, since $D > 4G/\sigma$. □

As we don't generally expect regret to be negative, this example suggests that the Quadratic Game is uninteresting without further constraints on the Player. While an explicit bound on the size of X is a possibility, it is easier for the analysis to place a slightly weaker restriction on the Player.

Assumption 3.5.1. *Let $\mathbf{x}_{t-1}^*$ be the minimizer of $F_{t-1}(\mathbf{x})$. We assume that the Player must choose $\mathbf{x}_t$ such that*

$$\sigma_t \|\mathbf{x}_t - \mathbf{x}_{t-1}^*\| < 2G_t.$$

This restriction is necessary for non-negative regret. Indeed, it can be shown that if we increase the size of the above ball by only ε, the method of Proposition 17 above shows that the regret will be negative for large enough T.

3.5.2 Minimax Analysis

With the above restriction in place, we now simply write the game as $\mathcal{G}'_{\text{quad}}(\langle G_t \rangle, \langle \sigma_t \rangle)$, omitting the input X. We now proceed to compute the value of this game exactly.

Theorem 18. *Under Assumption 3.5.1, the value of the game*

$$V_T(\mathcal{G}'_{quad}(\langle G_t \rangle, \langle \sigma_t \rangle)) = \sum_{t=1}^{T} \frac{G_t^2}{2\boldsymbol{\sigma}_{1:t}}.$$

With uniform G_t and σ_t, we obtain the harmonic series, giving us our logarithmic regret bound. We note that this is *exactly* the upper bound proven in [15, 48], even with the constant.

Corollary 19. *For the uniform parameters of the game,*

$$\frac{G}{2\sigma} \log(T+1) \leq V_T(\mathcal{G}'_{quad}(G, \sigma)) \leq \frac{G}{2\sigma}(1 + \log T).$$

The main argument in the proof of Theorem 18 boils down to reducing the multiple round game to a single round game. The following lemma gives the value of this single round game. Since the proof is somewhat technical, we postpone it to the Appendix.

Lemma 20. *For arbitrary $G_t, \sigma_t, \boldsymbol{\sigma}_{1:t-1} > 0$,*

$$\inf_{\Delta : \|\Delta\| \leq \frac{2G_t}{\sigma_t}} \sup_{\delta} \left(G_t \|\Delta - \delta\| - \tfrac{1}{2}\sigma_t \|\Delta - \delta\|^2 - \tfrac{1}{2}\boldsymbol{\sigma}_{1:t-1} \|\delta\|^2 \right)$$
$$= \frac{G_t^2}{2\boldsymbol{\sigma}_{1:t}} = \frac{G_t^2}{2\boldsymbol{\sigma}_{1:t}},$$

and indeed the optimal strategy pair is $\Delta = \mathbf{0}$ and δ any vector for which $\|\delta\| = \frac{G_t}{\boldsymbol{\sigma}_{1:t}}$.

We now show how to "unwind" the recursive inf sup definition of $V_T(\mathcal{G}'_{\text{quad}}(\langle G_t \rangle, \langle \sigma_t \rangle))$, where the final term we chop off is the object we described in the above lemma.

Proof of Theorem 18. Let $\mathbf{x}^*_{t-1}$ be the minimizer of $F_{t-1}(\mathbf{x})$ and $\mathbf{z} \in X$ be arbitrary. Note that F_t is $\boldsymbol{\sigma}_{1:t}$-quadratic, so

$$\begin{aligned} F_t(\mathbf{z}) &= F_{t-1}(\mathbf{z}) + f_t(\mathbf{z}) \\ &= F_{t-1}(\mathbf{x}^*_{t-1} + (\mathbf{z} - \mathbf{x}^*_{t-1})) + f_t(\mathbf{z}) \\ &= F_{t-1}(\mathbf{x}^*_{t-1}) + \nabla F_{t-1}(\mathbf{x}^*_{t-1})(\mathbf{z} - \mathbf{x}^*_{t-1}) \\ &\quad + \tfrac{1}{2}\boldsymbol{\sigma}_{1:t-1}\|\mathbf{z} - \mathbf{x}^*_{t-1}\|^2 + f_t(\mathbf{z}) \\ &= F_{t-1}(\mathbf{x}^*_{t-1}) + \tfrac{1}{2}\boldsymbol{\sigma}_{1:t-1}\|\mathbf{z} - \mathbf{x}^*_{t-1}\|^2 + f_t(\mathbf{z}), \end{aligned}$$

where the last equality holds by the definition of $\mathbf{x}^*_{t-1}$. Hence,

$$\begin{aligned} \sum_{s=1}^{t} f_s(\mathbf{x}_s) - F_t(\mathbf{z}) = &\left(\sum_{s=1}^{t-1} f_s(\mathbf{x}_s) - F_{t-1}(\mathbf{x}^*_{t-1}) \right) \\ &+ \left(f_t(\mathbf{x}_t) - f_t(\mathbf{z}) - \tfrac{1}{2}\boldsymbol{\sigma}_{1:t-1}\|\mathbf{z} - \mathbf{x}^*_{t-1}\|^2 \right). \end{aligned}$$

Expanding f_t around $\mathbf{x}_t$,

$$f_t(\mathbf{x}_t) - f_t(\mathbf{z}) = -\nabla f_t(\mathbf{x}_t)(\mathbf{z} - \mathbf{x}_t) - \tfrac{1}{2}\sigma_t\|\mathbf{z} - \mathbf{x}_t\|^2.$$

Substituting,

$$\begin{aligned} \sum_{s=1}^{t} f_s(\mathbf{x}_s) - F_t(\mathbf{z}) = &\left(\sum_{s=1}^{t-1} f_s(\mathbf{x}_s) - F_{t-1}(\mathbf{x}^*_{t-1}) \right) \\ &+ \left(\nabla f_t(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{z}) - \tfrac{1}{2}\sigma_t\|\mathbf{z} - \mathbf{x}_t\|^2 - \tfrac{1}{2}\boldsymbol{\sigma}_{1:t-1}\|\mathbf{z} - \mathbf{x}^*_{t-1}\|^2 \right). \end{aligned}$$

Then

$$\begin{aligned} V_t &:= \inf_{\mathbf{x}_1} \sup_{f_1} \cdots \inf_{\mathbf{x}_t} \sup_{f_t} \left(\sum_{s=1}^{t} f_s(\mathbf{x}_s) - \inf_{\mathbf{z}} F_t(\mathbf{z}) \right) \\ &= \inf_{\mathbf{x}_1} \sup_{f_1} \cdots \inf_{\mathbf{x}_t} \sup_{f_t, \mathbf{z}} \left(\sum_{s=1}^{t} f_s(\mathbf{x}_s) - F_t(\mathbf{z}) \right) \\ &= \inf_{\mathbf{x}_1} \sup_{f_1} \cdots \inf_{\mathbf{x}_{t-1}} \sup_{f_{t-1}} \left[\left(\sum_{s=1}^{t-1} f_s(\mathbf{x}_s) - F_{t-1}(\mathbf{x}^*_{t-1}) \right) \right. \\ &\qquad + \inf_{\mathbf{x}_t} \sup_{f_t, \mathbf{z}} \left(\nabla f_t(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{z}) - \tfrac{1}{2}\sigma_t\|\mathbf{z} - \mathbf{x}_t\|^2 \right. \\ &\qquad\qquad \left. \left. - \tfrac{1}{2}\boldsymbol{\sigma}_{1:t-1}\|\mathbf{z} - \mathbf{x}^*_{t-1}\|^2 \right) \right]. \end{aligned}$$

However, we can simplify the final inf sup as follows. We note that the quantity $\nabla f_t(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{z})$ is maximized when $\nabla f_t(\mathbf{x}_t) = G_t \frac{\mathbf{x}_t - \mathbf{z}}{\|\mathbf{x}_t - \mathbf{z}\|}$. Second, we can instead use the variables $\Delta = \mathbf{x}_t - \mathbf{x}^*_{t-1}$ and $\delta = \mathbf{z} - \mathbf{x}^*_{t-1}$ in the optimization. Recall from Assumption 3.5.1 that $\|\mathbf{x}_t - \mathbf{x}^*_{t-1}\| = \|\Delta\| \leq \frac{2G_t}{\sigma_t}$. Then,

$$\begin{aligned}
V_t &= \inf_{\mathbf{x}_1} \sup_{f_1} \dots \inf_{\mathbf{x}_{t-1}} \sup_{f_{t-1}} \left[\left(\sum_{s=1}^{t-1} f_s(\mathbf{x}_s) - F_{t-1}(\mathbf{x}^*_{t-1}) \right) \right. \\
&\quad + \inf_{\Delta : \|\Delta\| \leq \frac{2G_t}{\sigma_t}} \sup_{\delta} \left(G_t \|\Delta - \delta\| \right. \\
&\qquad \left. \left. - \tfrac{1}{2}\sigma_t \|\Delta - \delta\|^2 - \tfrac{1}{2}\boldsymbol{\sigma}_{1:t-1} \|\delta\|^2 \right) \right] \\
&= \inf_{\mathbf{x}_1} \sup_{f_1} \dots \inf_{\mathbf{x}_{t-1}} \\
&\qquad \sup_{f_{t-1}} \left[\left(\sum_{s=1}^{t-1} f_s(\mathbf{x}_s) - F_{t-1}(\mathbf{x}^*_{t-1}) \right) + \frac{G_t^2}{2\boldsymbol{\sigma}_{1:t}} \right] \\
&= V_{t-1} + \frac{G_t^2}{2\boldsymbol{\sigma}_{1:t}},
\end{aligned}$$

where the second equality is obtained by applying Lemma 20. Unwinding the recursion proves the theorem.

□

Corollary 21. *The optimal Player strategy is to set $\mathbf{x}_t = \mathbf{x}^*_{t-1}$ on each round.*

Proof. In analyzing the game, we found that the optimal choice of $\Delta = \mathbf{x}_t - \mathbf{x}^*_{t-1}$ was shown to be $\mathbf{0}$ in Lemma 20. □

3.6 General Games

While the minimax results shown above are certainly interesting, we have only shown them to hold for the rather restricted games $\mathcal{G}_{\text{lin}}$ and $\mathcal{G}_{\text{quad}}$. For these particular cases, the class of functions that the Adversary may choose from is quite small: both the set of linear functions and the set quadratic functions can be parameterized by $O(n)$ variables. It would of course be more satisfying if our minimax analyses held for more richer loss function spaces.

Indeed, we prove in this section that both of our minimax results hold much more generally. In particular, we prove that even if the Adversary were able to choose *any* convex function on round t, with derivative bounded by G_t, then he can do no better than if he only had access to linear functions. On a similar note, if the Adversary is given the weak restriction that his functions be σ_t-strongly convex on round t, then he can do no better than if he could only choose σ_t-quadratic functions.

Theorem 22. *For fixed $X, \langle G_t \rangle$, and $\langle \sigma_t \rangle$, the values of the Quadratic Game and the Strongly Convex Game are equal*[2]*:*

$$V_T(\mathcal{G}_{st\text{-}conv}(X, \langle G_t \rangle, \langle \sigma_t \rangle)) = V_T(\mathcal{G}_{quad}(X, \langle G_t \rangle, \langle \sigma_t \rangle)).$$

For a fixed X and $\langle G_t \rangle$, the values of the Convex Game and the Linear Game are equal:

$$V_T(\mathcal{G}_{conv}(X, \langle G_t \rangle)) = V_T(\mathcal{G}_{lin}(X, \langle G_t \rangle)).$$

We need the following lemma whose proof is postponed to the appendix. Define the regret function

$$R(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_T, f_T) = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{x \in X} \sum_{t=1}^{T} f_t(\mathbf{x}).$$

Lemma 23. *Consider a sequence of sets $\{N_s\}_{s=1}^T$ and $M \subseteq N_t$ for some t. Suppose that for all $f_t \in N_t$ and $\mathbf{x}_t \in X$ there exists $f_t^* \in M$ such that for all*

$$(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_{t-1}, f_{t-1}, \mathbf{x}_{t+1}, f_{t+1}, \ldots, \mathbf{x}_T, f_T),$$

$$\begin{aligned} &R(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t, \ldots, \mathbf{x}_T, f_T) \\ &\quad \leq R(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t^*, \ldots, \ldots, \mathbf{x}_T, f_T). \end{aligned}$$

Then

$$\begin{aligned} &\inf_{\mathbf{x}_1} \sup_{f_1 \in N_1} \ldots \inf_{\mathbf{x}_t} \sup_{f_t \in N_t} \ldots \inf_{\mathbf{x}_T} \sup_{f_T \in N_T} R(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_T, f_T) \\ &= \inf_{\mathbf{x}_1} \sup_{f_1 \in N_1} \ldots \inf_{\mathbf{x}_t} \sup_{f_t \in M} \ldots \inf_{\mathbf{x}_T} \sup_{f_T \in N_T} R(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_T, f_T). \end{aligned}$$

Proof of Theorem 22. Given the sequences $\langle G_t \rangle, \langle \sigma_t \rangle$, let $L_t(\mathbf{x}_t)$ be defined as for the Strongly Convex Game (Definition 10) and $L_t^*(\mathbf{x}_t)$ be defined as for the Quadratic Game (Definition 9). Observe that $L_t^* \subseteq L_t$ for any t. Moreover, for any $f_t \in L_t$ and $\mathbf{x}_t \in X$, define $f_t^*(\mathbf{x}) = f_t(\mathbf{x}_t) + \nabla f_t(\mathbf{x}_t)^\top (\mathbf{x} - \mathbf{x}_t) + \frac{1}{2}\sigma_t \|\mathbf{x} - \mathbf{x}_t\|^2$. By definition, $f_t(\mathbf{x}_t) = f_t^*(\mathbf{x}_t)$ and $\nabla f_t(\mathbf{x}_t) = \nabla f_t^*(\mathbf{x}_t)$. Hence, $f_t^* \in L_t^*$. Furthermore, $f_t(\mathbf{x}) \geq f_t^*(\mathbf{x})$ for any $\mathbf{x} \in X$, and $\mathbf{x}^*$ in particular. Hence, for all $(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_{t-1}, f_{t-1}, \mathbf{x}_{t+1}, f_{t+1}, \ldots, \mathbf{x}_T, f_T)$,

$$\begin{aligned} &R(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t, \ldots, \mathbf{x}_T, f_T) \\ &\leq R(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t^*, \ldots, \ldots, \mathbf{x}_T, f_T). \end{aligned}$$

The statement of the first part of the theorem follows by Lemma 23, applied for every $t \in \{1, \ldots, T\}$. The second part is proved by analogous reasoning. □

[2] *We note that the computation of V_T for the Quadratic Game required a particular restriction on the player, Assumption 3.5.1, where here we only consider a fixed domain X.*

Appendix

Proof of Lemma 20. We write

$$P_t(\Delta, \delta) := G_t\|\Delta - \delta\| - \tfrac{1}{2}\sigma_t\|\Delta - \delta\|^2 - \tfrac{1}{2}\boldsymbol{\sigma}_{1:t-1}\|\delta\|^2$$

and

$$Q_t(\Delta) := \sup_{\delta} P_t(\Delta, \delta),$$

then our goal is to obtain $\inf_{\Delta:\|\Delta\| \le \frac{2G_t}{\sigma_t}} Q_t(\Delta)$. We now proceed to show that the choice $\Delta = \mathbf{0}$ is optimal. For this choice,

$$Q_t(\mathbf{0}) = \sup_{\delta}\ G_t\|\delta\| - \tfrac{1}{2}\boldsymbol{\sigma}_{1:t}\|\delta\|^2 = \frac{G_t^2}{2\boldsymbol{\sigma}_{1:t}}.$$

Here the optimal choice of δ is any vector such that $\|\delta\| = \frac{G_t}{\boldsymbol{\sigma}_{1:t}}$.

Now let us consider the case that $\Delta \neq \mathbf{0}$. First, suppose $\Delta \neq \delta$. Note that the optimum $\sup_\delta P_t(\Delta, \delta)$ will be obtained when the gradient with respect to δ is zero, i.e.

$$-G_t\frac{\Delta - \delta}{\|\Delta - \delta\|} - \sigma_t(\delta - \Delta) - \boldsymbol{\sigma}_{1:t-1}\delta = \mathbf{0}$$

implying that δ is a linear scaling of Δ, i.e. $\delta = c\Delta$. The second case, $\Delta = \delta$, also implies that δ is a linear scaling of Δ. Substituting this optimal form of δ,

$$\begin{aligned} Q_t(\Delta) = \sup_{c \in \mathcal{R}} \big[&G_t|1 - c| \cdot \|\Delta\| \\ &- \tfrac{1}{2}\sigma_t(1 - c)^2\|\Delta\|^2 - \tfrac{1}{2}\boldsymbol{\sigma}_{1:t-1}c^2\|\Delta\|^2\Big]. \end{aligned}$$

We now claim that the supremum over $c \in \mathcal{R}$ occurs at some $c^* \le 1$ for any choice of Δ. Assume by contradiction that $c^* > 1$ for some Δ. Then $\tilde{c} = -c^* + 2$ achieves at least the same value as c^* since $|1 - c^*| = |1 - \tilde{c}|$ while $(c^*)^2 > (\tilde{c})^2$, making the last term larger, which is a contradiction. Hence, $c \le 1$ and, collecting the terms,

$$\begin{aligned} Q_t(\Delta) = \sup_{c \le 1} \Big[&\Big(G_t\|\Delta\| - \tfrac{1}{2}\sigma_t\|\Delta\|^2\Big) \\ &+ c \cdot \big(\sigma_t\|\Delta\|^2 - G_t\|\Delta\|\big) - c^2 \cdot \Big(\tfrac{1}{2}\boldsymbol{\sigma}_{1:t}\|\Delta\|^2\Big)\Big]. \end{aligned}$$

Since we now assume $\|\Delta\| \neq \mathbf{0}$, we see that the supremum is achieved for $c^* =$

$\frac{\sigma_t\|\Delta\|^2 - G_t\|\Delta\|}{\boldsymbol{\sigma}_{1:t}\|\Delta\|^2} = \frac{\sigma_t\|\Delta\| - G_t}{\boldsymbol{\sigma}_{1:t}\|\Delta\|} \leq 1$ and

$$\begin{aligned}
Q_t(\Delta) &= \frac{\left(\sigma_t\|\Delta\|^2 - G_t\|\Delta\|\right)^2}{2\boldsymbol{\sigma}_{1:t}\|\Delta\|^2} + \left(G_t\|\Delta\| - \tfrac{1}{2}\sigma_t\|\Delta\|^2\right) \\
&= \frac{\sigma_t^2\|\Delta\|^2 - \sigma_t\|\Delta\|G_t + G_t^2}{2\boldsymbol{\sigma}_{1:t}} \\
&\qquad + \left(G_t\|\Delta\| - \tfrac{1}{2}\sigma_t\|\Delta\|^2\right) \\
&= \frac{\sigma_t}{\boldsymbol{\sigma}_{1:t}}\left(\tfrac{1}{2}\sigma_t\|\Delta\|^2 - \|\Delta\|G_t\right) \\
&\qquad + \left(G_t\|\Delta\| - \tfrac{1}{2}\sigma_t\|\Delta\|^2\right) + \frac{G_t^2}{2\boldsymbol{\sigma}_{1:t}} \\
&= \frac{\boldsymbol{\sigma}_{1:t-1}}{\boldsymbol{\sigma}_{1:t}}\left(G_t - \tfrac{1}{2}\sigma_t\|\Delta\|\right)\|\Delta\| + \frac{G_t^2}{2\boldsymbol{\sigma}_{1:t}} > \frac{G_t^2}{2\boldsymbol{\sigma}_{1:t}},
\end{aligned}$$

where the last inequality holds by because $\|\Delta\| \leq \frac{2G_t}{\sigma_t}$. Hence, the value $Q_t(\Delta)$ is strictly larger than $G_t^2/(2\boldsymbol{\sigma}_{1:t})$ whenever $\|\Delta\| > 0$ and is equal to this value if $\Delta = \mathbf{0}$. Hence, the optimal choice for the Player is to choose $\Delta = \mathbf{0}$. □

Proof of Lemma 23. Fix $f_t \in L_t$ and $\mathbf{x}_t \in X$. Let $f_t^* \in M$ be as in the statement of the lemma. Define

$$\begin{aligned}
&h_1(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_{t-1}, f_{t-1}, \mathbf{x}_{t+1}, f_{t+1}, \ldots, \mathbf{x}_T, f_T) \\
&\qquad := R(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t, \ldots, \mathbf{x}_T, f_T)
\end{aligned}$$

$$\begin{aligned}
&h_2(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_{t-1}, f_{t-1}, \mathbf{x}_{t+1}, f_{t+1}, \ldots, \mathbf{x}_T, f_T) \\
&\qquad := R(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t^*, \ldots, \mathbf{x}_T, f_T).
\end{aligned}$$

By assumption, $h_1 \leq h_2$. Hence, we can inf/sup over the variables $\mathbf{x}_{t+1}, f_{t+1}, \ldots, \mathbf{x}_T, f_T$, obtaining

$$\begin{aligned}
&\inf_{\mathbf{x}_{t+1}} \sup_{f_{t+1}\in N_{t+1}} \cdots \inf_{\mathbf{x}_T} \\
&\qquad \sup_{f_T \in N_T} R(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t, \ldots, \mathbf{x}_T, f_T) \\
&\leq \inf_{\mathbf{x}_{t+1}} \sup_{f_{t+1}\in N_{t+1}} \cdots \inf_{\mathbf{x}_T} \\
&\qquad \sup_{f_T \in N_T} R(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t^*, \ldots, \mathbf{x}_T, f_T)
\end{aligned}$$

for any $(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_{t-1}, f_{t-1})$. Hence, since $f_t^* \in M$

$$\begin{aligned}
&\sup_{f_t\in N_t} \inf_{\mathbf{x}_{t+1}} \sup_{f_{t+1}\in N_{t+1}} \cdots \inf_{\mathbf{x}_T} \\
&\qquad \sup_{f_T \in N_T} R(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t, \ldots, \mathbf{x}_T, f_T) \\
&\leq \sup_{f_t\in M} \inf_{\mathbf{x}_{t+1}} \sup_{f_{t+1}\in N_{t+1}} \cdots \inf_{\mathbf{x}_T} \\
&\qquad \sup_{f_T \in N_T} R(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_t, f_t, \ldots, \mathbf{x}_T, f_T)
\end{aligned}$$

for all $(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_{t-1}, f_{t-1}, \mathbf{x}_t)$. Since $M \subseteq N_t$, the above is in fact an equality. Since the two functions of the variables $(\mathbf{x}_1, f_1, \ldots, \mathbf{x}_{t-1}, f_{t-1}, \mathbf{x}_t)$ are equal, taking inf's and sup's over these variables we obtain the statement of the lemma. □

Lemma 24. *The expression*

$$\frac{FG\sin\alpha}{\sqrt{F^2+G^2+K^2}} + \sqrt{F^2+G^2+K^2-2FG\sin\alpha}$$

is no more than $\sqrt{F^2+G^2+K^2}$ *for constants* $F, G, K > 0$ *and any* α.

Proof. We are interested in proving that the supremum of

$$\phi(\alpha) = \frac{FG\sin\alpha}{\sqrt{F^2+G^2+K^2}} + \sqrt{F^2+G^2+K^2-2FG\sin\alpha}$$

over $[-\pi/2, \pi/2]$ is attained at $\alpha = 0$. Setting the derivative of $\Phi(\alpha)$ to zero,

$$\frac{FG\cos\alpha}{\sqrt{F^2+G^2+K^2}} - \frac{FG\cos\alpha}{\sqrt{F^2+G^2+K^2-2FG\sin\alpha}} = 0$$

which implies that either $\cos\alpha = 0$ or $\sin\alpha = 0$, i.e. $\alpha \in \{-\pi/2, 0, \pi/2\}$. Taking the second derivative, we get

$$\begin{aligned}\phi''(\alpha) = &-\frac{FG\sin\alpha}{\sqrt{F^2+G^2+K^2}} \\ &- \left(-\frac{FG\sin\alpha}{\sqrt{F^2+G^2+K^2-2FG\sin\alpha}}\right. \\ &\qquad \left. + \frac{(FG\cos\alpha)(FG\cos\alpha)}{(F^2+G^2+K^2-2FG\sin\alpha)^{3/2}}\right).\end{aligned}$$

Thus, $\phi''(0) < 0$. We conclude that the optimum is attained at $\alpha = 0$ and therefore

$$\phi(\alpha) \leq \sqrt{F^2+G^2+K^2}$$

□

Chapter 4

Online Linear Optimization in the Bandit Setting

We stick with our focus on the problem of Online Linear Optimization, but we now consider the so-called *Bandit* version of the problem. In this version, once the decision maker commits to his choice $\mathbf{x}_t$ he does not have access to the linear cost functions $\mathbf{f}_t$ chosen by Adversary and, instead, observes only the scalar-valued cost $\mathbf{f}_t^\top \mathbf{x}_t$. The Bandit setting is obviously more difficult, as the Player has strictly less information, yet under many circumstances it is the more realistic problem. A classic example of this problem is often referred to as the *bandit shortest path* problem, also known as the "driving to working problem," which we now sketch.

Formally, the bandit shortest path problem is defined as the following repeated game. Given a directed graph $G = (V, E)$ and a source-sink pair $s, t \in V$, at each time step $t = 1$ to T,

- Player chooses a path $p_t \in \mathcal{P}_{s,t}$, where $\mathcal{P}_{s,t} \subseteq \{0,1\}^E$ is all s,t-paths in the graph
- Adversary independently chooses weights on the edges of the graph $\mathbf{f}_t \in \mathbb{R}^m$
- Player suffers and observes loss, which is the weighted length of the chosen path $\sum_{e \in p_t} \mathbf{f}_t(e)$

The problem is transformed into an instance of bandit linear optimization by associating each path with a vector $\mathbf{x} \in \{0,1\}^{|E|}$, where $\mathbf{x}(i)$ indicates the presence of the ith edge. The loss is then defined through the dot product $\mathbf{f}^\top \mathbf{x}$. Define the set $\mathcal{K}$ as the convex hull of the set of paths. It is well-known that this set is the set of flows in the graph and can be defined using $O(m)$ constraints: positivity constraints and conservation of in-flow and out-flow for every vertex other than source/sink (which have unit out-flow and in-flow, respectively). Hence, to convert this into a problem of Online Linear Optimization, we must allow the decision maker to choose $\mathbf{x}_t$ in the convex hull, which is equivalent to a flow or "randomized path" in the graph. Of course, in a shortest path problem we need to obtain a discrete path and not a flow, but the latter can be accomplished (efficiently) via randomized rounding.

Why should we consider this to be a key example for the Bandit setting? If we think about a fellow that must choose a route from home to work on each of a sequence of days, the feedback this commuter receives on each day is quite limited, he only observes the traffic on the selected route. Unfortunately, he does not have access to the traffic patterns on the roads not included in his chosen route.

This Bandit Linear Optimization problem is very well-studied [26, 27, 9, 33, 10, 56, 14] and many algorithms have been proposed. The result in the present Chapter settled an open problem proposed by Awerbuch and Kleinberg [11], who asked whether there is an *efficient* algorithm for Bandit Linear Optimization which achieve the optimal $\Theta(\sqrt{T})$ regret bound, and we answer this in the affirmative with a constructive solution.

A somewhat surprising fact about the result is that we required using tools from the optimization literature, from the area known as Interior Point Methods (IPM). In particular, we introduce the use of *self-concordant barrier functions* as a regularizer for the FTRL problem. Our proposed algorithm and proof require several tools from this area, and we begin with a summary of these techniques and a handful of helpful existing results developed by the IPM community.

4.1 Convex Optimization: Self-concordant Barriers and the Dikin ellipsoid

An unconstrained convex optimization problem consists of finding the value $\mathbf{x} \in \mathbb{R}^n$ that minimizes some given convex objective $g(\mathbf{x})$. Unconstrained optimization has generally been considered an "easy" problem, as straightforward techniques such as gradient descent and Newton's Method can be readily applied, and the solution admits a simple certificate, namely when $\nabla g = \mathbf{0}$. On the other hand, when the objective $g()$ must be minimized on some convex set $\mathcal{K}$, known as constrained optimization, the problem becomes significantly more difficult.

Interior-point methods were designed for precisely this problem and they are arguably one of the greatest achievements in the field of Convex Optimization in the past two decades. These iterative polynomial-time algorithms for Convex Optimization find the solution by adding a barrier function to the objective such that the barrier diverges at the boundary of the set. We may now interpret the resulting optimization problem, on the modified objective function, as an unconstrained minimization problem which, as mentioned, can now be solved quickly. Roughly speaking, this approximate solution can be iteratively improved by gradually reducing the weight of the barrier function as one approaches the true optimum. In work pioneered by Karmarkar in the context of linear programming [50], and greatly generalized to constrained convex optimization by Nesterov and Nemirovskii, it has been shown that this technique admits a polynomial-time complexity as long as the barrier function is *self-concordant*, a property we soon define explicitly.

In the present work we will borrow several tools from the Interior-point literature, foremost among these is the use of self-concordant barrier functions. The utility of such functions

is somewhat surprising, as our ultimate goal is not polynomial-time complexity but rather low-regret learning algorithms. While learning algorithms often involved adding "regularization" to a particular objective function, for the special case of learning with "bandit" feedback , as we shall see in Section 4.3, the self-concordant regularizer provides the missing piece in obtaining a near-optimal regret guarantee.

The construction of barrier functions for general convex sets has been studied extensively, and we refer the reader to [62, 16] for a thorough treatment on the subject. To be more precise, most of the results of this section can be found in [61], page 22-23, as well as in the aforementioned texts. We also refer the reader to the survey of Nemirovskii and Todd [60].

4.1.1 Definitions and Properties

In what follows, we list the relevant definitions and results on the theory of Interior Point Methods that will be used later in the present chapter. Let $\mathcal{K} \subset \mathbb{R}^n$ be a convex compact set with non-empty interior $\text{int}(\mathcal{K})$.

Basic Properties of Self-concordant Functions

Definition 25. *A* self-concordant function $\mathcal{R} : int(\mathcal{K}) \to \mathbb{R}$ *is a* C^3 *convex function such that*

$$|D^3\mathcal{R}(\mathbf{x})[\mathbf{h},\mathbf{h},\mathbf{h}]| \leq 2\left(D^2\mathcal{R}(\mathbf{x})[\mathbf{h},\mathbf{h}]\right)^{3/2}.$$

A ϑ-self-concordant barrier $\mathcal{R}$ *is a self-concordant function with*

$$|D\mathcal{R}(\mathbf{x})[\mathbf{h}]| \leq \vartheta^{1/2}\left[D^2\mathcal{R}(\mathbf{x})[\mathbf{h},\mathbf{h}]\right]^{1/2}.$$

Here, the third-order differential is defined as

$$D^3\mathcal{R}(\mathbf{x})[\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3] := \frac{\partial^3}{\partial t_1 \partial t_2 \partial t_3}|_{t_1=t_2=t_3=0}\mathcal{R}(\mathbf{x}+t_1\mathbf{h}_1+t_2\mathbf{h}_2+t_3\mathbf{h}_3).$$

We will further assume that the function approaches infinity for any sequence of points approaching the boundary of $\mathcal{K}$. Also, since $\mathcal{K}$ is compact, we can assume that $\mathcal{R}$ is non-degenerate.

A central fact about interior point methods is that they can be applied quite generally, as any arbitrary n-dimensional closed convex set admits an $O(n)$-self-concordant barrier [62]. Hence, throughout this chapter, $\vartheta = O(n)$, but can even be independent of the dimension, as for the sphere.

As self-concordant functions are used as a tool in optimization via iterative updates, there are a few objects used to "measure" the region around every point $\mathbf{x} \in K$ as well as the progress of the optimization.

Definition 26. *Let $\mathcal{R}$ be a self-concordant function. For any $\mathbf{x} \in int(\mathcal{K})$, we define an associated norm $\|\cdot\|_{\mathbf{x}}$ as*

$$\|\mathbf{h}\|_{\mathbf{x}} = (\mathbf{h}^\top \nabla^2 \mathcal{R}(\mathbf{x})\mathbf{h})^{1/2}.$$

We can also define $\|\cdot\|_{\mathbf{x}}^*$, the dual norm *to $\|\cdot\|_{\mathbf{x}}$, as*[1]

$$\|\mathbf{h}\|_{\mathbf{x}}^* = (\mathbf{h}^\top (\nabla^2 \mathcal{R}(\mathbf{x}))^{-1}\mathbf{h})^{1/2}.$$

For any $\mathbf{x} \in int(\mathcal{K})$ we define the Dikin ellipsoid *of radius r*

$$W_r(\mathbf{x}) := \{\mathbf{y} \in \mathcal{K} : \|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} < r\},$$

that is, the $\|\cdot\|_{\mathbf{x}}$-norm ball around $\mathbf{x}$. Finally, we define the Newton decrement *for $\mathcal{R}$ at $\mathbf{x}$ as*

$$\lambda(\mathbf{x}, \mathcal{R}) := \|\nabla \mathcal{R}(\mathbf{x})\|_{\mathbf{x}}^* = \|\nabla^2 \mathcal{R}(\mathbf{x})^{-1} \nabla \mathcal{R}(\mathbf{x})\|_{\mathbf{x}}$$

When we use the term Dikin Ellipsoid it will be implied that the radius is 1 unless otherwise noted. This ellipsoid $W_1(\mathbf{x})$ is a key piece of our main result, in particular due to the following nontrivial fact (See Nemirovskii [61] on page 23 for proof):

$$\forall\, \mathbf{x} \in \text{int}(\mathcal{K}) \quad W_1(\mathbf{x}) \subset \mathcal{K}. \tag{4.1}$$

In other words, the inverse Hessian of the self-concordant function $\mathcal{R}$ stretches the space in such a way that the eigenvectors of $\nabla^2 \mathcal{R}^{-1}$ fall in the set $\mathcal{K}$.

Self-concordant functions are used as a tool in a well-developed iterative algorithm for convex optimization known as the *damped Newton method*. While optimization is not the primary focus of the present work, we shall employ a modification of the damped Newton method as a more efficient alternative to one of our main algorithms, so we now briefly sketch the technique.

Given a current point $\mathbf{x} \in \mathcal{K}$, one first computes the *Newton direction*

$$\mathbf{e}(\mathbf{x}, \mathcal{R}) = -[\nabla^2 \mathcal{R}(x)]^{-1} \nabla \mathcal{R}(x),$$

and then a damped Newton iteration is performed, where the updated point is then

$$DN(\mathbf{x}, \mathcal{R}) = \mathbf{x} - \frac{1}{1 + \lambda(\mathbf{x}, \mathcal{R})} \mathbf{e}(\mathbf{x}, \mathcal{R}),$$

While not necessarily clear at first glance, this iterative process converges *very* quickly. It is convenient to measure the progress to the minimizer in terms of the Newton decrement, which leads us to the following Theorem.

Theorem 27 (e.g. [60])**.** *For any self-concordant function $\mathcal{R}$, let $\mathbf{x}$ be any point in the interior of $\mathcal{K}$ and let $\mathbf{x}^* := \arg\min \mathcal{R}$. Then $DN(\mathbf{x}, \mathcal{R}) \in \mathcal{K}$ and whenever $\lambda(\mathbf{x}, \mathcal{R}) \leq 1/4$ we have*

$$\begin{aligned} \|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{x}} &\leq 2\lambda(\mathbf{x}, \mathcal{R}) \\ \|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{x}^*} &\leq 2\lambda(\mathbf{x}, \mathcal{R}) \\ \lambda(DN(\mathbf{x}, \mathcal{R}), \mathcal{R}) &\leq 2\lambda(\mathbf{x}, \mathcal{R})^2 \end{aligned}$$

[1] *This is equivalent to the usual definition of the dual norm, namely $\|\mathbf{h}\|_{\mathbf{x}}^* := \sup\{\mathbf{h} \cdot \mathbf{z} : \|\mathbf{z}\|_{\mathbf{x}} \leq 1\}$.*

The key here is that the Newton decrement, which bounds the distance to the minimizer, decreases at a doubly-exponential rate from iteration to iteration. As soon as $\lambda(\mathbf{x}, \mathcal{R}) \leq 1/4$, we require only $O(\log\log \varepsilon^{-1})$ iterations to arrive at an ε-nearby point to $\mathbf{x}^*$.

Self-concordant Barriers and the Minkowsky Function

The final result we state is that a self-concordant *barrier* function on a compact convex set $\mathcal{K}$ does not grow excessively quickly despite that it must approach ∞ towards the boundary of $\mathcal{K}$. Ultimately, the crucial piece we shall need is that the growth is *logarithmic* as a function of the inverse distance to the boundary. Towards this aim let us define, for any $\mathbf{x}, \mathbf{y} \in \text{int}(\mathcal{K})$, the *Minkowsky function* $\pi_{\mathbf{x}}(\mathbf{y})$ on $\mathcal{K}$ as

$$\pi_{\mathbf{x}}(\mathbf{y}) = \inf\{t \geq 0 : \mathbf{x} + t^{-1}(\mathbf{y} - \mathbf{x}) \in \mathcal{K}\}.$$

The Minkowsky function measures distance from $\mathbf{x}$ to $\mathbf{y}$ as a portion of the total distance on the ray from $\mathbf{x}$ to the boundary of $\mathcal{K}$ that goes through the point $\mathbf{y}$. Hence $\pi_{\mathbf{x}}(\mathbf{y}) \in [0, 1]$ always and when $\mathbf{x}$ is considered the "center" of $\mathcal{K}$ then $1 - \pi_{\mathbf{x}}(\mathbf{y})$ can be interpreted as the distance from $\mathbf{y}$ to the boundary of $\mathcal{K}$.

Theorem 28. *For any ϑ-self-concordant barrier on $\mathcal{K}$, and for any $\mathbf{x}, \mathbf{y} \in$ int$(\mathcal{K})$, we have that*

$$\mathcal{R}(\mathbf{y}) - \mathcal{R}(\mathbf{x}) \leq \vartheta \ln\left(\frac{1}{1 - \pi_{\mathbf{x}}(\mathbf{y})}\right).$$

A proof can be found in the lecture notes of Nemirovskii [61] and elsewhere.

It is important to notice that any linear perturbation $\mathcal{R}'(\mathbf{x}) := \mathcal{R}(\mathbf{x}) + \mathbf{h} \cdot \mathbf{x}$ of a self-concordant function $\mathcal{R}$ is again a self-concordant function. Indeed, the linear term disappears in the 2nd and 3rd derivatives in the first requirement of Definition 25. In the same vein, the norm induced by such $\mathcal{R}'$ is identical to that of $\mathcal{R}$.

4.1.2 Examples of Self-Concordant Functions

We note a straightforward fact that illuminates how self-concordant barriers can be combined.

Lemma 29. *Let R_1 be a ϑ_1-self-concordant barrier function for the set $\mathcal{K}_1$ and let R_2 be a ϑ_2-self-concordant barrier function for the set $\mathcal{K}_2$, then $\mathcal{R} := \mathcal{R}_1 + \mathcal{R}_2$ is a $(\vartheta_1 + \vartheta_2)$-self-concordant barrier function for the set $\mathcal{K}_1 \cap \mathcal{K}_2$.*

The above Lemma is most useful for constructing self-concordant barriers on sets defined by the intersection of simpler sets. For example, on the set $[0, \infty]$ there exists a very simple barrier, namely $\mathcal{R}(x) = -\log x$. A quick check verifies that this function satisfies both the self-concordance and the barrier property with equality with $\vartheta = 1$. In addition, we can easily extend this to any half-space H in $\mathbb{R}^n$ by letting $\mathcal{R}(\mathbf{x}) = -\log \delta(\mathbf{x}, H)$, where $\delta(\cdot, H)$

is the Euclidean distance to the half-space. Finally, if the set $\mathcal{K}$ is a polytope in $\mathbb{R}^n$, then it is defined as the intersection of a number of halfspaces. Equivalently, it can be defined by linear inequalities $A\mathbf{x} \succeq \mathbf{b}$ for some $m \times n$ matrix A, which leads us immediately to the *log-barrier* function of this polytope, namely

$$\mathcal{R}(\mathbf{x}) := \sum_{i=1}^{m} -\log(A_i\mathbf{x} - b_i).$$

We note that this choice of $\mathcal{R}$ is m-self-concordant, as it is the sum of m 1-self-concordant barriers.

For the n-dimensional ball,

$$\mathcal{B}_n = \{\mathbf{x} \in \mathbb{R}^n \ , \ \sum_i \mathbf{x}_i^2 \leq 1\},$$

the barrier function $\mathcal{R}(\mathbf{x}) = -\log(1 - \|\mathbf{x}\|^2)$ is 1-self-concordant. In particular, this leads to the linear dependence of the regret bound in Section 4.3.3 on the dimension n, as $\vartheta = 1$.

4.2 Improved Bounds via Interior Point Methods

In Section 2.2 we defined the problem of Online Linear Optimization, and developed a technique for proving regret bounds via a *regularization* technique. In Section 4.1 we presented a brief summary of known results from the literature on interior point methods and self-concordant functions. In the present section we bring these seemingly dissimilar topics together and show that, by utilizing a self-concordant barrier as a regularization function, one can obtain much improved bounds for an array of problems. In particular, the introduction of these interior point techniques leads to a novel efficient algorithm for the Bandit setting with an essentially-optimal regret guarantee, resolving what was an open question for several years.

4.2.1 A refined regret bound: measuring $\mathbf{f}_t$ locally

We return our attention to proving regret bounds as in Section 2.2, but we now add a twist. The conclusions from that Section can be summarized as follows. For any FTRL algorithm, we achieve the fully general (yet unsatisfying) bound in Proposition 1. We can also apply Hölder's Inequality and, with the assumption that $\mathcal{R}$ is strongly convex, we arrive at Proposition 6.

The analysis of Proposition 6 is the typical approach, and indeed it can be shown that the above bound is tight (within a small constant factor from optimal), for instance, in the setting of prediction with expert advice [24]. On the other hand, there are times when we cannot make the assumption that $\mathbf{f}_t$ is bounded with respect to a fixed norm. This is particularly relevant in the bandit setting, when we will be estimating the functions $\mathbf{f}_t$ yet our estimates will blow up depending on the location of the point $\mathbf{x}_t$. In such cases, to obtain

tighter bounds, it will be necessary to measure the size of $\mathbf{f}_t$ with respect to a *changing norm*. While it may not be obvious at present, the ideal way to measure $\mathbf{f}_t$ is with the quadratic form defined by *the inverse Hessian of $\mathcal{R}$ at the point $\mathbf{x}_t$*. Indeed, this is precisely the norm defined in Section 4.1.1.

Theorem 30. *Suppose for all t we have $\eta\|\mathbf{f}_t\|^*_{\mathbf{x}_t} \leq \frac{1}{4}$, and $\mathcal{R}$ is a self-concordant barrier. Then for any $\mathbf{u} \in \mathcal{K}$*

$$\text{Regret}^{\mathbf{u}}(FTRL(\mathcal{R}, \eta); \mathbf{f}_{1:T}) \leq 2\eta \sum_{t=1}^{T} (\|\mathbf{f}_t\|^*_{\mathbf{x}_t})^2 + \eta^{-1}\mathcal{R}(\mathbf{u}).$$

Before proceeding to the proof let us emphasize a key point, namely that the function $\mathcal{R}$ is playing two distinct roles: first, $\mathcal{R}$ is the regularization function for FTRL and, second, when we refer to the norms $\|\cdot\|_{\mathbf{x}}$ and $\|\cdot\|^*_{\mathbf{x}}$, these are with respect to the function $\mathcal{R}$.

Proof of Theorem 30. Since $\mathcal{R}$ is a barrier, the minimization problem in (2.3) is unconstrained. As with Proposition 6, we can apply Hölders inequality to the term $\mathbf{f}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1})$. As the inequality holds for any primal-dual norm pair, we bound

$$\mathbf{f}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}) \leq \|\mathbf{f}_t\|^*_{\mathbf{x}_t}\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_{\mathbf{x}_t}. \tag{4.2}$$

We can write Φ_t as the objective used to obtain $\mathbf{x}_{t+1}$ in the FTRL algorithm, that is

$$\Phi_t(\mathbf{x}) := \eta \sum_{s=1}^{t} \mathbf{f}_s^\top \mathbf{x} + \mathcal{R}(\mathbf{x}).$$

We can then bound

$$\begin{aligned} \|\mathbf{x}_t - \mathbf{x}_{t+1}\|_{\mathbf{x}_t} &= \|\mathbf{x}_t - \arg\min \Phi_t\|_{\mathbf{x}_t} \\ \text{(By Theorem 27)} \quad &\leq 2\lambda(\mathbf{x}_t, \Phi_t) = 2\|\nabla\Phi_t(\mathbf{x}_t)\|^*_{\mathbf{x}_t} \end{aligned}$$

Recall that Theorem 27 requires $\lambda(\mathbf{x}_t, \Phi_t) = \|\nabla\Phi_t(\mathbf{x}_t)\|^*_{\mathbf{x}_t} \leq 1/4$. However, since $\mathbf{x}_t$ minimizes Φ_{t-1}, and because $\Phi_t(\mathbf{x}) = \Phi_{t-1}(\mathbf{x}) + \eta\mathbf{f}_t^\top\mathbf{x}$, it follows that $\nabla\Phi_t(\mathbf{x}_t) = \eta\mathbf{f}_t$. By assumption, $\eta\|\mathbf{f}_t\|^*_{\mathbf{x}_t} \leq \frac{1}{4}$. Furthermore, we have now shown that

$$\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_{\mathbf{x}_t} \leq 2\eta\|\mathbf{f}_t\|^*_{\mathbf{x}_t},$$

and when applied to (4.2) gives

$$\mathbf{f}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}) \leq 2\eta(\|\mathbf{f}_t\|^*_{\mathbf{x}_t})^2.$$

Combining this inequality with Proposition 1 finishes the proof. □

4.2.2 Improvement compared to previous bounds

Assuming that $\mathcal{R}$ is strongly convex, modulo multiplicative $\log T$ terms the bound obtained in Theorem 30 are never asymptotically worse than previous bounds, and at times are significantly tighter. We will briefly demonstrate this point in the present section.

The key advantage is that, by measuring the loss functions $\mathbf{f}_t$ using $\|\mathbf{f}_t\|_{\mathbf{x}_t}^{*2} = \mathbf{f}_t^\top \nabla^{-2}\mathcal{R}\mathbf{f}_t$, the bound may depend on something that does not scale with T. In particular, if a majority of the points $\mathbf{x}_t$ are close to the boundary, where the regularizer $\mathcal{R}$ has large curvature and the inverse hessian $\nabla^{-2}\mathcal{R}$ is tiny, we can expect the terms $\|\mathbf{f}_t\|_{\mathbf{x}_t}^{*2}$ to be miniscule.

As a simple example, consider an OLO problem in which the convex set is the real line segment $\mathcal{K} = [-1, 1]$, and we shall use FTRL with the simple logarithmic barrier $\mathcal{R}(\mathbf{x}) = -\log(1-\mathbf{x}) - \log(1+\mathbf{x})$. Let us now imagine a natural scenario in which our sequence of cost vectors $\mathbf{f}_1, \mathbf{f}_2, \ldots$ has some positive or negative bias, and hence for some $c > 0$ we have $\|\mathbf{f}_1 + \ldots + \mathbf{f}_t\| \geq ct$ for large enough t, say $t > t_0$ for constant t_0. It is easily checked that the FTRL optimization for our chosen regularization will lead to $\|\mathbf{x}_t\| \geq 1 - \frac{1}{c\eta t}$ for $t > t_0$, which implies that $\nabla^2\mathcal{R}(\mathbf{x}_t) \geq \frac{4}{c^2\eta^2 t^2}$. Pick a constant B so that $\sum_{t=1}^{t_0} 2\|\mathbf{f}_t\|_{\mathbf{x}_t}^{*\,2} \leq B$. For $t > t_0$, we can now bound

$$\|\mathbf{f}_t\|_{\mathbf{x}_t}^{*\,2} = \mathbf{f}^T\nabla^{-2}\mathcal{R}(\mathbf{x}_t)\mathbf{f} \leq \frac{4}{c^2\eta^2 t^2}.$$

Depending on the sign of $\mathbf{f}_1 + \ldots + \mathbf{f}_T$, set the comparator according to $\mathbf{u} = \pm(1 - 1/T)$ so that $\mathcal{R}(\mathbf{u}) \leq \log T$. We arrive at the following bound on the regret via theorem 30:

$$\begin{aligned}\text{Regret}^{\mathbf{u}}(\text{FTRL}(\mathcal{R}, \eta); \mathbf{f}_{1:T}) &\leq 2\eta \sum_{t=1}^{T} \left[\|\mathbf{f}_t\|_{\mathbf{x}_t}^*\right]^2 + \eta^{-1}\mathcal{R}(\mathbf{u}) \\ &\leq \eta B + \eta^{-1}\log T + \eta^{-1}C,\end{aligned}$$

where C hides the constant $\frac{8}{c^2}\sum_{t=t_0+1}^{T}\frac{1}{t^2}$. With an appropriately tuned[2] η, we obtain a bound on the order of $O(\sqrt{\log T})$.

We may now compare this to to previous "fixed-norm" regret bounds, such as those for online gradient descent of Zinkevich [77], where the value $\|\mathbf{f}_t\|$ does not change. The corresponding bound for this algorithm would be $O(\eta T + \eta^{-1})$ which, even when optimally tuned, must grow at a rate at least $\Theta(\sqrt{T})$.

4.2.3 An iterative interior point algorithm

Algorithm 1 requires the solution of a convex program every iteration. In this section we give a more efficient iterative algorithm.

Define $\Phi_t(\mathbf{x}) := \eta\sum_{s=1}^{t}\mathbf{f}_s^\top\mathbf{x} + \mathcal{R}(\mathbf{x})$, the FTRL objective at time t.

[2]In this short example, we must assume that the value of η is initially tuned for the given loss sequence. For a bound that is robust to arbitrary sequences of loss vectors an adaptive selection of η is necessary, but such issues are beyond the scope of this chapter.

Algorithm 2 Itertive FTRL (I-FTRL)$(\mathcal{R}, \eta)$

Input: $\eta > 0$, regularization $\mathcal{R}$.
Initialize $\mathbf{y}_1 = \arg\min_{\mathbf{x}\in\mathcal{K}} \mathcal{R}(\mathbf{x})$
On round $t+1$, play

$$\mathbf{y}_{t+1} := DN(\Phi_t, \mathbf{y}_t) = \mathbf{y}_t + \frac{1}{1+\lambda(\mathbf{y}_t, \Phi_t)}\mathbf{e}(\mathbf{y}_t, \Phi_t). \tag{4.3}$$

and observe $\mathbf{f}_{t+1}$.

Computationally, to generate $\mathbf{y}_{t+1}$, it only requires storing the previous point $\mathbf{y}_t$ and computing the Newton direction and Newton decrement. This latter vector can be computed by inverting a single matrix – the Hessian of the regularization function $\mathcal{R}$ at $\mathbf{y}_t$ – and a single matrix-vector product with the gradient of Φ_t at point $\mathbf{y}_t$.

While Algorithm 2 may seem very different from the FTRL method, it can be viewed as an efficient implementation of the same algorithm, and hence borrow the almost same regret bounds. Notice in the bound below that for $\eta = \tilde{O}(\frac{1}{\sqrt{T}})$, as is the optimal setting of parameter in Theorem 30, the additive term is a constant independent of the number of iterations.

Theorem 31. *Let $\mathcal{K}$ be a compact convex set and $\mathcal{R}$ be a ϑ-self-concordant barrier on $\mathcal{K}$. Assume $\|\mathbf{f}_t\|^*_{\mathbf{y}_t} \le C$ for all t and $\eta C \le \frac{1}{8}$. Then for any $\mathbf{u} \in \mathcal{K}$*

$$\begin{aligned}&\mathrm{Regret}^{\mathbf{u}}(\textit{I-FTRL}(\mathcal{R}, \eta); \mathbf{f}_{1:T}) \\ &\le \mathrm{Regret}^{\mathbf{u}}(\textit{FTRL}(\mathcal{R}, \eta); \mathbf{f}_{1:T}) + 16C^3\eta^2 T\ .\end{aligned} \tag{4.4}$$

To prove this theorem, we show that the predictions generated by Algorithm 2 are very close to those generated by Algorithm 1. More formally, we prove the following lemma, where $\{\mathbf{x}_t\}$ denotes the sequence of vectors generated by the FTRL algorithm as defined in equation (2.3).

Lemma 32.

$$\|\mathbf{y}_t - \mathbf{x}_t\|_{\mathbf{y}_t} \le 2\lambda(\mathbf{y}_t, \Phi_{t-1}) \le 4\lambda^2(\mathbf{y}_{t-1}, \Phi_{t-1}) \le 16\eta^2 C^2$$

Before proving this lemma, let us show how it immediately implies Theorem 31:

$$\begin{aligned}&\mathrm{Regret}^{\mathbf{u}}(\text{I-FTRL}(\mathcal{R}, \eta); \mathbf{f}_{1:T}) \\ &= \sum_{t=1}^{T} \mathbf{f}_t^\top(\mathbf{y}_t - \mathbf{u}) = \sum_{t=1}^{T} \mathbf{f}_t^\top(\mathbf{x}_t - \mathbf{u}) + \sum_{t=1}^{T} \mathbf{f}_t^\top(\mathbf{y}_t - \mathbf{x}_t) \\ &= \mathrm{Regret}^{\mathbf{u}}(\text{FTRL}(\mathcal{R}, \eta); \mathbf{f}_{1:T}) + \sum_{t=1}^{T} \|\mathbf{f}_t\|^*_{\mathbf{y}_t} \|\mathbf{y}_t - \mathbf{x}_t\|_{\mathbf{y}_t} \\ &\le \mathrm{Regret}^{\mathbf{u}}(\text{FTRL}(\mathcal{R}, \eta); \mathbf{f}_{1:T}) + 16\eta^2 C^3 T\end{aligned}$$

We can now proceed to prove Lemma 32:

Proof of Lemma 32. The proof is by induction on t. For $t = 1$ the result is true because $\mathbf{x}_1, \mathbf{y}_1$ are chosen to minimize $\mathcal{R}$. Suppose the statement holds for t, we prove for $t+1$. By definition,

$$\begin{aligned}\lambda^2(\mathbf{y}_t, \Phi_t) &= \nabla\Phi_t(\mathbf{y}_t)[\nabla^2\Phi_t(\mathbf{y}_t)]^{-1}\nabla\Phi_t(\mathbf{y}_t) \\ &= \nabla\Phi_t(\mathbf{y}_t)[\nabla^2\mathcal{R}(\mathbf{y}_t)]^{-1}\nabla\Phi_t(\mathbf{y}_t).\end{aligned}$$

Note that

$$\nabla\Phi_t(\mathbf{y}_t) = \nabla\Phi_{t-1}(\mathbf{y}_t) + \eta\mathbf{f}_t^\top.$$

Using $(x+y)^T A(x+y) \le 2x^T Ax + 2y^T Ay$ we obtain

$$\begin{aligned}\frac{1}{2}\lambda^2(\mathbf{y}_t, \Phi_t) &\le \nabla\Phi_{t-1}(\mathbf{y}_t)[\nabla^2\mathcal{R}(\mathbf{y}_t)]^{-1}\nabla\Phi_{t-1}(\mathbf{y}_t) \\ &\quad + \eta^2\mathbf{f}_t^\top[\nabla^2\mathcal{R}(\mathbf{y}_t)]^{-1}\mathbf{f}_t \\ &= \lambda^2(\mathbf{y}_t, \Phi_{t-1}) + \eta^2(\|\mathbf{f}_t\|^*_{\mathbf{y}_t})^2.\end{aligned}$$

The first term can be bounded by the induction hypothesis:

$$\lambda^2(\mathbf{y}_t, \Phi_{t-1}) \le 64\eta^4 C^4. \tag{4.5}$$

As for the second term, by our assumption on $\|\mathbf{f}_t\|^*_{\mathbf{y}_t}$

$$\eta^2(\|\mathbf{f}_t\|^*_{\mathbf{y}_t})^2 \le \eta^2 C^2 \ .$$

Combining the results,

$$\lambda^2(\mathbf{y}_t, \Phi_t) \le 2 \cdot (64\eta^4 C^4 + \eta^2 C^2) \le 4\eta^2 C^2, \tag{4.6}$$

where the last inequality follows since $\eta^2 C^2 \le \frac{1}{64}$. In particular, this implies that $\lambda(\mathbf{y}_t, \Phi_t) \le \frac{1}{4}$ and, therefore,

$$\lambda(\mathbf{y}_{t+1}, \Phi_t) \le 2\lambda^2(\mathbf{y}_t, \Phi_t) \le 8\eta^2 C^2 \le \frac{1}{8}$$

according to Theorem 27. The induction step is completed by applying Theorem 27 again:

$$\begin{aligned}\|\mathbf{y}_{t+1} - \mathbf{x}_{t+1}\|_{\mathbf{y}_{t+1}} &= \|\mathbf{y}_{t+1} - \arg\min\Phi_t\|_{\mathbf{y}_{t+1}} \\ &\le 2\lambda(\mathbf{y}_{t+1}, \Phi_t) \ .\end{aligned}$$

□

4.3 Bandit Feedback

We now return our attention to the *bandit version* of the online linear optimization problem that we have discussed. The additional difficulty in the bandit setting rests in the feedback model. As before, an $\mathbf{x}_t$ is chosen at round t, an Adversary chooses $\mathbf{f}_t$, and the cost $\mathbf{f}_t^\top\mathbf{x}_t$ is paid. But, instead of receiving the entire vector $\mathbf{f}_t$, the learner *may only observe* the

scalar value ($\mathbf{f}_t^\top \mathbf{x}_t$). Recall that, in our Follow The Regularized Leader template, the point $\mathbf{x}_t$ is computed with access to $(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{t-1})$ whereas an algorithm in the bandit setting is given only $(\mathbf{f}_1^\top \mathbf{x}_1, \mathbf{f}_2^\top \mathbf{x}_2, \ldots, \mathbf{f}_{t-1}^\top \mathbf{x}_{t-1})$ as input.

Let us emphasize that the bandit model is difficult not only because the feedback has been reduced from a vector to a scalar but also because the content of the feedback actually depends on the chosen action. This present an added dilemma for the algorithm : is it better to select $\mathbf{x}_t$ in order to gather better information or, alternatively, is it better to choose $\mathbf{x}_t$ to exploit previously obtained information? This is typically referred to as an *exploration-exploitation* trade-off, and arises in a range of problems.

In this section, we make an additional assumption that the adversary is *oblivious*. That is, the sequence $\mathbf{f}_1, \ldots, \mathbf{f}_T$ is fixed ahead of the game. For results in bandit optimization against non-oblivious adversaries, we refer the reader to [5].

4.3.1 Constructing a Bandit Algorithm

A large number of bandit linear optimization algorithms have been proposed, but essentially all make use of a generic template algorithm. This template has three key ingredients:

1. A **full-information algorithm** $\mathcal{A}$ which takes as input a sequence of loss vectors $\mathbf{f}_t$ and returns points $\mathbf{x} \in \mathcal{K}$; that is,

$$\mathbf{x}_t \leftarrow \mathcal{A}(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{t-1})$$

2. A **sampling scheme** $\texttt{sampler}(\mathbf{x})$ for each $\mathbf{x}$ that defines a distribution on $\mathcal{K}$ with the property that

$$\mathop{\mathbb{E}}_{\mathbf{y} \sim \texttt{sampler}(\mathbf{x})} \mathbf{y} = \mathbf{x} \tag{4.7}$$

3. A corresponding **estimation scheme** $\texttt{guesser}(\ell, \mathbf{y}, \mathbf{x})$ which uses the randomly chosen $\mathbf{y}$ and the observed value $\ell = \mathbf{f}^\top \mathbf{y}$ to produce a "guess" of $\mathbf{f}$. For every linear function $\mathbf{f}$, `guesser` must satisfy

$$\mathop{\mathbb{E}}_{\mathbf{y} \sim \texttt{sampler}(\mathbf{x})} [\texttt{guesser}(\mathbf{f}^\top \mathbf{y}, \mathbf{y}, \mathbf{x})] = \mathbf{f}. \tag{4.8}$$

   For the remainder of this chapter, we will use $\tilde{\mathbf{f}}_t$ to denote the random variable $\texttt{guesser}(\mathbf{f}_t^\top \mathbf{y}_t, \mathbf{y}_t, \mathbf{x}_t)$ when the definition of `sampler` and `guesser` are clear.

These ingredients are combined into the following recipe, which describes the generic construction taking a full-information algorithm $\mathcal{A}$ for online linear optimization and produces a new algorithm for the bandit setting. We shall refer to this bandit algorithm as BanditReduction($\mathcal{A}$, `sampler`, `guesser`).

Algorithm 3 BanditReduction($\mathcal{A}, \texttt{sampler}, \texttt{guesser}$)

Input: full-info algorithm $\mathcal{A}$, sampling scheme $\texttt{sampler}(\cdot)$, estimation scheme $\texttt{guesser}(\cdot,\cdot,\cdot)$

Initialize $\mathbf{x}_1 \leftarrow \mathcal{A}(\{\})$
for $t = 1 \dots T$ **do**
  Randomly sample $\mathbf{y}_t \sim \texttt{sampler}(\mathbf{x}_t)$
  Play $\mathbf{y}_t$, observe $\mathbf{f}_t^\top \mathbf{y}_t$
  Construct $\tilde{\mathbf{f}}_t \leftarrow \texttt{guesser}(\mathbf{f}_t^\top \mathbf{y}_t, \mathbf{y}_t, \mathbf{x}_t)$
  Update $\mathbf{x}_{t+1} \leftarrow \mathcal{A}(\tilde{\mathbf{f}}_1, \tilde{\mathbf{f}}_2, \dots, \tilde{\mathbf{f}}_t)$
end for

What justifies this reduction? In short, the unbiased sampling and unbiased estimation scheme allow us to bound the expected regret of BanditReduction($\mathcal{A}, \texttt{sampler}, \texttt{guesser}$) *in terms of* the regret of $\mathcal{A}$ on the estimated functions. Let us denote $\mathcal{A}' :=$ BanditReduction($\mathcal{A}, \texttt{sampler}, \texttt{guesser}$), and for simplicity, let $\mathbb{E}_t[\cdot]$ be the expectation over the algorithm's random draw of $\mathbf{y}_t \sim \texttt{sampler}(\mathbf{x}_t)$ conditioned on the history, i.e. the random $\mathbf{y}_1, \dots, \mathbf{y}_{t-1}$. In the following, the assumption that $\mathbf{f}_t$'s are fixed ahead of the game is crucial. For any $\mathbf{u} \in \mathcal{K}$, the expected regret of $\mathcal{A}'$ is

$$\begin{aligned}
\mathbb{E}[\text{Regret}_{\mathbf{u}}(\mathcal{A}'; \mathbf{f}_1, \dots, \mathbf{f}_T)] &= \mathbb{E}\left[\sum_{t=1}^T \mathbf{f}_t^\top (\mathbf{y}_t - \mathbf{u})\right] \\
\text{(By tower rule)} \quad &= \mathbb{E}\left[\sum_{t=1}^T \mathop{\mathbb{E}}_t[\mathbf{f}_t^\top (\mathbf{y}_t - \mathbf{u})]\right] \\
\text{(By (4.7))} \quad &= \mathbb{E}\left[\sum_{t=1}^T \mathop{\mathbb{E}}_t[\mathbf{f}_t^\top (\mathbf{x}_t - \mathbf{u})]\right] \\
\text{(By (4.8))} \quad &= \mathbb{E}\left[\sum_{t=1}^T \mathop{\mathbb{E}}_t[\tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u})]\right] \\
\text{(By tower rule)} \quad &= \mathbb{E}\left[\sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u})\right] \\
&= \mathbb{E}[\text{Regret}_{\mathbf{u}}(\mathcal{A}; \tilde{\mathbf{f}}_1, \dots, \tilde{\mathbf{f}}_T)]
\end{aligned}$$

Notice, however, that the last expression within the $\mathbb{E}[\cdot]$ is exactly the regret of $\mathcal{A}$ when the input functions are $\tilde{\mathbf{f}}_1, \dots, \tilde{\mathbf{f}}_T$. This leads us directly to the following Lemma:

Lemma 33. *Assume we are given any full-information algorithm $\mathcal{A}$ and any sampling and estimation schemes $\texttt{sampler}, \texttt{guesser}$. If we let the associated bandit algorithm be $\mathcal{A}' :=$ BanditReduction($\mathcal{A}, \texttt{sampler}, \texttt{guesser}$), then the expected regret of the (randomized) algorithm $\mathcal{A}'$ on the fixed sequence $\{\mathbf{f}_t\}$ is equal to the expected regret of the (deterministic[3]) *

[3] *Although we do not consider these here, there do exist randomized algorithms for the full-information setting. In terms of regret, randomized algorithms provide no performance improvement over deterministic algorithms, yet randomization may lead to other benefits, e.g. computation. In the bandit setting, however, randomization is entirely necessary for vanishing regret.*

algorithm $\mathcal{A}$ on the random sequence $\{\tilde{\mathbf{f}}_t\}$. That is,

$$\mathbb{E}[\text{Regret}_{\mathbf{u}}(\mathcal{A}'; \mathbf{f}_1, \dots, \mathbf{f}_T)] = \mathbb{E}[\text{Regret}_{\mathbf{u}}(\mathcal{A}; \tilde{\mathbf{f}}_1, \dots, \tilde{\mathbf{f}}_T)]$$

This Lemma is quite powerful: it says that we can construct a bandit algorithm from a full-information one, achieve a bandit regret bound in terms of the full-information bound, and we need only construct sampling and estimation schemes which satisfy the properties in (4.7) and (4.8).

4.3.2 The Dilemma of Bandit Optimization

At first glance, Lemma 33 may appear to be a slam dunk: as long as we have a full-information algorithm $\mathcal{A}$ with a low-regret guarantee, we can seemingly construct a Bandit version $\mathcal{A}'$ with an identical regret guarantee in expectation. The remaining difficulty, which may not be so obvious, is that the regret of $\mathcal{A}$ is taken with respect to the random estimates $\{\tilde{\mathbf{f}}_t\}$, and these estimates can unfortunately have *very high variance*! In general, the typical bound on $\text{Regret}(\mathcal{A}; \mathbf{f}_1, \dots, \mathbf{f}_T)$ will scale with the magnitude of the $\mathbf{f}_t$'s, and this can be quite bad if the $\mathbf{f}_t$'s can grow arbitrarily large.

Let us illustrate this issue with a simple example. Assume $\mathcal{K} = \Delta_2 = \{\alpha\mathbf{e}_1 + (1-\alpha)\mathbf{e}_2 \ : \ \forall \alpha \in [0,1]\}$. We need to construct a sampling scheme and an estimation scheme, and we give a natural choice. Assume $\mathbf{x} = \alpha\mathbf{e}_1 + (1-\alpha)\mathbf{e}_2$ and assume the unobserved cost function is $\mathbf{f}$, then let

$$\mathbf{y} = \begin{cases} \mathbf{e}_1, & \text{w.p. } \alpha \\ \mathbf{e}_2, & \text{w.p. } 1-\alpha \end{cases} \qquad \tilde{\mathbf{f}} = \begin{cases} \frac{\mathbf{f}^\top \mathbf{y}}{\alpha}\mathbf{e}_1, & \text{when } \mathbf{y} = \mathbf{e}_1 \\ \frac{\mathbf{f}^\top \mathbf{y}}{1-\alpha}\mathbf{e}_2, & \text{when } \mathbf{y} = \mathbf{e}_2. \end{cases}$$

It is easily checked that these sampling and estimation schemes satisfy the desired requirements (4.8) and (4.7). The downside that the magnitude of $\tilde{\mathbf{f}}$ can grow with $\max\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\}$ (assuming here that $\|\mathbf{f}\| = O(1)$). While the careful reader may notice that things are not so bad in expectation, as $\mathbb{E}\,\|\tilde{\mathbf{f}}\| = O(1)$, the typical regret bound generally depends on $\mathbb{E}\,\|\tilde{\mathbf{f}}\|^2$ which grows with $\max\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\}$. If we apply the strong-convexity result from Section 2.2.2, and by correctly choosing η, we would have a regret bound scaling with the quantity $\mathbb{E}\left[\sqrt{\sum_{t=1}^T \|\tilde{\mathbf{f}}_t\|^{*2}}\right] \leq \sqrt{\sum_{t=1}^T \mathbb{E}\,\|\tilde{\mathbf{f}}_t\|^{*2}}$. To obtain a rate of roughly $O(\sqrt{T})$ it is necessary that we have $\mathbb{E}\,\|\tilde{\mathbf{f}}_t\|^{*2} = O(1)$.

Perhaps our sampling and estimation schemes could have been better designed? Unfortunately no: the variance of $\tilde{\mathbf{f}}$ can not be untethered from $\max\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\}$. This example sheds light on a crucial issue of Bandit optimization: how does one handle estimation variance when $\mathbf{x}$ is *close to the boundary*? Note that the aforementioned example does not lead to difficulty when $\max\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\} = O(1)$. A common approach, used in various forms throughout the literature [27, 9, 33, 11, 56, 42], is simply to restrict $\mathbf{x} = \alpha\mathbf{e}_1 + (1-\alpha)\mathbf{e}_2$ away from the boundary, requiring that $\alpha \in [\gamma, 1-\gamma]$ for some appropriately chosen $\gamma \in (0, 1/2)$. This restriction does have the benefit of guaranteeing $\mathbb{E}\,\|\tilde{\mathbf{f}}\|^2 = O(1/\gamma)$, but this comes at a price: this γ-perturbation means we can only compete with a suboptimal comparator, and this approximation shall give an additive $O(\gamma T)$ in the regret bound.

The solution, which we present in the following section, is based on measuring the function $\tilde{\mathbf{f}}_t$ with a *local* norm. This was our original aim in developing the FTRL algorithms based on self-concordant barrier functions: they allows us to obtain a regret bound which measures each $\tilde{\mathbf{f}}_t$ in such a way that depends on the current hypothesis $\mathbf{x}_t$. Indeed, the norm $\|\cdot\|_{\mathbf{x}_t}$, which *locally* measures $\tilde{\mathbf{f}}_t$ is precisely what we shall need. Ultimately we will show that, with the correct choice of sampling scheme, we can always guarantee that as we shall show that $\|\tilde{\mathbf{f}}_t\|_{\mathbf{x}_t} = O(1)$.

4.3.3 Main Result

We now describe the primary contribution of this chapter, which is an efficient algorithm for Bandit linear optimization that achieves a $\sqrt{T}$-regret bound. We call this algorithm `SCRiBLe`, standing for Self-Concordant Regularization in Bandit Learning.

We have now developed all necessary techniques to describe the result and prove the desired bound. The key ingredients of our algorithm, that help overcome the previously-discussed difficulties, are:

1. A self-concordant barrier function $\mathcal{R}$ for the set $\mathcal{K}$ (Section 4.1.1)
2. The full-information algorithm FTRL (Section 2.2.1) using the barrier $\mathcal{R}$ as the regularization
3. A sampling scheme `sampler`$(\mathbf{x})$ based on the Dikin ellpisoid $W_1(\mathbf{x})$ (Section 4.1.1) chosen according to $\mathcal{R}$. Specifically, if we denote $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ and $\{\lambda_1, \ldots, \lambda_n\}$ as the eigenvalues and eigenvectors of $\nabla^2\mathcal{R}(\mathbf{x}_t)$, the algorithm will sample $\mathbf{y}_t \leftarrow \mathbf{x}_t \pm \lambda_i^{-1/2}\mathbf{e}_i$, one of the $2n$ poles of the Dikin ellipsoid, uniformly at random.
4. An estimation scheme `guesser`$(\cdot,\cdot,\cdot)$ which produces estimates aligned with eigenpoles of $W_r(\mathbf{x})$. Specifically, corresponding to the eigenpole chosen by `sampler`, `guesser` outputs
$$\tilde{\mathbf{f}}_t \leftarrow \pm n \left(\mathbf{f}_t^\top \mathbf{y}_t\right) \lambda_i^{1/2} \cdot \mathbf{e}_i$$
5. An improved regret bound for self-concordant functions using local norms (Section 4.2.1)

We now state the main result of the chapter.

Theorem 34. *Let $\mathcal{K}$ be a compact convex set and $\mathcal{R}$ be a ϑ-self-concordant barrier on $\mathcal{K}$. Assume $|\mathbf{f}_t^\top \mathbf{x}| \le 1$ for any $\mathbf{x} \in \mathcal{K}$ and any t. Setting $\eta = \sqrt{\frac{\vartheta \log T}{2n^2 T}}$, the regret of `SCRiBLe` (Algorithm 4) is bounded as*

$$\mathbb{E}[\mathrm{Regret}_{\mathbf{u}}(\textit{SCRiBLe}; \mathbf{f}_1, \ldots, \mathbf{f}_T)] \le \sqrt{8n^2\vartheta T \log T} + 2$$

whenever $\frac{T}{\log T} > 8\vartheta$.

Algorithm 4 `SCRiBLe`

Input: $\eta > 0$, ϑ-self-concordant $\mathcal{R}$
Let $\mathbf{x}_1 = \arg\min_{\mathbf{x}\in\mathcal{K}} [\mathcal{R}(\mathbf{x})]$.
for $t = 1$ to T **do**
  Let $\{\mathbf{e}_1, \dots, \mathbf{e}_n\}$ and $\{\lambda_1, \dots, \lambda_n\}$ be the set of eigenvectors and eigenvalues of $\nabla^2\mathcal{R}(\mathbf{x}_t)$.
  Choose i_t uniformly at random from $\{1, \dots, n\}$ and $\varepsilon_t = \pm 1$ with probability $1/2$.
  Predict $\mathbf{y}_t = \mathbf{x}_t + \varepsilon_t \lambda_{i_t}^{-1/2} \mathbf{e}_{i_t}$.
  Observe the gain $\mathbf{f}_t^\top \mathbf{y}_t \in \mathbb{R}$.
  Define $\tilde{\mathbf{f}}_t := n\left(\mathbf{f}_t^\top \mathbf{y}_t\right) \varepsilon_t \lambda_{i_t}^{1/2} \cdot \mathbf{e}_{i_t}$.
  Update

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{K}} \left[\eta \sum_{s=1}^{t} \tilde{\mathbf{f}}_s^\top \mathbf{x} + \mathcal{R}(\mathbf{x}) \right].$$

end for

Proof. `SCRiBLe` is exactly in the template of Algorithm 3, using the full-information algorithm $\mathrm{FTRL}_{\mathcal{R}}$ and with `sampler`$(\cdot)$ and `guesser`$(\cdot,\cdot,\cdot)$ that satisfy properties (4.7) and (4.8) respectively. By Lemma 33, we can write

$$\mathbb{E}[\mathrm{Regret}_{\mathbf{u}}(\mathcal{A}; \mathbf{f}_1, \dots, \mathbf{f}_T)] = \mathbb{E}[\mathrm{Regret}_{\mathbf{u}}(\mathrm{FTRL}_{\mathcal{R}}; \tilde{\mathbf{f}}_1, \dots, \tilde{\mathbf{f}}_T)].$$

We then apply Theorem 30 to obtain for any $\mathbf{u} \in \mathcal{K}$

$$\begin{aligned}
& \mathbb{E}[\mathrm{Regret}_{\mathbf{u}}(\mathrm{FTRL}_{\mathcal{R}}; \tilde{\mathbf{f}}_1, \dots, \tilde{\mathbf{f}}_T)] \\
\leq\ & 2\eta\, \mathbb{E}\left(\sum_{t=1}^{T} \left[\|\tilde{\mathbf{f}}_t\|_{\mathbf{x}_t}^* \right]^2 \right) + \eta^{-1}\mathcal{R}(\mathbf{u}) \\
=\ & 2\eta\, \mathbb{E}\left(\sum_{t=1}^{T} \mathop{\mathbb{E}}_{t} \left(\|\tilde{\mathbf{f}}_t\|_{\mathbf{x}_t}^* \right)^2 \right) + \eta^{-1}\mathcal{R}(\mathbf{u}) \\
=\ & 2\eta\, \mathbb{E}\left[\sum_{t=1}^{T} \mathop{\mathbb{E}}_{t} \left(\tilde{\mathbf{f}}_t^\top \nabla_{\mathbf{x}_t}^{-2} \mathcal{R} \tilde{\mathbf{f}}_t \right) \right] + \eta^{-1}\mathcal{R}(\mathbf{u}) \\
=\ & 2\eta\, \mathbb{E}\left[\sum_{t=1}^{T} \mathop{\mathbb{E}}_{t} \left((n\mathbf{f}_t^\top \mathbf{y}_t)^2 \lambda_{i_t} \mathbf{e}_{i_t}^\top \nabla_{\mathbf{x}_t}^{-2} \mathcal{R} \mathbf{e}_{i_t} \right) \right] + \eta^{-1}\mathcal{R}(\mathbf{u}) \\
=\ & 2\eta\, \mathbb{E}\left[\sum_{t=1}^{T} \mathop{\mathbb{E}}_{t} (n\mathbf{f}_t^\top \mathbf{y}_t)^2 \right] + \eta^{-1}\mathcal{R}(\mathbf{u}) \\
\leq\ & 2\eta n^2 T + \eta^{-1}\mathcal{R}(\mathbf{u})
\end{aligned}$$

If $\mathbf{u}$ is such that $\pi_{\mathbf{x}_1}(\mathbf{u}) \leq 1 - \frac{1}{T}$ then by Theorem 28 we have that

$$\mathcal{R}(\mathbf{u}) \leq \vartheta \log T. \tag{4.9}$$

If, on the other hand, $\pi_{\mathbf{x}_1}(\mathbf{u}) > 1 - \frac{1}{T}$ then we can define $\mathbf{u}' := (1 - 1/T)\mathbf{u} + (1/T)\mathbf{x}_1$. Certainly,

$$\begin{aligned}
\text{Regret}^{\mathbf{u}}(\mathcal{A}; \mathbf{f}_{1:T}) &= \text{Regret}^{\mathbf{u}'}(\mathcal{A}; \mathbf{f}_{1:T}) + \sum_{t=1}^{T} \mathbf{f}_t^\top (\mathbf{u}' - \mathbf{u}) \\
&= \text{Regret}^{\mathbf{u}'}(\mathcal{A}; \mathbf{f}_{1:T}) + \frac{1}{T} \sum_{t=1}^{T} \mathbf{f}_t^\top (\mathbf{x}_1 - \mathbf{u}) \\
&\leq 2\eta n^2 T + \eta^{-1} \mathcal{R}(\mathbf{u}') + 2 \\
&\leq 2\eta n^2 T + \vartheta \eta^{-1} \log T + 2
\end{aligned}$$

□

4.4 Conclusion

We have given the first efficient algorithm for bandit online linear optimization with optimal regret bound. For this purpose, we introduce the fascinating tool of self-concordant barriers from interior point optimization and give a new algorithm for full-information online linear optimization with strong regret bounds.

In the full information case, we have given an iterative version of our algorithm which is preferable computationally, and a similar iterative algorithm can be derived for the bandit case as well.

Chapter 5

Blackwell Approachability

5.1 Introduction

Von Neumann's minimax theorem (1928) establishes a central result in the theory of two-player zero-sum games, essentially by providing a prescription to both players. This prescription is in the form of a pair of optimal strategies, either of which attains the optimal worst-case value of the game even without knowledge of the opponent's strategy. However, the theorem fundamentally requires that both players have utility that can be expressed as a *scalar*. In 1956, in response to von Neumann's result, David Blackwell posed an intriguing question: what guarantee can we hope to achieve when playing a two-player game with a *vector-valued payoff*?

When our payoffs are non-scalar quantities, it does not make sense to ask "can we earn at least x?". A sensible generalization is, "can we guarantee that our vector payoff lies in some convex set S?" In this case the story is more difficult, and Blackwell observed that an oblivious strategy does not suffice—in short, we do not achieve "minimax duality" for vector-payoff games as we can when the payoff is a scalar. Blackwell was able to prove that this negative result applies only for *one-shot games*. In his celebrated Approachability Theorem [18], one can achieve a duality statment *in the limit* when the game is played repeatedly, and the player may learn from his opponent's prior actions. Blackwell constructed an algorithm (that is, an adaptive strategy) that guarantees the average payoff vector "approaches" S.

Blackwell's Approachability Theorem has the flavor of learning in repeated games, a topic which has received much interest. In particular, there are a wealth of recent results on so-called *no-regret learning algorithms* for making repeated decisions given an arbitrary (and potentially adversarial) sequence of cost functions. The first no-regret algorithm for a "discrete action" setting was given in a seminal paper by James Hannan in 1956 [43]. That same year, David Blackwell pointed out [17] that his Approachability result leads, as a special case, to an algorithm with essentially the same low-regret guarantee proven by Hannan.

Over the years several other problems have been reduced to Blackwell approachability, including asymptotic calibration [35], online learning with global cost functions [30] and more

[54]. Indeed, it has been presumed that approachability, while establishing the existence of a no-regret algorithm, is strictly more powerful than regret-minimization; hence its utility in such a wide range of problems. In the present chapter we prove, to the contrary, that Blackwell's Approachability Theorem is equivalent, in a very strong sense, to no-regret learning for the setting of *Online Linear Optimization*. This shows that the connection discovered by Blackwell, between regret and approachability, is much stronger than originally supposed.

More specifically, we show how any no-regret algorithm can be converted into an algorithm for Approachability and vice versa. This algorithmic equivalence is achieved via the use of *conic duality*: an approachability problem over a convex cone K can be reduced to an online linear optimization instance where we must "learn" within the *polar cone* K^0. The reverse direction is similar. This equivalence provides a range of benefits and one such is "asymptotic calibrated forecasting". The calibration problem was reduced to Blackwell's Approachability Theorem by Foster [34], and a handful of other calibration techniques have been proposed, yet none have provided any efficiency guarantees on the strategy. Using a similar reduction from calibration to approachability, and by carefully constructing the reduction from approachability to online linear optimization, we achieve the first efficient calibration algorithm.

Related work There is by now vast literature on all three main topics of this chapter: approachability, online learning and calibration, see [24] for an excellent exposition.

Calibration is a fundamental notion in prediction theory and has found numerous applications in economics and learning. Dawid [28] was the first to define calibration, with numerous algorithms later given by Foster and Vohra [35], Fudenberg and Levine [39], Hart and Mas-Colell [44] and more (see e.g. [68, 66]). Foster has given a calibration algorithm based on approachability [34]. There are numerous definitions (mostly asymptotic) of calibration in the literature. In this chapter we give precise finite-time rates of calibration. Furthermore, we give the first *efficient* algorithm for calibration: attaining ε-calibration (formally defined later) required a running time of $poly(\frac{1}{\varepsilon})$ for all previous algorithms, whereas our algorithm runs in time proportional to $\log \frac{1}{\varepsilon}$.

5.2 Game Theory Preliminaries

5.2.1 Two-Player Games

Formally, a two-player normal-form game is defined by a pair of action sets $[n]$ and $[m]$, for natural numbers n, m, and a pair of utility functions $u_1, u_2 : [n] \times [m] \to \mathbb{R}$. When player 1 chooses action i and player 2 chooses action j, player 1 and player 2 receive utilities $u_1(i, j)$ and $u_2(i, j)$ respectively. An important class of two-player games are known as *zero-sum*, in that $u_1 \equiv -u_2$. For zero-sum games we drop the subcripts on u_1, u_2 and simply write $u(i, j)$ for player 1's utility, and $-u(i, j)$ for player 2's utility. For the remainder of this section, we shall be concerned entirely with zero-sum games, hence we will refer to player 1 as the Player and player 2 as the Adversary.

It is natural to assume that the players in a game may include randomness in their choice of action; simple games such as Rock-Paper-Scissors require randomness to achieve optimality. When the players choose their actions randomly according to the distributions $p \in \Delta_n$ and $q \in \Delta_m$, respectively, the *expected utility* for the Player is $\sum_{i,j} p(i)q(j)u(i,j)$. Von Neumann's minimax theorem, widely considered the first key result in game theory, tells us that both the Player and the Adversary have an "optimal" randomized strategy that can be played without knowledge of the strategy of their respective opponent.

Theorem 35 (Von Neumann's Minimax Theorem [63])**.** *For any integers $n, m > 0$ and any utility function $u : [n] \times [m] \to \mathbb{R}$,*

$$\max_{p \in \Delta_n} \min_{q \in \Delta_m} \sum_{i,j} p(i)q(j)u(i,j) = \min_{q \in \Delta_m} \max_{p \in \Delta_n} \sum_{i,j} p(i)q(j)u(i,j)$$

The statement of the minimax theorem is often referred to as *duality* as it swaps the min and max. This result can be used to establish strong duality for linear programming. It was proven by Maurice Sion in the 1950's that von Neumann's notion of duality can be extended further, for a much larger class of input spaces and a more general class of functions.

Theorem 36 (Sion[1], 1958 [71])**.** *Given convex compact sets $\mathcal{X} \subset \mathbb{R}^n, \mathcal{Y} \subset \mathbb{R}^m$, and a function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ convex and concave in its first and second arguments respectively, we have*

$$\inf_{\mathbf{x} \in \mathcal{X}} \sup_{\mathbf{y} \in \mathcal{Y}} f(\mathbf{x}, \mathbf{y}) = \sup_{\mathbf{y} \in \mathcal{Y}} \inf_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}, \mathbf{y}).$$

In the present work we shall not need anything quite so general, although we use this theorem to generalize slightly the class of two-player zero-sum games. Rather than define the actions of our players as being drawn randomly from discrete sets $[n]$ and $[m]$, let the players' decision space be characterized by given compact convex sets $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{Y} \subset \mathbb{R}^m$ respectively. In addition, we shall assume that the utility is characterized by a *biaffine* function $u : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$; that is, $u(\alpha\mathbf{x} + (1-\alpha)\mathbf{x}', \mathbf{y}) = \alpha u(\mathbf{x}, \mathbf{y}) + (1-\alpha)u(\mathbf{x}', \mathbf{y})$ and $u(\mathbf{x}, \alpha\mathbf{y} + (1-\alpha)\mathbf{y}') = \alpha u(\mathbf{x}, \mathbf{y}) + (1-\alpha)u(\mathbf{x}, \mathbf{y}')$ for every $0 \leq \alpha \leq 1$, $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ and $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$. Following Sion's theorem, we arrive at the following.

Corollary 37. *For compact convex sets $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{Y} \subset \mathbb{R}^m$ and any biaffine function $u : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, we have*

$$\max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{y} \in \mathcal{Y}} u(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{y} \in \mathcal{Y}} \max_{\mathbf{x} \in \mathcal{X}} u(\mathbf{x}, \mathbf{y})$$

This alternative description of a zero-sum game has two advantages. First, we now assume that both players are deterministic. That is, we have converted the notion of a randomized strategy on a discrete action space to a deterministic strategy $\mathbf{x}$ inside of a convex set $\mathcal{X}$. Rather than evaluate the expected utility of a randomized action, this expectation is now incorporated via the linearity of $u(\cdot, \cdot)$. Note, crucially, that the assumptions that u

[1]The original paper of Sion proves an even more general statement than what we give.

is biaffine and $\mathcal{X}$ and $\mathcal{Y}$ are convex imply that neither player gains from randomness, as $\mathbb{E}_{\mathbf{x}}\,\mathbb{E}_{\mathbf{y}}\; u(\mathbf{x},\mathbf{y}) = u(\mathbb{E}_{\mathbf{x}}\,\mathbf{x}, \mathbb{E}_{\mathbf{y}}\,\mathbf{y})$.

A second advantage of this framework is that it allows us to work with action spaces that might seem prohibitively large. For example, we can imagine a game in which each player must select a route in a graph G between two endpoints, and the utility is the amount of overlap of their paths. The set of paths in a graph is exponential, and even counting the number of such paths is $\#P$-hard. However, we may instead set $\mathcal{X}$ and $\mathcal{Y}$ to be the *flow polytope* of G. The flow polytope can be described by a polynomially-sized number of constraints, and hence is much easier to work with.

5.2.2 Vector-Valued Games

Let us now turn our attention to Blackwell's question: what can be guaranteed when the utility function of the zero-sum game is *vector-valued*? Following the definition in the previous section, we can define a vector-valued game in terms of some biaffine utility function $\mathbf{v} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ from a product of two convex compact decision spaces $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{Y} \subset \mathbb{R}^m$ to d-dimensional space. The biaffine property is defined in the natural way.

Note that we may not apply our usual notions of utility maximization when dealing with vector-valued games—what does it mean to "maximize" a vector? Furthermore, the concept of "zero-sum" is not immediately clear. Blackwell proposed the following framework: suppose that the Player, who selects $\mathbf{x} \in \mathcal{X}$, would like his vector payoff $\mathbf{v}(\mathbf{x},\mathbf{y})$ to land inside of a particular closed convex set $S \subset \mathbb{R}^d$, where S is fixed and known to both players. We shall say that the Player wants to *satisfy* S. The Adversary, who selects $\mathbf{y} \in \mathcal{Y}$, would like to prevent the Player from satisfying S.

Let us return our attention to the simple case of scalar-valued games discussed in Section 5.2.1. The duality statement achieved in the Minimax Theorem, typically stated in terms of swapping the order of min and max, can instead be formulated in terms of swapping quantifiers $\forall$ and $\exists$.

Proposition 38. *For any convex compact sets $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{Y} \subset \mathbb{R}^m$, and any biaffine utility function $u : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, we have the following implication for any $c \in \mathbb{R}$:*

$$\forall \mathbf{y} \in \mathcal{Y}\; \exists \mathbf{x} \in \mathcal{X} :\quad u(\mathbf{x},\mathbf{y}) \in [c,\infty) \quad\Longrightarrow\quad \exists \mathbf{x} \in \mathcal{X}\; \forall \mathbf{y} \in \mathcal{Y} :\quad u(\mathbf{x},\mathbf{y}) \in [c,\infty).$$

This proposition is simply another way to state duality, in the following form:

$$\min_{\mathbf{y}\in\mathcal{Y}} \max_{\mathbf{x}\in\mathcal{X}} u(\mathbf{x},\mathbf{y}) \geq c \quad\Longrightarrow\quad \max_{\mathbf{x}\in\mathcal{X}} \min_{\mathbf{y}\in\mathcal{Y}} u(\mathbf{x},\mathbf{y}) \geq c.$$

Put another way, if the Player can earn c by choosing his strategy *with knowledge of* the Adversary's strategy, then he can earn c obliviously as well.

Here we have simply taken the Minimax Theorem and stated it in terms of satisfying a set, namely the set $S = [c,\infty)$ for some value c. This interpretation begs the question: can

we achieve a similar "duality" statement for vector-valued games? In other words, given a biaffine utility function $\mathbf{v} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ and any convex set $S \subset \mathbb{R}^d$, does the statement

$$\forall \mathbf{y} \in \mathcal{Y}\ \exists \mathbf{x} \in \mathcal{X} :\ \ \mathbf{v}(\mathbf{x}, \mathbf{y}) \in S \quad \Longrightarrow \quad \exists \mathbf{x} \in \mathcal{X}\ \forall \mathbf{y} \in \mathcal{Y} :\ \ \mathbf{v}(\mathbf{x}, \mathbf{y}) \in S$$

hold in general? The answer, unfortunately, is *no!* Consider the following easy example: $\mathcal{X} = \mathcal{Y} := [0, 1]$, the payoff is simply $\mathbf{v}(x, y) := (x, y)$ for $x, y \in [0, 1]$, and the set in question is $S := \{(z, z)\ \forall z \in [0, 1]\}$. Certainly the premise is true, since for every y there exists an x, namely $x = y$, such that $\mathbf{v}(x, y) \in S$. On the other hand, there is no such single x for which $\mathbf{v}(x, y) \in S$ for any y.

5.2.3 Blackwell Approachability

While we might hope that minimax duality, framed in terms of set satisfiability, would extend from scalar-valued games to vector-valued games, the previous example appears to be a nail in the coffin. But in fact the story is not quite so bad: the proposed example is difficult because it is a *one-shot* game. What Blackwell observed, and led to the Approchability Theorem, is that if the game is played *repeatedly* then one can achieve duality "in the limit." To make this precise we introduce some definitions.

Definition 39. *A* Blackwell instance *is a tuple $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot, \cdot), S)$, with $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{Y} \subset \mathbb{R}^m$ compact and convex, $\mathbf{v} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ biaffine, and $S \subset \mathbb{R}^d$ convex and closed. For any instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot, \cdot), S)$, we say that*

- *S is* satisfiable *if $\exists \mathbf{x} \in \mathcal{X}\ \forall \mathbf{y} \in \mathcal{Y} :\ \mathbf{v}(\mathbf{x}, \mathbf{y}) \in S$.*
- *S is* response-satisfiable *if $\forall \mathbf{y} \in \mathcal{Y}\ \exists \mathbf{x} \in \mathcal{X} :\ \mathbf{v}(\mathbf{x}, \mathbf{y}) \in S$.*
- *S is* halfspace-satisfiable *if, for any halfspace $H \supseteq S$, H is satisfiable.*

To recap, when our utility function $\mathbf{v}$ is scalar-valued, i.e. for zero-sum games where $d = 1$, then minimax duality holds and, according to Proposition 38, this be rephrased as "If $S := [c, \infty)$ is response-satisfiable then S is satisfiable." On the other hand, for vector-valued games it is not the case in general that "S is response-satisfiable $\implies$ S is satisfiable" for arbitrary sets S. What Blackwell showed is that response-satisfiability does lead to a weaker condition, termed *approachability*. Before we define this precisely, let us use the notation $\texttt{dist}(\mathbf{z}, U)$ to denote the distance between a point $\mathbf{z}$ and some convex set U, that is $\inf_{\mathbf{x} \in U} \|\mathbf{z} - \mathbf{x}\|$.

Definition 40. *Given a Blackwell instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot, \cdot), S)$, we say that S is* approachable *if there exists some algorithm $\mathcal{A}$ which selects points in $\mathcal{X}$ such that, for any sequence $\mathbf{y}_1, \mathbf{y}_2, \ldots \in \mathcal{Y}$, we have*

$$\texttt{dist}\Big(\tfrac{1}{T} \textstyle\sum_{t=1}^{T} \mathbf{v}(\mathbf{x}_t, \mathbf{y}_t), S\Big) \to 0 \quad \text{as} \quad T \to \infty,$$

where $\mathbf{x}_t \leftarrow \mathcal{A}(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{t-1})$.

Under this new notion, we now allow the Player to implement an *adaptive strategy* for a repeated version of the game, and we require that the average utility vector becomes arbitrarily close to S. Intuitively, we may think of approachability as "satisfiability in the limit".

Theorem 41 (Blackwell's Approachability Theorem [18])**.** *For any Blackwell instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S)$, S is approachable if and only if it is response-satisfiable.*

The beauty of this theorem is that, while we may not be able to satisfy S in a one-shot version of the game, we can satisfy the set "on average" if we may play the game indefinitely.

This version of the theorem, which appears in Evan-Dar et al. [30], is not the one usually attributed to Blackwell. The original theorem uses the concept of halfspace satisfiability. It is not difficult to establish the equivalence of the two statements via the following lemma, whose proof uses a nice application of minimax duality.

Lemma 42. *For any Blackwell instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S)$, S is response-satisfiable if and only if it is halfspace-satisfiable.*

Proof. ($\Longrightarrow$) Assume that S is response-satisfiable. Hence, for any $\mathbf{y}$ there is an $\mathbf{x}_\mathbf{y}$ such that $\mathbf{v}(\mathbf{x}_\mathbf{y}, \mathbf{y}) \in S$. Now take any halfspace $H \supset S$ parameterized by $\boldsymbol{\theta}, c$, that is $H = \{\mathbf{z} : \langle \boldsymbol{\theta}, \mathbf{z} \rangle \leq c\}$. Then let us define a scalar-valued game with utility $u(\mathbf{x}, \mathbf{y}) = \langle \boldsymbol{\theta}, \mathbf{v}(\mathbf{x}, \mathbf{y}) \rangle$. Notice that $H \supset S$ implies that $\langle \boldsymbol{\theta}, \mathbf{z} \rangle \leq c$ for all $\mathbf{z} \in S$. Since S is response-satisfiable, for every $\mathbf{y}$ there is an $\mathbf{x}_\mathbf{y}$ such that $\mathbf{v}(\mathbf{x}_\mathbf{y}, \mathbf{y}) \in S \implies u(\mathbf{x}_\mathbf{y}, \mathbf{y}) \leq c$. We then immediately see that

$$\max_{\mathbf{y} \in \mathcal{Y}} \min_{\mathbf{x} \in \mathcal{X}} u(\mathbf{x}, \mathbf{y}) \leq \max_{\mathbf{y} \in \mathcal{Y}} u(\mathbf{x}_\mathbf{y}, \mathbf{y}) \leq c.$$

It follows from Corollary 37 that $\min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbf{y} \in \mathcal{Y}} u(\mathbf{x}, \mathbf{y}) \leq c$. Let $\mathbf{x}^* \in \mathcal{X}$ be any minimizer of the latter expression and notice that, for any $\mathbf{y} \in \mathcal{Y}$, we have that $u(\mathbf{x}^*, \mathbf{y}) \leq c$. It follows immediately that H is satisfiable.

($\Longleftarrow$) Assume that S is not response-satisfiable. Hence, there must exists some $\mathbf{y}_0 \in \mathcal{Y}$ such that $\mathbf{v}(\mathbf{x}, \mathbf{y}_0) \notin S$ for every $\mathbf{x} \in \mathcal{X}$. Consider the set $U := \{\mathbf{v}(\mathbf{x}, \mathbf{y}_0) \text{ for all } \mathbf{x} \in \mathcal{X}\}$ and notice that U is convex since $\mathcal{X}$ is convex and $\mathbf{v}(\cdot, \mathbf{y}_0)$ is affine. Furthermore, because S is convex and $S \cap U = \emptyset$ by assumption, there must exist some halfspace H separating the two sets, that is $S \subseteq H$ and $H \cap U = \emptyset$. By construction, we see that for any $\mathbf{x}$, $\mathbf{v}(\mathbf{x}, \mathbf{y}_0) \notin H$ and hence H is not satisfiable. It follows immediately that S is not halfspace-satisfiable. □

Although it is not posed in this language, Blackwell's original theorem uses the concept of a *halfspace oracle*. Given a Blackwell instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S)$, define a halfspace oracle to be a function $\mathcal{O}$ that takes as input any halfspace $H \supset S$ and returns a point $\mathcal{O}(H) = \mathbf{x}_H \in \mathcal{X}$, and we shall refer to a halfspace oracle as *valid* if it satisfies that for each halfspace $H \supset S$, $\mathbf{v}(\mathbf{x}_H, \mathbf{y}) \in H$ for any $\mathbf{y} \in \mathcal{Y}$.

Theorem 43. *For any Blackwell instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S)$, the set S is approachable if and only if there exists a valid halfspace oracle.*

Notice that the existence of a valid halfspace oracle is equivalent to the halfspace-satisfiability condition. Hence, via Lemma 42, this theorem is equivalent to Theorem 41.

To achieve approachability, following Definition 40 one must construct an algorithm $\mathcal{A}$ that maps the observed subsequence $\mathbf{y}_1, \ldots, \mathbf{y}_{t-1} \in \mathcal{Y}$ to a point $\mathbf{x}_t \in \mathcal{X}$. By the previous theorem, in order for the set S to be approachable, there must be a valid halfspace oracle $\mathcal{O}$, and hence $\mathcal{A}$ may make calls to $\mathcal{O}$. Blackwell actually provides such an algorithm, quite elegant for its simplicity, which can be found in his original work [18] as well as in the book of Cesa-Bianchi and Lugosi [24].

We note that, when an approachability algorithm $\mathcal{A}$ is adapted to a Blackwell instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S)$, and makes calls to a halfspace oracle $\mathcal{O}$, we may write $\mathcal{A}^{\mathcal{O}}_{\mathcal{X},\mathcal{Y},\mathbf{v},S}$ to make the dependence clear.

5.3 Online Linear Optimization

Online Convex Optimization (OCO) has become a popular topic within Machine Learning since it was introduced by Zinkevich in 2003 [77], and there has been much followup work [69, 67, 47, 1]. It provides a generic problem template and was shown to generalize several existing problems in the realm of online learning and repeated decision making. Among these are online pattern classification, the "experts" or "hedge" setting, and sequential portfolio optimization [37, 46].

In the OCO setting, we imagine an online game between Player and Nature. Assume the Player is given a convex decision set $\mathcal{K} \subset \mathbb{R}^d$ and must make a sequence of a decisions $\mathbf{x}_1, \mathbf{x}_2, \ldots \in \mathcal{K}$. After committing to $\mathbf{x}_t$, Nature reveals a convex loss function ℓ_t, and Player pays $\ell_t(\mathbf{x}_t)$. The performance of the Player is typically measured by *regret* which we shall define below. In the present work we shall be concerned with the more specific problem of Online Linear Optimization (OLO) where the loss functions are assumed to be linear, $\ell_t(\mathbf{x}) = \langle \mathbf{f}_t, \mathbf{x} \rangle$ for some $\mathbf{f}_t \in \mathbb{R}^d$.

We define the Player's adaptive strategy $\mathcal{L}$, which we refer to as an *OLO algorithm*, as a function which takes as input a subsequence of loss vectors $\mathbf{f}_1, \ldots, \mathbf{f}_{t-1}$ and returns a point $\mathbf{x}_t \leftarrow \mathcal{L}(\mathbf{f}_1, \ldots, \mathbf{f}_{t-1})$, where $\mathbf{x}_t \in \mathcal{K}$.

Definition 44. *Given an OLO algorithm $\mathcal{L}$ and a sequence of loss vectors $\mathbf{f}_1, \mathbf{f}_2, \ldots \in \mathbb{R}^d$, let* $\mathrm{Regret}(\mathcal{L}; \mathbf{f}_{1:T}) := \sum_{t=1}^{T} \langle \mathbf{f}_t, \mathbf{x}_t \rangle - \min_{\mathbf{x} \in K} \sum_{t=1}^{T} \langle \mathbf{f}_t, \mathbf{x} \rangle$. *When the sequence of loss vectors is clear, we may simply write* $\mathrm{Regret}_T(\mathcal{L})$.

An important question is whether an OLO algorithm has a regret rate which scales *sublinearly* in T. A sublinear regret is key, for then our average performance, in the long run, is essentially no worse than the best in hindsight. We use the term *no-regret* algorithm when it possesses this property.

Theorem 45. *For any bounded decision set $\mathcal{K} \subset \mathbb{R}^d$ there exists an algorithm $\mathcal{L}_{\mathcal{K}}$ such that* $\mathrm{Regret}_T(\mathcal{L}_{\mathcal{K}}) = o(T)$ *for any sequence of loss vectors $\{\mathbf{f}_t\}$ with bounded norm.*

Later in the chapter we provide one such algorithm, known as Online Gradient Descent, proposed by Zinkevich [77].

Before proceeding, let us demonstrate the value of no-regret algorithms by proving an aforementioned result. We shall sketch a proof of the minimax statement of Corollary 37. Assume we are given convex and compact decision space $\mathcal{X} \subset \mathbb{R}^n$ and $\mathcal{Y} \subset \mathbb{R}^m$, and without loss of generality assume we have a utility function $u : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ of the form $u(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top M \mathbf{y}$ for some $M \in \mathbb{R}^{n \times m}$. Weak duality, i.e. $\min_{\mathbf{y} \in \mathcal{Y}} \max_{\mathbf{x} \in \mathcal{X}} \mathbf{x}^\top M \mathbf{y} \geq \max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{y} \in \mathcal{Y}} \mathbf{x}^\top M \mathbf{y}$ is trivial, and so we turn our attention to the reverse inequality. We shall imagine our game is played repeatedly, where on round t the first player chooses $\mathbf{x}_t$ and the second chooses $\mathbf{y}_t$, but where both players select their strategies according to a no-regret algorithm. For every t we shall set $\mathbf{x}_t \leftarrow \mathcal{L}_{\mathcal{X}}(\mathbf{f}_1, \ldots, \mathbf{f}_{t-1})$ and $\mathbf{y}_t \leftarrow \mathcal{L}_{\mathcal{Y}}(\mathbf{g}_1, \ldots, \mathbf{g}_{t-1})$, where we define the vectors $\mathbf{f}_t := -M\mathbf{y}_t$ and $\mathbf{g}_t^\top := \mathbf{x}_t^\top M$. By applying the definition of regret twice, we have

$$\frac{1}{T}\sum_{t=1}^T \mathbf{x}_t^\top M \mathbf{y}_t = \min_{\mathbf{y} \in \mathcal{Y}} \left(\frac{1}{T}\sum_{t=1}^T \mathbf{x}_t\right)^\top M\mathbf{y} + \frac{\text{Regret}_T(\mathcal{L}_{\mathcal{Y}})}{T} \leq \max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{y} \in \mathcal{Y}} \mathbf{x}^\top M \mathbf{y} + \frac{o(T)}{T}, \quad (5.1)$$

$$\frac{1}{T}\sum_{t=1}^T \mathbf{x}_t^\top M \mathbf{y}_t = \max_{\mathbf{x} \in \mathcal{X}} \mathbf{x}^\top M \left(\frac{1}{T}\sum_{t=1}^T \mathbf{y}_t\right) - \frac{\text{Regret}(\mathcal{L}_{\mathcal{X}})}{T} \geq \min_{\mathbf{y} \in \mathcal{Y}} \max_{\mathbf{x} \in \mathcal{X}} \mathbf{x}^\top M \mathbf{y} - \frac{o(T)}{T}. \quad (5.2)$$

Combining these two statements gives $\min_{\mathbf{y} \in \mathcal{Y}} \max_{\mathbf{x} \in \mathcal{X}} \mathbf{x}^\top M \mathbf{y} \leq \max_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{y} \in \mathcal{Y}} \mathbf{x}^\top M \mathbf{y} + \frac{o(T)}{T}$. Of course, we can let $T \to \infty$ which immediately gives the desired inequality.

The previous example foreshadows a key result of this chapter, which is that any no-regret learning algorithm can be converted into an approachability strategy. If we interpret Blackwell Approachability as a generalized form of Minimax Duality for vector-valued games then it may come as no surprise that regret-minimizing algorithms would provide a tool in establishing both game-theoretic results. However, in a certain sense regret-minimization is too heavy a hammer for proving Minimax Duality. For one, the above proof requires that we imagine a repeated version of the game, whereas scalar-valued game duality holds even for one-shot. Indeed, more standard proofs of von Neumann's result do not rely on repeated play. Blackwell Approachability, on the other hand, fundamentally involves repeated play, and in fact we shall show that regret-minimization is the perfectly-sized hammer, as it is *algorithmically equivalent* to approachability.

5.4 Equivalence of Approachability and Regret Minimization

5.4.1 Convex Cones and Conic Duality

We shall define some basic notions and then state some simple lemmas. Henceforth we use the notation $B_2(r)$ to refer to the ℓ_2-norm ball of radius r. The notation $\mathbf{x}' \oplus \mathbf{x}$ is the vector concatenation of $\mathbf{x}$ and $\mathbf{x}'$.

Definition 46. *A set $X \subset \mathbb{R}^d$ is a* cone *if it is closed under multiplication by nonnegative scalars, and X is a convex cone if it is also closed under element addition. Given any set*

$K \subset \mathbb{R}^d$, define the conic hull $\texttt{cone}(K) := \{\alpha \mathbf{x} : \alpha \in \mathbb{R}_+, \mathbf{x} \in K\}$ *which is also a cone in $\mathbb{R}^d$. Also, given any convex cone $C \subset \mathbb{R}^d$, we can define the* polar cone *of C as*

$$C^0 := \{\boldsymbol{\theta} \in \mathbb{R}^d : \langle \boldsymbol{\theta}, \mathbf{x} \rangle \leq 0 \text{ for all } \mathbf{x} \in C\}.$$

It is easily checked that if K is convex then $\texttt{cone}(K)$ is also convex. The following Lemma is folklore.

Lemma 47. *If C is a convex cone then (1) $(C^0)^0 = C$ and (2) supporting hyperplanes in C^0 correspond to points $\mathbf{x} \in C$, and vice versa. That is, given any supporting hyperplane H of C^0, H can be written exactly as $\{\boldsymbol{\theta} \in \mathbb{R}^d : \langle \boldsymbol{\theta}, \mathbf{x} \rangle = 0\}$ for some vector $\mathbf{x} \in C$ that is unique up to scaling.*

The distance to a cone can conveniently be measure via a "dual formulation," as we now show.

Lemma 48. *For every convex cone C in $\mathbb{R}^d$*

$$\texttt{dist}(\mathbf{x}, C) = \max_{\boldsymbol{\theta} \in C^0 \cap B_2(1)} \langle \boldsymbol{\theta}, \mathbf{x} \rangle \tag{5.3}$$

Proof. We need two simple observations. Define $\pi_C(\mathbf{x})$ as the projection of $\mathbf{x}$ onto C. Then clearly, for any $\mathbf{x}$,

$$\texttt{dist}(\mathbf{x}, C) = \|\mathbf{x} - \pi_C(\mathbf{x})\| \tag{5.4}$$
$$\langle \mathbf{x} - \pi_C(\mathbf{x}), \mathbf{y} \rangle \leq 0 \ \forall \mathbf{y} \in C \text{ and hence } \mathbf{x} - \pi_C(\mathbf{x}) \in C^0 \tag{5.5}$$
$$\langle \mathbf{x} - \pi_C(\mathbf{x}), \pi_C(\mathbf{x}) \rangle = 0 \tag{5.6}$$

Given any $\boldsymbol{\theta} \in C^0$ with $\|\boldsymbol{\theta}\| \leq 1$, since $\pi_C(\mathbf{x}) \in C$ we have that

$$\langle \boldsymbol{\theta}, \mathbf{x} \rangle \leq \langle \boldsymbol{\theta}, \mathbf{x} - \pi_C(\mathbf{x}) \rangle \leq \|\boldsymbol{\theta}\| \|\mathbf{x} - \pi_C(\mathbf{x})\| \leq \|\mathbf{x} - \pi_C(\mathbf{x})\|,$$

which immediately implies that $\max_{\boldsymbol{\theta} \in C^0, \|\boldsymbol{\theta}\| \leq 1} \langle \boldsymbol{\theta}, \mathbf{x} \rangle \leq \texttt{dist}(\mathbf{x}, C)$. Furthermore, by selecting $\boldsymbol{\theta} = \frac{\mathbf{x} - \pi_C(\mathbf{x})}{\|\mathbf{x} - \pi_C(\mathbf{x})\|}$ which has norm one and, by (5.4), is in C^0, we see that

$$\max_{\boldsymbol{\theta} \in C^0, \|\boldsymbol{\theta}\| \leq 1} \langle \boldsymbol{\theta}, \mathbf{x} \rangle \geq \left\langle \frac{\mathbf{x} - \pi_C(\mathbf{x})}{\|\mathbf{x} - \pi_C(\mathbf{x})\|}, \mathbf{x} \right\rangle = \left\langle \frac{\mathbf{x} - \pi_C(\mathbf{x})}{\|\mathbf{x} - \pi_C(\mathbf{x})\|}, \mathbf{x} - \pi_C(\mathbf{x}) \right\rangle = \|\mathbf{x} - \pi_C(\mathbf{x})\|,$$

which implies that $\max_{\boldsymbol{\theta} \in C^0, \|\boldsymbol{\theta}\| \leq 1} \langle \boldsymbol{\theta}, \mathbf{x} \rangle \geq \texttt{dist}(\mathbf{x}, C)$ and hence we are done. □

Our results require looking at convex cones rather than convex sets, hence we must consider the process of converting a set into a cone. In order to not lose information about the underlying set $\mathcal{K} \subset \mathbb{R}^d$, we shall embed the set into a higher dimension, and instead look at $\texttt{cone}(\{\kappa\} \times \mathcal{K}) \subset \mathbb{R}^{d+1}$, where $\kappa := \max_{\mathbf{x} \in \mathcal{K}} \|\mathbf{x}\|$ is the diameter of $\mathcal{K}$. We prove that this process of "lifting" and conifying does not perturb distances by more than a constant.

Lemma 49. *Consider a compact convex set $\mathcal{K} \subseteq \mathcal{H}$ in $\mathbb{R}^d$ and $\mathbf{x} \notin \mathcal{K}$. Let $\tilde{\mathbf{x}} := \kappa \oplus \mathbf{x}$ and $\tilde{\mathcal{K}} := \{\kappa\} \times \mathcal{K}$. Then we have*

$$\texttt{dist}(\tilde{\mathbf{x}}, \texttt{cone}(\tilde{\mathcal{K}})) \leq \texttt{dist}(\mathbf{x}, \mathcal{K}) \leq 2\texttt{dist}(\tilde{\mathbf{x}}, \texttt{cone}(\tilde{\mathcal{K}})) \tag{5.7}$$

Proof. Since $\texttt{dist}(\tilde{\mathbf{x}}, \tilde{\mathcal{K}}) = \texttt{dist}(\mathbf{x}, \mathcal{K})$ and $\tilde{\mathcal{K}} \subset \texttt{cone}(\tilde{\mathcal{K}})$, the first inequality follows immediately.

For notational convenience let $\mathbf{w} = \pi_{\texttt{cone}(\tilde{\mathcal{K}})}(\mathbf{y})$ be the projection of $\mathbf{y}$ onto $\texttt{cone}(\tilde{\mathcal{K}})$ and $\mathbf{v} = \pi_{\tilde{\mathcal{K}}}(\mathbf{y})$ be the projection onto $\tilde{\mathcal{K}}$. Consider the plane determined by the three points $\tilde{\mathbf{x}}, \mathbf{w}, \mathbf{v}$. Notice that the triangle $\Delta(\tilde{\mathbf{x}}, \mathbf{w}, \mathbf{v})$ is similar to the triangle $\Delta(\mathbf{0}, \kappa \oplus \mathbf{0}, \mathbf{v})$, and hence by triangle similarity

$$\frac{\|\mathbf{v}\|}{\|\kappa \oplus \mathbf{0}\|} = \frac{\|\tilde{\mathbf{x}} - \mathbf{v}\|}{\|\tilde{\mathbf{x}} - \mathbf{w}\|} = \frac{\texttt{dist}(\tilde{\mathbf{x}}, \tilde{\mathcal{K}})}{\texttt{dist}(\tilde{\mathbf{x}}, \texttt{cone}(\tilde{\mathcal{K}}))}$$

For a visual aid, we provide a picture of this triangle similarity in Figure 5.1. Since $\mathbf{v} \in \tilde{\mathcal{K}}$ we have $\|\mathbf{v}\| \leq \|\tilde{\mathcal{K}}\| \leq 2\kappa$. In addition $\|\kappa \oplus \mathbf{0}\| = \kappa$ and the result follows. □

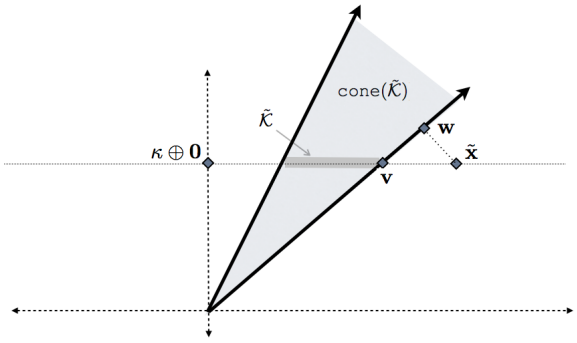

Figure 5.1. A geometric interpretation of the proof of Lemma 49.

5.4.2 Duality Theorems

In the previous sections we have presented two sequential decision problems, summarized in Figure 5.2. We now show that these two decision problems are *algorithmically equivalent*: any strategy (algorithm) that achieves approachability can be converted into an algorithm that achieves low-regret, and vice versa.

We present this equivalence as a pair of reductions. In Algorithm 5 we show how a learner, presented with a OLO problem characterized by a decision set $\mathcal{K}$ and an arriving sequence of loss vectors $\mathbf{f}_1, \mathbf{f}_2, \ldots$, can minimize regret with only oracle access to some approachability algorithm $\mathcal{A}$. In Algorithm 6 we show how a player, presented with a Blackwell instance

Blackwell Approachability Problem	**Online Linear Optimization Problem**
Given a Blackwell instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S)$ and a valid halfspace oracle $\mathcal{O} : H \mapsto \mathbf{x}_H \in \mathcal{X}$, construct an algorithm $\mathcal{A}$ so that, for any sequence $\mathbf{y}_1, \mathbf{y}_2, \ldots \in \mathcal{Y}$, $$\texttt{dist}(\tfrac{1}{T}\sum_{t=1}^{T} \mathbf{v}(\mathbf{x}_t, \mathbf{y}_t), S) \to 0$$ where $\mathbf{x}_t \leftarrow \mathcal{A}(\mathbf{y}_1, \ldots, \mathbf{y}_{t-1})$.	Given a compact convex set $\mathcal{K} \subset \mathbb{R}^d$, construct a learning algorithm $\mathcal{L}$ so that, for any sequence of loss vectors $\mathbf{f}_1, \mathbf{f}_2, \ldots \in \mathbb{R}^d$ we have vanishing regret, that is $$\sum_{t=1}^{T}\langle \mathbf{f}_t, \mathbf{x}_t\rangle - \min_{\mathbf{x}\in K}\sum_{t=1}^{T}\langle \mathbf{f}_t, \mathbf{x}\rangle = o(T),$$ where $\mathbf{x}_t \leftarrow \mathcal{L}(\mathbf{f}_1, \ldots, \mathbf{f}_{t-1})$.

Figure 5.2. A summary of Blackwell Approachability and Online Linear Optimization

$(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S)$ and a valid halfspace oracle $\mathcal{O}$, can achieve approachability when only given oracle access to a no-regret OLO algorithm $\mathcal{L}$. For the remainder of the chapter, for a given Blackwell instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S)$ and approachability algorithm $\mathcal{A}$, $D(\mathcal{A}; \mathbf{y}_1, \ldots, \mathbf{y}_T)$ shall refer to the rate of approachability $\texttt{dist}\left(\frac{1}{T}\sum_{t=1}^{T} \mathbf{v}(\mathbf{x}_t, \mathbf{y}_t), S\right)$. We shall write $D_T(\mathcal{A})$ when the input sequence is clear. For the convex set $\mathcal{K}$, we shall let $\kappa := \max_{\mathbf{x}\in\mathcal{K}} \|\mathbf{x}\|$, the "norm" of the set $\mathcal{K}$.

Algorithm 5 Conversion of Approachability Alg. $\mathcal{A}$ to Online Linear Optimization Alg. $\mathcal{L}$

1: Input: compact convex decision set $\mathcal{K} \subset \mathbb{R}^d$
2: Input: sequence of cost functions $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_T \in B_2(1)$
3: Input: approachability oracle $\mathcal{A}$
4: Set: Blackwell instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S)$, where $\mathcal{X} := \mathcal{K}$, $\mathcal{Y} := B_2(1)$, $\mathbf{v}(\mathbf{x}, \mathbf{f}) = \frac{\langle \mathbf{f}, \mathbf{x}\rangle}{\kappa} \oplus -\mathbf{f}$, and $S := \texttt{cone}(\{\kappa\} \times \mathcal{K})^0$
5: Construct: valid halfspace oracle $\mathcal{O}$ // Existence established in Lemma 50
6: **for** $t = 1, \ldots, T$ **do**
7: Let: $\mathcal{L}(\mathbf{f}_1, \ldots, \mathbf{f}_{t-1}) := \mathcal{A}^{\mathcal{O}}_{\mathcal{X},\mathcal{Y},\mathbf{v},S}(\mathbf{f}_1, \ldots, \mathbf{f}_{t-1})$
8: Receive: cost function $\mathbf{f}_t$
9: **end for**

In Algorithm 5 we require the construction of a valid halfspace oracle. In the lemma below we give one such oracle and prove that it is valid, but we note that this construction may not be the most efficient in general; any particular scenario may give rise to a simpler and faster construction.

Lemma 50. *There exists a valid halfspace oracle for the Blackwell instance in Algorithm 5.*

Proof. Assume we have some halfspace H which contains $S = \texttt{cone}(\{\kappa\} \times \mathcal{K})^0$. We can assume without loss of generality that H is tangent to S and, since S is a cone, H meets the origin; that is, $H = \{\boldsymbol{\theta} : \langle \boldsymbol{\theta}, \mathbf{z}_H\rangle \le 0\}$ for some $\mathbf{z}_H \in \mathbb{R}^d$. Furthermore, $H \supset \texttt{cone}(\{\kappa\} \times \mathcal{K})^0$ implies that $\mathbf{z}_H \in (\texttt{cone}(\{\kappa\} \times \mathcal{K})^0)^0 = \texttt{cone}(\{\kappa\} \times \mathcal{K})$. Equivalently, $\mathbf{z}_H = \alpha(\kappa \oplus \mathbf{x}_H)$

for some $\mathbf{x}_H \in \mathcal{K}$ and some $\alpha > 0$. With this in mind, we construct our oracle by setting $\mathbf{x}_H \leftarrow \mathcal{O}(H)$.

It remains to prove that this halfspace oracle is valid. We compute $\langle \mathbf{v}(\mathbf{x}_H, \mathbf{f}), \mathbf{z}_H \rangle$:

$$\langle \mathbf{v}(\mathbf{x}_H, \mathbf{f}), \mathbf{z}_H \rangle = \langle \kappa^{-1}\langle \mathbf{f}, \mathbf{x}_H \rangle \oplus -\mathbf{f}, \alpha\kappa \oplus \alpha\mathbf{x}_H \rangle = \alpha\langle \mathbf{f}, \mathbf{x}_H \rangle + \langle -\mathbf{f}, \alpha\mathbf{x}_H \rangle = 0.$$

By definition, $\langle \mathbf{v}(\mathbf{x}_H, \mathbf{f}), \mathbf{z}_H \rangle \le 0$ implies that $\mathbf{v}(\mathbf{x}_H, \mathbf{f}) \in H$ for any $\mathbf{f}$ and we are done. □

Theorem 51. *The reduction defined in Algorithm 5, for any input algorithm $\mathcal{A}$, produces an OLO algorithm $\mathcal{L}$ such that $\frac{\text{Regret}(\mathcal{L})}{T} \le 2\kappa D_T(\mathcal{A})$.*

Proof. Applying Lemmas 48 and 47 to the definition of $D_T(\mathcal{A})$ gives

$$D_T(\mathcal{A}) \equiv \texttt{dist}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{v}(\mathbf{x}_t, \mathbf{f}_t), S\right) = \max_{\mathbf{w} \in \texttt{cone}(\kappa \oplus \mathcal{K}) \cap B_2^d(1)} \left\langle \frac{1}{T}\sum_{t=1}^{T}\mathbf{v}(\mathbf{x}_t, \mathbf{f}_t), \mathbf{w} \right\rangle \tag{5.8}$$

Notice that, in this optimization, we can assume w.l.o.g. that $\|\mathbf{w}\| = 1$, or $\mathbf{w} = \mathbf{0}$. In the former case we can write $\mathbf{w} = \frac{\kappa \oplus \mathbf{x}}{\|\kappa \oplus \mathbf{x}\|}$ for some $\mathbf{x} \in \mathcal{K}$, and we drop the latter case to obtain the inequality

$$\begin{aligned} D_T(\mathcal{A}) \ge \max_{\mathbf{x} \in \mathcal{K}} \left\langle \frac{1}{T}\sum_{t=1}^{T}\mathbf{v}(\mathbf{x}_t, \mathbf{f}_t), \frac{\kappa \oplus \mathbf{x}}{\|\kappa \oplus \mathbf{x}\|} \right\rangle &= \frac{1}{T}\max_{\mathbf{x} \in \mathcal{K}} \frac{\left(\sum_{t=1}^{T}\langle \mathbf{f}_t, \mathbf{x}_t \rangle - \sum_{t=1}^{T}\langle \mathbf{f}_t, \mathbf{x} \rangle\right)}{\|\kappa \oplus \mathbf{x}\|} \\ &\ge \frac{\frac{1}{T}\left(\sum_{t=1}^{T}\langle \mathbf{f}_t, \mathbf{x}_t \rangle - \sum_{t=1}^{T}\langle \mathbf{f}_t, \mathbf{x}^* \rangle\right)}{\|\kappa \oplus \mathbf{x}^*\|} \\ &\ge \frac{\frac{1}{T}\text{Regret}_T(\mathcal{A})}{2\kappa}, \end{aligned}$$

where we set $\mathbf{x}^* := \arg\min_{\mathbf{x} \in \mathcal{K}} \sum_{t=1}^{T}\langle \mathbf{f}_t, \mathbf{x} \rangle$. □

We turn our attention to the second reduction.

Algorithm 6 Conversion of Online Linear Optimization Alg. $\mathcal{L}$ to Approachability Alg. $\mathcal{A}$

1: Input: Blackwell instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S)$, with S a cone; and a valid halfspace oracle $\mathcal{O}$
2: Input: Online Linear Optimization oracle $\mathcal{L}$
3: Set: $\mathcal{K} = S^0 \cap B_2(1)$
4: **for** $t = 1, \ldots, T$ **do**
5: Query $\mathcal{L}$: $\boldsymbol{\theta}_t \leftarrow \mathcal{L}_{\mathcal{K}}(\mathbf{f}_1, \ldots, \mathbf{f}_{t-1})$, where $\mathbf{f}_s \leftarrow -\mathbf{v}(\mathbf{x}_s, \mathbf{y}_s)$
6: Query $\mathcal{O}$: $\mathbf{x}_t \leftarrow \mathcal{O}(H_{\boldsymbol{\theta}_t})$ where $H_{\boldsymbol{\theta}_t} := \{\mathbf{z} : \langle \boldsymbol{\theta}_t, \mathbf{z} \rangle \le 0\}$
7: Let: $\mathcal{A}(\mathbf{y}_1, \ldots, \mathbf{y}_{t-1}) := \mathbf{x}_t$
8: Receive: $\mathbf{y}_t \in \mathcal{Y}$
9: **end for**

We now prove a similar rate for reverse direction. Here we assume that S is a cone, but we relax this restriction next.

Theorem 52. *The reduction in Algorithm 6, when S is a cone, leads to a rate of approachability of algorithm $\mathcal{A}$ of $D_T(\mathcal{A}; \mathbf{y}_{1:T}) \leq \frac{\text{Regret}(\mathcal{L}_\mathcal{K}; \mathbf{f}_{1:T})}{T}$.*

Proof. We state precisely the halfspace oracle guarantee from line 6. We know that $\mathbf{v}(\mathbf{x}_t, \mathbf{y}) \in H_{\boldsymbol{\theta}_t}$ or equivalently $\langle \boldsymbol{\theta}_t, \mathbf{v}(\mathbf{x}_t, \mathbf{y}) \rangle \leq 0$ for any $\mathbf{y} \in \mathcal{Y}$. In particular, since $\mathbf{v}(\mathbf{x}_t, \mathbf{y}_t) = -\mathbf{f}_t$, we have $\langle \boldsymbol{\theta}_t, \mathbf{f}_t \rangle \geq 0$. We bound $D_T(\mathcal{A})$ by applying Lemma 48 to obtain:

$$\begin{aligned} D_T(\mathcal{A}) = \texttt{dist}\left(\frac{1}{T}\sum_{t=1}^T \mathbf{v}(\mathbf{x}_t, \mathbf{y}_t), S\right) &= \max_{\boldsymbol{\theta} \in \mathcal{K}} \left\langle \frac{1}{T}\sum_{t=1}^T \mathbf{v}(\mathbf{x}_t, \mathbf{y}_t), \boldsymbol{\theta} \right\rangle = \frac{1}{T}\max_{\boldsymbol{\theta} \in \mathcal{K}} \left(-\sum_{t=1}^T \langle \mathbf{f}_t, \boldsymbol{\theta} \rangle \right) \\ &\leq \frac{1}{T}\left(\sum_{t=1}^T \langle \mathbf{f}_t, \boldsymbol{\theta}_t \rangle - \min_{\boldsymbol{\theta} \in \mathcal{K}} \sum_{t=1}^T \langle \mathbf{f}_t, \boldsymbol{\theta} \rangle \right) \qquad (5.9) \\ &= \frac{1}{T}\text{Regret}_T(\mathcal{A}) \qquad (5.10) \end{aligned}$$

where the inequality follows by the halfspace oracle guarantee. □

For a Blackwell instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S)$, even when S is not a cone we can still use Algorithm 6 by *lifting* S: apply Algorithm 6 to the instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}'(\cdot,\cdot), S')$, where $S' := \texttt{cone}(\{\kappa\} \times S)$ and $\mathbf{v}'(\mathbf{x}, \mathbf{y}) := \kappa \oplus \mathbf{v}(\mathbf{x}, \mathbf{y})$.

Corollary 53. *Given a Blackwell instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S)$ with compact S, and let its lifted instance be $(\mathcal{X}, \mathcal{Y}, \mathbf{v}'(\cdot,\cdot), S')$ as described above. Then*

$$\texttt{dist}\left(\frac{1}{T}\sum_{t=1}^T \mathbf{v}(\mathbf{x}_t, \mathbf{y}_t), S\right) \leq 2 \cdot \texttt{dist}\left(\frac{1}{T}\sum_{t=1}^T \mathbf{v}'(\mathbf{x}_t, \mathbf{y}_t), S'\right) \leq \frac{2}{T}\text{Regret}_T(\mathcal{A})$$

Proof. Apply Lemma 49 to Theorem 52. □

We include the compactness assumption only because Lemma 49 requires it yet it is not necessary; the size of S does not enter into the bound. For any Blackwell instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S)$ with non-compact S, we may always consider a functionally equivalent instance $(\mathcal{X}, \mathcal{Y}, \mathbf{v}(\cdot,\cdot), S_0)$, where $S_0 \subset S$ is compact. Letting $U := \{\mathbf{v}(\mathbf{x}, \mathbf{y}) : \mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}\}$, which is compact, we may simply let S_0 be the convex hull of all projections of points in U onto S. Hence $\texttt{dist}(\mathbf{z}, S) = \texttt{dist}(\mathbf{z}, S_0)$ for all $\mathbf{z} \in U$.

5.5 Efficient Calibration via Approachability and OLO

Imagine a sequence of binary outcomes, say 'rain' or 'shine' on a given day, and imagine a forecaster, say the weatherman, that wants to predict the probability of this outcome on each day. A natural question to ask is, on the days when the weatherman actually predicts "30% chance of rain", does it actually rain (roughly) 30% of the time? This exactly the problem of *calibrated forecasting* which we now discuss.

There have been a range of definitions of calibration given throughout the literature, some equivalent and some not, but from a computational viewpoint there are significant differences. We thus give a clean definition of calibration, first introduced by Foster [34], which is convenient to asses computationally.

We let $y_1, y_2, \ldots \in \{0,1\}$ be a sequence of outcomes, and $p_1, p_2, \ldots \in [0,1]$ a sequence of probability predictions by a forecaster. We define for every T and every probability interval $[p-\varepsilon/2, p+\varepsilon/2)$ for $p \in [0,1]$ and $\varepsilon > 0$, the quantities

$$n_T(p,\varepsilon) := \sum_{t=1}^{T} \mathbb{I}[p_t \in [p-\varepsilon/2, p+\varepsilon/2)], \qquad \rho_T(p,\varepsilon) := \frac{\sum_{t=1}^{T} y_t \mathbb{I}[p_t \in [p-\varepsilon/2, p+\varepsilon/2)]}{n_T(p,\varepsilon)}.$$

The quantity $\rho_T(p,\varepsilon)$ should be interpreted as the empirical frequency of $y_t = 1$, up to round T, on only those rounds where the forecaster's prediction was within $\varepsilon/2$ of p. The goal of calibration, of course, is to have this empirical frequency $\rho_T(p,\varepsilon)$ be close to the estimated frequency p in the limit. The standard definition of a calibrated forecaster is one that satisfies

$$\text{for all } p \in [0,1], \varepsilon > 0: \quad \limsup_{T\to\infty} |\rho_T(p,\varepsilon) - p| \le O(\varepsilon) \quad \text{unless} \quad n_T(p,\varepsilon) = o(T). \tag{5.11}$$

Requiring that $n_T(p,\varepsilon)$ does not grow too slowly is an important condition, as we can not expect the forecaster to be calibrated in regions on which he predicts only a small number of times. On the other hand, this case-sensitive condition is somewhat awkward, and we instead use the following equivalent notion.

Definition 54. *Let the (ℓ_1,ε)-calibration rate for forecaster $\mathcal{A}$ be*

$$C_T^{\varepsilon}(\mathcal{A}) = \max\left\{0, \sum_{i=0}^{\lfloor \varepsilon^{-1} \rfloor} \frac{n_T(i\varepsilon,\varepsilon)}{T} |i\varepsilon - \rho_T(i\varepsilon,\varepsilon)| - \frac{\varepsilon}{2}\right\}.$$

We say that a forecaster is (ℓ_1,ε)-calibrated *if $C_T^{\varepsilon}(\mathcal{A}) = o(1)$, or alternatively* $\limsup_{T\to\infty} C_T^{\varepsilon}(\mathcal{A}) \le 0$.

The definition of asymptotic calibration considers the "total error" over an ε-grid, and it adjusts the normalization for each term to $\frac{1}{T}$. The benefit here is that we can ignore intervals in this grid for which $n_T(p,\varepsilon) = o(T)$. In addition, we subtract the constant $\varepsilon/2$ which is an artifact of the discretization by ε; this is the smallest constant which allows for $\limsup_{T\to\infty} C_T^{\varepsilon}(\mathcal{A}) \le 0$. A standard reduction in the literature (see e.g. [24]) shows that a fully-calibrated algorithm (i.e. one satisfying (5.11)) can be constructed from and (ℓ_1,ε)-calibrated algorithm. Henceforth we only consider the (ℓ_1,ε) condition.

As our goal is to minimize the calibration score C_T^{ε}, we can interpret this value instead as a distance to the ℓ_1-norm ball. Define the *calibration vector* $\mathbf{c}_T \in \mathbb{R}^{\lfloor \varepsilon^{-1} \rfloor}$ at time T as: $\mathbf{c}_T(i) = \frac{n_T(i\varepsilon,\varepsilon)}{T}(i\varepsilon - \rho_T(i\varepsilon,\varepsilon))$.

Claim 1. *Whenever $\mathbf{c}_T \notin B_1(\varepsilon/2)$, we have*

$$C_T^{\varepsilon} = \mathtt{dist}_1(\mathbf{c}_T, B_1(\varepsilon/2)).$$

Proof. Notice that for any $\mathbf{x}$: $\texttt{dist}_1(\mathbf{x}, B_1(\varepsilon/2)) := \min_{\mathbf{y}:\|\mathbf{y}\|_1 \leq \varepsilon/2} \|\mathbf{x} - \mathbf{y}\|_1 = \max\{0, -\varepsilon/2 + \|\mathbf{x}\|_1\}$. The second equality follows by noting that an optimally chosen $\mathbf{y}$ will lie in the same quadrant as $\mathbf{x}$. When we set $\mathbf{x} = \mathbf{c}_T$, it is clear that $\|\mathbf{c}_T\|_1 > \varepsilon/2$ given our assumption that $\mathbf{c}_T \notin B_1(\varepsilon/2)$. □

The utility of this claim shall be to convert the problem of (ℓ_1, ε)-calibration to a problem of approachability; that is, can we approach the set $B_1(\varepsilon/2)$ for a particular vector-valued game? In the following section we describe this construction in detail.

5.5.1 Existence of Calibrated Forecaster via Blackwell Approachability

A surprising fact is that it is possible to achieve calibration even when the outcome sequence $\{y_t\}$ is chosen by an adversary, although this requires a randomized strategy of the forecaster. Algorithms for calibrated forecasting under adversarial conditions have been given in Foster and Vohra [35], Fudenberg and Levine [39], and Hart and Mas-Colell [44].

Interestingly, the calibration problem was reduced to Blackwell's Approachability Theorem in a short paper by Foster in 1999 [34]. Foster's reduction uses Blackwell's original theorem, proving that a given set is halfspace-satisfiable, in particular by providing a construction for each such halfspace. Here we provide a reduction to Blackwell Approachability using the response-satisfiability condition – that is by using Theorem 41 – which is both significantly easier and more intuitive than Foster's construction[2]. We also show, using the reduction to Online Linear Optimization from the previous section, how to achieve the most efficient known algorithm for calibration by taking advantage of the Online Gradient Descent algorithm of Zinkevich [77], using the results of Section 5.4.

We now describe the construction that allows us to reduce calibration to approachability. For any $\varepsilon > 0$ we will show how to construct an (ℓ_1, ε)-calibrated forecaster. Notice that from here, it is straightforward to produce a well-calibrated forecaster [35]. For simplicity, assume $\varepsilon = 1/m$ for some positive integer m. On each round t, a forecaster will now randomly predict a probability $p_t \in \{0/m, 1/m, 2/m, \ldots, (m-1)/m, 1\}$, according to the distribution $\mathbf{w}_t$, that is $\Pr(p_t = i/m) = \mathbf{w}_t(i)$. We now define a vector-valued game. Let the player choose $\mathbf{w}_t \in \mathcal{X} := \Delta_{m+1}$, and the adversary choose $y_t \in \mathcal{Y} := [0, 1]$, and the payoff vector will be

$$\mathbf{v}(\mathbf{w}_t, y_t) := \left\langle \mathbf{w}_t(0)\left(y_t - \frac{0}{m}\right), \mathbf{w}_t(1)\left(y_t - \frac{1}{m}\right), \ldots, \mathbf{w}_t(m)(y_t - 1)\right\rangle \tag{5.12}$$

Lemma 55. *Consider the vector-valued game described above and let $S := B_1(\varepsilon/2)$. If we have a strategy for choosing $\mathbf{w}_t$ that guarantees approachability of S, that is $\frac{1}{T}\sum_{t=1}^{T} \mathbf{v}(\mathbf{w}_t, y_t) \to S$, then a randomized forecaster that selects p_t according to $\mathbf{w}_t$ is (ℓ_1, ε)-calibrated with high probability.*

[2] A similar existence proof was discovered concurrently by Mannor and Stoltz [55]

The proof of this lemma is straightforward, and is similar to the construction in Foster [34]. The key fact is that $\frac{1}{T}\sum_{t=1}^{T}\mathbf{v}(\mathbf{w}_t, y_t) = \mathbb{E}[\mathbf{c}_T]$, where the expectation is taken over the algorithms draws of every p_t according to the distribution $\mathbf{w}_t$. Since each p_t is drawn independently, by standard concentration arguments we can see that if $\frac{1}{T}\sum_{t=1}^{T}\mathbf{v}(\mathbf{w}_t, y_t)$ is close to the $\ell_1$ball of radius $\varepsilon/2$, then the (ℓ_1, ε)-calibration vector is close to the $\varepsilon/2$ ball with high probability.

We can now apply Theorem 41 to prove the existence of a calibrated forecaster.

Theorem 56. *For the vector-valued game defined in* (5.12)*, the set* $B_1(\varepsilon/2)$ *is response-satisfiable and, hence, approachable.*

Proof. To show response-satisfiability, we need only show that, for every strategy $y \in [0, 1]$ played by the adversary, there is a strategy $\mathbf{w} \in \Delta_m$ for which $\mathbf{v}(\mathbf{w}, y) \in S$. This can be achieved by simply setting i so as to minimize $|i\varepsilon - y|$, which can always be made smaller than $\varepsilon/2$. We then choose our distribution $\mathbf{w} \in \Delta_{m+1}$ to be a point mass on i, that is we set $w(i) = 1$ and $w(j) = 0$ for all $j \neq i$. Then $\mathbf{v}(\mathbf{w}, y)$ is identically 0 everywhere except the ith coordinate, which has the value $y - i/m$. By construction, $y - i/m \in [-1/m, 1/m]$, and we are done. □

5.5.2 Efficient Algorithm for Calibration via Online Linear Optimization

We now show how the results in the previous Section lead to the first efficient algorithm for calibrated forecasting. The previous theorem provides a natural existence proof for Calibration, but it does not immediately provide us with a simple and efficient algorithm. We proceed according to the reduction outlined in the previous section to prove:

Theorem 57. *There exists a* (ℓ_1, ε)*-calibration algorithm that runs in time* $O(\log \frac{1}{\varepsilon})$ *per iteration and satisfies* $C_T^{\varepsilon} = O\left(\frac{1}{\sqrt{\varepsilon T}}\right)$

The reduction developed in Theorem 52 has some flexibility, and we shall modify it for the purposes of this problem. The objects we shall need, as well as the required conditions, are as follows:

1. A convex set $\mathcal{K}$
2. An efficient learning algorithm $\mathcal{A}$ which, for any sequence $\mathbf{f}_1, \mathbf{f}_2, \ldots$, can select a sequence of points $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots \in \mathcal{K}$ with the guarantee that $\sum_{t=1}^{T}\langle \mathbf{f}_t, \boldsymbol{\theta}_t\rangle - \min_{\boldsymbol{\theta}\in\mathcal{K}}\sum_{t=1}^{T}\langle \mathbf{f}_t, \boldsymbol{\theta}\rangle = o(T)$. For the reduction, we shall set $\mathbf{f}_t \leftarrow -\mathbf{v}(\mathbf{w}_t, y_t)$.
3. An efficient oracle that can select a particular $\mathbf{w}_t \in \mathcal{X}$ for each $\boldsymbol{\theta}_t \in \mathcal{K}$ with the guarantee that

$$\texttt{dist}\left(\frac{1}{T}\sum_{t=1}^{T}\mathbf{v}(\mathbf{w}_t, y_t), S\right) \leq \frac{1}{T}\left(\sum_{t=1}^{T}\langle -\mathbf{v}(\mathbf{w}_t, y_t), \boldsymbol{\theta}_t\rangle - \min_{\boldsymbol{\theta}\in\mathcal{K}}\sum_{t=1}^{T}\langle -\mathbf{v}(\mathbf{w}_t, y_t), \boldsymbol{\theta}\rangle\right) \tag{5.13}$$

where the function $\texttt{dist}()$ can be with respect to any norm.

The Setup Let $\mathcal{K} = B_\infty(1) = \{\boldsymbol{\theta} \in \mathbb{R}^d : \|\boldsymbol{\theta}\|_\infty \leq 1\}$ be the unit cube. This is an appropriate choice because we can write $\texttt{dist}_1(\mathbf{x}, B_1(\varepsilon/2))$ for $\mathbf{x} \notin B_1(\varepsilon/2))$ as

$$\texttt{dist}_1(\mathbf{x}, B_1(\varepsilon/2)) := \min_{\mathbf{y}:\|\mathbf{y}\|_1 \leq \varepsilon/2} \|\mathbf{x} - \mathbf{y}\|_1 = -\varepsilon/2 + \|\mathbf{x}\|_1 = -\varepsilon/2 - \min_{\boldsymbol{\theta}:\|\boldsymbol{\theta}\|_\infty \leq 1} \langle -\mathbf{x}, \boldsymbol{\theta} \rangle; \quad (5.14)$$

The former equality was proved in Claim 1. Furthermore, we shall construct our oracle mapping $\boldsymbol{\theta} \mapsto \mathbf{w}$ with the following guarantee: $\langle \mathbf{v}(\mathbf{w}, y), \boldsymbol{\theta} \rangle \leq \varepsilon/2$ for any y. Using this guarantee, and if we plug in $\mathbf{x} = \frac{1}{T}\sum_{t=1}^T \mathbf{v}(\mathbf{w}_t, y_t)$ (5.14), we arrive at:

$$\begin{aligned} \texttt{dist}_1\left(\frac{\sum_{t=1}^T \mathbf{v}(\mathbf{w}_t, y_t)}{T}, B_1(\varepsilon/2)\right) &= -\varepsilon/2 - \min_{\boldsymbol{\theta}:\|\boldsymbol{\theta}\|_\infty \leq 1} \left\langle \frac{-\sum_{t=1}^T \mathbf{v}(\mathbf{w}_t, y_t)}{T}, \boldsymbol{\theta} \right\rangle \\ &\leq \frac{1}{T}\left(\sum_{t=1}^T \langle -\mathbf{v}(\mathbf{w}_t, y_t), \boldsymbol{\theta}_t \rangle - \min_{\boldsymbol{\theta} \in \mathcal{K}} \sum_{t=1}^T \langle -\mathbf{v}(\mathbf{w}_t, y_t), \boldsymbol{\theta} \rangle\right) \end{aligned}$$

This is precisely the necessary guarantee (5.13).

Constructing the Oracle We now turn our attention to designing the required oracle in an *efficient* manner. In particular, given any $\boldsymbol{\theta}$ with $\|\boldsymbol{\theta}\|_\infty \leq 1$ we must construct $\mathbf{w} \in \Delta_{m+1}$ so that $\langle \ell(\mathbf{w}, y), \boldsymbol{\theta} \rangle \leq \varepsilon/2$ for any y. The details of this oracle are given in Algorithm 7. It

Algorithm 7 Efficient Oracle mapping $\mathcal{O} : \mathbf{w} \mapsto \boldsymbol{\theta}$

Input: $\boldsymbol{\theta}$ such that $\|\boldsymbol{\theta}\|_\infty \leq 1$
if $\boldsymbol{\theta}(0) \leq 0$ **then**
  $\mathbf{w} \leftarrow \delta_0$ // That is, choose $\mathbf{w}$ to place all weight on the 0th coordinate
else if $\theta(m) \geq 0$ **then**
  $\mathbf{w} \leftarrow \delta_m$ // That is, choose $\mathbf{w}$ to place all weight on the last coordinate
else
  Binary search $\boldsymbol{\theta}$ to find coordinate i such that $\boldsymbol{\theta}(i) > 0$ and $\boldsymbol{\theta}(i+1) \leq 0$
  $\mathbf{w} \leftarrow \frac{\boldsymbol{\theta}(i)^{-1}}{\boldsymbol{\theta}(i)^{-1} - \boldsymbol{\theta}(i+1)^{-1}}\delta_i + \frac{-\boldsymbol{\theta}(i+1)^{-1}}{\boldsymbol{\theta}(i)^{-1} - \boldsymbol{\theta}(i+1)^{-1}}\delta_{i+1}$
end if
Return $\mathbf{w}$

is straightforward why, in the final **else** condition, there must be such a pair of coordinates $i, i+1$ satisfying the condition. We need not be concerned with the case that $\boldsymbol{\theta}(i+1) = 0$, where we can simply define $\frac{0}{\infty} = 0$ and $\frac{\infty}{\infty} = 1$ leading to $\mathbf{w} \leftarrow \delta_{i+1}$. It is also clear that, with the binary search, this algorithm requires at most $O(\log m) = O(\log 1/\varepsilon)$ computation.

In order to prove that this construction is valid we need to check the condition that, for any $y \in \{0, 1\}$, $\langle \mathbf{v}(\mathbf{w}, y), \boldsymbol{\theta} \rangle \leq \varepsilon/2$; or more precisely, $\sum_{i=1}^m \boldsymbol{\theta}(i)\mathbf{w}(i)\left(y - \frac{i}{m}\right) \leq \varepsilon/2$.

Recalling that $m = 1/\varepsilon$, this is trivially checked for the case when $\boldsymbol{\theta}(1) \leq 0$ or $\boldsymbol{\theta}(m) \geq 0$. Otherwise, we have

$$\begin{aligned}\langle \mathbf{v}(\mathbf{w}, y), \boldsymbol{\theta} \rangle &= \boldsymbol{\theta}(i) \frac{\boldsymbol{\theta}(i)^{-1}}{\boldsymbol{\theta}(i)^{-1} - \boldsymbol{\theta}(i+1)^{-1}} \left(y - \frac{i}{m} \right) + \boldsymbol{\theta}(i+1) \frac{-\boldsymbol{\theta}(i+1)^{-1}}{\boldsymbol{\theta}(i)^{-1} - \boldsymbol{\theta}(i+1)^{-1}} \left(y - \frac{i+1}{m} \right) \\ &= \frac{1}{\boldsymbol{\theta}(i)^{-1} - \boldsymbol{\theta}(i+1)^{-1}} \frac{1}{m} \leq \frac{\max(|\boldsymbol{\theta}(i)|, |\boldsymbol{\theta}(i+1)|)}{2} \varepsilon \leq \frac{\varepsilon}{2}\end{aligned}$$

The Learning Algorithm The final piece is to construct an efficient learning algorithm which leads to vanishing regret. That is, we need to construct a sequence of $\boldsymbol{\theta}_t$'s in the unit cube (denoted $B_\infty(1)$) so that

$$\sum_{t=1}^{T} \langle \mathbf{v}_t, \boldsymbol{\theta}_t \rangle - \min_{\boldsymbol{\theta} \in B_\infty(1)} \sum_{t=1}^{T} \langle \mathbf{v}_t, \boldsymbol{\theta} \rangle = o(T),$$

where $\mathbf{v}_t := \mathbf{v}(\mathbf{w}_t, y_t)$. There are a range of possible no-regret algorithms available, but we use the one given by Zinkevich known commonly as Online Gradient Descent [77]. The details are given in Algorithm 8. This algorithm can indeed be implemented efficiently, requiring

Algorithm 8 Online Gradient Descent

Input: convex set $\mathcal{K} \subset \mathbb{R}^d$
Initialize: $\boldsymbol{\theta}_1 = \mathbf{0}$
Set Parameter: $\eta = O(T^{-1/2})$
for $t = 1, \ldots, T$ **do**
  Receive $\mathbf{v}_t$
  $\boldsymbol{\theta}'_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta \mathbf{v}_t$ // Gradient Descent Step
  $\boldsymbol{\theta}_{t+1} \leftarrow \text{Project}_2(\boldsymbol{\theta}'_{t+1}, \mathcal{K})$ // L2 Projection Step
end for

only $O(1)$ computation on each round and $O(\min\{m, T\})$ memory. The main advantage is that the vectors $\mathbf{v}_t$ are generated via our oracle above, and these vectors are *sparse*, having only at most two nonzero coordinates. Hence, the Gradient Descent Step requires only $O(1)$ computation. In addition, the Projection Step can also be performed in an efficient manner. Since we assume that $\boldsymbol{\theta}_t \in B_\infty(1)$, the updated point $\boldsymbol{\theta}'_{t+1}$ can violate at most two of the ℓ_∞ constraints of the ball $B_\infty(1)$. An ℓ_2 projection onto the cube requires simply rounding the violated coordinates into $[-1, 1]$. The number of non-zero elements in $\boldsymbol{\theta}$ can increase by at most two every iteration, and storing $\boldsymbol{\theta}$ is the only state that online gradient descent needs to store, hence the algorithm can be implemented with $O(\min\{T, m\})$ memory. We thus arrive at an efficient no-regret algorithm for choosing $\boldsymbol{\theta}_t$.

Putting it all Together We can now fully specify our calibration algorithm given the subroutines defined above. The precise description is in Algorithm 9, which makes queries to Algorithms 7 and 8.

Algorithm 9 Efficient Algorithm for Asymptotic Calibration

Input: $\varepsilon = 1/m$ for some natural number m
Initialize: $\boldsymbol{\theta}_1 = \mathbf{0}$, $\mathbf{w}_1 \in \Delta_{m+1}$ arbitrarily
for $t = 1, \ldots, T$ **do**
  Sample $i_t \sim \mathbf{w}_t$, predict $p_t = \frac{i_t}{m}$, observe $y_t \in \{0, 1\}$
  Set $\mathbf{v}_t := \mathbf{v}(\mathbf{w}_t, y_t)$ // Vector-valued game defined in (5.12)

  Query learning algorithm: $\boldsymbol{\theta}_{t+1} \leftarrow \text{Update}(\boldsymbol{\theta}_t | \mathbf{v}_t)$ // Subroutine from Algorithm 8
  Query halfspace oracle: $\mathbf{w}_{t+1} \leftarrow \mathcal{O}(\boldsymbol{\theta}_{t+1})$ // Subroutine from Algorithm 7
end for

of Theorem 57. Here we have bounded the distance directly by the regret, using equation (5.13), which tells us that the calibration rate is bounded by the regret of the online learning algorithm. Online Gradient Descent guarantees the regret to be no more than $DG\sqrt{T}$, where D is the ℓ_2 diameter of the set, and G is the ℓ_2-norm of the largest cost vector. For the ball $B_\infty(1)$, the diameter $D = \sqrt{\frac{1}{\varepsilon}}$, and we can bound the norm of our loss vectors by $G = \sqrt{2}$. Hence:

$$C_T^\varepsilon \quad = \quad \texttt{dist}(c_T, B_1(\varepsilon/2)) \quad \leq \quad \frac{\text{Regret}_T}{T} \quad \leq \quad \frac{GD}{\sqrt{T}} \quad = \quad O\left(\frac{1}{\sqrt{\varepsilon T}}\right) \tag{5.15}$$

□

Chapter 6

Gambler versus Casino

6.1 Introduction

This chapter analyzes the problem of sequential prediction and decision making from the perspective of a two player game. The game is played by a learner, called here the Gambler, who makes a sequence of betting decisions. The Gambler's opponent is the Casino in which he plays.

Gambler vs. Casino:

1. On each day, the Gambler arrives at the Casino with \$1. The Casino presents n *events* and each event is played once per day. The Gambler chooses a distribution vector $\mathbf{w} \in [0,1]^n$, where $\sum w_i = 1$, and bets the portion w_i of his \$1 budget on event i.

2. On each day the Casino determines the outcome of each event with the objective of winning as much money from the Gambler as possible. In particular, after observing the distribution of the Gambler's bets the Casino decides between a *loss* or a *no loss* for all daily events. These choices are summarized by a *loss vector* $\boldsymbol{\ell} \in \{1,0\}^n$ where $\ell_i = 1$ implies that on event i, the Gambler lost. (For simplicity, we assume the only relevant quantities are losses. By shifting our baseline we can model *wins* as *non-losses*).

3. At the end of each day, the Gambler leaves the Casino having lost $\mathbf{w} \cdot \boldsymbol{\ell} = \sum_i w_i \ell_i$ and the cumulative loss of the gambler is updated as $L \leftarrow L + \mathbf{w} \cdot \boldsymbol{\ell}$. The Gambler also monitors the cumulative performance of each event with a state vector $\mathbf{s} \in \mathbb{N}^n$, where s_i is the current total loss of event i. After incurring loss $\boldsymbol{\ell}$ at the current day, the state vector is updated to $\mathbf{s} \leftarrow \mathbf{s} + \boldsymbol{\ell}$.

4. The Gambler stops playing as soon as he observes that each even has suffered more than k losses, where k is some fixed positive integer known to both. The Casino is aware of this decision and behaves accordingly.

Gambling against a casino may seem an unlikely starting point for a model of sequential decision making – we generally consider the typical environment for learning to be *stochastic*

rather than *adversarial*. Yet are these two environments necessarily incompatible? Among the objectives of this chapter is to address questions such as: "What will be the Gambler's worst-case cumulative loss?"; and "What is the optimal betting strategy?" These questions, while clearly game-theoretic, are ultimately answered here by considering a *randomized* Casino rather than an adversarial one. From this perspective, randomness may indeed be the Gambler's worst adversary.

Early work on sequential decision making focused on the problem of predicting a binary outcome given advice from a set of n *experts*. In that setting, the goal of the predictor is to combine the predictions of the experts to make his own prediction, with the objective of performing well, in hindsight, compared to the best expert. The performance of both the learner and the experts is measured by a loss function that compares predictions to outcomes. One of the early algorithms, the Weighted Majority algorithm [53], utilizes a distribution corresponding to the degree of *trust* in each expert.

It was observed by Freund and Schapire [38] that the analysis of the Weighted Majority algorithm can be applied to the so-called *hedge setting*. Rather than predict a binary outcome, the learner now plays some distribution over the experts on every round, a loss value is assigned to each expert independently, and the learner suffers the expected loss according to his chosen distribution. In this case, the learner bears the exact burden of the Gambler - that of "hedging" his bets so as to minimize his cumulative loss. To emphasize that the Gambler/Casino game is useful for settings other than prediction, we use the term "event" rather than "expert".

A central theme of much of the sequential decision making literature is the use of so-called "exponential weights" to determine the learner's distribution on each round. Use of the exponential weighting scheme in the case of the Casino game results in the following strategy for the Gambler: At a state $\mathbf{s}$, bet

$$w_i = \frac{\beta^{s_i}}{\sum_j \beta^{s_j}} \quad \text{on event } i, \tag{6.1}$$

where the factor β lies in $[0, 1)$.

From the analysis of the Weighted Majority algorithm it follows that the cumulative loss of the Gambler using the above strategy is bounded by

$$\frac{\ln n + k \ln \frac{1}{\beta}}{1 - \beta}.$$

Under the assumption that the loss of the best event is at most k, the factor β can be tuned [38] so that the above bound becomes

$$k + \sqrt{2k \ln n} + \ln n.$$

The exponential weights framework, as well as other online learning techniques, can be motivated using the method of relative entropy regularization [51]. While the resulting algorithms are elegant and in some cases can be shown to be asymptotically optimal [22], they

do not optimally solve the underlying game. Some improvements have been made using, for example, binomial weights that lead to slightly better but still non-optimal solutions [22] in a setting where the experts must produce a prediction. While it is formally easy to define the optimal algorithms using minimax expressions, it has generally been assumed that actually computing an efficient solution is quite challenging [23]. More recently, however, a minimax result [4] was obtained for the specific game of prediction with absolute loss. The resulting algorithm, Binning, is efficient and optimal in a slightly relaxed setting.

In this chapter we show that the minimax solution to the Gambler/Casino prediction game, which is identical to the underlying game of the hedge setting with binary losses, can be obtained efficiently. In addition, the game can be fully analyzed using a simple Markov process: a random walk on an n-dimensional lattice. The value of the game, that is the cumulative loss of an optimal Gambler, can be interpreted as the expected length of such a random walk. The Gambler's optimal play, the portion of his budget he should bet on a given event, can similarly be interpreted as manifesting an assessment of the probability of a specific random outcome of this walk.

The game's stopping criterion, that is when all events have lost at least $k+1$ times, may seem unusual at first yet fits quite naturally within the experts framework. Indeed, online learning bounds are often tuned with an explicit a priori knowledge of the cumulative loss of the best expert, which here would be k[1]. While perhaps not realistic in practice, k can be estimated and various techniques such as successive doubling can be used to obtain near-optimal bounds [23].

The chapter is structured as follows. In Section 6.2 we give a minimax definition of the optimal value of the game considered here. In Section 6.2.1 we modify the game by restricting the adversary's choices to unit loss vectors. In Section 6.3, we then turn our attention to a specific Markov process with a number of relevant properties. We apply this randomized approach to the Casino game in Section 6.4, where we prove our main results. In Section 6.5 we give recurrences and exact formulas, based on sums over multinomials, for the value of the game and for the optimal probabilities. We set out an efficient method to compute both the optimal strategy of the Gambler and the value of the game. In Section 6.6 we compare the optimal regret bound to previous results, and in Section 6.7 we draw a connection between our game and a well studied version of the coupon collector problem. We also briefly summarize what is known about the asymptotics of this problem. We conclude with a discussion of our results and list open problems (Section 6.8).

6.2 The Value of the Game

Assume that in each event the Gambler has already suffered some losses specified by state vector $\mathbf{s}$. Define $V(\mathbf{s})$ to be the total money lost by an optimal Gambler playing against an

[1]Strictly speaking, in the expert setting it is assumed that at least one expert has not crossed the k-mistake threshold, while here we stop the Gambler/Casino game when the loss of the *last* expert/event goes beyond this threshold. It is easy to show that this slight modification, made for convenience, increases the worst-case loss of the Gambler by exactly 1.

optimal Casino *starting from the state* $\mathbf{s}$. That is, $V(\mathbf{s})$ is the amount of money that the optimal Gambler will lose (against an optimal Casino) from now until the end of the game. Roughly speaking, the value of the game is computed as:

$$V(\mathbf{s}) \stackrel{?}{=} \min_{\text{dist. } \mathbf{w}} \max_{\boldsymbol{\ell} \in \{0,1\}^n} \mathbf{w} \cdot \boldsymbol{\ell} + V(\mathbf{s} + \boldsymbol{\ell})$$

The Gambler chooses $\mathbf{w}$ to minimize the loss while the Casino chooses $\boldsymbol{\ell}$ to maximize the loss, where the loss is computed as the loss $\mathbf{w} \cdot \boldsymbol{\ell}$ on this round plus the worst case loss $V(\mathbf{s}+\boldsymbol{\ell})$ on future rounds. However, we have to be careful, as this recursive definition doesn't address the following issues:

- When is the game over? What is the base case of $V(\cdot)$?
- Is this recursion bounded?
- Do we need to record the losses $\mathbf{s}_i$ that go above k?

We address these issues beginning with some simplifications and notational conventions. First, we assume that the state vector $\mathbf{s}$ lies within the set $\mathcal{S} = \{0, 1, \ldots, k+1\}^n$. Note that it is not necessary to record the losses of events that have already crossed the k threshold. We call such events *dead*. Since the losses of dead events are not restricted, having loss $k + 3$ is the same as loss $k + 100$. We therefore "round" all states $\mathbf{s}$ into the state space $\{0, 1, \ldots, k+1\}^n$ using the notation $\dot{+}$ which we define below. We use the notation $\lambda(\mathbf{s})$ to record the set of live events; the statement $i \notin \lambda(\mathbf{s})$ is exactly the statement $s_i = k + 1$.

Second, as the game is defined recursively, we must guarantee that this recursion terminates. If the Casino repeatedly chose $\boldsymbol{\ell} = \langle 0, \ldots, 0 \rangle$, for example, the game would make no progress. The same problem occurs if the Casino causes losses on only dead events. We must therefore place additional restrictions so that the dead state is reached eventually. The simplest way to ensure this is to forbid the Casino from inflicting loss on only dead events. Yet this is not sufficient: with this restriction alone the Gambler would have a guaranteed non-losing strategy by betting solely on dead events. We thus assume that neither can the Casino can inflict losses on dead events nor can the Gambler bet money on them (keeping in mind that all such bets are in any case non-optimal). We must enforce this explicitly in order to have a well-defined game.

We use two notational conventions to describe the above restrictions. First, we write $\mathbf{w} \sim \lambda(\mathbf{s})$ to describe the set $\{\mathbf{w} \in \Delta_n \mid w_i = 0 \ \forall i \notin \lambda(\mathbf{s})\}$ where Δ_n is the n-simplex. We also abuse notation slightly and write $\boldsymbol{\ell} \subset \lambda(\mathbf{s})$ to mean that $\boldsymbol{\ell} \in \{0, 1\}^n$ and $\ell_i = 0$ for all $i \notin \lambda(\mathbf{s})$.

We now define the value of the game precisely.

Definition 58. *Define the value $V(\mathbf{s})$ of the game as follows.*

- *At the dead state, $V(\mathbf{d}) := 0$.*

- *For any other* $\mathbf{s} \in \mathcal{S}$*, we define* $V(\mathbf{s})$ *recursively as*

$$V(\mathbf{s}) := \min_{\mathbf{w} \sim \lambda(\mathbf{s})} \max_{\mathbf{0} \neq \boldsymbol{\ell} \subset \lambda(\mathbf{s})} \mathbf{w} \cdot \boldsymbol{\ell} + V(\mathbf{s} + \boldsymbol{\ell}). \tag{6.2}$$

In our notation, we commonly make use of several special states. The state where the game begins is the "initial" state, $\mathbf{s} = \mathbf{0}$. Once all events have lost more than k times the game is over and we refer to this as the *dead* state $\mathbf{d}$. It will also be useful to consider *one-live* states $\mathbf{o}_i$, where all events except i are dead, and the remaining event has exactly k losses. By the game definition, it is easy to check that $V(\mathbf{o}_i) = 1$, since the Gambler must bet all of his money on this event, and the Casino must inflict a corresponding loss, charging the Gambler \$1 and ending the game.

Below, we include a list of notations for reference:

Notation:

$\mathcal{S} := \{0, \ldots, k+1\}^n$	(the state space)
$\mathbf{0} := \langle 0, 0, \ldots, 0 \rangle = \langle 0^n \rangle$	(the initial state)
$\mathbf{d} := \langle (k+1)^n \rangle$	(the dead state)
$\mathbf{o}_i := \mathbf{d} - \mathbf{e}_i$	(ith one-live state)
$\lambda(\mathbf{s}) := \{i \in [n] : s_i \leq k\}$	(set of live events)
$\mathbf{s} \dot{+} \boldsymbol{\ell} := \langle \min(s_i + \ell_i, k+1) \rangle$	("rounded" addition)
$\lvert\mathbf{s}\rvert := \sum s_i$	(elementwise sum)
$\Delta_n := \{\mathbf{w} \in \mathbf{R}^n_+ : \lvert\mathbf{w}\rvert = 1\}$	(the n-simplex)

6.2.1 The Modified Game

We also consider a modified game that we make easier for the Gambler. In this new game, we restrict the Casino to inflict loss on exactly *one* event in each round, i.e. $\boldsymbol{\ell}$ must be a basis vector $\mathbf{e}_1, \ldots, \mathbf{e}_n$. So for $\boldsymbol{\ell} = \mathbf{e}_i$ we have $\mathbf{w} \cdot \boldsymbol{\ell} = w_i$. We can then precisely define the value $\widehat{V}(\cdot)$ of the modified game:

Definition 59. *Define* $\widehat{V}(\mathbf{d}) := V(\mathbf{d}) = 0$*. Otherwise*

$$\widehat{V}(\mathbf{s}) := \min_{\mathbf{w} \sim \lambda(\mathbf{s})} \max_{i \in \lambda(\mathbf{s})} w_i + \widehat{V}(\mathbf{s} + \mathbf{e}_i). \tag{6.3}$$

One of the central results of this chapter is that the above game, while seemingly more restricted, is ultimately just as difficult for the Gambler as the original game. It is easy to show that $V(\mathbf{s}) \geq \widehat{V}(\mathbf{s})$, since the Casino has strictly more choices in the original game. We go further and prove as our main result in Theorem 69 that

$$V(\mathbf{s}) = \widehat{V}(\mathbf{s}).$$

Thus both games have the same worst-case outcome.

Both the analysis of the modified game, as well as the proof of the above result, requires a different formulation of the Casino's actions.

6.3 A Randomized Casino

In Section 6.2 we presented a game-theoretic analysis of a well-known sequential prediction problem characterized as a game between a Gambler and a Casino. In the present section, we consider a different framework, in which the Casino uses random events. We will show that introducing a randomized strategy of the Casino enables us to specify the optimal strategy of the Gambler.

6.3.1 A Random Walk on the State Graph

Let us now imagine that our Casino does not fix outcomes deterministically, but instead chooses the outcome of each event using the following random process. Assume we are at state $\mathbf{s}$ and that, on each day, an event i is chosen uniformly at random from $\{1, \ldots, n\}$ and a loss is assigned to event i. In other words, the loss vector $\boldsymbol{\ell}$ is a uniformly sampled unit vector $\mathbf{e}_i$, and after the loss the new state is $\mathbf{s} \dot{+} \mathbf{e}_i$. This process continues until we reach the dead state $\mathbf{d}$.

We can model this behavior as a Markov process on the state space as follows. Consider any sequence of indices $I_1, I_2, \ldots \in [n]$, and let $S_t := \sum_{m=1}^{t} \mathbf{e}_{I_m}$, where $S_0 := \mathbf{0}$. Assuming that we start at state $\mathbf{s}$, this induces a sequence of states

$$\mathbf{s} = \mathbf{s} \dot{+} S_0 \to \mathbf{s} \dot{+} S_1 \to \mathbf{s} \dot{+} S_2 \to \ldots \to \mathbf{s} \dot{+} S_t.$$

Notice that this process has "self-loops"; i.e. it is quite possible that $\mathbf{s} \dot{+} S_t = \mathbf{s} \dot{+} S_{t+1}$. This occurs when $(\mathbf{s} \dot{+} S_t)_{I_{t+1}}$ is already at $k+1$.

If we imagine the state space $\mathcal{S}$ as an n-dimensional lattice, which we will call the *state lattice*, then the Markov process above can be interpreted as a random walk on this lattice. The walker starts at the initial state $\mathbf{0}$, and on every time interval a positively directed single step is taken along an axis drawn uniformly at random. If the walker has already reached the $k+1$ boundary in this dimension, he remains in place. The walk stops once the dead state $\mathbf{d}$ is reached. We will show that the value V is $1/n$ times the expected total number of random draws that achieves this position. Thus V is the expected walk/path length from $\mathbf{s}$ to $\mathbf{d}$.

6.3.2 Survival Probabilities

We now define a *survival probability* at a state $\mathbf{s}$. We will show in the next section that such probabilities are the basis for the Gambler's optimal strategy.

Definition 60. *Assume we are at state* $\mathbf{s}$*, and let the random state* $\mathbf{s}\dot{+}S_t$ *be the result of the above random walk after* t *steps. Define the* i*th survival probability* $\widehat{p}_i(\mathbf{s})$ *to be the probability that*

$$\exists t: \ \mathbf{s}\dot{+}S_t = \mathbf{o}_i.$$

Equivalently,

$$\widehat{p}_i(\mathbf{s}) \ = \ Pr(\lambda(\mathbf{s}\dot{+}S_t) = \{i\} \text{ for some } t).$$

We call these survival probabilities since $\widehat{p}_i(\mathbf{s})$ is the probability that, if the losses were assigned randomly to the events in sequence, the ith event would be the last non-dead event.

Lemma 61. *For any* $\mathbf{s} \neq \mathbf{d}$*, the vector*

$$\widehat{p}(\mathbf{s}) := \langle \widehat{p}_i(\mathbf{s}) \rangle_{i=1}^{n}$$

defines a distribution on $\{1, \ldots, n\}$*.*

Proof. The quantity $\sum_i \widehat{p}_i(\mathbf{s})$ is the probability that eventually there is exactly one live event. This probability is exactly 1, given that the current state is not the dead state $\mathbf{d}$. □

We list some examples of survival probabilities:

- When $\mathbf{s} = \mathbf{0}$ (or any other symmetric state), we have

  $$\widehat{p}_i(\mathbf{s}) = \frac{1}{n}, \ \forall i$$

  bexause there is a uniform chance of survival.

- When i is a dead event, i.e. $s_i = k + 1$, then

  $$\widehat{p}_i(\mathbf{s}) = 0$$

  because no dead event can be the last remaining live event.

- If there is only one remaining live event, i.e. $\lambda(\mathbf{s}) = \{i\}$, then

  $$\widehat{p}_i(\mathbf{s}) = 1.$$

Computing $\widehat{p}_i(\mathbf{s})$ for more general $\mathbf{s}$ requires a recursion, and we leave this discussion for Section 6.5.

6.3.3 Expected Path Lengths

Another important quantity we consider is the *length of a random path*, i.e. the number of steps in the random walk on the state lattice required until the dead state $\mathbf{d}$ is reached.

Definition 62. *For a sequence $S_0, S_1, \ldots$, let*

$$T(\mathbf{s}) := \min\{t \geq 0 : \mathbf{s}\dot{+}S_t = \mathbf{d}\}.$$

That is, $T(\mathbf{s})$ is the length *of the random path starting at $\mathbf{s}$ and just entering $\mathbf{d}$. Furthermore, let*

$$\tau(\mathbf{s}) := \mathbb{E}\ T(\mathbf{s})$$

be the expected path length.

We note that paths may be infinitely long due to self-loops, yet such paths occur with probability 0. A key fact is that the expected path length $\tau(\mathbf{s})$ can be rewritten using indicator variables:

$$T(\mathbf{s}) = \sum_{t=0}^{\infty} \mathbf{1}[\mathbf{s}\dot{+}S_t \neq \mathbf{d}], \tag{6.4}$$

i.e. $T(\mathbf{s})$ is the number of initial segments (including the empty segment) of a random path starting at $\mathbf{s}$ that has not reached the dead state $\mathbf{d}$.

We now prove a relationship between expected path length $\tau(\mathbf{s})$ and survival probabilities $\widehat{p}_i(\mathbf{s})$:

Lemma 63. *For any state $\mathbf{s}$ and event i,*

$$\widehat{p}_i(\mathbf{s}) = \frac{1}{n}(\tau(\mathbf{s}) - \tau(\mathbf{s}\dot{+}\mathbf{e}_i)).$$

Proof. When $i \notin \lambda(\mathbf{s})$, then $\mathbf{s} = \mathbf{s}\dot{+}\mathbf{e}_i$ and it is trivially true that

$$\widehat{p}_i(\mathbf{s}) = 0 = \frac{1}{n}(\tau(\mathbf{s}) - \tau(\mathbf{s}\dot{+}\mathbf{e}_i)).$$

The interesting case is when $i \in \lambda(\mathbf{s})$. Indeed, Using (6.4), we have

$$\begin{aligned}
&\tau(\mathbf{s}) - \tau(\mathbf{s} + \mathbf{e}_i) \\
&\quad = \mathbb{E}\ T(\mathbf{s}) - \mathbb{E}\ T(\mathbf{s} + \mathbf{e}_i) \\
&\quad = \mathbb{E}\left[\sum_{t=0}^{\infty} \mathbf{1}[\mathbf{s}\dot{+}S_t \neq \mathbf{d}] - \mathbf{1}[(\mathbf{s} + \mathbf{e}_i)\dot{+}S_t \neq \mathbf{d}]\right].
\end{aligned}$$

Since the dead state $\mathbf{d}$ is an absorbing state we have that for any path S, if $\mathbf{s}\dot{+}S = \mathbf{d}$, then $\mathbf{s} + \mathbf{e}_i\dot{+}S = \mathbf{d}$ as well. Equivalently, if $(\mathbf{s} + \mathbf{e}_i)\dot{+}S \neq \mathbf{d}$, then $\mathbf{s}\dot{+}S \neq \mathbf{d}$. Thus in the difference between the expectations, we only need be concerned with sequences S_t that are accounted for in the first expectation but not in the second. Therefore the above difference becomes

$$= \mathbb{E}\left[\sum_{t=0}^{\infty} \mathbf{1}[(\mathbf{s}\dot{+}S_t \neq \mathbf{d}) \wedge ((\mathbf{s} + \mathbf{e}_i)\dot{+}S_t) = \mathbf{d}]\right].$$

We claim that any sequence S_t that satisfies the conjunction must have the property that $(S_t)_i = k - s_i$. This is true because $(\mathbf{s} + \mathbf{e}_i)\dot{+}S_t = \mathbf{d}$ and therefore $(S_t)_i \geq k + 1 - s_i$. Also $(S_t)_j \geq k + 1 - s_j$, for $j \neq i$. This implies that $\mathbf{s}\dot{+}S_t = \mathbf{o}_i$ and the above difference becomes

$$\mathbb{E}\left[\sum_{t=0}^{\infty} \mathbf{1}[\mathbf{s}\dot{+}S_t = \mathbf{o}_i]\right].$$

The last term is exactly $\widehat{p}_i(\mathbf{s})$, the probability that $\mathbf{s}\dot{+}S_t$ eventually arrives at $\mathbf{o}_i$, times the expected number of iterations spent in state $\mathbf{o}_i$ before arriving at $\mathbf{d}$. To leave $\mathbf{o}_i$, the random walk must make a step in the ith direction, and thus the expected "waiting time" at $\mathbf{o}_i$ is can be computed as

$$\sum_{q=1}^{\infty} q \underbrace{(1 - \frac{1}{n})^{q-1}}_{\text{prob. of } q-1 \text{ loops}} \underbrace{\frac{1}{n}}_{\text{prob. of leaving}} = n.$$

□

The last lemma implies an important fact about the state lattice. Interpret the state lattice as a directed graph with directed edges at all pairs $(\mathbf{s}, \mathbf{s} + \mathbf{e}_i)$ for each $i \in \lambda(\mathbf{s})$. Also associate the edge $(\mathbf{s}, \mathbf{s} + \mathbf{e}_i)$ with the survival probability $\widehat{p}_i(\mathbf{s})$. Consider starting at state $\mathbf{s}$ and walking through this directed graph:

$$\mathbf{s} \to \mathbf{s} + \mathbf{e}_{i_1} \to \mathbf{s} + \mathbf{e}_{i_1} + \mathbf{e}_{i_2} \to \ldots$$

Corollary 64. *Consider any two states $\mathbf{s}, \mathbf{s}'$. For any path from $\mathbf{s}$ to $\mathbf{s}'$ through the directed state graph, the sum of all edge weights $\widehat{p}_i(\cdot)$ along this path is* independent *of the choice of path.*

Proof. Assume the path $\mathbf{s} = \mathbf{s}^1, \mathbf{s}^2, \ldots, \mathbf{s}^T, \mathbf{s}^{T+1} = \mathbf{s}'$ defined by a sequence of moves is $i_1, i_2, \ldots, i_T$, where $\mathbf{s}^{t+1} = \mathbf{s}^t + \mathbf{e}_{i_t}$. By Lemma 63 the total weight sum is

$$\sum_{t=1}^{T} \widehat{p}_{i_t}(\mathbf{s}^t) = \sum_{t=1}^{T} \frac{1}{n}(\tau(\mathbf{s}^t) - \tau(\mathbf{s}^{t+1})) = \frac{1}{n}(\tau(\mathbf{s}) - \tau(\mathbf{s}')),$$

which is independent of the choice of path. □

Note that in the definition of the directed state graph above and in the corollary we ignore loops, which occur when $\mathbf{s} = \mathbf{s}\dot{+}\mathbf{e}_i$ (or equivalently $i \notin \lambda(\mathbf{s})$). Such loops out of state $\mathbf{s}$ are immaterial because they correspond to dead events, and $i \notin \lambda(\mathbf{s})$ iff $\widehat{p}_i(\mathbf{s}) = 0$.

6.4 The Optimal Strategy

We now have the all the tools to express $\widehat{V}(\mathbf{s})$ in terms of the expected path length $\tau(\mathbf{s})$, prove that $V(\mathbf{s}) = \widehat{V}(\mathbf{s})$, and show that the optimal betting strategy for the gambler is $\widehat{p}(\mathbf{s})$.

We prove two major theorems in this section. We provide the mathematically precise argument for each but, as formality often obscures the true intuition, we also provide an

"English Version" so that the reader sees a rough sketch. Our mathematical proofs require induction on the state space $\mathcal{S}$, so we need a "measure of progress" for state vectors $\mathbf{s}$. For any $\mathbf{s} \in \mathcal{S}$, define $m(\mathbf{s}) := n(k+1) - |\mathbf{s}|$, the number of steps required before reaching the dead state. Clearly $m(\mathbf{s}) = 0$ if and only if $\mathbf{s} = \mathbf{d}$.

Theorem 65. *For all states* $\mathbf{s}$*,*

$$\widehat{V}(\mathbf{s}) = \frac{1}{n}\tau(\mathbf{s}).$$

Proof. (*English Version*) Assume that the Gambler always plays according to the distribution vector $\widehat{p}(\mathbf{s})$. Then we may think of the Casino's choices as a walk around the state graph and, as we discussed at the end of Section 6.3, a collection of the "weights" $\widehat{p}_i(\cdot)$ along the way, ending at $\mathbf{d}$. But as we proved in Corollary 64 for the weights $\widehat{p}(.)$, it doesn't matter what path is taken: the Casino will always receive $\frac{1}{n}(\tau(\mathbf{s}) - \tau(\mathbf{d})) = \frac{1}{n}\tau(\mathbf{s})$ on any path from $\mathbf{s}$ that just ended in $\mathbf{d}$.

If the Gambler ever chooses a distribution $\mathbf{w}$ different from $\widehat{p}(\mathbf{s})$ at some state $\mathbf{s}$, then the Casino can simply let $\boldsymbol{\ell} = \mathbf{e}_j$ for any j for which $w_j > \widehat{p}_j(\mathbf{s})$, and on this round the casino will force loss *greater* than $\widehat{p}_j(\mathbf{s})$. This means that on some path starting from $\mathbf{s}$, the Casino will accrue total weight/loss larger than $\frac{1}{n}\tau(\mathbf{s})$, and therefore that the distribution $\mathbf{w}$ at $\mathbf{s}$ was non-optimal for the Gambler. We conclude that for the Gambler $\widehat{p}(.)$ is the only optimal assignment of distributions to states. □

Proof. (*Formal Version*) We induct on $m(\mathbf{s})$. First we check the base case $\mathbf{s} = \mathbf{d}$. In this case, the expected path length is exactly 0 since we have already reached the dead state. Thus $\frac{\tau(\mathbf{s})}{n} = 0 = \widehat{V}(\mathbf{d})$ as desired.

Now assume that $m(\mathbf{s}) > 0$. Then

$$\begin{aligned}
\widehat{V}(\mathbf{s}) &= \min_{\mathbf{w}\sim\lambda(\mathbf{s})} \max_{i\in\lambda(\mathbf{s})} w_i + \widehat{V}(\mathbf{s}+\mathbf{e}_i) \\
\text{(induc.)} \quad &= \min_{\mathbf{w}\sim\lambda(\mathbf{s})} \max_{i\in\lambda(\mathbf{s})} w_i + \frac{1}{n}\tau(\mathbf{s}+\mathbf{e}_i) \\
&\leq \max_{i\in\lambda(\mathbf{s})} \widehat{p}_i(\mathbf{s}) + \frac{1}{n}\tau(\mathbf{s}+\mathbf{e}_i) \\
\text{(Lem. 63)} \quad &= \max_{i} \frac{1}{n}\left(\tau(\mathbf{s}) - \tau(\mathbf{s}+\mathbf{e}_i)\right) + \frac{1}{n}\tau(\mathbf{s}+\mathbf{e}_i) \\
&= \frac{1}{n}\tau(\mathbf{s}).
\end{aligned}$$

We prove $\widehat{V}(\mathbf{s}) \geq \frac{1}{n}\tau(\mathbf{s})$ by a similar induction. Assume that the Gambler chooses the optimal distribution $\mathbf{w}^*$ which may indeed be different from $\widehat{p}(\mathbf{s})$. For any $i \notin \lambda(\mathbf{s})$, $\widehat{p}_i(\mathbf{s})$ is defined as zero. For the optimal strategy $w_i^* = 0$ as well because otherwise the Casino can incurr unbounded loss by playing $\mathbf{e}_i$ repeatedly. Since $\mathbf{w}^*$ and $\widehat{p}(\mathbf{s})$ are different distributions

on the live events $\lambda(\mathbf{s})$, there must exist some $j \in \lambda(\mathbf{s})$ for which $w_j^* > \widehat{p}_j(\mathbf{s})$. We now have

$$\begin{aligned}
\widehat{V}(\mathbf{s}) &= \max_{i \in \lambda(\mathbf{s})} w_i^* + \widehat{V}(\mathbf{s} + \mathbf{e}_i) \\
\text{(induc.)} &= \max_{i \in \lambda(\mathbf{s})} w_i^* + \frac{1}{n}\tau(\mathbf{s} + \mathbf{e}_i) \\
&\geq w_j^* + \frac{1}{n}\tau(\mathbf{s} + \mathbf{e}_i) \\
&> \widehat{p}_j(\mathbf{s}) + \frac{1}{n}\tau(\mathbf{s} + \mathbf{e}_i) \\
\text{(Lem.63)} &= \frac{1}{n}(\tau(\mathbf{s}) - \tau(\mathbf{s} + \mathbf{e}_i)) + \frac{1}{n}\tau(\mathbf{s} + \mathbf{e}_i) \\
&= \frac{1}{n}\tau(\mathbf{s}).
\end{aligned}$$

□

Corollary 66. *For any $\mathbf{s} \neq \mathbf{d}$, $\widehat{p}(\mathbf{s})$ is the unique optimal probability vector for the learner for the game related to $\widehat{V}$.*

Proof. See end of last proof. □

Corollary 67. *For all $\mathbf{s}$ and all $i \in [n]$,*

$$\widehat{p}_i(\mathbf{s}) = \widehat{V}(\mathbf{s}) - \widehat{V}(\mathbf{s}\dot{+}\mathbf{e}_i)$$

Proof. This follows from the previous theorem and Lemma 63. □

We need one more lemma before we can prove our main result.

Lemma 68. *For any state $\mathbf{s}$ and distinct events $i, j \in \lambda(\mathbf{s})$, we have*

$$\widehat{p}_i(\mathbf{s}) < \widehat{p}_i(\mathbf{s} + \mathbf{e}_j).$$

This fact is intuitive: if losses are randomly assigned then the probability that the ith event will survive last *strictly increases* when another event suffers a loss. We prove this precisely below.

Proof. To show that $\widehat{p}_i(\mathbf{s}) \leq \widehat{p}_i(\mathbf{s} + \mathbf{e}_j)$ is straightforward. Any sequence $S_0, S_1, S_2, \ldots$ that brings $\mathbf{s}$ to the one-live state $\mathbf{o}_i$ also brings $\mathbf{s} + \mathbf{e}_j$ to $\mathbf{o}_i$. Indeed, if $\mathbf{s}\dot{+}S_t = \mathbf{o}_i$ for some t then certainly $(\mathbf{s} + \mathbf{e}_j)\dot{+}S_t = \mathbf{o}_i$ as well.

To show that this inequality is strict, we need only find one random sequence for which $\mathbf{s} + \mathbf{e}_j$ is brought to $\mathbf{o}_i$ but not $\mathbf{s}$. Take any sequence $S_0, S_1, \ldots$ such that $\mathbf{s}\dot{+}S_t = \mathbf{d} - \mathbf{e}_i - \mathbf{e}_j$ (where the only events remaining are i and j) and where $S_{t+1} = S_t + \mathbf{e}_i$. Then $(\mathbf{s}+\mathbf{e}_j)\dot{+}S_t = \mathbf{o}_i$ but $\mathbf{s}\dot{+}S_{t+1} = \mathbf{s}\dot{+}(S_t + \mathbf{e}_i) = \mathbf{o}_j$. □

Theorem 69. *For all states* $\mathbf{s}$*,*

$$V(\mathbf{s}) = \widehat{V}(\mathbf{s}) = \frac{1}{n}\tau(\mathbf{s}).$$

Proof. (*English Version*) Imagine a gambler who plays the distribution $\widehat{p}(\mathbf{s})$ at every state $\mathbf{s}$. We already know that the Casino can use its modified game strategy and simply play unit vectors $\boldsymbol{\ell} = \mathbf{e}_i$ on each round to force $\frac{1}{n}\tau(\mathbf{s})$ loss. Yet since $\boldsymbol{\ell}$ is unrestricted, can it obtain more? The answer is No: consider what happens if the Casino decides to choose $\boldsymbol{\ell}$ larger than a unit vector, e.g. let $\boldsymbol{\ell} = \mathbf{e}_i + \mathbf{e}_j$ for simplicity. Then on this round it obtains $\widehat{p}_i(\mathbf{s}) + \widehat{p}_j(\mathbf{s})$, but it can do better! We proved in Lemma 75 that survival probabilities strictly increase and therefore $\widehat{p}_i(\mathbf{s}) < \widehat{p}_i(\mathbf{s} + \mathbf{e}_j)$. Thus, a more patient Casino could choose $\boldsymbol{\ell} = \mathbf{e}_j$ on this round, obtain $\widehat{p}_j(\mathbf{s})$, and then choose $\boldsymbol{\ell} = \mathbf{e}_i$ on the next round to obtain $\widehat{p}_i(\mathbf{s} + \mathbf{e}_j)$. As $\widehat{p}_j(\mathbf{s}) + \widehat{p}_i(\mathbf{s} + \mathbf{e}_j) > \widehat{p}_j(\mathbf{s}) + \widehat{p}_i(\mathbf{s})$, the Casino only does worse by playing non-unit vectors. Indeed, this suggests that the Gambler has a strategy by which the Casino can inflict only as much loss as in the modified game, and thus the value $V(\mathbf{s})$ is no different from $\widehat{V}(\mathbf{s})$. □

Proof. (*Formal Version*) Certainly $V(\mathbf{s}) \geq \widehat{V}(\mathbf{s})$, since the Casino is given strictly fewer choices in the modified game. Thus we are left to show that $V(\mathbf{s}) \leq \widehat{V}(\mathbf{s})$. We proceed via induction on $m(\mathbf{s})$. By definition, $V(\mathbf{s}) = \widehat{V}(\mathbf{s})$ for the case $\mathbf{s} = \mathbf{d}$. Now assume that, for all successive states $\mathbf{s}'$ where $m(\mathbf{s}') < m(\mathbf{s})$, $V(\mathbf{s}') = \widehat{V}(\mathbf{s}')$. We proceed by directly analyzing the recursive definition (6.2). Assume that the Gambler has chosen the (possibly non-optimal) distribution $\mathbf{w} = \widehat{p}(\mathbf{s})$ to distribute his wealth on the live events $\lambda(\mathbf{s})$, and let $\boldsymbol{\ell}^* \in \{0,1\}^n$ be an optimal choice of the Casino (which can depend on the Gambler's choice). By definition (58) of $V(\mathbf{s})$, the chosen loss vector can't be $\mathbf{0}$ and all events with loss one must be in $\lambda(\mathbf{s})$. More precisely,

$$\begin{aligned}
V(\mathbf{s}) &= \min_{\mathbf{w}\sim\lambda(\mathbf{s})} \max_{\mathbf{0}\neq\boldsymbol{\ell}\subset\lambda(\mathbf{s})} \mathbf{w}\cdot\boldsymbol{\ell} + V(\mathbf{s}+\boldsymbol{\ell}) \\
&\overset{\text{(ind.)}}{=} \min_{\mathbf{w}\sim\lambda(\mathbf{s})} \max_{\mathbf{0}\neq\boldsymbol{\ell}\subset\lambda(\mathbf{s})} \mathbf{w}\cdot\boldsymbol{\ell} + \widehat{V}(\mathbf{s}+\boldsymbol{\ell}) \\
&\leq \max_{\mathbf{0}\neq\boldsymbol{\ell}\subset\lambda(\mathbf{s})} \widehat{p}(\mathbf{s})\cdot\boldsymbol{\ell} + \widehat{V}(\mathbf{s}+\boldsymbol{\ell}) \\
&= \widehat{p}(\mathbf{s})\cdot\boldsymbol{\ell}^* + \widehat{V}(\mathbf{s}+\boldsymbol{\ell}^*)
\end{aligned}$$

If $\boldsymbol{\ell}^*$ is any unit vector $\mathbf{e}_i$, s.t. $i \in \lambda(\mathbf{s})$, then

$$\begin{aligned}
V(\mathbf{s}) &\leq \widehat{p}(\mathbf{s})\cdot\mathbf{e}_i + \widehat{V}(\mathbf{s}+\mathbf{e}_i) \\
&= \widehat{p}_i(\mathbf{s}) + \widehat{V}(\mathbf{s}+\mathbf{e}_i) \;=\; \widehat{V}(\mathbf{s})
\end{aligned}$$

and in this case, $V(\mathbf{s}) = \widehat{V}(\mathbf{s})$ and we are done. We now prove by contradiction that $\boldsymbol{\ell}^*$ can have no more than one non-zero coordinate. Assume indeed that $|\boldsymbol{\ell}^*| > 1$, i.e. it admits a decomposition $\boldsymbol{\ell}^* = \mathbf{e}_i + \bar{\boldsymbol{\ell}}$ for some i and bit vector $\bar{\boldsymbol{\ell}} \neq \mathbf{0}$ with $\bar{\boldsymbol{\ell}}_i = 0$. Applying Lemma 75

repeatedly, we have that $\widehat{p}_i(\mathbf{s}) < \widehat{p}_i(\mathbf{s}+\bar{\boldsymbol{\ell}})$ and therefore

$$\begin{aligned}
&\widehat{p}(\mathbf{s})\cdot\boldsymbol{\ell}^* + \widehat{V}(\mathbf{s}+\boldsymbol{\ell}^*) \\
&\quad = \widehat{p}_i(\mathbf{s}) + \widehat{p}(\mathbf{s})\cdot\bar{\boldsymbol{\ell}} + \widehat{V}(\mathbf{s}+\boldsymbol{\ell}^*) \\
\text{(Lem. 75)} &\quad < \widehat{p}_i(\mathbf{s}+\bar{\boldsymbol{\ell}}) + \widehat{p}(\mathbf{s})\cdot\bar{\boldsymbol{\ell}} + \widehat{V}(\mathbf{s}+\boldsymbol{\ell}^*) \\
\text{(Cor. 67)} &\quad = \widehat{V}(\mathbf{s}+\bar{\boldsymbol{\ell}}) - \widehat{V}(\mathbf{s}+\boldsymbol{\ell}^*) + \widehat{p}(\mathbf{s})\cdot\bar{\boldsymbol{\ell}} + \widehat{V}(\mathbf{s}+\boldsymbol{\ell}^*) \\
&\quad = \widehat{p}(\mathbf{s})\cdot\bar{\boldsymbol{\ell}} + \widehat{V}(\mathbf{s}+\bar{\boldsymbol{\ell}}).
\end{aligned}$$

But the statement $\widehat{p}(\mathbf{s})\cdot\boldsymbol{\ell}^* + \widehat{V}(\mathbf{s}+\boldsymbol{\ell}^*) < \widehat{p}(\mathbf{s})\cdot\bar{\boldsymbol{\ell}} + \widehat{V}(\mathbf{s}+\bar{\boldsymbol{\ell}})$ implies $\boldsymbol{\ell}^*$ is a non-optimal choice for the Casino and this contradicts our assumption that $\boldsymbol{\ell}^*$ was optimum. □

Corollary 70. *For any $\mathbf{s} \neq \mathbf{d}$, if the learner plays with the optimum probability vector $\widehat{p}(\mathbf{s})$, then the only optimal responses of the adversary in the recurrence* (6.2) *for V is to choose a unit vector of a live event.*

Proof. Proved at the end of the last theorem. □

6.5 Recurrences, Combinatorics and Randomized Algorithms

The quantities $V(\mathbf{s})$, $\tau(\mathbf{s})$ and $\widehat{p}_i(\mathbf{s})$ have a number of interesting properties that we lay out in this section.

6.5.1 Some Recurrences

The expected path length, $\tau(\mathbf{s})$ satisfies a very natural recursion. When $\mathbf{s} = \mathbf{d}$, then the path length is deterministically 0 and therefore $\tau(\mathbf{d}) = 0$. Otherwise, we see that the expected path length is

$$\tau(\mathbf{s}) = 1 + \frac{\sum_{i=1}^{n} \tau(\mathbf{s}\dot{+}\mathbf{e}_i)}{n}. \tag{6.5}$$

That is, the expected path length is 1, for the current step in the path, plus the expected path length of the next random state. Since the next state is chosen randomly from the set $\{\mathbf{s}\dot{+}\mathbf{e}_i : i = 1, \ldots, n\}$, the probability of any given state is $\frac{1}{n}$, hence the normalization factor.

Of course, our original quantity of interest is $V(\mathbf{s})$, and as we showed in Theorem 69 $V(\mathbf{s}) = \frac{1}{n}\tau(\mathbf{s})$. This immediately gives us a recursion for V:

$$\begin{aligned}
V(\mathbf{s}) &= \frac{1}{n}\left(1 + \frac{1}{n}\sum_{i=1}^{n} \tau(\mathbf{s}\dot{+}\mathbf{e}_i)\right) \\
&= \frac{1 + \sum_{i=1}^{n} V(\mathbf{s}\dot{+}\mathbf{e}_i)}{n}.
\end{aligned}$$

This recurrence, while true for the function $V(\cdot)$, is ambiguous because $V(\mathbf{s})$ can occur on both sides of the equation. Indeed, whenever $i \notin \lambda(\mathbf{s})$, $V(\mathbf{s}\dot{+}\mathbf{e}_i) = V(\mathbf{s})$. However, we can rearrange all $V(\mathbf{s})$ terms to obtain the following well-defined recursion:

$$V(\mathbf{s}) = \frac{1 + \sum_{i \in \lambda(\mathbf{s})} V(\mathbf{s} + \mathbf{e}_i)}{|\lambda(\mathbf{s})|}. \tag{6.6}$$

We can find a similar recurrence for $\widehat{p}_i(\cdot)$. For the one-live states $\mathbf{o}_i$ we have $\widehat{p}_j(\mathbf{o}_i) = 1$ if $i = j$ and 0 otherwise. If $|\lambda(\mathbf{s})| > 1$, then

$$\widehat{p}_i(\mathbf{s}) = \frac{\sum_{j=1}^{n} \widehat{p}_i(\mathbf{s}\dot{+}\mathbf{e}_j)}{n}.$$

As $\widehat{p}_i(\mathbf{s})$ is the probability of ending at state $\mathbf{o}_i$ after executing the Markov chain, this formula is obtained by conditioning on one step of the Markov process. That is, the probability of ending at state $\mathbf{o}_i$ is

$$\sum_j Pr(j \text{ chosen}) Pr(\text{random process takes } \mathbf{s}\dot{+}\mathbf{e}_j \text{ to } \mathbf{o}_i).$$

This recurrence suffers from the same problem as did our initial recurrence for $V(\cdot)$: $\widehat{p}(\mathbf{s})$ can occur on both sides of the equality. We again solve this problem by rearranging terms and obtain

$$\widehat{p}_i(\mathbf{s}) = \frac{\sum_{j \in \lambda(\mathbf{s})} \widehat{p}_i(\mathbf{s} + \mathbf{e}_j)}{|\lambda(\mathbf{s})|}.$$

6.5.2 Combinatorial Sums

A further analysis gives us exact expressions for both $\widehat{p}_i(\mathbf{s})$ and $V(\mathbf{s})$ in terms of infinite sums of multinomials.

Proposition 71. *For any state* $\mathbf{s} \in \mathcal{S}$*,*

$$\widehat{p}_i(\mathbf{s}) = \sum_{\mathbf{r}:\mathbf{s}\dot{+}\mathbf{r}=\mathbf{o}_i} \binom{|\mathbf{r}|}{r_1, r_2, \ldots, r_n} \left(\frac{1}{n}\right)^{|\mathbf{r}|+1}.$$

Proof. By definition, $\widehat{p}_i(\mathbf{s})$ is the probability that $\mathbf{s}$ reaches the one-live state $\mathbf{o}_i$ eventually. To compute this probability, we consider at what point the Markov process *exits* the state $\mathbf{o}_i$ and into $\mathbf{d}$. Recall the random variable S_t defined in Section 6.3. Take any $\mathbf{r}$ for which $\mathbf{s}\dot{+}\mathbf{r} = \mathbf{o}_i$ and condition on $S_t = \mathbf{r}$. Then

$$\widehat{p}_i(\mathbf{s}) = \sum_{\mathbf{r}:\mathbf{s}\dot{+}\mathbf{r}=\mathbf{o}_i} Pr(S_t = \mathbf{r}) Pr(S_{t+1} = \mathbf{r} + \mathbf{e}_i | S_t = \mathbf{r})$$

The first probability is exactly $\binom{|\mathbf{r}|}{r_1, r_2, \ldots, r_n} n^{-|\mathbf{r}|}$ and the second probability is exactly $1/n$. □

Since $V(\mathbf{s})$ can be written as an expected path length, we can obtain a similar expression as a sum of multinomials for $V(\mathbf{s})$:

Proposition 72.

$$V(\mathbf{s}) = \sum_{i=1}^{n} \sum_{\mathbf{r}:\mathbf{r}\dot{+}\mathbf{s}=\mathbf{o}_i} (|\mathbf{r}|+1)\binom{|\mathbf{r}|}{r_1, r_2, \ldots, r_n}\left(\frac{1}{n}\right)^{|\mathbf{r}|+1}.$$

6.5.3 Randomized Approximations

Computing the exact value $V(\mathbf{s})$ for large but non-asymptotic values of the state vector is difficult because we have no polynomial time algorithm. On the other hand, finding a randomized approximation to $V(\mathbf{s})$ can be done very efficiently. Indeed, as we now have a representation of $V(\mathbf{s})$ in terms of the length of a random walk, we can simply run the random walk $S_1, S_2, \ldots$ several times, note the length $T(\mathbf{s})$, and return the mean. Such random approximations require that the distribution on $T(\mathbf{s})$ has low-variance, yet this certainly holds in the case at hand. While the random walk requires at least $n(k+1)$ iterations to finish, a simple argument shows that with probability $1-\delta$ the random walk completes in less than $nk\log(nk/\delta)$ rounds.

Algorithm 10 Random Approximation to $V(\mathbf{s})$

```
Input: state s
t ← 0
for i = 1, ..., NUMITER do
  z ← s
  repeat
    Sample i ∈ {1, ..., n} u.a.r.
    z ← z ∔ e_i
    t ← t + 1
  until z = d
end for
Return t / (n · NUMITER).
```

If $R(\mathbf{s})$ is the random variable returned by the above algorithm, then clealy $\mathbb{E}\, R(\mathbf{s}) = V(\mathbf{s})$. By increasing `NUMITER`, the variance of this estimate can be reduced quickly.

A randomized approximation for $\widehat{p}(\mathbf{s})$ can be obtained similarly. Again the above algorithm approximately computes $\widehat{p}(\mathbf{s})$ in the following sense: If $R(\mathbf{s})$ is the random variable returned by the above algorithm, then clearly $\mathbb{E}\, R(\mathbf{s}) = \widehat{p}(\mathbf{s})$. Again increasing `NUMITER`, reduces the variance of the estimate.

Algorithm 11 Random Approximation to $\widehat{p}(\mathbf{s})$

Input: state $\mathbf{s} \neq \mathbf{d}$
$\mathbf{p} \leftarrow \mathbf{0}$
for $i = 1, \ldots,$ `NUMITER` **do**
  $\mathbf{z} \leftarrow \mathbf{s}$
  repeat
    Sample $i \in \{1, \ldots, n\}$ u.a.r.
    $\mathbf{z} \leftarrow \mathbf{z} \dot{+} \mathbf{e}_i$
  until $\mathbf{z} = \mathbf{o}_j$ for some j
  $\mathbf{p} \leftarrow \mathbf{p} + \mathbf{e}_j$
end for
Return $\frac{\mathbf{p}}{\texttt{NUMITER}}$.

6.5.4 A Simple Strategy in a Randomized Setting

In the particular case of betting against the Casino, it may be necessary for the Gambler to compute $\widehat{p}_i(\mathbf{s})$ in order to place his bets optimally. In an alternative setting, however, a randomized algorithm may be sufficient. Let us consider the case in which the Gambler chooses to bet according to the outcome of several coin tosses. Further assume that the Casino can observe his strategy but cannot see the outcome of the coin tosses or his final bets. In this scenario, the Gambler can even bet all of his money on a random event $I \in \{1, \ldots, n\}$ drawn according to some distribution as long as $\mathbb{E}\,\mathbf{1}[I = i] = \widehat{p}_i(\mathbf{s})$ for all i, and indeed his expected loss would be $\widehat{p}(\mathbf{s}) \cdot \ell$.

For this scenario, randomly approximating $\widehat{p}$ is not necessary: *only one sample is needed!* To be precise, the Gambler can take the state $\mathbf{s}$, run the random walk until the state reaches $\mathbf{o}_i$ for some i, and then bet his full dollar on event i. This bet will be correct in expectation, i.e. he will pick event i with probability $\widehat{p}_i(\mathbf{s})$, and thus his expected loss will be exactly $\widehat{p} \cdot \ell$. The key here is that sampling from the distribution $\widehat{p}(\mathbf{s})$ may be quite easy even when computing it exactly may take more time.

Note that the above method based on one sample is similar to the way the Randomized Weighted Majority algorithm approximates the Weighted Majority algorithm (more precisely the WMC algorithm of [53]). More precisely `NUMITER=1` of Algorithm 6.5.3 corresponds to WMR, and `NUMITER` $\to \infty$ corresponds to WMC.

6.6 Comparison to Previous Bounds

As mentioned in the introduction, the bound obtainable base on exponential weights [38] is

$$k + \sqrt{2k \log n} + \log n \tag{6.7}$$

and can be shown to be asymptotically optimal [75]. Having computed the minimax solution to the same game, we can compute the game-theoretically optimal bound of $V(\mathbf{0})$ using

Algorithm 10. For small values of n and k, these bounds do differ quite substantially. We present in Figure **??** a comparison of the regret for $n = 2, 10, 100$ and $k = 1, \ldots, 20$.

6.7 Connections to classic problems of probabilistic enumerative combinatorics.

Theorem 69 shows that an optimal strategy for the Casino requires unit vector plays. This leads to alternative interpretations of the game in terms of well studied random processes.

For example, one can easily confirm that our game also describes the random process underlying a generalized form of the Coupon Collector's Problem [32] in which the collector buys cereal boxes one by one in order to obtain $K = k + 1$ complete sets of n baseball cards, assuming one card is randomly placed within each cereal box. The value of our game, $V(0, 0)$, is in fact the expected number of cereal boxes, per baseball card, needed to obtain the desired K complete sets.

Specifically, the probability generating function for the generalized Coupon Collector's Problem is [59]

$$G_{n,K}(z) = \frac{n}{(K-1)!} \int_0^\infty e^{-nt/z} t^{K-1} \left[\sum_{j \geq k} \frac{t^j}{j!} \right]^{n-1} dt.$$

Taking the derivative at $z = 1$ and dividing by n, we derive the expected number of steps to obtain K sets, which is also the value of our game, viz.

$$V(0^n) = \frac{n}{(K-1)!} \int_0^\infty t^K e^{-nt} \left[\sum_{j \geq k} \frac{t^j}{j!} \right]^{n-1} dt. \tag{6.8}$$

Equation (6.8) gives us an elegant closed form for the two-card case ($n = 2$):

$$V(\langle 0, 0 \rangle) = K + \frac{K}{2^{2K}} \binom{2K}{K}$$

From (6.8) we also obtain the well known asymptotic expression for the value, for large n and fixed K,

$$V(0^n) \to_{n \to \infty} \log n + (K-1) \log \log n [1 + o(1)].$$

The same asymptotic form appears in the analysis of an evolving random graph. [29] The random walk on the state lattice provides yet another interpretation of the same dynamics.

For $K >> n >> 1$, the law of large numbers gives [64]

$$V(\langle 0^n \rangle) = K + O(K^{1/2}).$$

6.8 Conclusion

We showed in Corollary 70 that against the optimal learning algorithm the optimal strategy of the adversary is to choose one of the unit loss vectors as his response. Curiously enough it can be show that this is also true of the Weighted Majority algorithm (6.1). That is, any trial in which $q > 1$ experts incurred a unit of loss can be split into q trials in which a single expert has a unit of loss, and doing this always increases the loss of the algorithm for all update factor $\beta \in [0, 1)$. This observation about the Weighted Majority algorithm might actually lead to improved loss bounds for this algorithm, perhaps in the way the parameter β is tuned.

There remains also a deep question regarding the techniques introduced in this chapter: how general is this method of computing the value of a game based on a random path? Can it handle slightly more involved problems? Examples we have considered include competing against m-sized sets of experts, discussed in [76], in which the loss of the algorithm is compared to the loss of the best m-subset. Another example is the problem of competing against permutations of n objects [49], where the loss of a permutation is linearly assigned. Our preliminary investigation suggests that similar techniques can be adapted to also handle such more complex problems.

Chapter 7

Repeated Games and Budgeted Adversaries

7.1 Introduction

How can we reasonably expect to learn given possibly adversarial data? Overcoming this obstacle has been one of the major successes of the Online Learning framework or, more generally, the so-called competitive analysis of algorithms: rather than measure an algorithm only by the cost it incurs, consider this cost *relative* to an optimal "comparator algorithm" which has knowledge of the data in advance. A classic example is the so-called "experts setting": assume we must predict a sequence of binary outcomes and we are given access to a set of *experts*, each of which reveals their own prediction for each outcome. After each round we learn the true outcome and, hence, which experts predicted correctly or incorrectly. The expert setting is based around a simple assumption, that while some experts' predictions may be adversarial, we have an a priori belief that there is at least one good expert whose predictions will be reasonably accurate. Under this relatively weak good-expert assumption, one can construct algorithms that have quite strong loss guarantees.

Another way to interpret this sequential prediction model is to treat it as a repeated two-player zero-sum game against an adversary *on a budget*; that is, the adversary's sequence of actions is restricted in that play ceases once the adversary exceeds the budget. In the experts setting, the assumption "there is a good expert" can be reinterpreted as a "nature shall not let the best expert err too frequently", perhaps more than some fixed number of times.

In the present chapter, we develop a general framework for repeated game-playing against an adversary on a budget, and we provide a simple randomized strategy for the learner/player for a particular class of these games. The proposed algorithms are based on a technique, which we refer to as a "random playout", that has become a very popular heuristic for solving games with massively-large state spaces. Roughly speaking, a random playout in an extensive-form game is a way to measure the likely outcome at a given state by finishing the game randomly from this state. Random playouts, often known simply as Monte Carlo

methods, have become particularly popular for solving the game of Go [20], which has led to much follow-up work for general games [41, 40]. The Budgeted Adversary game we consider also involves exponentially large state spaces, yet we achieve efficiency using these random playouts. The key result of this chapter is that the proposed random playout is not simply a good heuristic, it is indeed *minimax optimal* for the games we consider.

Abernethy et al [6] was the first to use a random playout strategy to optimally solve an adversarial learning problem, namely for the case of the so-called Hedge Setting introduced by Freund and Schapire [38]. Indeed, their model can be interpreted as a particular special case of a Budgeted Adversary problem. The generalized framework that we give in the first half of the chapter, however, has a much larger range of applications. We give three such examples, described briefly below. More details are given in the second half of the chapter.

Cost-sensitive Hedge Setting. In the standard Hedge setting, it is assumed that each expert suffers a cost in $[0, 1]$ on each round. But a surprisingly-overlooked case is when the cost ranges differ, where expert i may suffer per-round cost in $[0, c_i]$ for some fixed $c_i > 0$. The vanilla approach, to use a generic bound of $\max_i c_i$, is extremely loose, and we know of no better bounds for this case. Our results provide the optimal strategy for this cost-sensitive Hedge setting.

Metrical Task Systems (MTS). The MTS problem is decision/learning problem similar to the Hedge Setting above but with an added difficulty: the learner is required to pay the cost of moving through a given metric space. Finding even a near-optimal generic algorithm has remained elusive for some time, with recent encouraging progress made in one special case [12], for the so-called "weighted-star" metric. Our results provide a simple minimax optimal algorithm for this problem.

7.2 Preliminaries

Notation: We shall write $[n]$ for the set $\{1, 2, \ldots, n\}$, and $[n]^*$ to be the set of all finite-length sequences of elements of $[n]$. We will use the greek symbols ρ and σ to denote such sequences $i_1 i_2 \ldots i_T$, where $i_t \in [n]$. We let $\emptyset$ denote the empty sequence. When we have defined some T-length sequence $\rho = i_1 i_2 \ldots i_T$, we may write ρ_t to refer to the t-length prefix of ρ, namely $\rho_t = i_1 i_2 \ldots i_t$, and clearly $t \leq T$. We will generally use $\mathbf{w}$ to refer to a distribution in Δ_n, the n-simplex, where w_i denotes the ith coordinate of $\mathbf{w}$. We use the symbol $\mathbf{e}_i$ to denote the ith basis vector in n dimensions, namely a vector with a 1 in the ith coordinate, and 0's elsewhere. We shall use $\mathbf{1}[\cdot]$ to denote the "indicator function", where $\mathbf{1}[\texttt{predicate}]$ is 1 if `predicate` is true, and 0 if it is false. It may be that `predicate` is a random variable, in which case $\mathbf{1}[\texttt{predicate}]$ is a random variable as well.

7.2.1 The Setting: Budgeted Adversary Games

We will now describe the generic sequential decision problem, where a problem instance is characterized by the following triple: an $n \times n$ loss matrix M, a monotonic "cost function"

cost : $[n]^* \to \mathcal{R}_+$, and a cost budget k. A cost function is *monotonic* as long as it satisfies the relation $\text{cost}(\rho\sigma) \leq \text{cost}(\rho i\sigma)$ for all $\rho, \sigma \in [n]^*$ and all $i \in [n]$. Play proceeds as follows:

1. On each round t, the player chooses a distribution $\mathbf{w}_t \in \Delta_n$ over his action space.
2. An outcome $i_t \in [n]$ is chosen by Nature (potentially an adversary).
3. The player suffers $\mathbf{w}_t^\top M \mathbf{e}_{i_t}$.
4. The game proceeds until the first round in which the budget is spent, i.e. the round T when $\text{cost}(i_1 i_2 \dots i_{T-1}) \leq k < \text{cost}(i_1 i_2 \dots i_{T-1} i_T)$.

The goal of the Player is to choose each $\mathbf{w}_t$ in order to minimize the total cost of this repeated game on all sequences of outcomes. Note, importantly, that the player can *learn* from the past, and hence would like an efficiently computable function $\mathbf{w} : [n]^* \to \Delta_n$, where on round t the player is given $\rho_{t-1} = (i_1 \dots i_{t-1})$ and sets $\mathbf{w}_t \leftarrow \mathbf{w}(\rho_{t-1})$. We can define the worst-case cost of an algorithm $\mathbf{w} : [n]^* \to \Delta_n$ by its performance against a worst-case sequence, that is

$$\text{WorstCaseLoss}(\mathbf{w}; M, \text{cost}, k) \;:=\; \max_{\substack{\rho = i_1 i_2 \dots \in [n]^* \\ \text{cost}(\rho_{T-1}) \leq k < \text{cost}(\rho_T)}} \sum_{t=1}^{T} \mathbf{w}(\rho_{t-1})^\top M \mathbf{e}_{i_t}.$$

Note that above T is a parameter chosen according to ρ and the budget. We can also define the minimax loss, which is defined by choosing the $\mathbf{w}(\cdot)$ which minimizes WorstCaseLoss$(\cdot)$. Specifically,

$$\text{MinimaxLoss}(M, \text{cost}, k) \;:=\; \min_{\mathbf{w}:[n]^* \to \Delta_n} \max_{\substack{\rho = i_1 i_2 \dots \in [n]^* \\ \text{cost}(\rho_{T-1}) \leq k < \text{cost}(\rho_T)}} \sum_{t=1}^{T} \mathbf{w}(\rho_{t-1})^\top M \mathbf{e}_{i_t}.$$

In the next section, we describe the optimal algorithm for a restricted class of M. That is, we obtain the mapping $\mathbf{w}$ which optimizes WorstCaseLoss$(\mathbf{w}; M, \text{cost}, k)$.

7.3 The Algorithm

We will start by assuming that M is a nonnegative diagonal matrix, that is $M = \text{diag}(c_1, c_2, \dots, c_n)$, and $c_i > 0$ for all i. With these values c_i, define the distribution $\mathbf{q} \in \Delta_n$ with $q_i := \frac{1/c_i}{\sum_j 1/c_j}$.

Given a current state ρ, the algorithm will rely heavily on our ability to compute the following function $\Phi(\cdot)$. For any $\rho \in [n]^*$ such that $\text{cost}(\rho) > k$, define $\Phi(\rho) := 0$. Otherwise, let

$$\Phi(\rho) \;:=\; \frac{1}{\sum_i 1/c_i} \mathop{\mathbb{E}}_{\forall t: i_t \sim \mathbf{q}} \left[\sum_{t=0}^{\infty} \mathbf{1}[\text{cost}(\rho i_1 \dots i_t) \leq k] \right]$$

Notice, this is the expected length of a random process. Of course, we must impose the natural condition that the length of this process has a finite expectation. Also, since we assume that the cost increases, it is reasonable to require that the distribution over the length, i.e. $\min\{t : \text{cost}(\rho i_1 \dots i_t) > k\}$, has an exponentially decaying tail. Under these weak conditions, the following m-trial Monte Carlo method will provide a high probability estimate to error within $O(m^{-1/2})$.

Algorithm 12 Efficient Estimation of $\Phi(\rho)$

for i=1...m **do**
  Sample: infinite random sequence $\sigma := i_1 i_2 \dots$ where $\Pr(i_t = i) = q_i$
  Let: $T_i = \max\{t : \text{cost}(\rho\sigma_{t-1}) \leq k\}$
end for
Return $\frac{\sum_{i=1}^{m} T_i}{m}$

Notice that the infinite sequence σ does not have to be fully generated. Instead, we can continue to sample the sequence and simply stop when the condition $\text{cost}(\rho\sigma_{t-1}) \geq k$ is reached. We can now define our algorithm in terms of $\Phi(\cdot)$.

Algorithm 13 Player's optimal strategy

Input: state ρ
Compute: $\Phi(\rho), \Phi(\rho, 1), \Phi(\rho, 2), \dots, \Phi(\rho, n)$
Let: set $\mathbf{w}(\rho)$ with values $w_i(\rho) = \frac{\Phi(\rho) - \Phi(\rho, i)}{c_i}$

7.4 Minimax Optimality

Now we prove that Algorithm 13 is both "legal" and minimax optimal.

Lemma 73. *The vector $\mathbf{w}(\rho)$ computed in Algorithm 13 is always a valid distribution.*

Proof. It must first be established that $w_i(\rho) \geq 0$ for all i and ρ. This, however, follows because we assume that the function cost() is monotonic, which implies that $\text{cost}(\rho\sigma) \leq \text{cost}(\rho i\sigma)$ and hence $\text{cost}(\rho i\sigma) \leq k \implies \text{cost}(\rho\sigma) \leq k$, and hence $\mathbf{1}[\text{cost}(\rho i\sigma) \leq k] \leq \mathbf{1}[\text{cost}(\rho\sigma) \leq k]$. Taking the expected difference of the infinite sum of these two indicators leads to $\Phi(\rho) - \Phi(\rho i) \geq 0$, which implies $w_i(\rho) \geq 0$ as desired.

We must also show that $\sum_i w_i(\rho) = 1$. We claim that the following recurrence relation holds for the function $\Phi(\rho)$ whenever $\text{cost}(\rho) \leq k$:

$$\Phi(\rho) = \underbrace{\frac{1}{\sum_i 1/c_i}}_{\text{first step}} + \underbrace{\sum_i q_i \Phi(\rho i)}_{\text{remaining steps}}, \text{ for any } \rho \text{ s.t. } \text{cost}(\rho) < k.$$

This is clear from noticing that Φ is an expected random walk length, with transition probabilities defined by $\mathbf{q}$, and scaled by the constant $(\sum_i 1/c_i)^{-1}$. Hence,

$$\begin{aligned}\sum_i w_i(\rho) &= \sum_i \frac{\Phi(\rho) - \Phi(\rho i)}{c_i} = \left(\sum_i 1/c_i\right)\Phi(\rho) - \sum_i \frac{\Phi(\rho i)}{c_i} \\ &= \left(\sum_i 1/c_i\right)\left(\frac{1}{\sum_i 1/c_i} + \sum_i q_i \Phi(\rho i)\right) - \sum_i \frac{\Phi(\rho i)}{c_i} \quad = \quad 1\end{aligned}$$

where the last equality holds because $q_i = \frac{1/c_i}{\sum_j 1/c_j}$. □

Theorem 74. *For $M = diag(c_1, \ldots, c_n)$, Algorithm 13 is minimax optimal for the Budgeted Adversary problem. Furthermore, $\Phi(\emptyset) = \text{MinimaxLoss}(M, \text{cost}, k)$.*

Proof. First we prove an upper bound. Notice that, for an sequence $\rho = i_1 i_2 i_3 \ldots i_T$, the total cost of Algorithm 13 will be

$$\sum_{t=1}^{T} \mathbf{w}(\rho_{t-1})^\top M \mathbf{e}_{i_t} = \sum_{t=1}^{T} w_{i_t}(\rho_{t-1}) c_{i_t} = \sum_{t=1}^{T} \frac{\Phi(\rho_{t-1}) - \Phi(\rho_t)}{c_{i_t}} c_{i_t} = \Phi(\emptyset) - \Phi(\rho_T) \leq \Phi(\emptyset)$$

and hence the total cost of the algorithm is always bounded by $\Phi(\emptyset)$.

On the other hand, we claim that $\Phi(\emptyset)$ can always be achieved by an adversary for any algorithm $\mathbf{w}'(\cdot)$. Construct a sequence ρ as follows. Given that ρ_{t-1} has been constructed so far, select any coordinate $i_t \in [n]$ for which $w_{i_t}(\rho_{t-1}) \leq w'_{i_t}(\rho_{t-1})$, that is, where the the algorithm $\mathbf{w}'$ places at least as much weight on i_t as the proposed algorithm $\mathbf{w}$ we defined in Algorithm 13. This must always be possible because both $\mathbf{w}(\rho_{t-1})$ and $\mathbf{w}'(\rho_{t-1})$ are distributions and neither can fully dominate the other. Set $\rho_t \leftarrow \rho_{t-1} i$. Continue constructing ρ until the budget is reached, i.e. $\text{cost}(\rho) > k$. Now, let us check the loss of $\mathbf{w}'$ on this sequence ρ:

$$\sum_{t=1}^{T} \mathbf{w}'(\rho_{t-1})^\top M \mathbf{e}_{i_t} = \sum_{t=1}^{T} w'_{i_t}(\rho_{t-1}) c_{i_t} \geq \sum_{t=1}^{T} w_{i_t}(\rho_{t-1}) c_{i_t} = \Phi(\emptyset) - \Phi(\rho) = \Phi(\emptyset)$$

Hence, an adversary can achieve at least $\Phi(\emptyset)$ loss for any algorithm $\mathbf{w}'$. □

7.4.1 Extensions

For simplicity of exposition, we proved Theorem 74 under a somewhat limited scope: only for diagonal matrices M, known budget k and cost(). But with some work, these restrictions can be lifted. We sketch a few extensions of the result, although we omit the details due to lack of space.

First, the concept of a cost() function and a budget k is not entirely necessary. Indeed, we can redefine the Budgeted Adversary game in terms of an arbitrary stopping criterion

$\delta : [n]^* \to \{0, 1\}$, where $\delta(\rho) = 0$ is equivalent to "the budget has been exceeded". The only requirement is that $\delta()$ is monotonic, which is naturally defined as $\delta(\rho i \sigma) = 1 \implies \delta(\rho\sigma) = 1$ for all $\rho, \sigma \in [n]^*$ and all $i \in [n]$. This alternative budget interpretation lets us consider the sequence ρ as a path through a game tree. At a given node ρ_t of the tree, the adversary's action i_{t+1} determines which branch to follow. As soon as $\delta(\rho_t) = 0$ we have reached a terminal node of this tree.

Second, we need not assume that the budget k, or even the generalized stopping criterion $\delta()$, is known in advance. Instead, we can work with the following generalization: the stopping criterion δ is drawn from a known prior λ and given to the adversary before the start of the game. The resulting optimal algorithm depends simply on estimating a new version of $\Phi(\rho)$. $\Phi(\rho)$ is now redefined as both an expectation over a random σ and a random δ drawn from the *posterior* of λ, that is where we condition on the event $\delta(\rho) = 1$.

Third, Theorem 74 can be extended to a more general class of M, namely *inverse-nonnegative matrices*, where M is invertible and M^{-1} has all nonnegative entries. (In all the examples we give we need only diagonal M, but we sketch this generalization for completeness). If we let $\mathbf{1}_n$ be the vector of n ones, then define $D = \text{diag}^{-1}(M^{-1}\mathbf{1}_n)$, which is a nonnegative diagonal matrix. Also let $N = DM^{-1}$ and notice that the rows of N are the normalized rows of M^{-1}. We can use Algorithm 13 with the diagonal matrix D, and attain distribution $\mathbf{w}'(\rho)$ for any ρ. To obtain an algorithm for the matrix M (not D), we simply let $\mathbf{w}(\rho) = (\mathbf{w}'(\rho)^\top N)^\top$, which is guaranteed to be a distribution. The loss of $\mathbf{w}$ is identical to $\mathbf{w}'$ since $\mathbf{w}(\rho)^\top M = \mathbf{w}'(\rho)^\top D$ by construction.

Fourth, we have only discussed minimizing *loss* against a budgeted adversary. But all the results can be extended easily to the case where the player is instead maximizing gain (and the adversary is minimizing). A particularly surprising result is that the minimax strategy is *identical* in either case; that is, the the recursive definition of $w_i(\rho)$ is the same whether the player is maximizing or minimizing. However, the termination condition might change depending on whether we are minimizing or maximizing. For example in the expert setting, the game stops when all experts have cost larger than k versus at least one expert has gain at least k. Therefore for the same budget size k, the minimax value of the gain version is typically smaller than the value of the loss version.

Simplified Notation. For many examples, including two that we consider below, recording the entire sequence ρ is unnecessary—the only relevant information is the *number* of times each i occurs in ρ and not where it occurs. This is the case precisely when the function $\text{cost}(\rho)$ is unchanged up to permutations of ρ. In such situations, we can consider a smaller state space, which records the "counts" of each i in the sequence ρ. We will use the notation $\mathbf{s} \in \mathbb{N}^n$, where $\mathbf{s}_t = \mathbf{e}_{i_1} + \ldots + \mathbf{e}_{i_t}$ for the sequence $\rho_t = i_1 i_2 \ldots i_t$.

7.5 The Cost-Sensitive Hedge Setting

A straightforward application of Budgeted Adversary games is the "Hedge setting" introduced by Freund and Schapire [38], a version of the aforementioned experts setting. The minimax algorithm for this special case was already thoroughly developed by Abernethy et al [6]. We describe an interesting extension that can be achieved using our techniques which has not yet been solved.

The Hedge game goes as follows. A learner must predict a sequence of distributions $\mathbf{w}_t \in \Delta_n$, and receive a sequence of loss vectors $\ell_t \in \{0,1\}^n$. The total loss to the learner is $\sum_t \mathbf{w}_t \cdot \ell_t$, and the game ceases only once the best expert has more than k errors, i.e. $\min_i \sum_t \ell_{t,i} > k$. The learner wants to minimize his total loss.

The natural way to transform the Hedge game into a Budgeted Adversary problem is as follows. We'll let $\mathbf{s}$ be the state, defined as the vector of cumulative losses of all the experts.

$$M = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} \qquad \text{cost}(\mathbf{s}) = \min_i s_i \qquad \sum_t \mathbf{w}_t \cdot \ell_t = \sum_t \mathbf{w}_t^\top M \mathbf{e}_{i_t}$$

The proposed reduction *almost* works, except for one key issue: this only allows cost vectors of the form $\ell_t = M\mathbf{e}_{i_t} = \mathbf{e}_{i_t}$, since by definition Nature chooses columns of M. However, as shown in Abernethy et al, this is not a problem.

Lemma 75 (Lemma 11 and Theorem 12 of [6])**.** *In the Hedge game, the worst case adversary always chooses* $\ell_t \in \{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$.

The standard and more well-known, although non-minimax, algorithm for the Hedge setting [38] uses a simple modification of the Weighted Majority Algorithm [53], and is described simply by setting $w_i(\mathbf{s}) = \frac{\exp(-\eta s_i)}{\sum_j \exp(-\eta s_j)}$. With the appropriate tuning of η, it is possible to bound the total loss of this algorithm by $k + \sqrt{2k \ln n} + \ln n$, which is known to be roughly optimal in the limit. Abernethy et al [6] provide the minimax optimal algorithm, but state the bound in terms of an expected length of a random walk. This is essentially equivalent to our description of the minimax cost in terms of $\Phi(\emptyset)$.

A significant drawback of the Hedge result, however, is that it requires the losses to be uniformly bounded in $[0,1]$, that is $\ell_t \in [0,1]^n$. Ideally, we would like an algorithm and a bound that can handle non-uniform cost ranges, i.e. where expert i suffers loss in some range $[0, c_i]$. The $\ell_{t,i} \in [0,1]$ assumption is fundamental to the Hedge analysis, and we see no simple way of modifying it to achieve a tight bound. The simplest trick, which is just to take $c_{\max} := \max_i c_i$, leads to a bound of the form $k + \sqrt{2c_{\max} k \ln n} + c_{\max} \ln n$ which we know to be very loose. Intuitively, this is because only a single "risky" expert, with a large c_i, should not affect the bound significantly.

In our Budgeted Adversary framework, this case can be dealt with trivially: letting $M = \text{diag}(c_1, \ldots, c_n)$ and $\text{cost}(\mathbf{s}) = \min_i s_i c_i$ gives us immediately an optimal algorithm that, by Theorem 74, we know to be minimax optimal. According to the same theorem, the minimax loss bound is simply $\Phi(\emptyset)$ which, unfortunately, is in terms of a random walk

length. We do not know how to obtain a closed form estimate of this expression, and we leave this as an intriguing open question.

7.6 Metrical Task Systems

A classic problem from the Online Algorithms community is known as Metrical Task Systems (MTS), which we now describe. A player (decision-maker, algorithm, etc.) is presented with a finite metric space and on each of a sequence of rounds will occupy a single state (or point) within this metric space. At the beginning of each round the player is presented with a *cost vector*, describing the cost of occupying each point in the metric space. The player has the option to remain at the his present state and pay this states associated cost, or he can decide to switch to another point in the metric and pay the cost of the new state. In the latter case, however, the player must also pay the *switching cost* which is exactly the metric distance between the two points.

The MTS problem is a useful abstraction for a number of problems; among these is job-scheduling. An algorithm would like to determine on which machine, across a large network, it should process a job. At any given time point, the algorithm observes the number of available cycles on each machine, and can choose to migrate the job to another machine. Of course, if the subsequent machine is a great distance, then the algorithm also pays the travel time of the job migration through the network.

Notice that, were we given a sequence of cost vectors in advance, we could compute the optimal path of the algorithm that minimized total cost. Indeed, this is efficiently solved by dynamic programming, and we will refer to this as the *optimal offline cost*, or just the offline cost. What we would like is an algorithm that performs well relative to the offline cost without knowledge of the sequence of cost vectors. The standard measure of performance for an online algorithm is the *competitive ratio*, which is the ratio of cost of the online algorithm to the optimal offline cost. For all the results discussed below, we assume that the online algorithm can maintain a *randomized* state—a distribution over the metric—and pays the expected cost according to this random choice (Randomized algorithms tend to exhibit much better competitive ratios than deterministic algorithms).

When the metric is uniform, i.e. where all pairs of points are at unit distance, it is known that the competitive ratio is $O(\log n)$, where n is the number of points in the metric; this was shown by Borodin, Linial and Saks who introduced the problem [19]. For general metric spaces, Bartal et al achieved a competitive ratio of $O(\log^6 n)$ [13], and this was improved to $O(\log^2 n)$ by Fiat and Mendel [31]. The latter two techniques, however, rely on a scheme of randomly approximating the metric space with a hierarchical tree metric, adding a (likely-unnecessary) multiplicative cost factor of $\log n$. It is widely believed that the minimax competitive ratio is $O(\log n)$ in general, but this gap has remained elusive for at least 10 years.

The most significant progress towards $O(\log n)$ is the 2007 work of Bansal et al [12] who achieved such a ratio for the case of "weighted-star metrics". A weighted star is a metric such that each point i has a fixed distance d_i from some "center state", and traveling between any

state i and j requires going through the center, hence incurring a switching cost of $d_i + d_j$. For weighted-star metrics, Bansal et al managed to justify two simplifications which are quite useful:

1. We can assume that the cost vector is of the form $\langle 0, \ldots, \infty, \ldots, 0 \rangle$; that is, all state receive 0 cost, except some state i which receives an infinite cost.

2. When the online algorithm is currently maintaining a distribution $\mathbf{w}$ over the metric, and an infinite cost occurs at state i, we can assume[1] that algorithm incurs exactly $2d_i w_i$, exactly the cost of having w_i probability weight enter and leave i from the center.

Bansal et al provide an efficient algorithm for this setting using primal-dual techniques developed for solving linear programs. With the methods developed in the present chapter, however, we can give the minimax optimal online algorithm under the above simplifications. Notice that the adversary is now choosing a sequence of states $i_1, i_2, i_3 \ldots \in [n]$ at which to assign an infinite cost. If we let $\rho = i_1 i_2 i_3 \ldots$, then the online algorithm's job is to choose a sequence of distributions $\mathbf{w}(\rho_t)$, and pays $2d_{i_{t+1}} w_{i_{t+1}}(\rho_t)$ at each step. In the end, the online algorithm's cost is compared to the offline MTS cost of ρ, which we will call $\text{cost}(\rho)$. Assume[2] we know the cost of the offline in advance, say it's k, and let us define $M = \text{diag}(2d_1, \ldots, 2d_n)$. Then the player's job is to select an algorithm $\mathbf{w}$ which minimizes

$$\max_{\substack{\rho = (i_1, \ldots, i_T) \\ \text{cost}(\rho) \leq k}} \sum_{t=1}^{T} \mathbf{w}(\rho_{t-1})^\top M \mathbf{e}_{i_t}.$$

As we have shown, Algorithm 13 is minimax optimal for this setting. The competitive ratio of this algorithm is precisely $\limsup_{k \to \infty} \left(\frac{1}{k} \text{MinimaxLoss}(M, \text{cost}, k) \right)$. Notice the convenient trick here: by bounding a priori the cost of the offline at k, we can simply imagine playing this repeated game until the budget k is achieved. Then the competitive ratio is just the worst-case loss over the offline cost, k. On the downside, we don't know of any easy way to bound the worst-case loss $\Phi(\emptyset)$.

[1] Precisely, they claim that it should be upper-bounded by $4d_i$. We omit the details regarding this issue, but it only contributes a multiplicative factor of 2 to the competitive ratio.

[2] Even when we do not know the offline cost in advance, standard "doubling tricks" allow you to guess this value and increase the guess as the game proceeds. For space, we omit these details.

Bibliography

[1] J. Abernethy, A. Agarwal, P. Bartlett, and A. Rakhlin, "A stochastic view of optimal regret through minimax duality," in *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

[2] J. Abernethy, P. Bartlett, and E. Hazan, "Blackwell approachability and low-regret learning are equivalent," in *COLT*, 2011.

[3] J. Abernethy, P. Bartlett, A. Rakhlin, and A. Tewari, "Optimal strategies and minimax lower bounds for online convex games," in *Proceedings of the Nineteenth Annual Conference on Computational Learning Theory*. Citeseer, 2008.

[4] J. Abernethy, J. Langford, and M. K. Warmuth, "Continuous experts and the Binning algorithm," in *Proceedings of the 19th Annual Conference on Learning Theory (COLT06)*. Springer, June 2007, pp. 544–558.

[5] J. Abernethy and A. Rakhlin, "Beating the adaptive bandit with high probability," in *COLT*, 2009.

[6] J. Abernethy, M. K. Warmuth, and J. Yellin, "Optimal strategies from random walks," in *Proceedings of the 21st Annual Conference on Learning Theory (COLT 08)*, July 2008, pp. 437–445.

[7] J. Abernethy and M. Warmuth, "Repeated games against budgeted adversaries," *Advances in Neural Information Processing Systems*, vol. 22, 2010.

[8] J. Abernethy, E. Hazan, and A. Rakhlin, "Competing in the dark: An efficient algorithm for bandit linear optimization," in *COLT*, 2008, pp. 263–274.

[9] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multi-armed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, 2003.

[10] B. Awerbuch and R. Kleinberg, "Online linear optimization and adaptive routing," *J. Comput. Syst. Sci.*, vol. 74, no. 1, pp. 97–114, 2008.

[11] B. Awerbuch and R. D. Kleinberg, "Adaptive routing with end-to-end feedback: distributed learning and geometric approaches," in *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 2004, pp. 45–53.

[12] N. Bansal, N. Buchbinder, and J. S. Naor, "A Primal-Dual randomized algorithm for weighted paging," in *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society, 2007, pp. 507–517. [Online]. Available: http://portal.acm.org/citation.cfm?id=1333875.1334222

[13] Y. Bartal, A. Blum, C. Burch, and A. Tomkins, "A polylog (n)-competitive algorithm for metrical task systems," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, 1997, p. 711719.

[14] P. Bartlett, V. Dani, T. Hayes, S. Kakade, A. Rakhlin, and A. Tewari, "High-probability bounds for the regret of bandit online linear optimization," 2008, in submission to COLT 2008.

[15] P. Bartlett, E. Hazan, and A. Rakhlin, "Adaptive online gradient descent," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008.

[16] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, ser. MPS/SIAM Series on Optimization. Philadelphia: SIAM, 2001, vol. 2.

[17] D. Blackwell, "Controlled random walks," in *Proceedings of the International Congress of Mathematicians*, vol. 3, 1954, pp. 336–338.

[18] D. Blackwell, "An analog of the minimax theorem for vector payoffs," *Pacific Journal of Mathematics*, vol. 6, no. 1, 1956.

[19] A. Borodin, N. Linial, and M. E. Saks, "An optimal on-line algorithm for metrical task system," *Journal of the ACM (JACM)*, vol. 39, no. 4, p. 745763, 1992.

[20] B. Brügmann, "Monte carlo go," *Master's Thesis, Unpublished*, 1993.

[21] N. Cesa-Bianchi, A. Conconi, and C. Gentile, "On the generalization ability of on-line learning algorithms," *Information Theory, IEEE Transactions on*, vol. 50, no. 9, pp. 2050 – 2057, sept. 2004.

[22] N. Cesa-Bianchi, Y. Freund, D. P. Helmbold, and M. K. Warmuth, "On-line prediction and conversion strategies," *Machine Learning*, vol. 25, pp. 71–110, 1996.

[23] N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth, "How to use expert advice," *J. ACM*, vol. 44, no. 3, pp. 427–485, 1997.

[24] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[25] T. Cover, "Universal portfolios," *Mathematical Finance*, vol. 1, no. 1, pp. 1–29, January 1991. [Online]. Available: citeseer.ist.psu.edu/article/cover96universal.html

[26] V. Dani, T. Hayes, and S. Kakade, "The price of bandit information for online optimization," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008.

[27] V. Dani and T. P. Hayes, "Robbing the bandit: less regret in online geometric optimization against an adaptive adversary," in *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*. New York, NY, USA: ACM, 2006, pp. 937–943.

[28] A. Dawid, "The well-calibrated Bayesian," *Journal of the American Statistical Association*, vol. 77, pp. 605–613, 1982.

[29] P. Erdos and A. Renyi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci.*, vol. 5A, pp. 17–61, 1960.

[30] E. Even-Dar, R. Kleinberg, S. Mannor, and Y. Mansour, "Online learning for global cost functions," in *22nd Annual Conference on Learning Theory (COLT)*, 2009.

[31] A. Fiat and M. Mendel, "Better algorithms for unfair metrical task systems and applications," in *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, 2000, p. 725734.

[32] P. Flajolet and R. Sedgewick, *Analytic Combinatorics*. Cambridge U. Press, 2008.

[33] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: gradient descent without a gradient," in *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2005, pp. 385–394.

[34] D. P. Foster, "A proof of calibration via blackwell's approachability theorem," *Games and Economic Behavior*, vol. 29, no. 1-2, p. 7378, 1999.

[35] D. P. Foster and R. V. Vohra, "Asymptotic calibration," *Biometrika*, vol. 85, no. 2, p. 379, 1998.

[36] D. Foster, "Prediction in the worst case," *The Annals of Statistics*, pp. 1084–1090, 1991.

[37] Y. Freund and R. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational learning theory*. Springer, 1995, pp. 23–37.

[38] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to Boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997, special Issue for EuroCOLT '95.

[39] D. Fudenberg and D. K. Levine, "An easier way to calibrate* 1," *Games and economic behavior*, vol. 29, no. 1-2, p. 131137, 1999.

[40] S. Gelly and D. Silver, "Combining online and offline knowledge in UCT," in *Proceedings of the 24th international conference on Machine learning*, 2007, p. 280.

[41] S. Gelly, Y. Wang, R. Munos, and O. Teytaud, "Modification of UCT with patterns in Monte-Carlo go," 2006.

[42] A. György, T. Linder, G. Lugosi, and G. Ottucsák, "The on-line shortest path problem under partial monitoring," *Journal of Machine Learning Research*, vol. 8, pp. 2369–2403, 2007.

[43] J. Hannan, "Approximation to Bayes risk in repeated play," *Contributions to the Theory of Games*, vol. 3, pp. 97–139, 1957.

[44] S. Hart and A. Mas-Colell, "A simple adaptive procedure leading to correlated equilibrium," *Econometrica*, vol. 68, no. 5, p. 11271150, 2000.

[45] D. Haussler, J. Kivinen, and M. Warmuth, "Sequential prediction of individual sequences under general loss functions," *Information Theory, IEEE Transactions on*, vol. 44, no. 5, pp. 1906 –1925, sep 1998.

[46] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Machine Learning*, vol. 69, no. 2, pp. 169–192, 2007.

[47] E. Hazan, "The convex optimization approach to regret minimization." in *To appear in Optimization for Machine Learning*. MIT Press, 2010.

[48] E. Hazan, A. Kalai, S. Kale, and A. Agarwal, "Logarithmic regret algorithms for online convex optimization." in *COLT*, 2006, pp. 499–513.

[49] D. Helmbold and M. K. Warmuth, "Learning permutations with exponential weights," in *Proceedings of the 20th Annual Conference on Learning Theory (COLT07)*. Springer, 2007.

[50] N. Karmarkar, "New polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, pp. 373–395, 1984.

[51] J. Kivinen and M. K. Warmuth, "Averaging expert predictions," in *Computational Learning Theory, 4th European Conference, EuroCOLT '99, Nordkirchen, Germany, March 29-31, 1999, Proceedings*, ser. Lecture Notes in Artificial Intelligence, vol. 1572. Springer, 1999, pp. 153–167.

[52] N. Littlestone, "Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm," *Machine learning*, vol. 2, no. 4, pp. 285–318, 1988.

[53] N. Littlestone and M. K. Warmuth, "The Weighted Majority algorithm," *Inform. Comput.*, vol. 108, no. 2, pp. 212–261, 1994, preliminary version in FOCS 89.

[54] S. Mannor and N. Shimkin, "Regret minimization in repeated matrix games with variable stage duration," *Games and Economic Behavior*, vol. 63, no. 1, pp. 227–258, 2008.

[55] S. Mannor and G. Stoltz, "A Geometric Proof of Calibration," *arXiv*, Dec 2009. [Online]. Available: http://arxiv.org/abs/0912.3604

[56] H. B. McMahan and A. Blum, "Online geometric optimization in the bandit setting against an adaptive adversary," in *COLT*, 2004, pp. 109–123.

[57] N. Merhav and M. Feder, "Universal schemes for sequential decision from individual data sequences," *Information Theory, IEEE Transactions on*, vol. 39, no. 4, pp. 1280 –1292, jul 1993.

[58] N. Merhav and M. Feder, "Universal prediction," *Information Theory, IEEE Transactions on*, vol. 44, no. 6, pp. 2124–2147, 1998.

[59] A. Myers and H. S. Wilf, "Some new aspects of the Coupon-Collector's problem," *SIAM J. Disc. Math.*, vol. 17, pp. 1–17, 2003.

[60] A. Nemirovski and M. Todd, "Interior-point methods for optimization," *Acta Numerica*, pp. 191—234, 2008.

[61] A. Nemirovskii, "Interior point polynomial time methods in convex programming," 2004, lecture Notes.

[62] Y. E. Nesterov and A. S. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming*. Philadelphia: SIAM, 1994.

[63] J. V. Neumann, O. Morgenstern, H. W. Kuhn, and A. Rubinstein, *Theory of games and economic behavior*. Princeton university press Princeton, NJ, 1947.

[64] D. Newman and L. Shepp, "The Double Dixie Cup problem," *Amer. Math Monthly.*, vol. 67, pp. 541–574, 1960.

[65] E. Ordentlich and T. M. Cover, "The cost of achieving the best portfolio in hindsight," *Math. Oper. Res.*, vol. 23, no. 4, pp. 960–982, 1998.

[66] V. Perchet, "Calibration and internal no-regret with random signals," in *Proceedings of the 20th international conference on Algorithmic learning theory*. Springer-Verlag, 2009, pp. 68–82.

[67] A. Rakhlin, K. Sridharan, and A. Tewari, "Online Learning: Beyond Regret," *Arxiv preprint arXiv:1011.3168*, 2010.

[68] A. Sandroni, R. Smorodinsky, and R. Vohra, "Calibration with many checking rules," *Mathematics of Operations Research*, vol. 28, no. 1, pp. 141–153, 2003.

[69] S. Shalev-Shwartz and Y. Singer, "Convex repeated games and Fenchel duality," *Advances in Neural Information Processing Systems*, vol. 19, p. 1265, 2007.

[70] S. Shalev-Shwartz and Y. Singer, "A primal-dual perspective of online learning algorithms," *Mach. Learn.*, vol. 69, no. 2-3, pp. 115–142, 2007.

[71] M. Sion, "On general minimax theorems," *Pacific J. Math*, vol. 8, no. 1, pp. 171–176, 1958.

[72] E. Takimoto and M. K. Warmuth, "The minimax strategy for gaussian density estimation. pp," in *COLT '00: Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 100–106.

[73] V. Vovk, "Aggregating strategies," in *Proceedings of the Third Annual Workshop on Computational Learning Theory*. Morgan Kaufmann, 1990, pp. 371–383.

[74] V. Vovk, "Competitive on-line linear regression," in *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*. Cambridge, MA, USA: MIT Press, 1998, pp. 364–370.

[75] V. Vovk, "A game of prediction with expert advice," *J. of Comput. Syst. Sci.*, vol. 56, no. 2, pp. 153–173, 1998, special Issue: Eighth Annual Conference on Computational Learning Theory.

[76] M. K. Warmuth and D. Kuzmin, "Randomized PCA algorithms with regret bounds that are logarithmic in the dimension," in *Advances in Neural Information Processing Systems 19 (NIPS 06)*. MIT Press, December 2006.

[77] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *International Conference on Machine Learning*, vol. 20, 2003, p. 928.

[78] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent." in *ICML*, 2003, pp. 928–936.