Online Decision Problems with Large Strategy Sets

by

Robert David Kleinberg

B.A., Cornell University, 1997

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2005

Author .....
Department of Mathematics
May 12, 2005

Certified by .....
F. Thomson Leighton
Professor of Mathematics
Thesis Supervisor

Accepted by .....
Rodolfo Ruben Rosales
Chairman, Applied Mathematics Committee

Accepted by .....
Pavel I. Etingof
Chairman, Department Committee on Graduate Students

Online Decision Problems with Large Strategy Sets

by

Robert David Kleinberg

Submitted to the Department of Mathematics
on May 12, 2005, in partial fulfillment of the
requirements for the degree of
DOCTOR OF PHILOSOPHY

Abstract

In an online decision problem, an algorithm performs a sequence of trials, each of which involves selecting one element from a fixed set of alternatives (the "strategy set") whose costs vary over time. After T trials, the combined cost of the algorithm's choices is compared with that of the single strategy whose combined cost is minimum. Their difference is called *regret*, and one seeks algorithms which are efficient in that their regret is sublinear in T and polynomial in the problem size.

We study an important class of online decision problems called *generalized multi-armed bandit problems.* In the past such problems have found applications in areas as diverse as statistics, computer science, economic theory, and medical decision-making. Most existing algorithms were efficient only in the case of a small (i.e. polynomial-sized) strategy set. We extend the theory by supplying non-trivial algorithms and lower bounds for cases in which the strategy set is much larger (exponential or infinite) and the cost function class is structured, e.g. by constraining the cost functions to be linear or convex. As applications, we consider adaptive routing in networks, adaptive pricing in electronic markets, and collaborative decision-making by untrusting peers in a dynamic environment.

Thesis Supervisor: F. Thomson Leighton
Title: Professor of Mathematics

Acknowledgments

I have frequently heard people compare the process of getting an MIT education with the process of trying to drink from a fire hose. In my own career as a graduate student, no experience illustrates this analogy more clearly than my interactions with my advisor, Tom Leighton. It was not unusual for me to spend the better part of a week unraveling the new ideas which Tom came up with during a single two-hour meeting. Tom's greatest gift as an advisor is not his deftness in devising and communicating these ideas, but his knack for doing it in a way which is enlivening rather than ego-shattering for the student. I thank Tom for instilling in me the confidence to do research on my own, for never allowing me to settle for doing less than the best job I was capable of, and for being a thoughtful advisor and friend.

I am fortunate to have grown up in a family of gifted educators, and it is impossible to overstate their influence on my development as a scholar. I thank my parents for sharing their tremendous love of life and fostering my intellectual curiosity from day one, and more recently for helping me through my darker moods while in graduate school and for showing much more interest in my math research than any grad student has a right to expect. Along with my parents, my brother Jon and Lillian Lee have been a source of indispensible support, encouragement, and humor. Jon has been teaching me throughout my life, and he had a particularly strong impact on this most recent stage of my career. His beautiful perspective on mathematics and computer science has been an inspiration to me for many years. I thank him for his wise, patient, and gentle guidance as I progressed through graduate school and into the next phase of my life.

It has been a pleasure to learn from many wonderful mathematicians and computer scientists during graduate school, including Baruch Awerbuch (whose ideas influenced much of the work in this thesis), Christian Borgs, Jennifer Chayes, Henry Cohn, Michel Goemans, Mike Hopkins, László Lovász, Assaf Naor, David Parkes, Rajmohan Rajaraman, Madhu Sudan, Ravi Sundaram, Balázs Szegedy, Chris Umans, and Santosh Vempala. It is both inspiring and humbling to work with such talented mentors and colleagues.

Working at Akamai Technologies from 1999 to 2002 was one of the most thrilling and transformative experiences of my life. I thank Arthur Berger, Avi and Noam Freedman, Rahul Hariharan, Leonidas Kontothanassis, Danny Lewin, Bruce Maggs, Daniel Stodolsky, Marcelo Torres, Joel Wein, Neal Young, and the many other people at Akamai who made my time there so memorable and who played a strong role in shaping my appreciation of computer science.

My years as a graduate student have been sweetened by the companionship of my friends James Kaplan and Erica Streit-Kaplan, David Brandt and Amy Vigorita,

Jeremy Bem, Andras Ferencz, Dan Murray, and Irene Perciali, and my fellow students Pramod Achar, Daniel Biss, Jesper Grodal, Nick Harvey, Tara Holm, Adam Klivans, April and Eric Lehman, Lenny Ng, and Carmen Young. I also thank Kathleen Dickey and Linda Okun for their help and friendship over the past few years.

The work in this thesis was supported by a Fannie and John Hertz Foundation Fellowship. I thank the Hertz Foundation not only for its generous financial support, but also for encouraging the unfettered pursuit of my intellectual interests.

Finally, I thank my wife, Miranda Phillips, for keeping me afloat through some of my most stormy periods, for being a true friend, for making me laugh, and for helping with all of my most important decision problems.

Contents

List of Figures

Chapter 1

Introduction

How should a decision-maker perform repeated choices so as to optimize the average cost or benefit of those choices in the long run? This thesis concerns a mathematical model for analyzing such questions, by situating them in the framework of *online decision problems*. An online decision problem involves an agent performing a series of trials, each of which requires choosing one element from a fixed set of alternatives, in a time-varying environment which determines the cost or benefit of the chosen alternative. The agent lacks information about the future evolution of the environment, and potentially about its past evolution as well. This uncertainty, rather than limitations on computational resources such as space and processing time, is typically the main source of difficulty in an online decision problem.

Most of the online decision problems we consider here are generalizations of the *multi-armed bandit problem*, a problem which has been studied extensively in statistics and related fields, and more recently in theoretical machine learning. The problem is most often motivated in terms of a gambling scenario, which also explains the derivation of the name "multi-armed bandit." (A slot machine is sometimes whimsically called a "one-armed bandit.") Imagine a casino with K slot machines. Each of these machines, when operated, produces a random payoff by sampling from a probability distribution which does not vary over time, but which may differ from one slot machine to the next. A gambler, who does not know the payoff distribution of each machine, performs a series of trials, each of which consists of operating one slot machine and observing its payoff. How should the gambler decide which slot machine to choose in each trial, so as to maximize his or her expected payoff?

While gambling supplies the clearest metaphor for stating the multi-armed bandit problem, the original motivation for studying such problems came from medicine, specifically the design of clinical trials [18]. Here, the decision-maker is an experimenter who administers one of K experimental treatments sequentially to a pop-

ulation of patients. How should one decide which treatment to administer to each patient, so as to maximize the expected benefit to the population, given that the efficacy of each treatment is a random variable whose distribution is initially unknown?

The problems introduced above assume that the decision-maker has a fixed, finite set of K alternatives in each trial. But if we consider the problem of designing clinical trials more closely, it is clear that many of the natural questions involve an underlying decision-making problem with a very large (possibly even infinite) set of alternatives. For example, suppose that the question facing the experimenter in each trial is not *which* treatment to prescribe, but *what dosage* to prescribe. In this case, the underlying set of alternatives is more accurately modeled as a one-parameter interval rather than a discrete, finite set. One can easily imagine multi-parameter decision problems arising in this context as well, e.g. if the experimenter is administering a treatment which is a mixture of several ingredients whose dosages may be adjusted independently, or if there are other variables (frequency of treatment, time of day) which may influence the outcome of the treatment.

Online decision problems with large strategy sets arise naturally in many other contexts as well. The work in this thesis was motivated by applications in economic theory, electronic commerce, network routing, and collaborative decision systems. We will say much more about these applications in subsequent chapters.

The preceding examples illustrate that there are compelling reasons to study generalized multi-armed bandit problems in which the decision-maker has a large (potentially even infinite) set of strategies. In this work, we will develop the theory of such decision problems by supplying new algorithms and lower bound techniques, in some cases providing exponential improvements on the best previously-known bounds. The theoretical portion of this thesis is organized into three parts:

Online optimization in one-parameter spaces: We study online decision problems with a one-parameter strategy space, in which the cost or benefit of a strategy is a Lipschitz function of the control parameter.

Online optimization in vector spaces: We study online decision problems with a multi-parameter strategy space, in which the cost of benefit of a strategy is a linear or convex function of the control parameters.

Online optimization in measure spaces: Here we assume no special structure for the strategy set or the cost functions, only that the set of strategies forms a measure space of total measure 1, and that the cost functions are measurable functions.

In parallel with these theoretical contributions, we will illustrate the usefulness of our techniques via three concrete applications:

Adaptive pricing strategies: We study mechanisms for setting prices in a sequence of transactions with different buyers, so as to maximize the seller's expected profit.

Online routing: We study online algorithms for choosing a sequence of routing paths between a designated pair of terminals in a network with time-varying edge delays, so as to minimize the average delay of the chosen paths.

Collaborative learning: We study algorithms for a community of agents, each participating in an online decision problem, to pool information and learn from each other's mistakes in spite of the presence of malicious (Byzantine) participants as well as differences in taste.

The rest of the introduction is organized as follows. In Section 1.1 we present a precise mathematical formulation of online decision problems. We review some of the prior literature on multi-armed bandit problems and other online decision problems in Sections 1.2 and 1.3, followed by an exposition of our main theoretical results in Section 1.4. Turning to the applications, we discuss adaptive pricing in Section 1.5, online routing in Section 1.6, and collaborative learning in Section 1.7.

1.1 Problem formulation

1.1.1 Basic definitions

Definition 1.1 (Online decision domain). An *online decision domain* is an ordered pair $(\mathcal{S}, \Gamma)$ where $\mathcal{S}$ is a set and Γ is a class of functions mapping $\mathcal{S}$ to $\mathbb{R}$. We refer to elements of $\mathcal{S}$ as *strategies* and elements of Γ as *cost functions*.

Definition 1.2 (Feedback model, full feedback, opaque feedback). A *feedback model* for an online decision domain $(\mathcal{S}, \Gamma)$ consists of a set Φ and a function $F : \mathcal{S} \times \Gamma \to \Phi$. We refer to elements of Φ as *feedback values*, and we refer to $F(x, c)$ as the feedback received when playing strategy x against cost function c.

For any online decision domain $(\mathcal{S}, \Gamma)$, the *full feedback* model is defined by setting $\Phi = \Gamma$ and $F(x, c) = c$. The *opaque feedback* model is defined by setting $\Phi = \mathbb{R}$ and $F(x, c) = c(x)$.

Definition 1.3 (Online decision problem). An online decision problem is a quadruple $\Pi = (\mathcal{S}, \Gamma, \Phi, F)$ where $(\mathcal{S}, \Gamma)$ is an online decision domain and (Φ, F) is a feedback model for $(\mathcal{S}, \Gamma)$.

Classically, online decision problems were studied for the online decision domain $(\mathcal{S}, [0,1]^{\mathcal{S}})$ where $\mathcal{S}$ is a finite set and $\Gamma = [0,1]^{\mathcal{S}}$ is the set of all mappings from $\mathcal{S}$ to $[0,1]$. For the full feedback model, the resulting online decision problem is commonly known as the "best-expert" problem, owing to the metaphor of choosing an alternative based on expert advice. For the opaque feedback model, the resulting online decision problem is commonly known as the "multi-armed bandit" problem, owing to the metaphor of choosing a slot machine in a casino.

Definition 1.4 (Generalized best-expert, generalized multi-armed bandit). An online decision problem $\Pi = (\mathcal{S}, \Gamma, \Phi, F)$ is called the *generalized best-expert problem* for $(\mathcal{S}, \Gamma)$ if (Φ, F) is the full feedback model for $(\mathcal{S}, \Gamma)$. It is called the *generalized multi-armed bandit problem* for $(\mathcal{S}, \Gamma)$ if (Φ, F) is the opaque feedback model for $(\mathcal{S}, \Gamma)$.

Definition 1.5 (Algorithm). An *algorithm* for an online decision problem $\Pi = (\mathcal{S}, \Gamma, \Phi, F)$ consists of a probability space Ω_{alg} and a sequence of functions $X_t : \Omega_{\text{alg}} \times \Phi^{t-1} \to \mathcal{S}$ for $t = 1, 2, \ldots$. We interpret $X_t(r, y_1, \ldots, y_{t-1}) = x$ to mean that the algorithm chooses strategy x at time t if its random seed is r and the feedback values for trials $1, 2, \ldots, t-1$ are $y_1, \ldots, y_{t-1}$, respectively.

Note that Definition 1.5 doesn't place any limitations on the computational resources which the algorithm may use in computing the functions X_t. However, all of the algorithms introduced in this thesis will be computationally efficient, in that they require only polynomial computation time per trial.

Definition 1.6 (Adversary, oblivious adversary, i.i.d. adversary). An *adversary* for an online decision problem $\Pi = (\mathcal{S}, \Gamma, \Phi, F)$ consists of a probability space Ω_{adv} and a sequence of functions $C_t : \Omega_{\text{adv}} \times \mathcal{S}^{t-1} \to \Gamma$ for $t = 1, 2, \ldots$. We interpret $C_t(r', x_1, \ldots, x_{t-1}) = c$ to mean that the adversary chooses cost function c at time t if its random seed is r' and the algorithm has played strategies $x_1, \ldots, x_{t-1}$ in trials $1, \ldots, t-1$, respectively.

A *deterministic oblivious adversary* is an adversary such that each function C_t is a constant function mapping $\Omega_{\text{adv}} \times \mathcal{S}^{t-1}$ to some element $c_t \in \Gamma$. A *randomized oblivious adversary* is an adversary such that for all t, the value of $C_t(r', x_1, \ldots, x_{t-1})$ depends only on r', i.e. C_t is a random variable on Ω_{adv} taking values in Γ. An *i.i.d. adversary* is a randomized oblivious adversary such that the random variables $C_1, C_2, \ldots$ are independent and identically distributed.

Definition 1.7 (Transcript of play). If ALG and ADV are an algorithm and adversary for an online decision problem Π then we may define a probability space $\Omega = \Omega_{\text{alg}} \times \Omega_{\text{adv}}$ and sequences of random variables $(x_t), (c_t), (y_t)$ $(1 \leq t < \infty)$ on Ω

representing the strategies, cost functions, and feedback values, respectively, that are selected given the random seeds used by the algorithm and adversary. These random variables are defined recursively according to the formulae:

$$\begin{aligned} x_t(r, r') &= X_t(r, y_1(r, r'), \ldots, y_{t-1}(r, r')) \\ c_t(r, r') &= C_t(r', x_1(r, r'), \ldots, x_{t-1}(r, r')) \\ y_t(r, r') &= F(x_t(r, r'), c_t(r, r')). \end{aligned}$$

We refer to the probability space Ω and the random variables $(x_t), (c_t), (y_t)$ $(1 \leq t < \infty)$ collectively as the *transcript of play* for ALG and ADV.

1.1.2 Defining regret

Definition 1.8 (Regret, convergence time). Given an algorithm ALG and adversary ADV for an online decision problem $(\mathcal{S}, \Gamma, \Phi, F)$, a strategy $x \in \mathcal{S}$, and a positive integer T, the *regret* of ALG relative to x is defined by:

$$R(\mathsf{ALG}, \mathsf{ADV}; x, T) = \mathbf{E}\left[\sum_{t=1}^{T} c_t(x_t) - c_t(x)\right].$$

If $\mathcal{A}$ is a set of adversaries, the regret of ALG against ADV is defined by:

$$R(\mathsf{ALG}, \mathcal{A}; T) = \sup\{R(\mathsf{ALG}, \mathsf{ADV}; x, T) \,:\, \mathsf{ADV} \in \mathcal{A},\, x \in \mathcal{S}\}$$

and the *normalized regret* is defined by

$$\overline{R}(\mathsf{ALG}, \mathcal{A}; T) = \frac{1}{T} R(\mathsf{ALG}, \mathcal{A}; T).$$

If τ is a function from $(0, \infty)$ to $\mathbb{N}$, we say that ALG has *convergence time* τ if there exists a constant C such that for all $\delta > 0$ and all $T > \tau(\delta)$,

$$\overline{R}(\mathsf{ALG}, \mathcal{A}; T) < C\delta.$$

Definition 1.9 (Ex post regret). Given an algorithm ALG and a set of adversaries $\mathcal{A}$ for an online decision problem $(\mathcal{S}, \Gamma, \Phi, F)$, and given a positive integer T, the *ex post regret* of ALG against $\mathcal{A}$ at time T is defined by

$$\widehat{R}(\mathsf{ALG}, \mathcal{A}; T) = \sup_{\mathsf{ADV} \in \mathcal{A}} \left\{ \mathbf{E}\left[\sup_{x \in \mathcal{S}} \sum_{t=1}^{T} c_t(x_t) - c_t(x)\right]\right\}.$$

When we wish to distinguish the notion of regret defined in Definition 1.8 from the ex post regret, we will refer to the former as *ex ante regret*.

Note that the two definitions of regret differ by interchanging the $\sup_{x \in \mathcal{S}}$ with the $\mathbf{E}[\cdot]$ operator. Hence it is always the case that

$$\widehat{R}(\mathsf{ALG}, \mathcal{A}; T) \geq R(\mathsf{ALG}, \mathcal{A}; T),$$

with equality when $\mathcal{A}$ consists of deterministic oblivious adversaries.

1.1.3 Maximization versus minimization

In defining online decision problems, and particularly in formulating the definition of regret, we have implicitly assumed that the underlying optimization problem is a minimization problem rather than a maximization problem. In many applications, it is most natural to formulate the objective as a maximization problem. For example, we will consider an adaptive pricing problem where the objective is to maximize revenue.

There is quite often a close correspondence between the maximization and minimization versions of an online decision problem, so that the difference between maximization and minimization becomes immaterial from the standpoint of formulating definitions and theorems about algorithms for such problems. The details of this correspondence are usually fairly obvious; our goal in this section is to make these details explicit for the sake of mathematical precision.

Definition 1.10 (Maximization version of an online decision problem). If $(\mathcal{S}, \Gamma, \Phi, F)$ is an online decision problem, the *maximization version* of $(\mathcal{S}, \Gamma, \Phi, F)$ has exactly the same notions of "algorithm", "adversary", and "transcript of play". However, we redefine "regret" and "ex post regret" as follows.

$$\begin{aligned} R(\mathsf{ALG}, \mathsf{ADV}; x, T) &= \mathbf{E}\left[\sum_{t=1}^{T} c_t(x) - c_t(x_t)\right] \\ R(\mathsf{ALG}, \mathcal{A}; T) &= \sup\{R(\mathsf{ALG}, \mathsf{ADV}; x, T) : \mathsf{ADV} \in \mathcal{A},\, x \in \mathcal{S}\} \\ \widehat{R}(\mathsf{ALG}, \mathcal{A}; T) &= \sup_{\mathsf{ADV} \in \mathcal{A}} \left\{ \mathbf{E}\left[\sup_{x \in \mathcal{S}} \sum_{t=1}^{T} c_t(x) - c_t(x_t)\right] \right\} \end{aligned}$$

Definition 1.11 (Mirror image, mirror symmetry). If $\Pi_1 = (\mathcal{S}, \Gamma, \Phi, F)$ and $\Pi_2 = (\mathcal{S}, \Gamma', \Phi', F')$ are two online decision problems with the same strategy set, we say that Π_2 is a *mirror image* of Π_1 if there exists a constant C and one-to-one correspondences $\tau : \Gamma \to \Gamma'$, $\sigma : \Phi \to \Phi'$, such that

$$\begin{aligned} \forall c \in \Gamma \qquad & c + \tau(c) = C \\ \forall x \in \mathcal{S}, c \in \Gamma \qquad & \sigma(F(x, c)) = F'(x, \tau(c)). \end{aligned}$$

If $\mathcal{A}$ is a set of adversaries for Π_1, then $\tau(\mathcal{A})$ is the following set of adversaries for Π_2:

$$\tau(\mathcal{A}) = \{(\Omega_{\text{adv}}, \tau(C_1), \tau(C_2), \ldots) \ : \ (\Omega_{\text{adv}}, C_1, C_2, \ldots) \in \mathcal{A}\}.$$

We say that Π_1 has *mirror symmetry* if it is a mirror image of itself. If Π_1 has mirror symmetry and $\mathcal{A}$ is a set of adversaries for Π_1, we say $\mathcal{A}$ has *mirror symmetry* if $\tau(\mathcal{A}) = \mathcal{A}$.

Lemma 1.1. *If Π_1 and Π_2 are online decision problems such that Π_2 is a mirror image of Π_1 via mappings τ, σ, and if ALG is an algorithm for Π_1 achieving regret $R(T)$ against a set of adversaries $\mathcal{A}$, then there exists an algorithm ALG' for the maximization version of Π_2 achieving regret $R(T)$ against adversary set $\tau(\mathcal{A})$.*

Proof. Algorithm ALG' operates as follows. It initializes an instance of ALG and plays the strategies x_t selected by that algorithm. When it receives a feedback value y_t, it passes the feedback value $\sigma^{-1}(y_t)$ on to ALG. The verification that

$$R(\mathsf{ALG}', \tau(\mathsf{ADV}); x, T) = R(\mathsf{ALG}, \mathsf{ADV}; x, T)$$

is straightforward. □

Corollary 1.2. *If an online decision problem Π and an adversary set $\mathcal{A}$ have mirror symmetry, and if there exists an algorithm ALG for Π achieving regret $R(T)$ against $\mathcal{A}$, then there exists an algorithm ALG' for the maximization version of Π achieving regret $R(T)$ against $\mathcal{A}$.*

Many of the online decision problems and adversary sets studied in this work have mirror symmetry. For example, if $\mathcal{S}$ is any set and $I \subseteq \mathbb{R}$ is either $\mathbb{R}$ or a bounded subinterval of $\mathbb{R}$, then the best-expert and multi-armed bandit problems for $(\mathcal{S}, I^{\mathcal{S}})$ have mirror symmetry, as do the sets of adaptive, deterministic oblivious, randomized oblivious, and i.i.d. adversaries for these problems.

1.2 Prior work on multi-armed bandit problems

The earliest work on the multi-armed bandit problem assumed that the strategy set was finite and that the sequence of cost functions came from a prior distribution known to the decision-maker. In other words, one assumed a randomized oblivious adversary whose cost functions C_t are random variables whose joint distribution is known to the algorithm designer. (For a concrete example, it may be known that there are two slot machines, one of which always produces a payoff of 0.5, and the other of which generates $\{0, 1\}$-valued payoffs according to independent Bernoulli trials with

success probability p, where p is a random parameter uniformly distributed in $[0, 1]$.) The goal of this work was to characterize the optimal algorithm, i.e. the algorithm which precisely minimizes the expected cost or maximizes the expected payoff.

The most influential result in this area is a theorem of Gittins and Jones [35] on the existence of dynamic allocation indices for multi-armed bandit problems with geometric time-discounting. To state the theorem, let us make the following definitions.

Definition 1.12 (geometrically discounted adversary). Given an i.i.d. adversary ADV specified by probability space Ω_{adv} and cost functions C_t $(1 \leq t < \infty)$, and given a positive constant $\alpha < 1$, let ADV_α denote the adversary specified by probability space Ω_{adv} and cost functions $\widehat{C}_t = \alpha^t C_t$. We say that ADV_α is a *geometrically discounted* adversary with *discount factor* α. A *geometrically discounted adversary ensemble* with discount factor α is a probability distribution P on the set of geometrically discounted adversaries with discount factor α. An *optimal algorithm* for P is an algorithm which minimizes $\lim_{T\to\infty} \mathbf{E}_{\mathsf{ADV}\leftarrow P}[R(\mathsf{ALG}, \mathsf{ADV}; T)]$ where the expectation is over the choice of a random adversary ADV sampled from P.

Definition 1.13 (dynamic allocation index, index policy). For a multi-armed bandit problem with strategy set $\mathcal{S}$, a *dynamic allocation index* is a rule for assigning a real-valued score to each strategy $x \in \mathcal{S}$ depending only on the feedback observed in past trials when x was selected. More formally, it is a sequence of mappings $I_t : \mathcal{S} \times \mathcal{S}^t \times \Phi^t \to \mathbb{R}$ for $1 \leq t < \infty$, such that $I_t(x, \vec{x}, \vec{y}) = I_t(x, \vec{x}\,', \vec{y}\,')$ if $x_i = x'_i$ and $y_i = y'_i$ for all i such that $x_i = x$. We interpret $I_t(x, \vec{x}, \vec{y})$ as the score assigned to strategy x after trial t, if the strategies and feedbacks in trials $1, 2, \ldots, t$ are designated by $\vec{x}, \vec{y}$, respectively.

An algorithm ALG is an *index policy* if there exists a dynamic allocation index $\{I_t\}$ such that ALG always chooses the strategy of maximum index, i.e. for every adversary ADV, the transcript of play for $\mathsf{ALG}, \mathsf{ADV}$ satisfies

$$x_{t+1} = \arg\max_{x\in\mathcal{S}} I_t(x, (x_1, \ldots, x_t), (y_1, \ldots, y_t))$$

for all $t \geq 1$.

Theorem 1.3 (Gittins-Jones Index Theorem [35]). *Given a multi-armed bandit problem with a geometrically discounted adversary ensemble, the optimal algorithm is an index policy.*

Subsequent authors found simpler proofs [69, 70, 71] and studied methods for calculating dynamic allocation indices [34, 47]. A partial converse to the Gittins-Jones Index Theorem is proved in [17]: if the discount sequence is not geometric, there exist adversaries for which the optimal algorithm is *not* an index policy. A

very entertaining and lucid exposition of some of these topics can be found in [28], which explains Gittins index policies in terms of the metaphor of playing golf with more than one ball. (In this context the dynamic allocation index assigns a score to each golf ball which quantifies the desirability of hitting that ball, given its current position on the golf course.)

The work discussed above assumed a randomized oblivious adversary with a *known* prior distribution. For an i.i.d. adversary with an unknown distribution, Lai and Robbins [52] determined that the optimal multi-armed bandit algorithm has regret $\theta(\log T)$ as $T \to \infty$. Auer, Cesa-Bianchi, and Fischer [3] sharpened this asymptotic result by supplying a precise upper bound that holds for finite T. (Interestingly, although the input distribution is unknown and the adversary is not geometrically discounted, the algorithms achieving this upper bound in [3] are index policies.) A treatment of Auer, Cesa-Bianchi, and Fischer's upper bound is presented in Section 2.4 of this thesis. Mannor and Tsitsiklis [56] similarly sharpened the Lai-Robbins asymptotic *lower* bound by supplying a precise lower bound that holds for finite T.

Bandit problems with an infinite number of arms have received much less consideration in the literature. Banks and Sundaram [13] study properties of Gittins index policies for countable-armed bandit problems with geometrically discounted adversaries. For a very large class of such problems, they demonstrate several counterintuitive facts about the optimal algorithm:

- When a strategy x is chosen, there is a positive probability that x will be chosen forever thereafter. In particular, there is a positive probability that the algorithm only samples one strategy during the entire infinite sequence of trials.

- For some problem instances, the strategy which is optimal (under the adversary's cost function distribution) is discarded by the algorithm in finite time, with probability 1.

An excellent survey of these and related results appears in [67]. Berry *et. al.* [16] consider a bandit problem with a countable set of arms and $\{0,1\}$-valued reward functions; the objective is to maximize the long-run success proportion, i.e. the quantity $\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} c_t(x_t)$.

For multi-armed bandit problems with an uncountable strategy set, the only prior work we are aware of is by R. Agrawal [2], who introduced the continuum-armed bandit problem. In Agrawal's formulation of the problem, the strategy set $\mathcal{S}$ is a subinterval of $\mathbb{R}$ and the adversary is an i.i.d. adversary who samples from a probability distribution on cost functions $c : \mathcal{S} \to \mathbb{R}$ such that the expected cost $\bar{c}(x)$ is uniformly locally Lipschitz with exponent $\alpha > 0$. Agrawal presents an algorithm whose regret is $o\left(T^{(2\alpha+1)/(3\alpha+1)+\varepsilon}\right)$ for all $\varepsilon > 0$.

The study of multi-armed bandit algorithms for stronger adversarial models was initiated by Auer *et al.* in [4]. They presented a randomized algorithm Exp3 which achieves regret $O(\sqrt{TK\log(K)})$ against adaptive adversaries. They also supplied a lower bound which proves that no algorithm can achieve regret $o(\sqrt{TK})$, even against oblivious adversaries.

1.3 Prior work on other online decision problems

The work reviewed in Section 1.2 exclusively concerned multi-armed bandit problems, i.e. online decision problems in the opaque feedback model. There is also a rich body of literature on online decision problems in the full feedback model, e.g. the so-called "best expert" problem and its generalizations. (See [20] for an excellent survey of this literature.) Work in this area began with the study of mistake-bound learning and online prediction problems, in which a learning algorithm is required to repeatedly predict the next element in a sequence of labels, given access to a set of K "experts" who predict the next label in the sequence before it is revealed. An archetypical example of such a problem is discussed in Section 2.1: the input is a binary sequence $b_1, \ldots, b_T$ and each expert i makes a sequence of predictions $b_1(i), \ldots, b_T(i)$. At time t, the predictions $b_t(i)$ are revealed for each expert, then the algorithm predicts the value of b_t, then the true value of b_t is revealed. The goal is to make nearly as few mistakes as the best expert. To understand the relevance of this problem to machine learning, imagine that we are solving a binary classification problem in which each element of a set $\mathcal{X}$ is given a label in $\{0, 1\}$, and that we are trying to perform nearly as well as the best classifier in some set $\mathcal{C}$ of classifiers. The algorithm is presented with a sequence of elements $x_t \in X$ and can compute the output of each classifier i on example x_t (i.e. the prediction $b_t(i)$); it must then predict the label of x_t before the true label is revealed.

In Section 2.1 we will present two closely related algorithms for this binary prediction problem. The first, due to Littlestone and Warmuth [54], is called the *weighted majority* algorithm, while the second, due to Freund and Schapire [33], is called Hedge. Both are based on the principle of maintaining a weight for each expert which decays exponentially according to the number of mistakes the expert has made in the past. These techniques yield an algorithm which makes at most $O(\sqrt{T\log(K)})$ more mistakes than the best expert.

The best-expert problem is closely related to the binary prediction problem. Suppose that instead of assigning a cost of 0 or 1 to each expert at time t according to whether or not it correctly predicted a binary label b_t, we simply have a sequence of cost functions c_t ($t = 1, 2, \ldots, T$) which assign real-valued scores between 0 and 1 to

each expert. We now require the algorithm to specify an expert x_t at time t; after the algorithm chooses x_t, the cost function c_t is revealed and the algorithm is charged a cost of $c_t(x_t)$. The reader will recognize this as the full-feedback online decision problem with $\mathcal{S} = \{1, 2, \ldots, K\}$ and $\Gamma = [0,1]^{\mathcal{S}}$. It turns out that the Hedge algorithm can also be applied to this problem, again with regret $O(\sqrt{T \log(K)})$. Using the terminology of Section 1.1, we can express this by saying that Hedge has convergence time $O(\log K)$.

Because the best-expert problem has algorithms with convergence time $O(\log K)$, it is possible to achieve rapid (i.e. polynomial) convergence time even when the size of the strategy set is exponential in the problem size. (For a concrete example, consider the problem of choosing a route to drive to work each day. Here, the number of strategies — i.e. paths from home to work — may be exponential in the size of the network of roads.) However, a naive implementation of Hedge with an exponential-sized set of experts would require exponential computation time. In many cases, this difficulty can be circumvented using other algorithms. For example, when the set of strategies, $\mathcal{S}$, can be embedded in a low-dimensional vector space in such a way that the cost functions are represented by linear functions, then there is an elegant algorithm for the generalized best-expert problem, originally due to Hannan [39], which achieves polynomial convergence time and polynomial computation time, assuming there is a polynomial-time algorithm to optimize linear functions over $\mathcal{S}$ (as is the case, for example, when $\mathcal{S}$ is the solution set of a polynomial-sized linear program). This algorithm was rediscovered and generalized by Kalai and Vempala [44]. When $\mathcal{S}$ is a convex set and the cost functions are convex, an algorithm called *Greedy Projection*, due to Zinkevich [75], solves the generalized best-expert problem with polynomial convergence time and polynomial computation time, assuming there is a polynomial-time algorithm which computes, for any point x in Euclidean space, the point of $\mathcal{S}$ which is closest to x in the Euclidean metric. The algorithms of Kalai-Vempala and Zinkevich will be presented and analyzed in Sections 2.2 and 2.3.

1.4 Our contributions

Our contributions to the theory of online decision problems involve studying generalized multi-armed bandit problems at three progressively greater levels of generality.

- Chapter 4 concerns generalized bandit problems in which the strategy set is a bounded interval. Such problems are called *continuum-armed bandit problems*.

- Generalizing from a one-dimensional to a d-dimensional strategy set, in Chapter 5 we consider generalized bandit problems in vector spaces.

- Generalizing still further, Chapter 6 concerns generalized bandit problems in measure spaces.

Below, we give a detailed explanation of our results in each of these areas.

1.4.1 Continuum-armed bandit problems

Recall from Section 1.2 that there is an exponential gap between the best known bounds for the continuum-armed bandit problem and those for the K-armed bandit problem against i.i.d. adversaries. More specifically, for i.i.d. adversaries the best K-armed bandit algorithms have regret $\theta(\log T)$ and it is not possible to improve the dependence on T [52], whereas the best previously known continuum-armed bandit algorithm [2] has regret $O\left(T^{(2\alpha+1)/(3\alpha+1)+\varepsilon}\right)$, for arbitrarily small positive ε, when α is the exponent of Lipschitz continuity of the mean reward function. It was not known whether or not this exponential gap is inherent, i.e. whether the regret of the optimal continuum-armed bandit algorithm (against i.i.d. adversaries) is polynomial or polylogarithmic in T. In Chapter 4 we settle this question, by supplying nearly matching upper and lower bounds for the continuum-armed bandit problem with respect to i.i.d. adversaries as well as adaptive adversaries. The main theorem of Chapter 4 may be summarized as follows.

Theorem 1.4. *Let $\mathcal{S}$ be a bounded subinterval of $\mathbb{R}$ and let Γ denote the set of functions $\mathcal{S} \to [0,1]$ which are uniformly locally Lipschitz with exponent α, constant L, and restriction δ. Let $\mathcal{A}_{\text{adpt}}(\Gamma)$ and $\mathcal{A}_{\text{iid}}(\Gamma)$ denote the sets of adaptive and i.i.d. adversaries, respectively, for $(\mathcal{S}, \Gamma)$; note that $\mathcal{A}_{\text{iid}}(\Gamma) \subset \mathcal{A}_{\text{adpt}}(\Gamma)$. There exists an algorithm* **CAB** *satisfying*

$$R(\mathbf{CAB}, \mathcal{A}_{\text{adpt}}(\Gamma); T) = O(T^{\frac{\alpha+1}{2\alpha+1}} \log^{\frac{\alpha}{2\alpha+1}}(T)).$$

For any $\beta < \frac{\alpha+1}{2\alpha+1}$, there is no algorithm ALG *satisfying $R(\mathsf{ALG}, \mathcal{A}_{\text{iid}}(\Gamma); T) = O(T^{\beta})$.*

The theorem demonstrates that there really is a qualitative gap between the K-armed and continuum-armed bandit problems with an i.i.d. adversary: for the former problem, the regret of the best algorithms grows logarithmically with T; for the latter, it grows polynomially with T.

Independently and concurrently with our work, Eric Cope [25] also demonstrated that the regret of the optimal continuum-armed bandit algorithm grows polynomially with T. In fact he proved that no continuum-armed bandit algorithm can achieve regret $o(\sqrt{T})$ against i.i.d. adversaries provided that:

- The cost function class Γ contains a function $f(x)$ with a unique global maximum at $x = \theta$ satisfying

$$f(\theta) - f(x) \geq C|\theta - x|^p$$

for some constants $C < \infty$ and $p \geq 1$;

- Γ also contains a continuum of functions "close" to f in a sense made precise in [25].

The paper also presents a matching upper bound of $O(\sqrt{T})$ for the regret of the optimal algorithm when $p = 2$, using a modified version of Kiefer-Wolfowitz stochastic approximation [48]. In comparison with these results, our Theorem 1.4 makes a weaker assumption about the cost functions — i.e. Lipschitz continuity — and achieves a weaker upper bound on regret, i.e. $O(T^{(\alpha+1)/(2\alpha+1)} \log^{\alpha/(2\alpha+1)} T)$ rather than $O(\sqrt{T})$.

1.4.2 Bandit problems in vector spaces

Let us consider the prospect of generalizing Theorem 1.4 to a d-dimensional strategy set. The upper bound in Theorem 1.4 is proved using a trivial reduction to the K-armed bandit problem: for an appropriate value of ε, one selects a finite subset $X \subset \mathcal{S}$ such that every element of $\mathcal{S}$ is within ε of an element of X, and one runs the $|X|$-armed bandit algorithm with strategy set X. When we try to use the same reduction with a d-dimensional strategy set, we run into a problem: the size of the set X must be exponential in d, resulting in an algorithm with exponential convergence time. In fact, it is easy to see that this exponential convergence time is unavoidable, as is demonstrated by the following example.

Example 1.1. If the strategy set is $[-1, 1]^d$ then it may naturally be partitioned into 2^d orthants depending on the signs of the d coordinates. Suppose we choose one of these orthants $\mathcal{O}$ at random, choose a Lipschitz-continuous cost function c satisfying $c(x) = 1$ for $x \notin \mathcal{O}$ and $c(x) = 0$ for some $x \in \mathcal{O}$, and put $c_1 = c_2 = \ldots = c_T$. For any algorithm, the expected number of trials before the algorithm samples an element of $\mathcal{O}$ is $\Omega(2^d)$, and it follows that no algorithm can achieve convergence time $o(2^d)$.

While Example 1.1 illustrates that it is impossible for the convergence time of an algorithm to be polynomial in d when the cost functions are arbitrary Lipschitz functions, one can hope to achieve polynomial convergence time for generalized bandit problems in d dimensions when the class of cost functions is further constrained. The next two theorems illustrate that this is indeed the case.

Theorem 1.5. *If $\mathcal{S}$ is a compact subset of $\mathbb{R}^d$ and Γ is the set of linear functions mapping $\mathcal{S}$ to an interval $[-M, M]$, then there exists an algorithm achieving regret $O(T^{2/3}Md^{5/3})$ against oblivious adversaries in the generalized bandit problem for $(\mathcal{S}, \Gamma)$. The algorithm requires only polynomial computation time, given an oracle for minimizing linear functions on $\mathcal{S}$.*

Theorem 1.6. *If $\mathcal{S}$ is a bounded convex subset of $\mathbb{R}^d$ and Γ is a set of convex functions mapping $\mathcal{S}$ to an interval $[-M, M]$, and if the functions in Γ are twice continuously differentiable with bounded first and second partial derivatives, then there is an algorithm achieving regret $O(T^{3/4}d^{17/4})$ against oblivious adversaries in the generalized bandit problem for $(\mathcal{S}, \Gamma)$. The algorithm requires only polynomial computation time, given an oracle for minimizing the Euclidean distance from a point $x \in \mathbb{R}^d$ to $\mathcal{S}$.*

In proving both of these theorems, we introduce the notion of a *barycentric spanner*. This is a special type of basis for the vector space spanned by $\mathcal{S}$, with the property that every vector in $\mathcal{S}$ may be expressed as a linear combination of basis vectors with coefficients between -1 and 1. (If we relax this condition by allowing the coefficients to be between $-C$ and C, we call the basis a C-approximate barycentric spanner.) We demonstrate that every compact subset of $\mathbb{R}^d$ has a barycentric spanner, and that a C-approximate barycentric spanner may be computed in polynomial time, for any $C > 1$, given access to an oracle for minimizing linear functions on $\mathcal{S}$. Barycentric spanners arise naturally in problems which involve estimating linear functions based on noisy measurements, or approximating arbitrary functions on a compact subset of $\mathbb{R}^d$ with linear combinations of a finite number of basis functions. We will sketch one of these applications in Section 5.3.

Subsequent to our work on online linear optimization, McMahan and Blum [58] strengthened Theorem 1.5 to hold against an adaptive adversary, with a slightly weaker upper bound on regret, using a modified version of our algorithm. Independently and concurrently with our work on online convex optimization, Flaxman *et. al.* [32] obtained a stronger version of Theorem 1.6 which allows an adaptive adversary, requires no smoothness hypothesis on the cost functions, and achieves a stronger regret bound of $O(dT^{3/4})$. In Section 5.4.1 we elaborate on the comparison between these two algorithms.

1.4.3 Bandit problems in measure spaces

The lower bound in Example 1.1 demonstrates that if the cost functions are unconstrained and an exponentially small fraction of the strategy set achieves an average cost less than y, it is unreasonable to expect an algorithm, in a polynomial number of trials, to also achieve an average cost less than y. But if a *polynomially small* fraction

of the strategy set achieves an average cost less than y, one can hope to design an algorithm which achieves an average cost less than $y + \delta$ (for some small $\delta > 0$) in a polynomial number of trials. In order to make this idea precise, it must be possible to specify what fraction of the strategies in $\mathcal{S}$ are contained in a given subset X, i.e. $\mathcal{S}$ must be a measure space of finite total measure.

In considering generalized bandit problems whose strategy set is a measure space $(\mathcal{S}, \mu)$ of total measure 1, we are led to the notion of an *anytime bandit algorithm*; the name "anytime" refers to the fact that the algorithm satisfies a non-trivial performance guarantee when stopped at any time $T > 0$, and the quality of the guarantee improves as $T \to \infty$, eventually converging to optimality. Let us say that ALG is an anytime bandit algorithm for $(\mathcal{S}, \mu)$ with convergence time $\tau(\varepsilon, \delta)$ if it satisfies the following guarantee for all $\varepsilon, \delta > 0$ and for all $T > \tau(\varepsilon, \delta)$: if $\mathcal{S}$ contains a subset X of measure at least ε such that every $x \in X$ satisfies

$$\frac{1}{T} \sum_{t=1}^{T} c_t(x) \leq y,$$

then the sequence of strategies $x_1, x_2, \ldots, x_T$ chosen by ALG satisfies

$$\mathbf{E}\left[\frac{1}{T} \sum_{t=1}^{T} c_t(x_t)\right] \leq y + \delta.$$

Theorem 1.7. *For any measure space $(\mathcal{S}, \mu)$ of total measure 1, there is an anytime bandit algorithm for $(\mathcal{S}, \mu)$ whose convergence time is $O\left((1/\varepsilon)^{1+\alpha}\text{poly}(1/\delta)\right)$. No anytime bandit algorithm achieves convergence time $O((1/\varepsilon)\text{polylog}(1/\varepsilon)\text{poly}(1/\delta))$.*

An anytime bandit algorithm is applied in the collaborative learning algorithm of Chapter 7.

1.5 Adaptive pricing

The rising popularity of Internet commerce has spurred much recent research on market mechanisms which were either unavailable or impractical in traditional markets, because of the amount of communication or computation required. In Chapter 3 we will consider one such mechanism, the on-line posted-price mechanism, in which a seller with an unlimited supply of identical goods interacts sequentially with a population of T buyers. For each buyer, the seller names a price between 0 and 1; the buyer then decides whether or not to buy the item at the specified price, based on her privately-held valuation for the good. This transaction model is dictated by the following considerations:

- Following earlier authors [14, 31, 36, 65], we are interested in auction mechanisms which are *strategyproof*, meaning that buyers weakly maximize their utility by truthfully revealing their preferences. As shown in [14], this requirement in the on-line auction setting is equivalent to requiring that the seller charge buyer i a price which depends only on the valuations of previous buyers.

- Given that the price offered to buyer i does not depend on any input from that buyer, it is natural for the seller to announce this price before the buyer reveals any preference information. In fact, for reasons of trust, the buyers may not want to reveal their preferences before an offer price is quoted [21].

- For privacy reasons, the buyers generally do not wish to reveal any preference information *after* the price is quoted either, apart from their decision whether or not to purchase the good. Also, buyers are thus spared the effort of precisely determining their valuation, since the mechanism only requires them to determine whether it is greater or less than the quoted price.

The seller's pricing strategy will tend to converge to optimality over time, as she gains information about how the buyers' valuations are distributed. A natural question which arises is: what is the cost of not knowing the distribution of the buyers' valuations in advance? In other words, assume our seller pursues a pricing strategy $\mathbb{S}$ which maximizes her expected revenue $\rho(\mathbb{S})$. As is customary in competitive analysis of auctions, we compare $\rho(\mathbb{S})$ with the revenue $\rho(\mathbb{S}^{\text{opt}})$ obtained by a seller who knows the buyers' valuations in advance but is constrained to charge the same price to all buyers ([14, 21, 31, 36]). While previous authors have analyzed auctions in terms of their competitive ratio (the ratio between $\rho(\mathbb{S})$ and $\rho(\mathbb{S}^{\text{opt}})$), we instead analyze the regret, i.e. the difference $\rho(\mathbb{S}) - \rho(\mathbb{S}^{\text{opt}})$. This is a natural parameter to study for two reasons. First, it roughly corresponds to the amount the seller should be willing to pay to gain knowledge of the buyers' valuations, e.g. by doing market research. Second, it was shown by Blum et al in [21] that there are randomized pricing strategies achieving competitive ratio $1 + \varepsilon$ for any $\varepsilon > 0$; thus it is natural to start investigating the lower-order terms, i.e. the $o(1)$ term in the ratio $\rho(\mathbb{S})/\rho(\mathbb{S}^{\text{opt}})$ for the optimal pricing strategy $\mathbb{S}$.

One can envision several variants of this problem, depending on what assumptions are made about the buyers' valuations. We will study three valuation models.

Identical: All buyers' valuations are equal to a single price $p \in [0, 1]$. This price is unknown to the seller.

Random: Buyers' valuations are independent random samples from a fixed probability distribution on $[0, 1]$. The probability distribution is not known to the

seller.

Worst-case: The model makes no assumptions about the buyers' valuations. They are chosen by an adversary who is oblivious to the algorithm's random choices.

Our results are summarized in the following three theorems. In all of them, the term "pricing strategy" refers to a randomized on-line algorithm for choosing offer prices, unless noted otherwise.

Theorem 1.8. *Assuming identical valuations, there is a deterministic pricing strategy achieving regret $O(\log\log T)$. No pricing strategy can achieve regret $o(\log\log T)$.*

Theorem 1.9. *Assuming random valuations, there is a pricing strategy achieving regret $O(\sqrt{T\log T})$, under the hypothesis that the function*

$$f(x) = x \cdot \Pr(\textit{buyer's valuation is at least } x)$$

has a unique global maximum x^* in the interior of $[0,1]$, and that $f''(x^*) < 0$. No pricing strategy can achieve regret $o(\sqrt{T})$, even under the same hypothesis on the distribution of valuations.

Theorem 1.10. *Assuming worst-case valuations, there is a pricing strategy achieving regret $O((T^{2/3}(\log T)^{1/3})$. No pricing strategy can achieve regret $o(T^{2/3})$.*

The lower bound in the random-valuation model is the most difficult of the results stated above, and it represents the most interesting technical contribution in Chapter 3. Interestingly, our lower bound does not rely on constructing a contrived demand curve to defeat a given pricing strategy. Rather, we will show that for any family $\mathcal{D}$ of demand curves satisfying some reasonably generic axioms, and for any randomized pricing strategy, the probability of achieving regret $o(\sqrt{T})$ when the demand curve is chosen randomly from $\mathcal{D}$ is zero. Note the order of quantification here, which differs from the $\Omega(\sqrt{T})$ lower bounds which have appeared in the literature on the adversarial multi-armed bandit problem [4]. In that lower bound it was shown that, given foreknowledge of T, an adversary could construct a random sequence of payoffs forcing any strategy to have regret $\Omega(\sqrt{T})$. In our theorem, the demand curve is chosen randomly without foreknowledge of T or of the pricing strategy, and it is *still* the case that the pricing strategy has probability 0 of achieving regret $o(\sqrt{T})$ as $T \to \infty$.

We should also point out that in Theorem 1.9, it is not possible to eliminate the hypothesis that f has a well-defined second derivative at x^* and that $f''(x^*) < 0$. The proof of Theorem 1.4 — which demonstrates that continuum-armed bandit algorithms can not achieve regret $o(T^{2/3})$ against Lipschitz cost functions — can be modified to

yield an explicit example of a demand curve family such that the function $f(x)$ has a singularity at x^* and such that no pricing strategy can achieve $o(T^{2/3})$ in the random-valuations model. This example illustrates that the distinction between the $\widetilde{O}(\sqrt{T})$ bounds in Theorem 1.9 and the $\widetilde{O}(T^{2/3})$ bounds in Theorem 1.10 is primarily due to the distinction between smooth and singular demand curves, not the distinction between random and worst-case valuations.

1.5.1 Related work in computer science

There have been many papers applying notions from the theory of algorithms to the analysis of auction mechanisms. While much of this work focuses on combinatorial auctions — a subject not touched on here — there has also been a considerable amount of work on auction mechanisms for selling identical individual items, the setting considered in Chapter 3. In [36], the authors consider mechanisms for off-line auctions, i.e. those in which all buyers reveal their valuations before any goods are sold. The authors characterize mechanisms which are truthful (a term synonymous with "strategyproof", defined above), and show that no such mechanism can be constant-competitive with respect to the optimal single-price auction, assuming worst-case valuations. In contrast, they present several randomized off-line auction mechanisms which are truthful and constant-competitive with respect to the optimal auction which is constrained to set a single price *and to sell at least two copies of the good.*

On-line auctions were considered in [14, 21], in the posted-price setting considered here as well as the setting where buyers reveal their valuations but are charged a price which depends only on the information revealed by prior buyers. In the latter paper, techniques from the theory of online decision problems are applied to yield a $(1+\varepsilon)$-competitive on-line mechanism (for any $\varepsilon > 0$) under the hypothesis that the optimal single-price auction achieves revenue $\Omega(h \log h \log\log h)$, where $[1, h]$ is the interval which is assumed to contain all the buyers' valuations. In Section 3.3.1, we use their algorithm (with a very minor technical modification) to achieve expected regret $O(T^{2/3}(\log T)^{1/3})$ assuming worst-case valuations.

An interesting hybrid of the off-line and on-line settings is considered by Hartline in [40]. In that paper, the mechanism interacts with the set of buyers in two rounds, where prices in the second round may be influenced by the preferences revealed by buyers in the first round. Assuming the set of buyers participating in the first round is a uniform random subset of the pool of T buyers, the paper exhibits a posted-price mechanism which is 4-competitive against the optimal single-price auction.

On-line multi-unit auctions (in which buyers may bid for multiple copies of the item) are considered in [12], which presents a randomized algorithm achieving com-

petitive ratio $O(\log B)$ where B is the ratio between the highest and lowest per-unit prices offered. This result is sharpened in [53], where the optimal competitive ratio (as a function of B) is determined exactly.

1.5.2 Related work in economics

The preceding papers have all adopted the worst-case model for buyers' valuations, as is customary in the computer science literature. The traditional approach in the economics literature is to assume that buyers' valuations are i.i.d. samples from a probability distribution which is either known to the seller or which depends on some unknown parameters sampled from a Bayesian prior distribution which is known to the seller. Rothschild [63] introduced multi-armed bandit techniques into the economics literature on optimal pricing, in a paper which studies the optimal pricing policy of a monopolistic seller in an idealized model in which the demand curve depends on unknown parameters governed by a known Bayesian prior, the seller is constrained to choose one of two prices p_1, p_2, and the seller's revenue is geometrically discounted over time. Rothschild exhibited instances in which a seller using the optimal pricing policy has a positive probability of choosing just one of the two prices after a finite number of trials and charging this price *forever* thereafter; moreover, there is a positive probability that this price is the inferior price, i.e. it would have the smaller expected revenue if the demand curve were revealed. This pathology is referred to as *incomplete learning*; as we have seen in Section 1.2 it is a phenomenon associated with online decision problems that have a geometrically discounted adversary. Subsequent papers on optimal pricing in economics focused on the incomplete learning phenomenon revealed by Rothschild. McLennan [57] considers a model in which there is a one-parameter continuum of possible prices (rather than only two possible prices as in Rothschild's work); even if there are only two possible demand curves, McLennan demonstrates that incomplete learning may occur with positive probability. Easley and Kiefer [30] generalize these results still further, examining general online decision problems in which the strategy set $\mathcal{S}$ and feedback set Φ are compact and the adversary is geometrically discounted. They characterize conditions under which the optimal algorithm for such problems may exhibit incomplete learning with positive probability. Aghion *et. al.* [1] focus on online decision problems in which the payoff functions are geometrically discounted but do not vary over time, i.e. in our terminology, they study randomized oblivious adversaries satisfying $C_t(r') = \alpha^t C(r')$ for some constant $0 < \alpha \leq 1$ and some Γ-valued random variable C. They explore the role of smoothness, analyticity, and quasi-concavity of the payoff function C in determining whether the optimal algorithm may exhibit incomplete learning.

A recent paper by Ilya Segal [65] considers optimal pricing by a monopolistic seller

in an *off-line* auction. As in the other economics papers cited above, Segal assumes that the seller's beliefs about the buyers' valuations are represented by a Bayesian prior distribution. In Section V of [65], Segal compares the expected regret of the optimal strategyproof off-line mechanism with that of the optimal on-line posted-price mechanism (which he calls the "optimal experimentation mechanism") under three assumptions on the space $\mathcal{D}$ of possible demand curves:

- $\mathcal{D}$ is a finite set ("Hypothesis testing");
- $\mathcal{D}$ is parametrized by a finite-dimensional Euclidean space ("Parametric estimation");
- $\mathcal{D}$ is arbitrary ("Non-parametric estimation").

In this terminology, our Chapter 3 is concerned with bounding the expected regret of the optimal experimentation mechanism in the non-parametric case. Segal explicitly refrains from addressing this case, writing, "The optimal experimentation mechanism would be very difficult to characterize in [the non-parametric] setting. Intuitively, it appears that its convergence rate may be slower [than that of the optimal off-line mechanism] because the early purchases at prices that are far from p^* will prove useless for fine-tuning the price around p^*." This intuition is partly confirmed by the lower bound we prove in Section 3.2.2.

1.6 Online routing

Consider a network $G = (V, E)$ with a designated source s and sink r, and with edge delays which may vary over time. In each trial a decision-maker must choose a path from s to r with the objective of minimizing the total delay. (One may motivate this as the problem of choosing a route to drive to work each day, or of source-routing packets in a network so as to minimize the average transmission delay.) It is natural to model this problem as an online decision problem; in this case the strategy set is the set of paths from s to r and the cost functions are determined by specifying real-valued edge lengths and assigning to each path a cost equal to the sum of its edge lengths. Note that the strategy set is finite, but its size may be exponential in the size of G, in which case a naïve solution based on the multi-armed bandit algorithm would have exponential convergence time.

If we allow non-simple paths, then the online shortest path problem becomes a special case of online linear optimization, hence our online linear optimization algorithm from Section 5.2.3 may be applied to yield an online shortest path algorithm with polynomial convergence time.

Theorem 1.11. *Let $G = (V, E)$ be a directed graph with two distinguished vertices s, r. Let $\mathcal{S}$ denote the set of (not necessarily simple) directed paths from s to r of length at most H, and let Γ denote the set of all functions from $\mathcal{S}$ to $\mathbb{R}_+$ defined by assigning a cost between 0 and 1 to each edge of G and defining the cost of a path to be the sum of its edge costs. Let $\mathcal{A}$ denote the set of all oblivious adversaries in the generalized bandit problem for $(\mathcal{S}, \Gamma)$. There is an algorithm ALG whose regret and convergence time satisfy*

$$\begin{aligned} R(\mathsf{ALG}, \mathcal{A}; T) &= O(T^{2/3} H m^{5/3}) \\ T(\delta) &= O(H^3 m^5 / \delta^3). \end{aligned}$$

1.6.1 Related work

The online shortest path problem has received attention in recent years, but algorithms prior to ours either assumed full feedback or they assumed a feedback model which is intermediate between full and opaque feedback. Takimoto and Warmuth [68] studied the online shortest path problem with full feedback, demonstrating that the best-expert algorithm [33, 54] in this case can be simulated by an efficient (polynomial-time) algorithm in spite of the fact that there are exponentially many "experts", corresponding to all the $s - r$ paths in G. Kalai and Vempala [44] considered online linear optimization for a strategy set $\mathcal{S} \subseteq \mathbb{R}^d$, presenting an algorithm which achieves $O(\sqrt{T \log d})$ regret, as discussed earlier. (Note that this bound depends only on the dimension d, hence it is applicable even if the cardinality of $\mathcal{S}$ is exponential or infinite.) One may interpret this as an algorithm for the online shortest path problem, by considering the set of $s - r$ paths in G as the vertices of the polytope of unit flows from s to r; thus Kalai and Vempala's algorithm also constitutes a online shortest path algorithm with regret $O(\sqrt{T \log m})$ in the full feedback model.

Awerbuch and Mansour [8] considered an online path-selection problem in a model where the edges have binary costs (they either fail or they do not fail) and the cost of a path is 1 if any edge fails, 0 otherwise. They consider a "prefix" feedback model, where the feedback in each trial identifies the location of the *first* edge failure, if any edge failed. Using the best-expert algorithm [54] as a black box, their algorithm obtains regret $O\left(H(n \log(nT))^{1/2} T^{5/6}\right)$ assuming an oblivious adversary. The algorithm was strengthened to work against an adaptive adversary in subsequent work by Awerbuch, Holmer, Rubens, and Kleinberg [5]. These algorithms are structurally quite different from the online shortest path algorithm presented here in Section 5.2.4. Rather than treating the shortest path problem as a special case of a global linear optimization problem in a vector space determined by the edges of G, the algorithms in [5, 8] are based on factoring the problem into a set of local optimization problems, in which

each node v uses a best-expert algorithm to learn which of its incoming edges lies on the shortest $s - v$ path. A similar approach can be used to design an online shortest path algorithm in the opaque-feedback model against an adaptive adversary; see [6] for details.

Our approach to the online shortest path problem, which relies on estimating the length of a small number of paths and using linear algebra to reconstruct the lengths of all other paths, is strongly reminiscent of linear algebraic approaches to *network tomography* (e.g. [24, 66]), in which one attempts to deduce link-level or end-to-end path properties (e.g. delay, delay variance, packet loss rate) in a network by making a limited number of measurements. One of the conclusions which can be drawn from our work is that the output of these algorithms may be very sensitive to measurement errors if one does not use a carefully-chosen basis for the set of paths; moreover, good bases (i.e. approximate barycentric spanners) always exist and may be computed efficiently given knowledge of the network layout.

1.7 Collaborative learning

It is clear that leveraging trust or shared taste enables a community of users to be more productive, as it allows them to repeat each other's good decisions while avoiding unnecessary repetition of mistakes. Systems based on this paradigm are becoming increasingly prevalent in computer networks and the applications they support. Examples include reputation systems in e-commerce (e.g. eBay, where buyers and sellers rank each other), collaborative filtering (e.g. Amazon's recommendation system, where customers recommend books to other customers), and link analysis techniques in web search (e.g., Google's PageRank, based on combining links — i.e. recommendations — of different web sites). Not surprisingly, many algorithms and heuristics for such systems have been proposed and studied experimentally or phenomenologically [23, 49, 55, 60, 72, 73, 74]. Yet well-known algorithms (e.g. eBay's reputation system, the Eigentrust algorithm [45], the PageRank [23, 60] and HITS [49] algorithms for web search) have thus far not been placed on an adequate theoretical foundation.

In Chapter 7 we propose a theoretical framework for understanding the capabilities and limitations of such systems as a model of distributed computation, using the theory of online decision problems. Our approach aims to highlight the following challenges which confront the users of collaborative decision-making systems such as those cited above.

Malicious users. Since the Internet is open for anybody to join, the above systems

are vulnerable to fraudulent manipulation by dishonest ("Byzantine") participants.

Distinguishing tastes. Agents' tastes may differ, so that the advice of one honest agent may not be helpful to another.

Temporal fluctuation. The quality of resources varies of time, so past experience is not necessarily predictive of future performance.

To account for these challenges, we formulate collaborative decision problems as a generalization of the multi-armed bandit problem in which there are n agents and m resources with time-varying costs. In each trial each agent must select one resource and observe its cost. (Thus, the multi-armed bandit problem is the special case $n = 1$.) Assume that among the n agents, there are h "honest" agents who obey the decision-making protocol specified by the algorithm and report their observations truthfully; the remaining $n - h$ agents are Byzantine and may behave arbitrarily. Assume moreover that the honest agents may be partitioned into k coalitions with "consistent tastes," in the sense that agents in the same coalition observe the same expected cost for a resource y if they sample y in the same trial. In Chapter 7 we formalize these notions and also extend the definitions of "regret" and "convergence time" to this setting to obtain the following theorem.

Theorem 1.12. *Assume that the number of honest agents, h, and the number of resources, m, are comparable in magnitude to the number of agents, n, i.e. there exist positive constants c_1, c_2 such that h, m both lie between $c_1 n$ and $c_2 n$. Then there exists an algorithm* TrustFilter *for the collaborative learning problem whose regret $\overline{R}$ and convergence time $\tau(\delta)$ satisfy*

$$\overline{R} = O\left(k \cdot \frac{\log n \log T}{T^{1/3}}\right) \tag{1.1}$$

$$\tau(\delta) = O(k^3 \log^3 n \log^3(k \log n)). \tag{1.2}$$

While our online learning paradigm is different from prior approaches to collaborative decision systems, the resulting algorithms exhibit an interesting resemblance to algorithms previously proposed in the systems and information retrieval literature [23, 45, 49, 60] indicating that our approach may be providing a theoretical framework which sheds light on the efficacy of such algorithms in practice while suggesting potential enhancements to these algorithms.

1.7.1 Related work

The adversarial multi-armed bandit problem [4] forms the basis for our work; our model generalizes the existing multi-armed bandit model to the setting of collabora-

tive learning with dishonest users. Our work is also related to several other topics which we now discuss.

Collaborative filtering — spectral methods:

Our problem is similar, at least in terms of motivation, to the problem of designing collaborative filtering or recommendation systems. In such problems, one has a community of users selecting products and giving feedback on their evaluations of these products. The goal is to use this feedback to make recommendations to users, guiding them to subsequently select products which they are likely to evaluate positively. Theoretical work on collaborative filtering has mostly dealt with centralized algorithms for such problems. Typically, theoretical solutions have been considered for specific (e.g., stochastic) input models [11, 29, 37, 41, 61], In such work, the goal is typically to reconstruct the full matrix of user preferences based on small set of potentially noisy samples. This is often achieved using spectral methods. In constrast, we consider an adversarial input model. Matrix reconstruction techniques do not suffice in our model. They are vulnerable to manipulation by dishonest users, as was observed in [9, 10]. Dishonest users may disrupt the low-rank assumption which is crucial in matrix reconstruction approaches. Alternatively, they may report phony data so as to perturb the singular vectors of the matrix, directing all the agents to a particularly bad resource.

In contrast, our algorithm makes recommendations which are provably good even in the face of arbitrary malicious attacks by dishonest users. To obtain this stronger guarantee, we must make a stronger assumption about the users: honest users are assumed to behave like automata who always follow the recommendations provided by the algorithm. (The work on collaborative filtering cited above generally assumes that users will choose whatever resources they like; the algorithm's role is limited to that of a passive observer, taking note of the ratings supplied by users and making recommendations based on this data.)

Collaborative filtering — random sampling methods:

The only previous collaborative filtering algorithm which tolerates Byzantine behavior is the "Random Advice Random Sample" algorithm in [9, 10]; it achieves a logarithmic convergence time, assuming the costs of resources do not vary over time. (The only changes in the operating environment over time occur when resources arrive or depart.) This assumption of static costs allows the design of algorithms based on a particularly simple "recommendation" principle: once an agent finds a good resource, it chooses it forever and recommends it to others. The bounds on regret and conver-

gence time in [9] are analogous to ours, and are in fact polylogarithmically superior, to those in our Theorem 7.1. However, [9] does not handle costs which evolve dynamically as a function of time, and is limited to $\{0, 1\}$-valued rather than real-valued costs.

Reputation management in peer-to-peer networks:

Kamvar et al [45] proposed an algorithm, dubbed *EigenTrust*, for the problem of locating resources in peer-to-peer networks. In this problem, users of a peer-to-peer network wish to select other peers from whom to download files, with the aim of minimizing the number of downloads of inauthentic files by honest users; the problem is made difficult by the presence of malicious peers who may attempt to undermine the algorithm. Like our algorithm, EigenTrust defines reputation scores using a random walk on the set of agents, with time-varying transition probabilities which are updated according to the agents' observations. Unlike our algorithm, they use a different rule for updating the transition probabilities, and they demonstrate the algorithm's robustness against a limited set of malicious exploits, as opposed to the arbitrary adversarial behavior against which our algorithm is provably robust. The problem considered here is less general than the peer-to-peer resource location problem considered in [45]; for instance, we assume that in each trial, any agent may select any resource, whereas they assume that only a subset of the resources are available (namely, those peers who claim to have a copy of the requested file). Despite these differences, we believe that our work may shed light on the efficacy of EigenTrust while suggesting potential enhancements to make it more robust against Byzantine malicious users.

1.8 Outline of this thesis

The rest of this thesis is organized as follows. In Chapter 2 we present and analyze some algorithms from the prior literature on online decision problems; we will frequently rely on these algorithms when presenting our own algorithms later in this work. Chapter 3 presents upper and lower bounds for adaptive pricing problems. These problems are a special case of "continuum-armed bandit problems", in which the strategy space is a one-parameter interval; we consider the general case more fully in Chapter 4. In Chapter 5 we progress from one-parameter to multi-parameter optimization problems, presenting algorithms for generalized bandit problems in vector spaces with linear or convex cost functions. As a concrete application of these techniques, we specify an online routing algorithm with polynomial convergence time.

Chapter 6 passes to a still greater level of generality, in which we consider the strategy space to be a measure space and the cost functions to be measurable functions, and we design "anytime bandit algorithms" which have polynomial convergence time with respect to suitably relaxed definitions of regret and convergence time. One of these algorithms is applied in Chapter 7, which presents an algorithm for the collaborative bandit problem. In Chapter 8 we present some concluding remarks and open problems.

1.9 Bibliographic notes

The material in Chapter 3 is based on joint work with Tom Leighton which appeared in [51]. Chapter 4 and Section 5.4 are based on the extended abstract [50]. The remaining material in Chapter 5 and all of the material in Chapter 7 are based on joint work with Baruch Awerbuch; this work appeared in the extended abstracts [6, 7].

Chapter 2

Background material

2.1 The weighted majority and Hedge algorithms

The problem of *predicting based on expert advice* was the impetus for much of the early work on the worst-case analysis of online decision problems. Such problems are known as "best-expert" problems; they correspond to the *full feedback model* defined in Section 1.1. Here we present two best-expert algorithms: the weighted majority algorithm WMA [54] and the closely-related algorithm Hedge [4].

Suppose that we wish to predict a binary sequence $b_1, b_2, \ldots, b_T$, given access to a set $\mathcal{S}$ of K *experts* each of whom makes a sequence of predictions $b_1(i), b_2(i), \ldots, b_T(i)$. At the start of trial t, each expert i reveals its prediction $b_t(i)$, the algorithm makes a prediction $b_t(\mathsf{ALG})$ based on this advice, and then the true value of b_t is revealed. The algorithm and the experts are charged a cost of 1 for each mistake in predicting $b_t\,(1 \leq t \leq T)$, and the algorithm's regret is measured by comparing its charge with that of the best expert. Note that this problem doesn't quite fit the formulation of online decision problems specified in Section 1.1: the algorithm has only *two* choices in each trial, but its regret is measured by comparing it against K experts. In an online decision problem as formulated in Section 1.1, these two sets — the set of alternatives in each trial, and the set of strategies against which the algorithm is compared when evaluating its regret — would be identical. Nevertheless, the binary prediction problem considered here clearly bears a close relation to online decision problems as formulated in Section 1.1, and the algorithm WMA which we will analyze is important both historically and as an aid in developing intuition about online decision problems.

The weighted majority algorithm is actually a *one-parameter family* of algorithms parametrized by a real number $\varepsilon > 0$. We will use the notation $\mathsf{WMA}(\varepsilon)$ to denote the algorithm obtained by choosing a specific value of ε. This algorithm is presented in

Algorithm $\mathsf{WMA}(\varepsilon)$

/* Initialization */
$w_0(i) \leftarrow 1$ **for** $i = 1, 2, \ldots, K$.

/* Main loop */
for $t = 1, 2, \ldots, T$
    /* Make prediction by taking weighted majority vote */
    if $\sum_{i\,:\,b_t(i)=0} w_{t-1}(i) > \sum_{i\,:\,b_t(i)=1} w_{t-1}(i)$
        predict $b_t = 0$;
    else
        predict $b_t = 1$.

    Observe the value of b_t.

    /* Update weights multiplicatively */
    $E_t \leftarrow$ {experts who predicted incorrectly}
    $w_t(i) \leftarrow (1 - \varepsilon) \cdot w_{t-1}(i)$ **for** $i \in E_t$.
end

Figure 2-1: The weighted majority algorithm

Figure 2-1. The idea of the algorithm is quite simple: its prediction of b_t is based on taking a majority vote among the experts, where each expert is given a vote weighted according to its past performance. After the true value of b_t is revealed, we reduce the weight of each expert who made a mistake by a multiplicative factor of $1 - \varepsilon$, where $\varepsilon > 0$ is a parameter specified when the algorithm is invoked.

In the following theorem, and throughout the rest of this thesis, "log" refers to the natural logarithm function, unless otherwise noted.

Theorem 2.1. *For any sequence of bits $b_1, b_2, \ldots, b_T$ and any $j \in \{1, 2, \ldots, K\}$, let $b_t(j), b_t(\mathsf{WMA})$ denote the predictions made by expert j and by algorithm $\mathsf{WMA}(\varepsilon)$, respectively, in trial t on input sequence $b_1, b_2, \ldots, b_T$. Then*

$$\sum_{t=1}^{T} |b_t(\mathsf{WMA}) - b_t| \leq \frac{2}{1-\varepsilon} \sum_{t=1}^{T} |b_t(j) - b_t| + \frac{2 \log K}{\varepsilon}.$$

Proof. The analysis of $\mathsf{WMA}(\varepsilon)$ is based on the parameter $W_t = \sum_{i=1}^{K} w_t(i)$. Whenever the algorithm makes a mistake, the value of W_t shrinks by a constant factor.

On the other hand, W_t is bounded below by $w_t(i)$. These two bounds on W_t, taken together, will lead to the desired result.

If the algorithm makes a mistake at time t, then the set E_t of experts who made a mistake at time t accounts for a weighted majority of the votes at time t, i.e.

$$\sum_{i \in E_t} w_{t-1}(i) \geq \frac{1}{2} W_{t-1}$$

and consequently

$$W_t = \left[(1-\varepsilon) \sum_{i \in E_t} w_{t-1}(i) \right] + \left[\sum_{i \notin E_t} w_{t-1}(i) \right] = W_{t-1} - \varepsilon \sum_{i \in E_t} w_{t-1}(i) \leq \left(1 - \frac{\varepsilon}{2}\right) W_{t-1}.$$

Let $A = \sum_{t=1}^{T} |b_t(\mathsf{WMA}) - b_t|$ be the number of mistakes made by $\mathsf{WMA}(\varepsilon)$, and let $B = \sum_{t=1}^{T} |b_t(j) - b_t|$ be the number of mistakes made by expert j. We have

$$(1-\varepsilon)^B = w_T(j) < W_T \leq (1 - \frac{\varepsilon}{2})^A W_0 < e^{-\frac{\varepsilon}{2}A} W_0.$$

Substituting $W_0 = K$ and taking the logarithm of both sides, we obtain

$$B \log(1-\varepsilon) < -\frac{\varepsilon A}{2} + \log(K)$$

and upon rearranging terms this becomes

$$A \;<\; \left(\frac{-2\log(1-\varepsilon)}{\varepsilon} \right) B + \frac{2\log(K)}{\varepsilon} \;<\; \left(\frac{2}{1-\varepsilon} \right) B + \frac{2\log(K)}{\varepsilon},$$

using the inequality $-\frac{\log(1-x)}{x} < \frac{1}{1-x}$, valid for $x > 0$. □

The mistake bound in Theorem 2.1 is essentially the best possible such bound for deterministic algorithms, in a sense made precise by the following theorem.

Theorem 2.2. *Let* ALG *be any deterministic algorithm for the binary prediction problem with* $K \geq 2$ *experts. Suppose that for every input sequence* $b_1, \ldots, b_T$ *and every expert* $j \in \{1, 2, \ldots, K\}$*, the number of mistakes made by* ALG *satisfies a linear inequality of the form*

$$\sum_{t=1}^{T} |b_t(\mathsf{ALG}) - b_t| \leq \alpha \sum_{t=1}^{T} |b_t(j) - b_t| + \beta \tag{2.1}$$

where α *is a constant and* β *depends only on* K*. Then* $\alpha \geq 2$ *and* $\beta \geq \log_2(K)$*.*

Proof. To prove $\alpha \geq 2$, suppose there is at least one expert i who always predicts $b_t(i) = 0$ and at least one expert j who always predicts $b_t(j) = 1$. Since ALG is deterministic, it is possible to construct an input sequence $b_1, b_2, \ldots, b_T$ such that ALG makes T mistakes: always choose b_t to be the opposite of ALG's prediction at time t. Since i and j always make opposite predictions, one of them is correct at least half the time, i.e. makes at most $T/2$ mistakes. Thus $T \leq \alpha T/2 + \beta$. Letting $T \to \infty$, this proves that $\alpha \geq 2$.

To prove $\beta \geq \log_2(K)$, suppose $K = 2^T$ and suppose that for each binary sequence of length T, there is one expert who predicts that sequence. As before, since ALG is deterministic we may construct an input sequence $b_1, \ldots, b_T$ such that ALG makes T mistakes. By construction there is one expert who makes no mistakes on $b_1, \ldots, b_T$, hence $T \leq \beta$. Since $T = \log_2(K)$ we have $\beta \geq \log_2(K)$ as desired. □

To obtain significantly better performance in the binary prediction problem, we must use a randomized algorithm. The next algorithm we will present, Hedge(ε), achieves a mistake bound of the form (2.1) with $\alpha = 1 + O(\varepsilon)$ and $\beta = O(\log K/\varepsilon)$. In fact, Hedge($\varepsilon$) is actually a randomized algorithm for an online decision problem with strategy set $\mathcal{S} = \{1, 2, \ldots, K\}$, cost function class $[0,1]^{\mathcal{S}}$, and full feedback. The binary prediction problem reduces to this online decision problem by defining a cost function $c_t(i) = |b_t(i) - b_t|$, i.e. the cost of an expert at time t is 1 if it makes a mistake, 0 otherwise. In each trial, Hedge(ε) designates an expert $x_t \in \mathcal{S}$ based on its random seed and on the data observed in past trials, but not on the predictions of the experts in the present trial.

Theorem 2.3. *Let $\mathcal{S} = \{1, 2, \ldots, K\}$ and $\Gamma = [0,1]^{\mathcal{S}}$, and let $\mathcal{A}$ denote the set of all adaptive adversaries for $(\mathcal{S}, \Gamma)$. Then*

$$R(\text{Hedge}(\varepsilon), \mathcal{A}; T) \leq \left(\frac{\varepsilon}{1-\varepsilon}\right) T + \frac{\log(K)}{\varepsilon}.$$

Proof. Let $W_t = \sum_{i \in \mathcal{S}} w_t(i)$. The convexity of the function $(1-\varepsilon)^y$ implies that $(1-\varepsilon)^y \leq 1 - \varepsilon y$ for $y \in [0,1]$, from which we obtain the bound:

$$\begin{aligned} \frac{W_t}{W_{t-1}} &= \frac{\sum_{i \in \mathcal{S}} (1-\varepsilon)^{c_t(i)} w_{t-1}(i)}{\sum_{i \in \mathcal{S}} w_{t-1}(i)} \\ &\leq \frac{\sum_{i \in \mathcal{S}} (1 - \varepsilon c_t(i)) w_{t-1}(i)}{\sum_{i \in \mathcal{S}} w_{t-1}(i)} \\ &= 1 - \varepsilon \sum_{i \in \mathcal{S}} c_t(i) p_t(i) = 1 - \varepsilon \mathbf{E}(c_t(x_t)) \end{aligned}$$

Algorithm Hedge(ε)

/* Initialization */
$w_0(i) \leftarrow 1$ **for** $i \in \mathcal{S}$

/* Main loop */
for $t = 1, 2, \ldots, T$
    /* Define distribution for sampling random strategy */
    for $i \in \mathcal{S}$
        $p_t(i) \leftarrow w_{t-1}(i) \Big/ \left(\sum_{j \in \mathcal{S}} w_{t-1}(j) \right)$
    end
    Choose $x_t \in \mathcal{S}$ at random according to distribution p_t.
    Observe feedback c_t.

    /* Update score for each strategy */
    for $i \in \mathcal{S}$
        $w_t(i) \leftarrow w_{t-1}(i) \cdot (1 - \varepsilon)^{c_t(i)}$
    end
end

Figure 2-2: The algorithm Hedge(ε).

hence

$$\log(W_t/W_{t-1}) \leq \log(1 - \varepsilon \mathbf{E}(c_t(x_t)) \leq -\varepsilon \mathbf{E}(c_t(x_t))$$

and

$$\begin{aligned} \log(W_T) &= \log(W_0) + \sum_{t=1}^{T} \log\left(\frac{W_t}{W_{t-1}}\right) \\ &\leq \log(K) - \varepsilon \mathbf{E}\left(\sum_{t=1}^{T} c_t(x_t)\right). \end{aligned} \tag{2.2}$$

But for any $x \in \mathcal{S}$, $W_T \geq w_T(x) = (1 - \varepsilon)^{\sum_{t=1}^{T} c_t(x)}$, hence

$$\log(W_T) \geq \log(1 - \varepsilon) \sum_{t=1}^{T} c_t(x). \tag{2.3}$$

Combining (2.2) and (2.3) and rearranging terms we obtain

$$\begin{aligned}\mathbf{E}\left(\sum_{t=1}^{T} c_t(x_t)\right) &\leq \frac{-\log(1-\varepsilon)}{\varepsilon}\sum_{t=1}^{T} c_t(x) + \frac{\log K}{\varepsilon} \\ &< \frac{1}{1-\varepsilon}\sum_{t=1}^{T} c_t(x) + \frac{\log K}{\varepsilon} \qquad (2.4)\end{aligned}$$

using the inequality $-\frac{\log(1-x)}{x} < \frac{1}{1-x}$, valid for $x > 0$. The theorem now follows by subtracting $\sum_{t=1}^{T} c_t(x)$ from both sides of (2.4) and using the trivial upper bound $\sum_{t=1}^{T} c_t(x) \leq T$. □

2.2 The Kalai-Vempala algorithm

Figure 2-3 presents an algorithm of Kalai and Vempala [44] which solves online decision problems in which the strategy set is a compact set $\mathcal{S} \subseteq \mathbb{R}^d$ and the cost functions are linear functions on $\mathcal{S}$. We identify elements of the cost function set Γ with vectors in $\mathbb{R}^d$; such a cost vector c defines a function from $\mathcal{S}$ to $\mathbb{R}$ according to the rule $x \mapsto c \cdot x$.

Given a sequence of cost vectors $c_0, c_1, \ldots, c_T \in \mathbb{R}^d$, we will use the notation $c_{i..j}$, for $0 \leq i \leq j \leq T$, to refer to the vector

$$c_{i..j} = c_i + c_{i+1} + \ldots + c_j,$$

and we will also define

$$x_{i..j} = \arg\min_{x \in \mathcal{S}}(c_{i..j} \cdot x),$$

with the convention that if the function $c_{i..j} \cdot x$ is minimized at more than one point of $\mathcal{S}$, then $x_{i..j}$ is an arbitrary point of minimization.

Theorem 2.4. *Let Γ denote the class of all linear functions $x \mapsto c \cdot x$ on $\mathcal{S}$ represented by cost vectors c satisfying $\|c\|_1 \leq 1$. Let $\mathcal{A}$ denote the set of all oblivious adversaries for Γ. Assume that the L_1-diameter of $\mathcal{S}$ is at most D, i.e. $\|x - y\|_1 \leq D$ for all $x, y \in \mathcal{S}$. Then*

$$R(\mathsf{KV}(\varepsilon), \mathcal{A}; T) \leq \frac{D}{\varepsilon} + D\varepsilon T.$$

The proof of the theorem rests on the following lemma, which says that an algorithm which always picks $x^*_{i..j}$ has non-positive regret on the input sequence $c_i, c_{i+1}, \ldots, c_j$.

Algorithm $\mathsf{KV}(\varepsilon)$

/* Initialization */

$c_0 \leftarrow$ a uniform random vector in $[-\frac{1}{\varepsilon}, \frac{1}{\varepsilon}]^d$

/* Main loop */

for $t = 1, 2, \ldots, T$

Choose strategy $x_t = x^*_{0..t-1}$.

end

Figure 2-3: The Kalai-Vempala Algorithm

Lemma 2.5. *For $0 \le i \le j \le T$, if x is any element of $\mathcal{S}$,*

$$\sum_{t=i}^{j} c_t \cdot x^*_{i..t} \le \sum_{t=i}^{j} c_t \cdot x.$$

Proof. The proof is by induction on $j - i$, the base case $j = i$ being trivial. If $j > i$, then by the induction hypothesis

$$\sum_{t=i}^{j-1} c_t \cdot x^*_{i..t} \le \sum_{t=i}^{j-1} c_t \cdot x^*_{i..j}. \tag{2.5}$$

Adding $c_j \cdot x^*_{i..j}$ to both sides we obtain

$$\sum_{t=i}^{j} c_t \cdot x^*_{i..t} \le \sum_{t=i}^{j} c_t \cdot x^*_{i..j}, \tag{2.6}$$

and the lemma follows because for all $x \in \mathcal{S}$,

$$\sum_{t=i}^{j} c_t \cdot x^*_{i..j} \le \sum_{t=i}^{j} c_t \cdot x.$$

□

Proof of Theorem 2.4. Suppose we are given an oblivious adversary $\mathsf{ADV} \in \mathcal{A}$ represented by a sequence of cost functions $c_1, \ldots, c_T$. For any $x \in \mathcal{S}$ we may use Lemma 2.5:

$$\sum_{t=0}^{T} c_t \cdot x^*_{0..t} - \sum_{t=0}^{T} c_t \cdot x \le 0,$$

hence

$$\sum_{t=1}^{T} c_t \cdot x^*_{0..t} - \sum_{t=1}^{T} c_t \cdot x \le c_0 \cdot (x - x^*_{0..0}) \le \frac{D}{\varepsilon}. \tag{2.7}$$

This proves that if the algorithm always chose $x_t = x^*_{0..t}$ its regret would be at most $D\varepsilon$. However, instead the algorithm chooses $x_t = x^*_{0..t-1}$, incurring an additional regret equal to $\sum_{t=1}^{T} \left[\mathbf{E}(c_t \cdot x^*_{0..t-1}) - \mathbf{E}(c_t \cdot x^*_{0..t})\right]$. To finish proving the theorem, it therefore suffices to prove that

$$\mathbf{E}(c_t \cdot x^*_{0..t-1}) \le \mathbf{E}(c_t \cdot x^*_{0..t}) + D\varepsilon \tag{2.8}$$

for $1 \le t \le T$. To prove (2.8) we will produce a random variable $\tilde{x}^*_{0..t}$ with the same distribution as $x^*_{0..t}$, but coupled to $x^*_{0..t-1}$ in such a way that $\Pr(x^*_{0..t-1} = \tilde{x}^*_{0..t}) \ge 1-\varepsilon$. Let

$$\tilde{c}_0 = \begin{cases} c_0 - c_t & \text{if } c_0 - c_t \in [-1/\varepsilon, 1/\varepsilon]^d \\ -c_0 & \text{otherwise} \end{cases}.$$

The reader may verify that $\tilde{c}_0$ has the same distribution as c_0 and that $\Pr(\tilde{c}_0 + c_t = c_0) \ge 1 - \varepsilon$. Now let $\tilde{x}^*_{0..t} = \arg\min_{x \in \mathcal{S}}\{(\tilde{c}_0 + c_{1..t}) \cdot x\}$. From the properties of $\tilde{c}_0$ it follows immediately that $\tilde{x}^*_{0..t}$ has the same distribution as $x^*_{0..t}$, hence

$$\mathbf{E}(c_t \cdot \tilde{x}^*_{0..t}) = \mathbf{E}(c_t \cdot x^*_{0..t}), \tag{2.9}$$

and that $\Pr(\tilde{x}^*_{0..t} = x^*_{0..t-1}) \ge 1 - \varepsilon$, hence

$$\mathbf{E}(c_t \cdot (x^*_{0..t-1} - \tilde{x}^*_{0..t})) \le \varepsilon \sup_{x,y \in \mathcal{S}} \{c_t \cdot (x - y)\} = \varepsilon D. \tag{2.10}$$

Together, (2.9) and (2.10) establish (2.8) as desired. □

2.3 Zinkevich's algorithm

An *online convex programming problem* is an online decision problem in which the strategy set $\mathcal{S}$ is a convex subset of $\mathbb{R}^d$ (for some $d \ge 0$) and the cost function class Γ is a subset of the set of real-valued convex functions on $\mathcal{S}$. In this section, following Zinkevich [75], we make the following assumptions about $\mathcal{S}$ and Γ. Define $\|x\| = \sqrt{x \cdot x}$ and $d(x, y) = \|x - y\|$.

1. $\mathcal{S}$ is a nonempty, closed, bounded subset of $\mathbb{R}^d$. Let $\|\mathcal{S}\| = \max_{x,y \in \mathcal{S}} d(x, y)$.

2. The cost functions in Γ are differentiable convex functions, and their gradients are uniformly bounded. Let

$$\|\nabla c\| = \max\{\nabla c(x) \,|\, c \in \Gamma,\, x \in \mathcal{S}\}.$$

3. There exists an algorithm which computes, for any $y \in \mathbb{R}^d$, the vector

$$P(y) = \arg\min_{x \in \mathcal{S}} d(x, y).$$

Zinkevich, in [75], defines an algorithm called *Greedy Projection* for online convex programming problems with full feedback. The algorithm is invoked with a strategy set $\mathcal{S}$ and a sequence of *learning rates* $\eta_1, \eta_2, \ldots, \eta_T \in \mathbb{R}^+$. Greedy Projection operates as follows: it selects an arbitrary $x_1 \in \mathcal{S}$. In time step t, after learning the cost function c_t, the algorithm updates its vector by setting

$$x_{t+1} = P(x_t - \eta_t \nabla c_t(x_t)).$$

Theorem 2.6. *If $\eta_1 \geq \eta_2 \geq \ldots \geq \eta_T$, then the regret of the Greedy Projection algorithm* GP *satisfies the bound*

$$R(\mathsf{GP}, \mathcal{A}; T) \leq \frac{\|\mathcal{S}\|^2}{\eta_T} + \frac{\|\nabla c\|^2}{2} \sum_{t=1}^{T} \eta_t,$$

where $\mathcal{A}$ denotes the set of all oblivious adversaries for $(\mathcal{S}, \Gamma)$.

Proof. We will make use of the inequality

$$\|P(y) - x\|^2 \leq \|y - x\|^2, \tag{2.11}$$

valid for any $y \in \mathbb{R}^d$, $x \in \mathcal{S}$. To prove this inequality, assume without loss of generality that $P(y) = 0$, so that the inequality reduces to

$$x \cdot x \leq (y - x) \cdot (y - x) = (y \cdot y) - 2(y \cdot x) + (x \cdot x). \tag{2.12}$$

Clearly (2.12) will follow if we can prove that $y \cdot x \leq 0$. Consider the function $f(\lambda) = d(y, \lambda x)^2$ for $\lambda \in [0, 1]$. Since $\lambda x \in \mathcal{S}$ for all $\lambda \in [0, 1]$, and $0 = P(y)$, we know that the minimum of f on the interval $[0, 1]$ occurs at $\lambda = 0$; hence

$$\begin{aligned} 0 &\leq \left(\frac{\partial f}{\partial \lambda}\right)_{\lambda=0} \\ &= \frac{\partial}{\partial \lambda}\left[(y \cdot y) - 2\lambda(y \cdot x) + \lambda^2(x \cdot x)\right]_{\lambda=0} \\ &= -2y \cdot x \end{aligned}$$

which establishes that $y \cdot x \leq 0$ as desired.

Consider an oblivious adversary ADV defined by a sequence of cost functions $c_1, \ldots, c_T$, and let $x^* = \arg\min_{x \in \mathcal{S}} \sum_{t=1}^{T} c_t(x)$. Following [75], we define the potential

function $\Phi(t) = \|x_t - x^*\|^2$. To estimate the change in potential $\Phi(t+1) - \Phi(t)$, we define $y_{t+1} = x_t - \eta_t \nabla c_t(x_t)$, so that $x_{t+1} = P(y_{t+1})$. Now

$$\begin{aligned}
\Phi(t+1) - \Phi(t) &= \|P(y_{t+1}) - x^*\|^2 - \|x_t - x^*\|^2 \\
&\leq \|y_{t+1} - x^*\|^2 - \|x_t - x^*\|^2 \\
&= \|(x_t - x^*) - \eta_t \nabla c_t(x_t)\|^2 - \|x_t - x^*\|^2 \\
&= -2\eta_t \nabla c_t(x_t) \cdot (x_t - x^*) + \eta_t^2 \|\nabla c_t(x_t)\|^2,
\end{aligned}$$

i.e.

$$\begin{aligned}
\nabla c_t(x_t) \cdot (x_t - x^*) &\leq \frac{1}{2\eta_t}(\Phi(t) - \Phi(t+1)) + \frac{\eta_t}{2}\|\nabla c_t(x_t)\|^2 \\
&\leq \frac{1}{2\eta_T}(\Phi(t) - \Phi(t+1)) + \frac{\eta_t}{2}\|\nabla c\|^2. \qquad (2.13)
\end{aligned}$$

By the convexity of c_t we have

$$c_t(x) \geq c_t(x_t) + \nabla c_t(x_t) \cdot (x - x_t)$$

for all $x \in \mathcal{S}$, i.e.

$$c_t(x_t) - c_t(x) \leq \nabla c_t(x_t) \cdot (x_t - x). \qquad (2.14)$$

Combining (2.13) and (2.14) we obtain

$$\begin{aligned}
R(\mathsf{GP}, \mathsf{ADV}; T) &= \sum_{t=1}^{T} c_t(x_t) - c_t(x^*) \\
&\leq \sum_{t=1}^{T} \nabla c_t(x_t) \cdot (x_t - x^*) \\
&\leq \frac{1}{2\eta_T} \sum_{t=1}^{T} \Phi(t) - \Phi(t+1) + \frac{\|\nabla c\|^2}{2} \sum_{t=1}^{T} \eta_t \\
&= \frac{1}{2\eta_T}(\Phi(1) - \Phi(T+1)) + \frac{\|\nabla c\|^2}{2} \sum_{t=1}^{T} \eta_t \\
&\leq \frac{\|\mathcal{S}\|^2}{\eta_T} + \frac{\|\nabla c\|^2}{2} \sum_{t=1}^{T} \eta_t
\end{aligned}$$

using the fact that $\Phi(t) = \|x_t - x^*\|^2 \leq \|\mathcal{S}\|^2$ for all t. □

2.4 Multi-armed bandit algorithms I: The UCB1 algorithm

In this section we review an algorithm of Auer, Cesa-Bianchi, and Fischer [3] for the maximization version of the multi-armed problem with an i.i.d. adversary. We

assume that the strategy set $\mathcal{S}$ is $\{1, 2, \ldots, K\}$ for some K and that the set of cost functions, Γ, is equal to the set $[0,1]^K$ of all real-valued functions on $\mathcal{S}$ taking values in the interval $[0,1]$.

In fact, we will prove that the algorithm works under a somewhat more general adversarial model defined in [2], and we will require this more general result later on. To state the generalization, we make the following definition.

Definition 2.1. Let $\Gamma = \mathbb{R}^{\mathcal{S}}$. For a probability distribution P on functions $c \in \Gamma$, we define $\bar{c} \in \mathbb{R}^{\mathcal{S}}$ to be the function $\bar{c}(x) = \mathbf{E}_P(c(x))$, where $\mathbf{E}_P(\cdot)$ denotes the expectation operator defined by P. For positive real numbers ζ, s_0, we say that P is (ζ, s_0)-*bounded* if it is the case, for all $s \in [-s_0, s_0]$ and all $x \in \mathcal{S}$, that

$$\mathbf{E}_P\left(e^{s(c(x)-\bar{c}(x))}\right) \leq e^{\zeta^2 s^2/2}.$$

We assume that there exist positive constants ζ, s_0 such that the set of $\mathcal{A}$ of adversaries is a subset of the set of all i.i.d. adversaries defined by a (ζ, s_0)-bounded distribution P on $\Gamma = \mathbb{R}^K$. The following lemma justifies our assertion that this assumption generalizes the assumption that $\mathcal{A}$ is a set of i.i.d. adversaries for cost function class $[0,1]^K$

Lemma 2.7. *If P is a probability distribution on functions $c \in [0,1]^K$ then P is $(1,1)$-bounded.*

Proof. The lemma is equivalent to the following assertion: if y is a random variable taking values in $[0,1]$ and $\bar{y} = \mathbf{E}(y)$, then $\mathbf{E}(e^{s(y-\bar{y})}) \leq e^{s^2/2}$ for $|s| \leq 1$. The random variable $z = y - \bar{y}$ takes values in $[-1,1]$. Let $\lambda = \frac{z+1}{2}$, so that $\lambda \in [0,1]$ and $z = \lambda \cdot (-1) + (1-\lambda) \cdot 1$. By Jensen's inequality,

$$e^{sz} \leq \lambda \cdot e^{-s} + (1-\lambda)e^s,$$

so

$$\mathbf{E}(e^{sz}) \leq e^{-s}\mathbf{E}(\lambda) + e^s(1 - \mathbf{E}(\lambda)).$$

We have $\mathbf{E}(\lambda) = \frac{1}{2}(\mathbf{E}(z) + 1) = \frac{1}{2}$, since $\mathbf{E}(z) = 0$. Thus $\mathbf{E}(e^{sz}) \leq \frac{1}{2}(e^{-s} + e^s)$, and the lemma reduces to proving

$$\frac{1}{2}(e^{-s} + e^s) \leq e^{s^2/2}.$$

This is easily verified by writing both sides as a power series in s,

$$\sum_{n=0}^{\infty} \frac{s^{2n}}{(2n)!} \leq \sum_{n=0}^{\infty} \frac{s^{2n}}{2^n(n!)},$$

and observing that the series on both sides converge absolutely for all $s \in \mathbb{R}$ and that the series on the right dominates the one on the left term-by-term. □

Algorithm UCB1

/* Initialization */
$z(i) \leftarrow 0$ **for** $i \in \mathcal{S}$
$n(i) \leftarrow 0$ **for** $i \in \mathcal{S}$

/* Main loop */
for $t = 1, 2, \ldots, T$
if $\exists j \in \mathcal{S}$ such that $n(j) \leq 32(s_0\zeta)^{-2}\log(t)$
then
$i \leftarrow \arg\min_j n(j)$
else
$i \leftarrow \arg\max_i \left(\frac{z(i)}{n(i)} + 2\zeta\sqrt{\frac{2\log(t)}{n(i)}}\right)$
fi
Play strategy $x_t = i$.
Observe feedback y_t.
$z(i) \leftarrow z(i) + y_t$
$n(i) \leftarrow n(i) + 1$
end

Figure 2-4: The UCB1 algorithm

The algorithm UCB1 is defined in Figure 2-4. To bound its regret, we define the following notations. Suppose given a probability distribution P on $\Gamma = \mathbb{R}^K$. We define i^* to be any element of the set $\arg\max_{1\leq i\leq K} \bar{c}(i)$. For any $i \in \mathcal{S}$ we let $\Delta_i = \bar{c}(i^*) - \bar{c}(i)$. Given $\varepsilon \geq 0$ we let $\mathcal{S}_\varepsilon(\mathsf{ADV}) = \{i \in \mathcal{S} : \Delta_i > \varepsilon\}$, and we let $Z_\varepsilon(\mathsf{ADV}) = 1$ if $\mathcal{S}_\varepsilon(\mathsf{ADV}) \neq \mathcal{S}$, 0 otherwise.

Theorem 2.8. *Let $\mathcal{S} = \{1, \ldots, K\}$. Let P be a (ζ, s_0)-bounded probability distribution on $\mathbb{R}^{\mathcal{S}}$, and let* ADV *be the i.i.d. adversary with distribution P. For all $\varepsilon \geq 0$, the regret of* UCB1 *in the maximization version of the multi-armed bandit problem with adversary* ADV *satisfies*

$$R(\mathsf{UCB1}, \mathsf{ADV}; T) \leq Z_\varepsilon(\mathsf{ADV})\varepsilon T + \left[\sum_{i\in\mathcal{S}_\varepsilon(\mathsf{ADV})} \left(\frac{32\Delta_i}{s_0^2\zeta^2} + \frac{32\zeta^2}{\Delta_i}\right)\right]\log T + \left(1 + \frac{\pi^2}{3}\right)\sum_{i\in\mathcal{S}}\Delta_i.$$

Corollary 2.9. *Let $\mathcal{S} = \{1, \ldots, K\}$ and let $\mathcal{A}$ denote the set of all i.i.d. adversaries for the maximization version of the multi-armed bandit problem on $(\mathcal{S}, \mathbb{R}^{\mathcal{S}})$ defined by (ζ, s_0)-bounded probability distributions on $\mathbb{R}^{\mathcal{S}}$. Then for all $\mathsf{ADV} \in \mathcal{A}$*

$$\limsup_{T\to\infty} \frac{R(\mathsf{UCB1}, \mathsf{ADV}; T)}{\log(T)} < \infty. \tag{2.15}$$

Also,

$$R(\mathsf{UCB1}, \mathcal{A}; T) = O(K + \sqrt{KT\log(T)}). \tag{2.16}$$

Proof. Taking $\varepsilon = 0$ in Theorem 2.8 gives (2.15), and taking $\varepsilon = \sqrt{\frac{K\log(T)}{T}}$ gives (2.16). □

The proof of Theorem 2.8 depends on the following tail inequality which generalizes Azuma's martingale inequality [59]. Recall that a sequence of random variables $X_0, X_1, \ldots, X_n$ is a *martingale sequence* if it satisfies

$$\mathbf{E}(X_i \,\|\, X_1, \ldots, X_{i-1}) = X_{i-1}$$

for $1 \le i \le n$.

Lemma 2.10. *Suppose that $X_0, X_1, \ldots, X_n$ is a martingale sequence satisfying $X_0 = 0$ and*

$$\mathbf{E}(e^{s(X_i - X_{i-1})} \,\|\, X_1, \ldots, X_{i-1}) \le e^{\zeta^2 s^2/2}$$

for $|s| \le s_0$ and $1 \le i \le n$. Then

$$\Pr(X_n \ge \lambda) \le e^{-\frac{\lambda^2}{2\zeta^2 n}}$$

for $\lambda \le \frac{1}{2} s_0 \zeta^2 n$.

Proof. Let $z_i = X_i - X_{i-1}$ for $1 \le i \le n$, so that $X_n = \sum_{i=1}^n z_i$. Put $s = \frac{\lambda}{\zeta^2 n}$, and note that $|s| \le s_0$. We have

$$\begin{aligned}
\mathbf{E}\left(e^{sX_n}\right) &= \mathbf{E}\left(e^{sz_n} e^{sX_{n-1}}\right) \\
&= \mathbf{E}\left(\mathbf{E}\left(e^{sz_n} \,\|\, X_1, \ldots, X_{n-1}\right) e^{sX_{n-1}}\right) \\
&\le e^{\zeta^2 s^2/2} \mathbf{E}\left(e^{sX_{n-1}}\right).
\end{aligned}$$

By induction,

$$\mathbf{E}\left(e^{sX_n}\right) \le e^{n\zeta^2 s^2/2} = e^{s\lambda/2}.$$

By Markov's inequality,

$$\begin{aligned}\Pr(X_n \geq \lambda) &\leq \Pr(e^{sX_n} \geq e^{s\lambda}) \\ &\leq e^{-s\lambda}\mathbf{E}(e^{sX_n}) \\ &\leq e^{-s\lambda/2} \\ &= e^{-\frac{\lambda^2}{2\zeta^2 n}}.\end{aligned}$$

□

The following proof of Theorem 2.8 is a modification of the proof of Theorem 1 in [3].

Proof of Theorem 2.8. For a strategy $i \in \mathcal{S}$ and for $1 \leq t \leq T$, let $z_t(i), n_t(i)$ denote the values of the variables $z(i), n(i)$ in Algorithm UCB1 at the start of trial t, i.e. $z_t(i)$ is the sum of feedback values received for trials prior to t in which i was selected, and $n_t(i)$ is the number of such trials. Let $v_t(i) = \frac{z_t(i)}{n_t(i)} + 2\zeta\sqrt{\frac{2\log(t)}{n_t(i)}}$.

For $k \geq 0$ let $\tau_k(i) = \max\{t : n_t(i) \leq k\}$, i.e. $\tau_k(i)$ equals the number of the $(k+1)$-st trial in which UCB1 selected strategy i if there is such a trial, otherwise $\tau_k(i) = T$. Let

$$X_k(i) = z_{\tau_k(i)}(i) - n_{\tau_k(i)}(i)\bar{c}(i),$$

i.e. $X_k(i)$ is the amount by which the total feedback observed for strategy i exceeds its expected value (conditional on the number of times i has been selected) after the k-th trial in which i is selected, or at time T if i is selected fewer than k times. For any i, the sequence $0 = X_0(i), X_1(i), \ldots, X_T(i)$ is a martingale sequence satisfying the hypotheses of Lemma 2.10. This leads to the following claim.

Claim 2.11. *For any $t > 0$ and $k > 32(s_0\zeta)^{-2}\log(t)$,*

$$\Pr\left(\frac{1}{k}X_k(i) > 2\zeta\sqrt{\frac{2\log t}{k}}\right) < t^{-4} \qquad \textit{and} \qquad \Pr\left(\frac{1}{k}X_k(i) < -2\zeta\sqrt{\frac{2\log t}{k}}\right) < t^{-4}.$$

Proof. The hypothesis that $k > 32(s_0\zeta)^{-2}\log(t)$ ensures that $2\zeta\sqrt{2k\log(t)} \leq \frac{1}{2}s_0\zeta^2 k$. Applying Lemma 2.10 with $\lambda = 2\zeta\sqrt{k\log(t)}$, we obtain

$$\Pr\left(X_k(i) > 2\zeta\sqrt{2k\log(t)}\right) < e^{-\frac{8\zeta^2 k\log(t)}{2\zeta^2 k}} = t^{-4},$$

which proves the first half of the claim. The second half follows by applying the same argument to the martingale $-X_0(i), -X_1(i), \ldots, -X_T(i)$. □

The following manipulation allows us to reduce the theorem to the task of proving a bound on $n_T(i)$ for each $i \in \mathcal{S}$.

$$\begin{aligned}
\sum_{t=1}^{T} c_t(x_t) - c_t(i^*) &= \sum_{i\in\mathcal{S}} \sum_{t\,:\,x_t=i} c_t(i) - \sum_{t=1}^{T} c_t(i^*) \\
&= \sum_{i\in\mathcal{S}} n_T(i)\bar{c}(i) + \sum_{i\in\mathcal{S}} \sum_{t\,:\,x_t=i} (c_t(i) - \bar{c}(i)) - \sum_{t=1}^{T} c_t(i^*) \\
&= \sum_{i\in\mathcal{S}} n_T(i)\Delta_i + \sum_{i\in\mathcal{S}} X_T(i) + \sum_{t=1}^{T} (\bar{c}(i^*) - c_t(i^*)). \qquad (2.17)
\end{aligned}$$

When we take the expected value of both sides of (2.17), the second and third terms on the right side vanish and we are left with:

$$R(\mathsf{ALG}, \mathsf{ADV}; T) = \sum_{i\in\mathcal{S}} \mathbf{E}(n_T(i))\Delta_i.$$

We may split the sum on the right side into two sums according to whether or not $i \in \mathcal{S}_\varepsilon(\mathsf{ADV})$ to obtain:

$$\begin{aligned}
R(\mathsf{UCB1}, \mathsf{ADV}; T) &= \sum_{i\notin\mathcal{S}_\varepsilon(\mathsf{ADV})} \mathbf{E}(n_T(i))\Delta_i + \sum_{i\in\mathcal{S}_\varepsilon(\mathsf{ADV})} \mathbf{E}(n_T(i))\Delta_i \\
&\le Z_\varepsilon(\mathsf{ADV})\varepsilon T + \sum_{i\in\mathcal{S}_\varepsilon(\mathsf{ADV})} \mathbf{E}(n_T(i))\Delta_i
\end{aligned}$$

so the theorem reduces to proving

$$\mathbf{E}(n_T(i)) \le \left\lceil \frac{32\log T}{(s_0\zeta)^2} + \frac{32\zeta^2\log(T)}{\Delta_i^2} \right\rceil + 1 + \frac{\pi^2}{3}. \qquad (2.18)$$

Let

$$Q_i = \left\lceil \frac{32\log(T)}{(s_0\zeta)^2} + \frac{32\zeta^2\log(T)}{\Delta_i^2} \right\rceil.$$

Observe that

$$\begin{aligned}
\mathbf{E}(n_T(i)) - Q_i &= \sum_{k=Q_i}^{T} \Pr(\exists t\ n_t(i) = k \ \wedge\ x_t = i) \\
&\le \sum_{t=1}^{T} \sum_{k=Q_i}^{t} \sum_{\ell=1}^{t} \Pr(n_t(i) = k \ \wedge\ n_t(i^*) = \ell \ \wedge\ x_t = i) \qquad (2.19)
\end{aligned}$$

Let $\mathcal{E}(t,k,\ell)$ denote the event "$(n_t(i) = k) \ \wedge\ (n_t(i^*) = \ell) \ \wedge\ (x_t = i)$" for $1 \le t \le T$, $Q_i < k \le T$, $1 \le \ell \le T$. We claim that $\Pr(\mathcal{E}(t,k,\ell)) < 2t^{-4}$. Since $k > Q_i >$

$32(s_0\zeta)^{-2}\log(T)$, event $\mathcal{E}(t,k,\ell)$ implies that UCB1 chose, at time t, a strategy i satisfying $n_t(i) > 32(s_0\zeta)^{-2}\log(t)$. Due to the structure of the algorithm, this implies that $v_t(i) \geq v_t(i^*)$ and also that $n_t(i^*) > 32(s_0\zeta)^{-2}\log(t)$. Assuming $\mathcal{E}(t,k,\ell)$,

$$v_t(i) = \frac{z_t(i)}{n_t(i)} + 2\zeta\sqrt{\frac{2\log(t)}{n_t(i)}} = \frac{1}{k}X_k(i) + \bar{c}(i) + 2\zeta\sqrt{\frac{2\log(t)}{k}}$$

and similarly

$$v_t(i^*) = \frac{1}{\ell}X_\ell(i^*) + \bar{c}(i^*) + 2\zeta\sqrt{\frac{2\log(t)}{\ell}}.$$

Hence the inequality $v_t(i) - v_t(i^*) \geq 0$ implies

$$\left[\frac{1}{k}X_k(i) + 2\zeta\sqrt{\frac{2\log(t)}{k}} - \Delta_i\right] + \left[-\frac{1}{\ell}X_\ell(i^*) - 2\zeta\sqrt{\frac{2\log(t)}{\ell}}\right] \geq 0. \tag{2.20}$$

The fact that $k > Q_i > 32\zeta^2\log(T)/\Delta_i^2$ implies that $\Delta_i > 4\zeta\sqrt{\frac{2\log(t)}{k}}$, so Claim 2.11 implies that the first term on the left side of (2.20) has probability less than t^{-4} of being non-negative. Likewise, Claim 2.11 also implies that the second term on the right side of (2.20) has probability less than t^{-4} of being non-negative. But $\mathcal{E}(t,k,\ell)$ implies that at least one of these two terms is non-negative, so we conclude that $\Pr(\mathcal{E}(t,k,\ell)) < 2t^{-4}$ as claimed.

Plugging this estimate back into (2.19) one obtains

$$\mathbf{E}(n_T(i)) - Q_i \leq \sum_{t=1}^{T}\sum_{k=Q_i}^{t}\sum_{\ell=1}^{t} 2t^{-4} \leq \sum_{t=1}^{T} 2t^{-2} = \frac{\pi^2}{3},$$

which confirms (2.18) and completes the proof. □

2.5 Multi-armed bandit algorithms II: The Exp3 algorithm

This section introduces an algorithm of Auer, Cesa-Bianchi, Freund, and Schapire [4] for the multi-armed bandit problem with an adaptive adversary. As before, we assume that the strategy set $\mathcal{S}$ is $\{1, 2, \ldots, K\}$ for some K and that the set of cost functions, Γ, is equal to the set $[0,1]^K$ of all real-valued functions on $\mathcal{S}$ taking values in the interval $[0,1]$. The algorithm uses, as a subroutine, the algorithm Hedge presented in Section 2.1.

The regret of Exp3 satisfies an upper bound specified in the following theorem.

Algorithm Exp3

/* Initialization */
$\gamma \leftarrow \min\left\{\frac{1}{2}, \sqrt{\frac{K \log K}{T}}\right\}$
Initialize an instance of Hedge(γ/K) with strategy set $\mathcal{S} = \{1, 2, \ldots, K\}$.

/* Main loop */
for $t = 1, 2, \ldots, T$
Let p_t be the probability distribution on $\mathcal{S}$ reported by Hedge(γ/K).
$\hat{p}_t(i) \leftarrow (1-\gamma)p_t(i) + \gamma/K$ **for** $i \in \mathcal{S}$.
Sample $x_t \in \mathcal{S}$ using distribution $\hat{p}_t$.
Observe feedback y_t.

/* Create simulated cost function $\hat{c}_t$ to feed back to Hedge. */
$\hat{c}_t(x_t) \leftarrow y_t/\hat{p}_t(x_t)$
$\hat{c}_t(i) \leftarrow 0$ **for** $i \in \mathcal{S} \setminus \{x_t\}$

Present $\hat{c}_t$ as feedback to Hedge(γ/K).
end /* End of main loop */

Figure 2-5: The algorithm Exp3

Theorem 2.12. *The algorithm* Exp3 *satisfies*

$$R(\text{Exp3}, \mathcal{A}; T) = O(\sqrt{TK \log K}),$$

where $\mathcal{A}$ denotes the set of all adaptive adversaries for the multi-armed bandit problem with strategy set $\mathcal{S} = \{1, 2, \ldots, K\}$.

Before proving Theorem 2.12, we will need the following lemmas.

Lemma 2.13. *For $x \geq 0$ and $0 < \varepsilon < 1$,*

$$(1-\varepsilon)^x \leq 1 + \log(1-\varepsilon)x + \log^2(1-\varepsilon)x^2.$$

Proof. Let $f(x) = (1-\varepsilon)^x$ and $g(x) = 1 + \log(1-\varepsilon)x + \log^2(1-\varepsilon)x^2$. We have

$$f(0) = 1 = g(0)$$

$$f'(0) = \log(1-\varepsilon) = g'(0)$$

$$f''(0) = \log^2(1-\varepsilon) < 2\log^2(1-\varepsilon) = g''(0)$$

which demonstrates that $f(x) < g(x)$ for sufficiently small positive x. If $f(x) > g(x)$ for some $x > 0$, then by the intermediate value theorem there exists $x_0 > 0$ with $f(x_0) = g(x_0)$. By the mean value theorem applied to the function $f - g$, we conclude that there exists $x_1 \in (0, x_0)$ such that $f'(x_1) = g'(x_1)$. By the mean value theorem applied to the function $f' - g'$, there must exist $x_2 \in (0, x_1)$ such that $f''(x_2) = g''(x_2)$. But this is impossible, since $g''(x_2) = 2\log^2(1-\varepsilon)$ while $f''(x_2) = (1-\varepsilon)^{x_2}\log^2(1-\varepsilon) < 2\log^2(1-\varepsilon)$. □

Lemma 2.14. *The probability distributions $p_t(\cdot)$ computed by* Hedge(ε) *satisfy the following inequality for all $x \in \mathcal{S}$:*

$$\sum_{t=1}^{T}\sum_{i\in\mathcal{S}} p_t(i)c_t(i) \le \sum_{t=1}^{T} c_t(x) - \log(1-\varepsilon)\sum_{t=1}^{T}\sum_{i\in\mathcal{S}} p_t(i)c_t(i)^2 + \frac{\log K}{\varepsilon}.$$

Proof. The proof is parallel to the proof of Theorem 2.3. Let $W_t = \sum_{i\in\mathcal{S}} w_t(i)$. Using Lemma 2.13 we obtain the bound:

$$\begin{aligned}
\frac{W_t}{W_{t-1}} &= \frac{\sum_{i\in\mathcal{S}}(1-\varepsilon)^{c_t(i)}w_{t-1}(i)}{\sum_{i\in\mathcal{S}} w_{t-1}(i)} \\
&\le \frac{\sum_{i\in\mathcal{S}}[1 + \log(1-\varepsilon)c_t(i) + \log^2(1-\varepsilon)c_t(i)^2]w_{t-1}(i)}{\sum_{i\in\mathcal{S}} w_{t-1}(i)} \\
&= 1 + \log(1-\varepsilon)\sum_{i\in\mathcal{S}} p_t(i)c_t(i) + \log^2(1-\varepsilon)\sum_{i\in\mathcal{S}} p_t(i)c_t(i)^2
\end{aligned}$$

hence

$$\log(W_t/W_{t-1}) \le \log(1-\varepsilon)\sum_{i\in\mathcal{S}} p_t(i)c_t(i) + \log^2(1-\varepsilon)\sum_{i\in\mathcal{S}} p_t(i)c_t(i)^2$$

and

$$\begin{aligned}
\log(W_T) &= \log(W_0) + \sum_{t=1}^{T}\log\left(\frac{W_t}{W_{t-1}}\right) \\
&\le \log(K) + \log(1-\varepsilon)\sum_{t=1}^{T}\sum_{i\in\mathcal{S}} p_t(i)c_t(i) \\
&\qquad + \log^2(1-\varepsilon)\sum_{t=1}^{T}\sum_{i\in\mathcal{S}} p_t(i)c_t(i)^2. \qquad (2.21)
\end{aligned}$$

But for any $x \in \mathcal{S}$, $W_T \ge w_T(x) = (1-\varepsilon)^{\sum_{t=1}^{T} c_t(x)}$, hence

$$\log(W_T) \ge \log(1-\varepsilon)\sum_{t=1}^{T} c_t(x). \qquad (2.22)$$

Combining (2.21) and (2.22) and rearranging terms we obtain

$$\sum_{t=1}^{T}\sum_{i\in\mathcal{S}} p_t(i)c_t(i) \;\leq\; \sum_{t=1}^{T} c_t(x) - \log(1-\varepsilon)\sum_{t=1}^{T}\sum_{i\in\mathcal{S}} p_t(i)c_t(i)^2 - \frac{\log K}{\log(1-\varepsilon)}$$

and the lemma now follows using the inequality $-\frac{1}{\log(1-x)} < \frac{1}{x}$, valid for $0 < x < 1$. □

Proof of Theorem 2.12. If $T < 4K\log K$ then the theorem follows from the trivial observation that $R(\mathsf{Exp3}, \mathcal{A}; T) \leq T < 2\sqrt{TK\log K}$. So assume from now on that $T \geq 4K\log K$ and, consequently, $\gamma = \sqrt{K\log(K)/T}$. We have

$$\begin{aligned}
\mathbf{E}\left[\sum_{t=1}^{T} c_t(x_t)\right] &= \mathbf{E}\left[\sum_{t=1}^{T}\sum_{i\in\mathcal{S}} \hat{p}_t(i)c_t(i)\right] \\
&= (1-\gamma)\mathbf{E}\left[\sum_{t=1}^{T}\sum_{i\in\mathcal{S}} p_t(i)c_t(i)\right] + \frac{\gamma}{K}\mathbf{E}\left[\sum_{t=1}^{T}\sum_{i\in\mathcal{S}} c_t(i)\right] \\
&\leq \mathbf{E}\left[\sum_{t=1}^{T}\sum_{i\in\mathcal{S}} p_t(i)c_t(i)\right] + \gamma T
\end{aligned}$$

and $\gamma T = \sqrt{TK\log K}$, so it suffices to prove that

$$\mathbf{E}\left[\sum_{t=1}^{T}\sum_{i\in\mathcal{S}} p_t(i)c_t(i) - \sum_{t=1}^{T} c_t(x)\right] = O\left(\sqrt{TK\log K}\right) \tag{2.23}$$

for every $x \in \mathcal{S}$. Let $\mathcal{F}_{<t}$ denote the σ-field generated by the random variables $x_1, \ldots, x_{t-1}$ and $c_1, \ldots, c_t$. The reader may verify, from the definition of $\hat{c}_t$, that

$$\mathbf{E}[\hat{c}_t(i) \,\|\, \mathcal{F}_{<t}] = c_t(i)$$

and therefore,

$$\begin{aligned}
\mathbf{E}[p_t(i)c_t(i)] &= \mathbf{E}[p_t(i)\hat{c}_t(i)] \\
\mathbf{E}[c_t(x)] &= \mathbf{E}[\hat{c}_t(x)].
\end{aligned}$$

Hence, to prove (2.23) it suffices to prove that

$$\mathbf{E}\left[\sum_{t=1}^{T}\sum_{i\in\mathcal{S}} p_t(i)\hat{c}_t(i) - \sum_{t=1}^{T}\hat{c}_t(x)\right] = O\left(\sqrt{TK\log K}\right) \tag{2.24}$$

By Lemma 2.14, the left side of (2.24) is bounded above by

$$-\log\left(1-\frac{\gamma}{K}\right)\mathbf{E}\left[\sum_{t=1}^{T}\sum_{i\in\mathcal{S}} p_t(i)\hat{c}_t(i)^2\right] + \frac{\log K}{\gamma/K}.$$

The second term is $\sqrt{TK \log K}$, and the factor $-\log(1-\gamma/K)$ in the first term is bounded above by $2\log(2) \cdot \gamma/K$ using the inequality $-\log(1-x) \leq 2\log(2) \cdot x$, valid for $0 < x \leq 1/2$. Thus it remains to prove that

$$\frac{\gamma}{K} \mathbf{E}\left[\sum_{t=1}^{T} \sum_{i \in \mathcal{S}} p_t(i) \hat{c}_t(i)^2\right] = O(\sqrt{TK \log K}).$$

We have

$$\begin{aligned} \sum_{i \in \mathcal{S}} p_t(i) \hat{c}_t(i)^2 &= \sum_{i \in \mathcal{S}} \frac{p_t(i)}{\hat{p}_t(i)} c_t(i) \hat{c}_t(i) \\ &\leq \sum_{i \in \mathcal{S}} \frac{1}{1-\gamma} \hat{c}_t(i) \\ &\leq 2 \sum_{i \in \mathcal{S}} \hat{c}_t(i) \end{aligned}$$

and, consequently,

$$\begin{aligned} \frac{\gamma}{K} \mathbf{E}\left[\sum_{t=1}^{T} \sum_{i \in \mathcal{S}} p_t(i) \hat{c}_t(i)^2\right] &\leq \frac{2\gamma}{K} \mathbf{E}\left[\sum_{t=1}^{T} \sum_{i \in \mathcal{S}} \hat{c}_t(i)\right] \\ &= \frac{2\gamma}{K} \mathbf{E}\left[\sum_{t=1}^{T} \sum_{i \in \mathcal{S}} c_t(i)\right] \\ &\leq \frac{2\gamma}{K} \cdot TK \\ &= 2\sqrt{TK \log(K)} \end{aligned}$$

as desired. □

The reader who wishes to keep track of the constants in the proof of Theorem 2.12 will see that they are not bad: the proof actually establishes that

$$R(\mathsf{Exp3}, \mathcal{A}; T) \leq (2 + 4\log 2)\sqrt{TK \log K} < 5\sqrt{TK \log K}.$$

2.6 Known versus unknown time horizon

In designing algorithms for online decision problems, there is a simple but powerful *doubling* technique which usually enables us to transform algorithms which have foreknowledge of the time horizon T into algorithms which lack such foreknowledge, at the cost of only a constant factor in the regret bound. In other words, the technique allows us to take statements of the form, "For every T there exists an algorithm with regret $R(T)$," and transform them into statements of the form,"There exists an algorithm such that for every T, the regret is $O(R(T))$."

Theorem 2.15. *Let $(\mathcal{S}, \Gamma, \Phi, F)$ be an online decision problem, $\mathcal{A}$ a set of adversaries, and T_0 a non-negative integer. Suppose that for every $T > T_0$ there is an algorithm $\mathsf{ALG}(T)$ satisfying $R(\mathsf{ALG}(T), \mathcal{A}; T) \leq R(T)$. If R is non-decreasing and satisfies $R(2T) > C \cdot R(T)$ for some constant $C > 1$ and for all $T > T_0$, then there exists an algorithm ALG such that*

$$R(\mathsf{ALG}, \mathcal{A}; T) < \left(\frac{C}{C-1}\right) R(T)$$

for all T.

Proof. The algorithm ALG operates as follows. Whenever the time t is a power of 2, say $t = 2^k$, it initializes an instance of $\mathsf{ALG}(t)$ and runs this instance for the next t trials. Let T be a positive integer and let $\ell = \lfloor \log_2(T) \rfloor$. For any $\mathsf{ADV} \in \mathcal{A}$ and $x \in \mathcal{S}$, we have

$$\begin{aligned}
R(\mathsf{ALG}, \mathsf{ADV}; x, T) &= \mathbf{E}\left[\sum_{t=1}^{T} c_t(x_t) - c_t(x)\right] \\
&= \sum_{k=0}^{\ell} \mathbf{E}\left[\sum_{2^k \leq t < \min\{2^{k+1}, T+1\}} c_t(x_t) - c_t(x)\right] \\
&\leq \sum_{k=0}^{\ell} R(\mathsf{ALG}(2^k), \mathsf{ADV}; x, 2^k) \\
&\leq \sum_{k=0}^{\ell} R(2^k) \\
&< \sum_{k=0}^{\ell} C^{k-\ell} R(2^\ell) \\
&< \frac{C}{C-1} R(T).
\end{aligned}$$

□

2.7 Kullback-Leibler Divergence

The Kullback-Leibler divergence (also known as "KL-divergence" or "relative entropy") is a measure of the statistical distinguishability of two probability distributions. For probability distributions on finite sets, an excellent treatment of the Kullback-Leibler divergence can be found in [26]. In this work, we will have occasion to work with the Kullback-Leibler divergence of distributions on infinite sets. While

the relevant definitions and theorems do not differ substantially from the finite case, it is difficult to find an adequate exposition of such topics in the literature. Accordingly, we present here a self-contained treatment of the relevant theory.

2.7.1 Review of measure and integration theory

We first present some definitions from measure and integration theory. A more leisurely exposition of the same definitions may be found in [19] or [64].

Definition 2.2 (Measurable space, measure space, probability space). A *measurable space* is an ordered pair $(\Omega, \mathcal{F})$, where Ω is a set and $\mathcal{F}$ is a collection of subsets of Ω satisfying:

1. $\Omega \in \mathcal{F}$.
2. If $A \in \mathcal{F}$ then $\Omega \setminus A \in \mathcal{F}$.
3. If $\{A_i : i \in \mathbb{N}\}$ is a countable collection of elements of $\mathcal{F}$ then $\bigcup_{i\in\mathbb{N}} A_i \in \mathcal{F}$.

If $\mathcal{F}$ satisfies these properties, we say it is a σ-field on Ω.

A *measure space* is an ordered triple $(\Omega, \mathcal{F}, \mu)$ where $(\Omega, \mathcal{F})$ is a measurable space and $\mu : \mathcal{F} \to [0, \infty]$ is a real-valued set function satisfying

$$\mu\left(\bigcup_{i\in\mathbb{N}} A_i\right) = \sum_{i\in\mathbb{N}} \mu(A_i)$$

when $\{A_i : i \in \mathbb{N}\}$ is a countable collection of disjoint sets in $\mathcal{F}$. We refer to μ as a *measure* on $(\Omega, \mathcal{F})$. If $\mu(\Omega) = 1$ then we say that μ is a *probability measure* and that $(\Omega, \mathcal{F}, \mu)$ is a *probability space*.

Definition 2.3 (Measurable mapping, induced measure). If $(\Omega, \mathcal{F})$ and $(\Omega', \mathcal{F}')$ are measurable spaces, a function $f : \Omega \to \Omega'$ is called a *measurable mapping* if $f^{-1}(A) \in \mathcal{F}$ for every $A \in \mathcal{F}'$. If μ is a measure on $(\Omega, \mathcal{F})$ and $f : \Omega \to \Omega'$ is a measurable mapping, then there is a measure $f_*\mu$ defined on $(\Omega', \mathcal{F}')$ by

$$(f_*\mu)(A) = \mu(f^{-1}(A)).$$

We refer to $f_*\mu$ as the measure *induced* by $f : \Omega \to \Omega'$.

Definition 2.4 (Borel measurable function, simple function, σ-simple function, random variable). The Borel algebra on $\mathbb{R} \cup \{\pm\infty\}$ is the unique minimal σ-field $\mathfrak{R}$ containing all closed intervals. If $(\Omega, \mathcal{F})$ is a measure space, a function $f : \Omega \to \mathbb{R} \cup \{\pm\infty\}$ is *Borel measurable* if f is a measurable mapping from $(\Omega, \mathcal{F})$

to $(\mathbb{R}, \mathfrak{R})$. A Borel measurable function f is a *simple function* if its range is a finite subset of $\mathbb{R}$. It is a *σ-simple function* if its range is a countable subset of $\mathbb{R} \cup \{\pm\infty\}$. When $(\Omega, \mathcal{F}, \mu)$ is a probability space, we refer to a Borel measurable function f as a *random variable* defined on Ω.

Definition 2.5 (Lebesgue integral). If f is a simple function then it may be expressed as a finite sum

$$f = \sum_{i=1}^{n} \alpha_i \chi_{A_i}$$

where $A_1, A_2, \ldots, A_n$ are disjoint sets in $\mathcal{F}$ and χ_{A_i} is a function which takes the value 1 on A_i, 0 elsewhere. For a measure μ on $(\Omega, \mathcal{F})$, we define the *integral* of such a function f by

$$\int f \, d\mu = \sum_{i=1}^{n} \alpha_i \mu(A_i).$$

For a non-negative Borel measurable function f, we define the *Lebesgue integral* by

$$\int f \, d\mu = \sup_{g} \int g \, d\mu,$$

where the supremum is over all simple functions g satisfying $0 \leq g \leq f$. For a Borel measurable function f which assumes both positive and negative values, we set

$$\begin{aligned} f^+ &= \max\{0, f\} \\ f^- &= \max\{0, -f\} \\ \int f \, d\mu &= \int f^+ d\mu - \int f^- d\mu. \end{aligned}$$

The integral is well-defined as long as at least one of $\int f^+ d\mu$, $\int f^- d\mu$ is finite.

When f is a Borel measurable function on $(\Omega, \mathcal{F})$ and $A \in \mathcal{F}$, we will use the notation $\int_A f d\mu$ to denote $\int f \chi_A \, d\mu$.

We will need the following characterization of the Lebesgue integral in terms of σ-simple functions.

Theorem 2.16. *If f is an $\mathbb{R}$-valued Borel measurable function whose integral $\int f \, d\mu$ is well-defined, then*

$$\int f \, d\mu = \sup_{g} \int g \, d\mu,$$

where the supremum is over all σ-simple functions g satisfying $g \leq f$.

Proof. It suffices to prove that

$$\int f\,d\mu \leq \sup_g \int g\,d\mu, \tag{2.25}$$

since the reverse inequality is clear from the definition of the Lebesgue integral. If $\int f\,d\mu = -\infty$ then (2.25) is obvious. If $\int f\,d\mu > -\infty$ then we will construct a sequence of σ-simple functions $g_n \leq f$ $(n = 1, 2, \ldots)$ such that $\int g_n\,d\mu \to \int f\,d\mu$. Let

$$G(x) = \begin{cases} -2^{\lceil \log_2(-f(x)) \rceil} & \text{if } f(x) < 0 \\ 0 & \text{if } f(x) \geq 0, \end{cases}$$

i.e. $-G(x)$ is the smallest power of 2 in the interval $[-f(x), \infty)$ if $f(x) < 0$, otherwise $-G(x) = 0$. Let

$$G_n(x) = 2^{-n}\lfloor 2^n f(x) \rfloor,$$

i.e. $G_n(x)$ is the largest multiple of 2^{-n} in the interval $(-\infty, f(x)]$. Finally, let $g_n(x) = \max\{G(x), G_n(x)\}$. Note that $g_1 \leq g_2 \leq \ldots \leq f$ and that g_n converges pointwise to f. It follows, by Lebesgue's monotone convergence theorem [64], that $\int g_n^+ d\mu \to \int f^+ d\mu$. Moreover, we have

$$0 \leq g_n^-(x) \leq -G(x) \leq 2f^-.$$

By assumption $\int f\,d\mu > -\infty$ so $\int f^- d\mu < \infty$. This implies, by Lebesgue's dominated convergence theorem [64], that $\int g_n^-\,d\mu \to \int f^-\,d\mu$. □

Definition 2.6 (Absolutely continuous). If $(\Omega, \mathcal{F})$ is a measurable space and μ, ν are two measures on $(\Omega, \mathcal{F})$, we say that ν is *absolutely continuous* with respect to μ, denoted by $\nu \ll \mu$, if $\nu(A) = 0$ whenever $A \in \mathcal{F}$ and $\mu(A) = 0$.

Definition 2.7 (Radon-Nikodym derivative). If μ, ν are two measures on a measurable space $(\Omega, \mathcal{F})$, a measurable function $\rho : \Omega \to [0, \infty]$ is called a *Radon-Nikodym derivative* of ν with respect to μ if $\nu(A) = \int_A \rho\,d\mu$ for all $A \in \mathcal{F}$.

Theorem 2.17 (Radon-Nikodym Theorem). *If $\nu \ll \mu$ then ν has a Radon-Nikodym derivative with respect to μ. If ρ, τ are two Radon-Nikodym derivatives of ν with respect to μ then $\rho = \tau$ almost everywhere, i.e. $\mu(\{x : \rho(x) \neq \tau(x)\}) = 0$.*

2.7.2 Definition of KL-divergence

Our definition of Kullback-Leibler divergence parallels the definition of the Lebesgue integral.

Definition 2.8 (Simple measurable space, simple probability space). For any set X, the set 2^X of all subsets of X is a σ-field on X. If X is a finite set, we refer to the measurable space $(X, 2^X)$ as a *simple measurable space*. A probability space $(\Omega, \mathcal{F}, \mu)$ is a *simple probability space* if $(\Omega, \mathcal{F})$ is a simple measurable space. If $(\Omega, \mathcal{F}, \mu)$ is a simple probability space and $x \in \Omega$, we will sometimes write $\mu(x)$ as a shorthand for $\mu(\{x\})$.

Definition 2.9 (Kullback-Leibler divergence). If $(X, 2^X)$ is a simple measurable space and μ, ν are two probability measures on $(X, 2^X)$, satisfying $\nu \ll \mu$, their *simple Kullback-Leibler divergence* is the sum

$$\mathcal{KL}(\mu \,\|\, \nu) = \sum_{x \in X} \log\left(\frac{\mu(x)}{\nu(x)}\right) \mu(x).$$

Here, we interpret the term $\log\left(\frac{\mu(x)}{\nu(x)}\right)\mu(x)$ to be equal to 0 if $\mu(x) = \nu(x) = 0$, and to be equal to $+\infty$ if $\mu(x) > 0$, $\nu(x) = 0$.

If $(\Omega, \mathcal{F})$ is a measurable space and μ, ν are two probability measures on $(\Omega, \mathcal{F})$ satisfying $\nu \ll \mu$, their *Kullback-Leibler divergence* is defined by

$$KL(\mu \,\|\, \nu) = \sup_{(f,X)} \mathcal{KL}(f_*\mu \,\|\, f_*\nu),$$

where the supremum is taken over all pairs (f, X) such that X is a finite subset of $\mathbb{N}$ and f is a measurable mapping from $(\Omega, \mathcal{F})$ to $(X, 2^X)$. (The stipulation that $X \subset \mathbb{N}$ is necessary only in order to avoid set-theoretic difficulties, by ensuring that the collection of such pairs (f, X) is actually a set.)

The following lemma is sometimes referred to as the "data processing inequality for KL-divergence," since it may roughly be interpreted as saying that one cannot increase the KL-divergence of two probability measures by throwing away information about the sample points.

Lemma 2.18. *Suppose that $(\Omega, \mathcal{F})$ and $(\Omega', \mathcal{F}')$ are measurable spaces and that $f : \Omega \to \Omega'$ is a measurable mapping. If μ, ν are probability measures on $(\Omega, \mathcal{F})$ such that $\nu \ll \mu$ then $KL(\mu \,\|\, \nu) \geq KL(f_*\mu \,\|\, f_*\nu)$.*

Proof. If $(X, 2^X)$ is a simple measurable space and $g : \Omega' \to X$ is a measurable mapping, then the composition $g \circ f : \Omega \to X$ is a measurable mapping and

$$\begin{aligned} g_*(f_*\mu) &= (g \circ f)_*\mu \\ g_*(f_*\nu) &= (g \circ f)_*\nu. \end{aligned}$$

The lemma now follows immediately from the definition of KL-divergence. □

Some authors define the *differential Kullback-Leibler divergence* of two distributions on $\mathbb{R}$ with density functions f, g to be the integral

$$KL(f \,\|\, g) = \int \log\left(\frac{f(x)}{g(x)}\right) f(x)\, dx,$$

when this integral is well-defined. We relate this definition to ours via the following theorem.

Theorem 2.19. *If μ, ν are two probability measures on $(\Omega, \mathcal{F})$ satisfying $\nu \ll \mu$ and ρ is a Radon-Nikodym derivative of ν with respect to μ, then*

$$KL(\mu \,\|\, \nu) = -\int \log(\rho) d\mu. \tag{2.26}$$

Proof. Let $f = \log(\rho)$. We will first establish that $\int f d\mu$ is well-defined by proving that $\int f^+ d\mu < \infty$. Let $A = \{x : \rho(x) \geq 1\}$. Using the fact that $\log(x) \leq x - 1$ for $x > 0$, we find that

$$\int f^+ d\mu = \int_A \log(\rho) d\mu \leq \int_A (\rho - 1) d\mu = \nu(A) - \mu(A) \leq 1,$$

so $\int f^+ d\mu < \infty$ as claimed.

If $\mu \not\ll \nu$ then there exists a set B such that $\mu(B) > 0$ and $\nu(B) = 0$. For almost every $x \in B$ we have $\rho(x) = 0$ and $\log(\rho(x)) = -\infty$, hence $-\int \log(\rho) d\mu = \infty$. To confirm that $KL(\mu \,\|\, \nu) = \infty$, observe that the function χ_B is a measurable mapping from Ω to $\{0, 1\}$ and that it satisfies $\mathcal{KL}(f_*\mu \,\|\, f_*\nu) = \infty$.

It remains to verify (2.26) when $\mu \ll \nu$. We will first prove

$$KL(\mu \,\|\, \nu) \leq -\int \log(\rho) d\mu, \tag{2.27}$$

and then prove the reverse inequality. To prove (2.27) it suffices to show that $\mathcal{KL}(f_*\mu \,\|\, f_*\nu) \leq -\int \log(\rho) d\mu$ whenever $f : \Omega \to X$ is a measurable mapping from $(\Omega, \mathcal{F})$ to a simple measurable space $(X, 2^X)$. Given such a mapping f, define a function

$$\hat{\rho}(x) = \frac{\nu(f^{-1}(f(x)))}{\mu(f^{-1}(f(x)))}.$$

Note that $0 < \hat{\rho}(x) < \infty$ for all x outside a set of μ-measure zero, and that the definition of $\mathcal{KL}(f_*\mu \,\|\, f_*\nu)$ is equivalent to the formula:

$$\mathcal{KL}(f_*\mu \,\|\, f_*\nu) = -\int \log(\hat{\rho}) d\mu,$$

so it remains to prove that

$$-\int \log(\hat{\rho}) d\mu \leq -\int \log(\rho) d\mu,$$

i.e.

$$0 \leq \int \log\left(\frac{\hat{\rho}}{\rho}\right) d\mu.$$

Let λ be the measure defined by $\lambda(A) = \int_A (\rho/\hat{\rho})\, d\mu$. We claim $\int_\Omega d\lambda = 1$. To prove this, let $x_1, x_2, \ldots, x_n$ be the elements of X and let $A_i = f^{-1}(x_i)$ for $1 \leq i \leq n$. Note that $\hat{\rho}(x) = \nu(A_i)/\mu(A_i)$ for $x \in A_i$, so if $\mu(A_i) > 0$ then

$$\lambda(A_i) = \int_{A_i} \frac{\rho}{\nu(A_i)/\mu(A_i)} d\mu = \frac{\mu(A_i)}{\nu(A_i)} \int_{A_i} \rho\, d\mu = \frac{\mu(A_i)}{\nu(A_i)} \cdot \nu(A_i) = \mu(A_i).$$

Hence

$$\int_\Omega d\lambda = \sum_{i\,:\,\mu(A_i)>0} \mu(A_i) = 1,$$

as claimed. Now we may apply Jensen's inequality [64] to the convex function $\phi(x) = x\log(x)$ to conclude that

$$\begin{aligned} \int \log\left(\frac{\hat{\rho}}{\rho}\right) d\mu &= \int \left(\frac{\hat{\rho}}{\rho}\right) \log\left(\frac{\hat{\rho}}{\rho}\right) d\lambda \\ &= \int \phi\left(\frac{\hat{\rho}}{\rho}\right) d\lambda \\ &\geq \phi\left(\int \left(\frac{\hat{\rho}}{\rho}\right) d\lambda\right) = \phi\left(\int d\mu\right) = \phi(1) = 0, \end{aligned}$$

as desired.

We turn now to proving the reverse of (2.27), i.e. $KL(\mu \,\|\, \nu) \geq -\int \log(\rho) d\mu$ when $\mu \ll \nu$. By Theorem 2.16 and Lemma 2.18, it suffices to prove that

$$KL(s_*\mu \,\|\, s_*\nu) \geq \int s\, d\mu$$

whenever s is a σ-simple function satisfying $s \leq -\log(\rho)$. Represent s as a countable weighted sum of characteristic functions: $s = \sum_{i=1}^\infty \alpha_i \chi_{A_i}$, where the sets $A_1, A_2, \ldots$ are disjoint measurable subsets of Ω and $\alpha_i \neq \alpha_j$ for $i \neq j$. We have

$$\alpha_i \leq \inf\{-\log(\rho(x)) \,:\, x \in A_i\},$$

hence

$$e^{-\alpha_i}\mu(A_i) = \int_{A_i} e^{-\alpha_i}\, d\mu \geq \int_{A_i} \rho\, d\mu = \nu(A_i),$$

i.e. $\alpha_i \le \log\left(\frac{\mu(A_i)}{\nu(A_i)}\right)$. Hence

$$\int s\,d\mu = \sum_{i=1}^{\infty} \alpha_i \mu(A_i) \le \sum_{i=1}^{\infty} \log\left(\frac{\mu(A_i)}{\nu(A_i)}\right)\mu(A_i).$$

It remains to prove that

$$KL(s_*\mu \,\|\, s_*\nu) \ge \sum_{i=1}^{\infty} \log\left(\frac{\mu(A_i)}{\nu(A_i)}\right)\mu(A_i). \tag{2.28}$$

For $N \ge 1$ let $B_N = \bigcup_{i\ge N} A_i$, and let $t : \Omega \to \{1, 2, \ldots, N\}$ be the measurable mapping which sends A_i to $\{i\}$ for $i < N$ and sends B_N to $\{N\}$. We have $t = g \circ s$ for some measurable mapping $g : \mathbb{R} \to \{1, 2, \ldots, N\}$, so by Lemma 2.18,

$$\begin{aligned} KL(s_*\mu \,\|\, s_*\nu) &\ge KL(t_*\mu \,\|\, t_*\nu) \\ &= \sum_{i<N} \log\left(\frac{\mu(A_i)}{\nu(A_i)}\right)\mu(A_i) + \log\left(\frac{\mu(B_N)}{\nu(B_N)}\right)\mu(B_N) \\ &\ge \sum_{i<N} \log\left(\frac{\mu(A_i)}{\nu(A_i)}\right)\mu(A_i) + (\mu(B_N) - \nu(B_N)), \end{aligned}$$

where the last inequality follows from the fact that

$$\log\left(\frac{\mu(B_N)}{\nu(B_N)}\right) = -\log\left(\frac{\nu(B_N)}{\mu(B_N)}\right) \ge 1 - \frac{\nu(B_N)}{\mu(B_N)}.$$

Letting $N \to \infty$, both $\mu(B_N)$ and $\nu(B_N)$ converge to zero, proving (2.28). □

2.7.3 Properties of KL-divergence

Definition 2.10. If $(\Omega, \mathcal{F})$ is a measurable space and μ, ν are two probability measures on $(\Omega, \mathcal{F})$, we define their L_1-distance by

$$\|\mu - \nu\|_1 = 2 \sup_{A\in\mathcal{F}} (\mu(A) - \nu(A)). \tag{2.29}$$

When $(\Omega, \mathcal{F})$ is a simple measurable space, this definition of $\|\mu - \nu\|_1$ is related to the more conventional definition (i.e., $\sum_x |\mu(x) - \nu(x)|$) by the following observation. If $A \subseteq \Omega$ then

$$\begin{aligned} 2(\mu(A) - \nu(A)) &= \mu(A) - \nu(A) + (1 - \nu(A)) - (1 - \mu(A)) \\ &= \sum_{x\in A} \mu(x) - \nu(x) + \sum_{y\notin A} \nu(y) - \mu(y) \\ &\le \sum_{x\in\Omega} |\mu(x) - \nu(x)|, \end{aligned}$$

with equality when $A = \{x : \mu(x) > \nu(x)\}$. Hence $\|\mu - \nu\|_1 = \sum_{x\in\Omega} |\mu(x) - \nu(x)|$.

Theorem 2.20. *If $(\Omega, \mathcal{F})$ is a measurable space and μ, ν are two probability measures on $(\Omega, \mathcal{F})$ satisfying $\mu \ll \nu$, then*

$$2KL(\mu \,\|\, \nu) \geq \|\mu - \nu\|_1^2. \tag{2.30}$$

Proof. For $A \in \mathcal{F}$ with $\mu(A) = p$, $\nu(A) = q$, the function $f = \chi_A$ is a measurable mapping from Ω to $\{0, 1\}$ and we have

$$\begin{aligned} KL(\mu \,\|\, \nu) &\geq \mathcal{KL}(f_*\mu \,\|\, f_*\nu) \\ &= \log\left(\frac{p}{q}\right) p + \log\left(\frac{1-p}{1-q}\right)(1-p) \\ &= \int_p^q \left(\frac{1-p}{1-x} - \frac{p}{x}\right) dx \\ &= \int_p^q \frac{x-p}{x(1-x)}\, dx \\ &\geq \int_p^q 4(x-p)\, dx \end{aligned} \tag{2.31}$$

while

$$[2(\mu(A) - \nu(A))]^2 = 4(p-q)^2 = \int_p^q 8(x-p)\, dx. \tag{2.32}$$

If $q \geq p$ this confirms (2.30). If $q < p$, we rewrite the right sides of (2.31) and (2.32) as $\int_q^p 4(p-x)\, dx$ and $\int_q^p 8(p-x)\, dx$ to make the intervals properly oriented and the integrands non-negative, and again (2.30) follows. □

Theorem 2.21. *Let $(\Omega, \mathcal{F})$ be a measurable space with two probability measures μ, ν satisfying $\nu \ll \mu$, and let ρ be the Radon-Nikodym derivative of ν with respect to μ. If $0 < \varepsilon < 1/2$ and $1 - \varepsilon < \rho(x) < 1 + \varepsilon$ for all $x \in \Omega$, then $KL(\mu \,\|\, \nu) \leq \varepsilon^2$.*

Proof. Let

$$\lambda(x) = \frac{\rho(x) - (1-\varepsilon)}{2\varepsilon}$$

and note that

$$\int \lambda\, d\mu = \frac{1}{2\varepsilon}\left[\int \rho\, d\mu - (1-\varepsilon)\int d\mu\right] = \frac{1}{2\varepsilon}\left[\nu(\Omega) - (1-\varepsilon)\mu(\Omega)\right] = \frac{1}{2}$$

$$\int (1-\lambda) d\mu = \int d\mu - \int \lambda\, d\mu = 1 - \frac{1}{2} = \frac{1}{2}.$$

We have

$$\rho(x) = \lambda(x)(1-\varepsilon) + (1-\lambda(x))(1+\varepsilon).$$

Since $1-\varepsilon < \rho < 1+\varepsilon$ this implies $0 < \lambda < 1$ and, by Jensen's inequality,

$$\log\left(\frac{1}{\rho(x)}\right) \leq \lambda(x)\log\left(\frac{1}{1-\varepsilon}\right) + (1-\lambda(x))\log\left(\frac{1}{1+\varepsilon}\right).$$

Thus

$$\begin{aligned} KL(\mu \,\|\, \nu) &= \int \log\left(\frac{1}{\rho}\right) d\mu \\ &\leq \left(\int \lambda \, d\mu\right)\log\left(\frac{1}{1-\varepsilon}\right) + \left(\int (1-\lambda)\, d\mu\right)\log\left(\frac{1}{1+\varepsilon}\right) \\ &= \frac{1}{2}\log\left(\frac{1}{1-\varepsilon}\right) + \frac{1}{2}\log\left(\frac{1}{1+\varepsilon}\right) \\ &= \frac{1}{2}\log\left(\frac{1}{1-\varepsilon^2}\right) \;<\; \frac{1}{2}\left(\frac{1}{1-\varepsilon^2} - 1\right) \;<\; \varepsilon^2, \end{aligned}$$

since $\varepsilon < 1/2$.

□

2.7.4 The chain rule for KL-divergence

We assume the reader is familiar with the definition of conditional probabilities for probability spaces $(\Omega, \mathcal{F}, \mu)$ in which Ω is either finite or countable. When Ω is an uncountable set it is necessary to formulate the definitions much more carefully. We briefly review the necessary definitions here; the reader is advised to consult Chapter 33 of [19] for a more thorough and readable account of this topic.

Definition 2.11 (Conditional expectation, conditional probability). Suppose given a probability space $(\Omega, \mathcal{F}, \mu)$ and a sigma-field $\mathcal{G} \subseteq \mathcal{F}$. If $A \in \mathcal{F}$, a function $\Pr(A \,\|\, \mathcal{G}) : \Omega \to \mathbb{R}$ is called a *conditional probability* of A given $\mathcal{G}$ if it is $\mathcal{G}$-measurable and satisfies

$$\int_G \Pr(A \,\|\, \mathcal{G}) d\mu = \Pr(A \cap G) \tag{2.33}$$

for all $G \in \mathcal{G}$. Similarly, given a random variable $X : \Omega \to \mathbb{R}$ which is $\mathcal{F}$-measurable and integrable, we say that a $\mathcal{G}$-measurable function $\mathbf{E}[X \,\|\, \mathcal{G}] : \Omega \to \mathbb{R}$ is a *conditional expectation* of X given $\mathcal{G}$ if

$$\int_G \mathbf{E}[X \,\|\, \mathcal{G}] d\mu = \int_G X d\mu \tag{2.34}$$

for all $G \in \mathcal{G}$.

Note that when X is a non-negative integrable $\mathcal{F}$-measurable random variable, the function $\nu(G) = \int_G X d\mu$ constitutes a measure on $(\Omega, \mathcal{G})$ which satisfies $\nu \ll \mu$. The Radon-Nikodym Theorem thus guarantees that a conditional expectation $\mathbf{E}[X \,\|\, \mathcal{G}]$ exists, and that two such functions must agree almost everywhere.

The following theorem is proved in [19].

Theorem 2.22. *Let $(\Omega, \mathcal{F}, \mu)$ be a probability space, $\mathcal{G} \subseteq \mathcal{F}$ a σ-field, and X a random variable defined on Ω. There exists a function $\mu_X(H, \omega)$, defined for all Borel measurable sets $H \subseteq \mathbb{R} \cup \{\pm\infty\}$ and all points $\omega \in \Omega$, satisfying:*

- *For each $\omega \in \Omega$, the function $\mu_X(\cdot, \omega)$ is a probability measure on $(\mathbb{R}, \mathfrak{R})$.*
- *For each $H \in \mathfrak{R}$, the function $\mu_X(H, \cdot)$ is a conditional probability of $X \in H$ given $\mathcal{G}$.*

We call μ_X a conditional distribution *of X given $\mathcal{G}$.*

The function $\mu_X(\cdot, \omega)$, mapping points $\omega \in \Omega$ to probability measures on $(\mathbb{R}, \mathfrak{R})$, will be denoted by $X_*^{\mathcal{G}}\mu$. This notation is justified by the fact that when $\mathcal{G}$ is the σ-field $\{0, \Omega\}$, $X_*^{\mathcal{G}}\mu(\omega) = X_*\mu$ for all $\omega \in \Omega$.

The following theorem is called the "chain rule for Kullback-Leibler divergence." It is analogous to Theorem 2.5.3 of [26].

Theorem 2.23. *Suppose $(\Omega, \mathcal{F})$ is a measurable space. Let $(\Omega \times \mathbb{R}, \mathcal{F} \times \mathfrak{R})$ denote the product of the measurable spaces $(\Omega, \mathcal{F})$ and $(\mathbb{R}, \mathfrak{R})$, and let $p : \Omega \times R \to \Omega$, $q : \Omega \times R \to \mathbb{R}$ denote the projection mappings $(\omega, x) \mapsto \omega$ and $(\omega, x) \mapsto x$. Let $\mathcal{G} \subset \mathcal{F} \times \mathfrak{R}$ denote the σ-field consisting of all sets $A \times \mathbb{R}$ where $A \in \mathcal{F}$. Suppose μ, ν are two probability measures on $\Omega \times \mathbb{R}$ satisfying $\nu \ll \mu$. Then*

$$KL(\mu \,\|\, \nu) = KL(p_*\mu \,\|\, p_*\nu) + \int_\Omega KL(q_*^{\mathcal{G}}\mu(\omega) \,\|\, q_*^{\mathcal{G}}\nu(\omega)) d\mu.$$

Proof. Let ρ be a Radon-Nikodym derivative of ν with respect to μ, let τ be a Radon-Nikodym derivative of $p_*\nu$ with respect to $p_*\mu$, and let $\sigma(\omega, x)$ be a Radon-Nikodym derivative of $q_*^{\mathcal{G}}\nu(\omega)$ with respect to $q_*^{\mathcal{G}}\mu(\omega)$ for each $\omega \in \Omega$. We claim

$$\sigma(\omega, x)\tau(\omega) = \rho(\omega, x) \tag{2.35}$$

almost everywhere in $\Omega \times \mathbb{R}$. To verify this, it suffices to verify that for all $(\mathcal{F} \times \mathfrak{R})$-measurable, integrable functions f defined on $\Omega \times \mathbb{R}$,

$$\int_{\Omega \times \mathbb{R}} f(\omega, x)\sigma(\omega, x)\tau(\omega) d\mu = \int_{\Omega \times \mathbb{R}} f(\omega, x)\rho(\omega, x) d\mu.$$

Let us prove the following property of $q_*^{\mathcal{G}}\mu(\omega)$: for any $(\mathcal{F}\times\mathfrak{R})$-measurable, integrable function f,

$$\int_{\Omega\times\mathbb{R}} f(\omega,x)\,d\mu = \int_\Omega \left[\int_{\mathbb{R}} f(\omega,x) dq_*^{\mathcal{G}}\mu(\omega)\right] dp_*\mu. \tag{2.36}$$

When f is the characteristic function of a set $A\times H$, where $A\in\mathcal{F}$, $H\in\mathfrak{R}$,

$$\begin{aligned}
\int_\Omega \left[\int_{\mathbb{R}} f(\omega,x) dq_*^{\mathcal{G}}\mu(\omega)\right] dp_*\mu &= \int_A \Pr(q(\omega,x)\in H \parallel \mathcal{G}) dp_*\mu \\
&= \int_{A\times\mathbb{R}} \Pr(q(\omega,x)\in H \parallel \mathcal{G}) d\mu \\
&= \mu(A\times H) \\
&= \int_{\Omega\times\mathbb{R}} f(\omega,x)\,d\mu.
\end{aligned}$$

The sets $A\times H$ ($A\in\mathcal{F}$, $H\in\mathfrak{R}$) generate the σ-algebra $\mathcal{F}\times\mathfrak{R}$, so (2.36) holds whenever f is the characteristic function of a measurable subset of $\Omega\times\mathbb{R}$. By linearity, (2.36) holds whenever f is a simple function. Every integrable function is a pointwise limit of simple functions, so by dominated convergence (2.36) holds for all integrable f. Of course, by the same argument (2.36) also holds with ν in place of μ.

Applying (2.36), we discover that

$$\begin{aligned}
\int_{\Omega\times\mathbb{R}} f(\omega,x)\sigma(\omega,x)\tau(\omega)d\mu &= \int_\Omega \left[\int_R f(\omega,x)\sigma(\omega,x) dq_*^{\mathcal{G}}\mu(\omega)\right] \tau(\omega) dp_*\mu \\
&= \int_\Omega \left[\int_R f(\omega,x) dq_*^{\mathcal{G}}\nu(\omega)\right] dp_*\nu \\
&= \int_{\Omega\times\mathbb{R}} f(\omega,x) d\nu \\
&= \int_{\Omega\times\mathbb{R}} f(\omega,x)\rho(\omega,x) d\mu,
\end{aligned}$$

as claimed. Hence $\sigma(\omega,x)\tau(\omega)=\rho(\omega,x)$, and

$$\log(\rho(\omega,x)) = \log(\sigma(\omega,x)) + \log(\tau(\omega)). \tag{2.37}$$

Now by Theorem 2.19,

$$\begin{aligned} KL(\mu \,\|\, \nu) &= -\int_{\Omega\times\mathbb{R}} \log(\rho(\omega, x))d\mu \qquad (2.38) \\ KL(p_*\mu \,\|\, p_* \,\|\, \nu) &= -\int_{\Omega} \log(\tau(\omega))dp_*\mu \\ &= -\int_{\Omega\times\mathbb{R}} \log(\tau(\omega))d\mu \qquad (2.39) \\ \int_\Omega KL(q_*^{\mathcal{G}}\mu(\omega) \,\|\, q_*^{\mathcal{G}}\nu(\omega))dp_*\mu &= \int_\Omega \left[-\int_{\mathbb{R}} \log(\sigma(\omega, x))dq_*^{\mathcal{G}}\mu(\omega)\right] dp_*\mu \\ &= -\int_{\Omega\times\mathbb{R}} \log(\sigma(\omega, x))d\mu. \qquad (2.40) \end{aligned}$$

The last equation follows using (2.36). Combining (2.38)-(2.40) with (2.37), we obtain

$$KL(\mu \,\|\, \nu) = KL(p_*\mu \,\|\, p_* \,\|\, \nu) + \int_\Omega KL(q_*^{\mathcal{G}}\mu(\omega) \,\|\, q_*^{\mathcal{G}}\nu(\omega))dp_*\mu.$$

□

When $\Omega = \mathbb{R}^n$ and $\mathcal{F}$ is the Borel σ-field $\mathfrak{R}^n$, let us define $\mathcal{F}_k \subseteq \mathcal{F}$ (for $0 \le k \le n$) to be the σ-field consisting of all sets of the form $A \times \mathbb{R}^{n-k}$, where $A \in \mathfrak{R}^k$. Let $p_k : \Omega \to \mathbb{R}$ be the mapping which projects an n-tuple $(x_1, \ldots, x_n)$ onto its k-th coordinate x_k, and let $p_{1..k} : \Omega \to \mathbb{R}^k$ be the mapping $(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_k)$. Given a probability measure μ on $(\Omega, \mathcal{F})$ we will use μ_k as notational shorthand for $p_{k*}^{\mathcal{F}_{k-1}}\mu$. Note that μ_k is a measure-valued function of n-tuples $(x_1, \ldots, x_n)$ which only depends on the first $k-1$ coordinates; accordingly we will sometimes interpret it as a measure-valued function of $(k-1)$-tuples $(x_1, \ldots, x_{k-1})$. Conceptually one should think of $\mu_k(x_1, \ldots, x_{k-1})$ as the marginal distribution of the coordinate x_k conditional on the values of $x_1, \ldots, x_{k-1}$. By iterated application of Theorem 2.23 one obtains the following theorem.

Theorem 2.24. *If μ, ν are two probability measures on $(\mathbb{R}^n, \mathfrak{R}_n)$, then*

$$KL(\mu \,\|\, \nu) = \sum_{k=0}^{n-1} \int_{\mathbb{R}^k} KL(\mu_{k+1} \,\|\, \nu_{k+1})dp_{1..k*}\mu.$$

2.8 Lower bound for the K-armed bandit problem

In this section we present a proof, adapted from [4], that any algorithm for the K-armed bandit problem must have regret $\Omega(\sqrt{TK})$ against oblivious adversaries. The proof illustrates the use of KL-divergence in proving lower bounds for online decision

problems. We will use similar techniques to prove the lower bounds in Theorems 3.20 and 4.4.

Theorem 2.25. *Let $\mathcal{A}$ be the class of oblivious adversaries for the multi-armed bandit problem with strategy set $\mathcal{S} = \{1, 2, \ldots, K\}$ and cost function class $\Gamma = [0,1]^{\mathcal{S}}$. For any algorithm* ALG*,*

$$R(\mathsf{ALG}, \mathcal{A}; T) = \Omega(\sqrt{TK}).$$

Proof. Let $\varepsilon = \sqrt{K/8T}$. Our lower bound will be based on an i.i.d. adversary who samples $\{0,1\}$-valued cost functions which assign expected cost $\frac{1}{2}$ to all but one strategy $i \in \mathcal{S}$, and which assign expected cost $\frac{1}{2} - \varepsilon$ to strategy i. The idea of the proof will be to demonstrate, using properties of KL-divergence, that the algorithm is unlikely to gain enough information to distinguish the best strategy, i, from all other strategies before trial T.

Let σ denote the uniform distribution on the two-element set $\{0,1\}$, and let σ' denote the distribution which assigns probability $\frac{1}{2} - \varepsilon$ to 1 and probability $\frac{1}{2} + \varepsilon$ to 0. For $i = 0, 1, 2, \ldots, K$, define a probability distribution P_i on cost functions $c \in \Gamma$ by specifying that the random variables $c(j)$ $(1 \leq j \leq K)$ are independent $\{0,1\}$-valued random variables with distribution σ for $j \neq i$, σ' for $j = i$. (Note that P_0 is simply the uniform distribution on the set $\{0,1\}^{\mathcal{S}}$, while for $i > 0$, P_i is a non-uniform distribution on this set.) Let $\mathsf{ADV}_i \in \mathcal{A}$ denote the i.i.d. adversary defined by distribution P_i, for $i = 0, 1, \ldots, K$. Note that the expectation of the random cost function sampled by adversary ADV_i in each trial is

$$\bar{c}(x) = \begin{cases} \frac{1}{2} - \varepsilon & \text{if } x = i \\ \frac{1}{2} & \text{if } x \neq i. \end{cases}$$

We will prove that $R(\mathsf{ALG}, \mathsf{ADV}_i; T) = \Omega(\sqrt{TK})$ for some $i \in \{1, 2, \ldots, K\}$,

For a strategy $i \in \mathcal{S}$ and an adversary ADV, let $\chi_i(x_t)$ denote the Bernoulli random variable which equals 1 if $x_t = i$, 0 otherwise, and let

$$Q_i(\mathsf{ALG}, \mathsf{ADV}; T) = \mathbf{E}\left[\sum_{t=1}^{T} \chi_i(x_t)\right]$$

denote the expected number of times ALG selects strategy i during the first T trials,

when playing against ADV. Note that

$$\begin{aligned} R(\mathsf{ALG}, \mathsf{ADV}_i; T) &= \mathbf{E}\left[\sum_{t=1}^{T} c_t(x_t) - c_t(i)\right] \\ &= \mathbf{E}\left[\sum_{t=1}^{T} \bar{c}(x_t) - \bar{c}(i)\right] \\ &= \varepsilon[T - Q_i(\mathsf{ALG}, \mathsf{ADV}_i; T)]. \end{aligned}$$

Recalling that $\varepsilon = \sqrt{K/8T}$, we see that the theorem reduces to proving that there exists $i \in \{1, \ldots, K\}$ such that $T - Q_i(\mathsf{ALG}, \mathsf{ADV}_i; T) = \Omega(T)$.

We have $\sum_{i=1}^{K} Q_i(\mathsf{ALG}, \mathsf{ADV}_0; T) = T$, hence there is at least one $i \in \{1, \ldots, K\}$ such that $Q_i(\mathsf{ALG}, \mathsf{ADV}_0; T) \leq T/K$. We claim that this implies

$$Q_i(\mathsf{ALG}, \mathsf{ADV}_i; T) \leq \frac{T}{2} + \frac{T}{K}, \tag{2.41}$$

from which the theorem would follow. The transcripts of play for ALG against $\mathsf{ADV}_0, \mathsf{ADV}_i$ define two probability distributions μ, ν on sequences $(x_1, y_1, \ldots, x_T, y_T)$, where $x_1, \ldots, x_T$ denote the algorithm's choices in trials $1, \ldots, T$ and $y_1, \ldots, y_T$ denote the feedback values received. We see that

$$\begin{aligned} Q_i(\mathsf{ALG}, \mathsf{ADV}_i; T) - Q_i(\mathsf{ALG}, \mathsf{ADV}_0; T) &= \sum_{t=1}^{T} \nu(\{x_t = i\}) - \mu(\{x_t = i\}) \\ &\leq T\left(\frac{\|\nu - \mu\|_1}{2}\right). \end{aligned}$$

From Theorem 2.20 we know that

$$\frac{\|\nu - \mu\|_1}{2} \leq \sqrt{\frac{KL(\mu \,\|\, \nu)}{2}}$$

so we are left with proving $KL(\mu \,\|\, \nu) \leq \frac{1}{2}$.

Using the chain rule for Kullback-Leibler divergence (Theorem 2.24) we have

$$KL(\mu \,\|\, \nu) = \sum_{i=0}^{2T-1} \int_{\mathbb{R}^i} KL(\mu_{i+1} \,\|\, \nu_{i+1}) dp_{1..i*}\mu.$$

Now if $i = 2j$ is an even number, $\mu_{i+1}(x_1, y_1, \ldots, x_j, y_j) = \nu_{i+1}(x_1, y_1, \ldots, x_j, y_j)$ because both distributions are equal to the conditional distribution of the algorithm's choice x_{j+1}, conditioned on the strategies and feedbacks revealed in trials $1, \ldots, j$. If $i = 2j - 1$ is an odd number, then $\nu_{i+1}(x_1, y_1, \ldots, x_{j-1}, y_{j-1}, x_j)$ is the distribution of $y_j = c_j(x_j)$ for adversary ADV_i, so $\nu_{i+1}(x_1, y_1, \ldots, x_j) = \sigma'$ if $x_j = i$, σ otherwise.

Similarly $\mu_{i+1}(x_1, y_1, \ldots, x_j)$ is the distribution of $y_j = c_j(x_j)$ for adversary ADV_0, so $\mu_{i+1}(x_1, y_1, \ldots, x_j) = \sigma$. Therefore $KL(\mu_{i+1} \,\|\, \nu_{i+1}) = 0$ if $x_j \neq i$, and by Theorem 2.21 $KL(\mu_{i+1} \,\|\, \nu_{i+1}) \leq 4\varepsilon^2$ if $x_j = i$. In other words,

$$KL(\mu_{i+1} \,\|\, \nu_{i+1}) \leq 4\varepsilon^2 \chi_i(x_j).$$

Letting $\mathbf{E}_\mu$ denote the expectation operator defined by the measure μ, we find that

$$\int_{\mathbb{R}^i} KL(\mu_{i+1} \,\|\, \nu_{i+1}) dp_{1..i*}\mu \leq 4\varepsilon^2 \mathbf{E}_\mu[\chi_i(x_j)]$$

and therefore

$$\begin{aligned} KL(\mu \,\|\, \nu) &\leq \sum_{j=1}^{T} 4\varepsilon^2 \mathbf{E}_\mu[\chi_i(x_j)] \\ &= 4\varepsilon^2 Q_i(\mathsf{ALG}, \mathsf{ADV}_0; T) \\ &= 4\left(\frac{K}{8T}\right)\left(\frac{T}{K}\right) = \frac{1}{2} \end{aligned}$$

which completes the proof. □

Chapter 3

Online pricing strategies

In this chapter we study online pricing strategies, motivated by the question, "What is the value of knowing the demand curve for a good?" In other words, how much should a seller be willing to pay to obtain foreknowledge of the valuations held by a population of buyers, if the alternative is to attempt to converge to the optimal price using adaptive pricing? The study of these questions will serve as an introduction to the applications of online decision problems, while also highlighting many of the main technical themes which will reappear throughout this work.

Recall the pricing problem defined in Section 1.5. A seller with an unlimited supply of identical goods interacts sequentially with a population of T buyers. For each buyer, the seller names a price between 0 and 1; the buyer then decides whether or not to buy the item at the specified price, based on her privately-held valuation for the good. We will study three versions of this problem, which differ in the type of assumption made about the buyers' valuations:

Identical: All buyers' valuations are equal to a single price $p \in [0, 1]$. This price is unknown to the seller.

Random: Buyers' valuations are independent random samples from a fixed probability distribution on $[0, 1]$. The probability distribution is not known to the seller.

Worst-case: The model makes no assumptions about the buyers' valuations. They are chosen by an adversary who is oblivious to the algorithm's random choices.

In all cases, our aim is to derive nearly matching upper and lower bounds on the regret of the optimal pricing strategy.

3.1 Identical valuations

3.1.1 Upper bound

When all buyers have the same valuation $p \in [0, 1]$, every response gives the seller perfect information about a lower or upper bound on p, depending on whether the buyer's response was to accept or to reject the price offered. A pricing strategy $\mathbb{S}$ which achieves regret $O(\log \log T)$ may be described as follows. The strategy keeps track of a *feasible interval* $[a, b]$, initialized to $[0, 1]$, and a *precision parameter* ε, initialized to $1/2$. In a given phase of the algorithm, the seller offers the prices $a, a + \varepsilon, a + 2\varepsilon, \ldots$ until one of them is rejected. If $a + k\varepsilon$ was the last offer accepted in this phase, then $[a + k\varepsilon, a + (k + 1)\varepsilon]$ becomes the new feasible interval, and the new precision parameter is ε^2. This process continues until the length of the feasible interval is less than $1/T$; then the seller offers a price of a to all remaining buyers.

Theorem 3.1. *The regret of strategy $\mathbb{S}$ is bounded above by $2\lceil \log_2 \log_2 T \rceil + 4$.*

Proof. The number of phases is equal to the number of iterations of repeated squaring necessary to get from $1/2$ to $1/T$, i.e. $\lceil \log_2 \log_2 T \rceil + 1$. Let p denote the valuation shared by all buyers. The seller accrues regret for two reasons:

- Items are sold at a price $q < p$, accruing regret $p - q$.
- Buyers decline items, accruing regret p.

At most one item is declined per phase, incurring at most one unit of regret, so the declined offers contribute at most $\lceil \log_2 \log_2 T \rceil + 1$ to the total regret.

In each phase except the first and the last, the length $b - a$ of the feasible interval is $\sqrt{\varepsilon}$ (i.e. it is the value of ε from the previous phase), and the set of offer prices carves up the feasible interval into subintervals of length ε. There are $1/\sqrt{\varepsilon}$ such subintervals, so there are at most $1/\sqrt{\varepsilon}$ offers made during this phase. Each time one of them is accepted, this contributes at most $b - a = \sqrt{\varepsilon}$ to the total regret. Thus, the total regret contribution from accepted offers in this phase is less than or equal to $(1/\sqrt{\varepsilon}) \cdot \sqrt{\varepsilon} = 1$. The first phase is exceptional since the feasible interval is longer than $\sqrt{\varepsilon}$. The accepted offers in phase 1 contribute at most 2 to the total regret. Summing over all phases, the total regret contribution from accepted offers is $\leq \lceil \log_2 \log_2 n \rceil + 3$.

In the final phase, the length of the feasible interval is at most $1/T$, and each offer is accepted. There are at most T such offers, so they contribute at most 1 to the total regret. □

Remark 3.1. If the seller does not have foreknowledge of T, it is still possible to achieve regret $O(\log \log T)$ by modifying the strategy. At the start of a phase in which the feasible interval is $[a, b]$, the seller offers price a to the next $\lfloor 1/(b-a) \rfloor$ buyers. This raises the regret per phase from 2 to 3, but ensures that the number of phases does not exceed $\lceil \log_2 \log_2 T \rceil + 1$.

3.1.2 Lower bound

Theorem 3.2. *If $\mathbb{S}$ is any randomized pricing strategy, and p is randomly sampled from the uniform distribution on $[0, 1]$, the expected regret of $\mathbb{S}$ when the buyers' valuations are p is $\Omega(\log \log T)$.*

Proof. It suffices to prove the lower bound for a deterministic pricing strategy $\mathbb{S}$, since any randomized pricing strategy is a probability distribution over deterministic ones. At any stage of the game, let a denote the highest price that has yet been accepted, and b the lowest price that has yet been declined; thus $p \in [a, b]$. As before, we will refer to this interval as the *feasible interval*. It is counterproductive to offer a price less than a or greater than b, so we may assume that the pricing strategy works as follows: it offers an ascending sequence of prices until one of them is declined; it then limits its search to the new feasible interval, offering an ascending sequence of prices in this interval until one of them is declined, and so forth.

Divide the pool of buyers into phases (starting with phase 0) as follows: phase k begins immediately after the end of phase $k-1$, and ends after an addtional $2^{2^k} - 1$ buyers, or after the first rejected offer following phase $k-1$, whichever comes earlier. The number of phases is $\Omega(\log \log T)$, so it suffices to prove that the expected regret in each phase is $\Omega(1)$. This is established by the following three claims. □

Claim 3.3. *Let $\mathcal{I}_k$ denote the set of possible feasible intervals at the start of phase k. The cardinality of $\mathcal{I}_k$ is at most 2^{2^k}, and the intervals in $\mathcal{I}_k$ have disjoint interiors.*

Proof. The proof is by induction on k. The base case $k = 0$ is trivial. Now assume the claim is true for a particular value of k, and let Let $I_k = [a_k, b_k]$ be the feasible interval at the start of phase k. Let $x_1 \le x_2 \le \cdots \le x_j$ denote the ascending sequence of prices that $\mathbb{S}$ will offer during phase k if all offers are accepted. (Here $j = 2^{2^k} - 1$.) Then the feasible interval at the start of phase $k+1$ will be one of the subintervals $[a_k, x_1], [x_1, x_2], [x_2, x_3], \ldots, [x_{j-1}, x_j], [x_j, b_k]$. There are at most $j = 2^{2^k}$ such subintervals, and at most 2^{2^k} possible choices for I_k (by the induction hypothesis), hence there are at most $2^{2^{k+1}}$ elements of $\mathcal{I}_{k+1}$. Moreover, they all have disjoint interiors because we have simply taken the intervals in $\mathcal{I}_k$ and partitioned them into subintervals. □

Claim 3.4. *Let $|I|$ denote the length of an interval I. With probability at least 3/4, $|I_k| \geq \frac{1}{4} \cdot 2^{-2^k}$.*

Proof. The expectation of $1/|I_k|$ may be computed as follows:

$$\mathbf{E}(1/|I_k|) = \sum_{I \in \mathcal{I}_k} \Pr(p \in I)(1/|I|) = \sum_{I \in \mathcal{I}_k} |I|/|I| \leq 2^{2^k},$$

where the last inequality follows from Claim 3.3. Now use Markov's Inequality:

$$\Pr(|I_k| < \frac{1}{4} \cdot 2^{-2^k}) = \Pr(1/|I_k| > 4 \cdot 2^{2^k}) < 1/4.$$

□

Claim 3.5. *The expected regret in phase k is at least $\frac{1}{128}$.*

Proof. By Claim 3.3, with probability 1, p belongs to a unique interval $I_k = [a, b] \in \mathcal{I}_k$. Let $\mathcal{E}_k$ denote the event that $b \geq 1/4$ and $|I_k| \geq \frac{1}{4} \cdot 2^{-2^k}$. This is the intersection of two events, each having probability $\geq 3/4$, so $\Pr(\mathcal{E}_k) \geq 1/2$. It suffices to show that the expected regret in phase k, conditional on $\mathcal{E}_k$, is at least 1/64. So from now on, assume that $p \geq 1/4$ and $|I_k| \geq \frac{1}{4} \cdot 2^{-2^k}$. Also note that, conditional on the events $\mathcal{E}_k$ and $p \in I_k$, p is uniformly distributed in I_k.

Let $m = (a+b)/2$. As before, let $j = 2^{2^k} - 1$ and let $x_1 \leq x_2 \leq \cdots \leq x_j$ denote the ascending sequence of prices which $\mathbb{S}$ would offer in phase k if no offers were rejected. We distinguish two cases:

Case 1: $x_j \geq m$**.** With probability at least 1/2, $p < m$ and the phase ends in a rejected offer, incurring a regret of p, whose conditional expectation $\frac{a+m}{2}$ is at least 1/16. Thus the expected regret in this case is at least 1/32.

Case 2: $x_j < m$**.** The event $\{p > m\}$ occurs with probability 1/2, and conditional on this event the expectation of $p - m$ is $|I_k|/4 \geq 2^{-2^k}/16$. Thus with probability at least 1/2, there will be $2^{2^k} - 1$ accepted offers, each contributing $2^{-2^k}/16$ to the expected regret, for a total of $(1/2)(2^{2^k} - 1)(2^{-2^k})/16 \geq 1/64$.

□

3.2 Random valuations

3.2.1 Preliminaries

In this section we will consider the case each buyer's valuation v is an independent random sample from a fixed but unknown probability distribution on $[0, 1]$. It is

customary to describe this probability distribution in terms of its *demand curve*

$$D(x) = \Pr(v \geq x).$$

Given foreknowledge of the demand curve, but not of the individual buyers' valuations, it is easy to see what the optimal pricing strategy would be. The expected revenue obtained from setting price x is $xD(x)$. Since buyers' valuations are independent and the demand curve is known, the individual buyers' responses provide no useful information about future buyers' valuations. The best strategy is thus to compute

$$x^* = \arg\max_{x \in [0,1]} xD(x)$$

and to offer this price to every buyer. We denote this strategy by $\mathbb{S}^*$, and its expected revenue by $\rho(\mathbb{S}^*)$. There is also an "omniscient" pricing strategy $\mathbb{S}^{\text{opt}}$, defined as the maximum revenue obtainable by observing the valuations of all T buyers and setting a single price to be charged to all of them. Clearly, for any on-line pricing strategy $\mathbb{S}$, we have

$$\rho(\mathbb{S}) \leq \rho(\mathbb{S}^*) \leq \rho(\mathbb{S}^{\text{opt}}),$$

and it may be argued that in the context of random valuations it makes the most sense to compare $\rho(\mathbb{S})$ with $\rho(\mathbb{S}^*)$ rather than $\rho(\mathbb{S}^{\text{opt}})$. We address this issue by proving a lower bound on $\rho(\mathbb{S}^*) - \rho(\mathbb{S})$ and an upper bound on $\rho(\mathbb{S}^{\text{opt}}) - \rho(\mathbb{S})$.

A deterministic pricing strategy can be specified by a sequence of rooted planar binary trees $\mathbb{T}_1, \mathbb{T}_2, \ldots$, where the T-th tree specifies the decision tree to be applied by the seller when interacting with a population of T buyers. (Thus $\mathbb{T}_T$ is a complete binary tree of depth T.) We will use a to denote a generic internal node of such a decision tree, and ℓ to denote a generic leaf. The relation $a \prec b$ will denote that b is a descendant of a; here b may be a leaf or another internal node. If e is an edge of $\mathbb{T}$, we will also use $a \prec e$ (resp. $e \prec a$) to denote that e is below (resp. above) a in $\mathbb{T}$, i.e. at least one endpoint of e is a descendant (resp. ancestor) of a. The left subtree rooted at a will be denoted by $\mathbb{T}_l(a)$, the right subtree by $\mathbb{T}_r(a)$. Note that $\mathbb{T}_l(a)$ (resp. $\mathbb{T}_r(a)$) includes the edge leading from a to its left (resp. right) child.

The internal nodes of the tree are labeled with numbers $x_a \in [0,1]$ denoting the price offered by the seller at node a, and random variables $v_a \in [0,1]$ denoting the valuation of the buyer with whom the seller interacts at that node. The buyer's choice is represented by a random variable

$$\chi_a = \begin{cases} 1 & \text{if } v_a \geq x_a \\ 0 & \text{if } v_a < x_a \end{cases}.$$

In other words, χ_a is 1 if the buyer accepts the price offered, 0 otherwise.

The tree $\mathbb{T}_T$ specifies a pricing strategy as follows. The seller starts at the root r of the tree and offers the first buyer price x_r. The seller moves from this node to its left or right child depending on whether the buyer declines or accepts the offer, and repeats this process until reaching a leaf which represents the outcomes of all transactions.

A strategy as defined above is called a *non-uniform deterministic pricing strategy.* A uniform deterministic pricing strategy is one in which there is a single infinite tree $\mathbb{T}$ whose first T levels comprise $\mathbb{T}_T$ for each T. (This corresponds to a pricing strategy which is not informed of the value of T at the outset of the auction.) A randomized pricing strategy is a probability distribution over deterministic pricing strategies.

As mentioned above, the outcome of the auction may be represented by a leaf $\ell \in \mathbb{T}_T$, i.e. the unique leaf such that for all ancestors $a \prec \ell$, $\ell \in \mathbb{T}_r(a) \Leftrightarrow \chi_a = 1$. A probability distribution on the buyers' valuations v_a induces a probability distribution on outcomes ℓ. We will use $p_D(\ell)$ to denote the probability assigned to ℓ under the valuation distribution represented by demand curve D. For an internal node a, $p_D(a)$ denotes the probability that the outcome leaf is a descendant of a. We define $p_D(e)$ similarly for edges $e \in \mathbb{T}$.

3.2.2 Lower bound

A family of random demand curves

The demand curves D appearing in our lower bound will be random samples from a space $\mathcal{D}$ of possible demand curves. In this section we single out a particular random demand-curve model, and we enumerate the properties which will be relevant in establishing the lower bound. The choice of a particular random demand-curve model is done here for ease of exposition, and not because of a lack of generality in the lower bound itself. At the end of this section we will indicate that Theorem 3.14 applies to much broader classes $\mathcal{D}$ of demand curves. In particular we believe that it encompasses random demand-curve models which are realistic enough to be of interest in actual economics and e-commerce applications.

For now, however, $\mathcal{D}$ denotes the one-parameter family of demand curves $\{D_t : 0.3 \leq t \leq 0.4\}$ defined as follows. Let

$$\tilde{D}_t(x) = \max\left\{1 - 2x, \frac{2t - x}{7t^2}, \frac{1 - x}{2}\right\}.$$

In other words, the graph of $\tilde{D}_t$ consists of three line segments: the middle segment is tangent to the curve $xy = 1/7$ at the point $(t, 1/7t)$, while the left and right segments belong to lines which lie below that curve and are independent of t. Now we obtain D_t

by smoothing $\tilde{D}_t$. Specifically, let $b(x)$ be a non-negative, even C^∞ function supported on the interval $[-0.01, 0.01]$ and satisfying $\int_{-0.01}^{0.01} b(x)\,dx = 1$. Define D_t by convolving $\tilde{D}_t$ with b, i.e.

$$D_t(x) = \int_{-\infty}^{\infty} \tilde{D}_t(y) b(x-y) dy.$$

We will equip $\mathcal{D} = \{D_t : 0.3 \le t \le 0.4\}$ with a probability measure by specifying that t is uniformly distributed in $[0.3, 0.4]$.

Let $x_t^* = \arg\max_{x \in [0,1]} x D_t(x)$. It is an exercise to compute that $x_t^* = t$. (With $\tilde{D}_t$ in place of D_t this would be trivial. But $D_t(x) = \tilde{D}_t(x)$ unless x is within 0.01 of one of the two points where $\tilde{D}_t'$ is discontinuous, and these two points are far from maximizing $x\tilde{D}_t(x)$, so $xD_t(x)$ is also maximized at $x = t$.)

The specifics of the construction of $\mathcal{D}$ are not important, except insofar as they enable us to prove the properties specified in the following lemma.

Lemma 3.6. *There exist constants $\alpha, \beta > 0$ and $\gamma < \infty$ such that for all $D = D_{t_0} \in D$ and $x \in [0,1]$:*

1. $\frac{d}{dt}(x_t^*)|_{t=t_0} > \alpha$;
2. $x^* D(x^*) - xD(x) > \beta(x^* - x)^2$;
3. $|\dot{D}(x)/D(x)| < \gamma|x^* - x|$ *and* $|\dot{D}(x)/(1 - D(x))| < \gamma|x^* - x|$;
4. $|D^{(k)}(x)/D(x)| < \gamma$ *and* $|D^{(k)}(x)/(1 - D(x))| < \gamma$, *for* $k = 2, 3, 4$.

Here x^* denotes $x_{t_0}^*$, $D^{(k)}(x)$ denotes the k-th t-derivative of $D_t(x)$ at $t = t_0$, and $\dot{D}(x)$ denotes $D^{(1)}(x)$.

The proof of the lemma is elementary but tedious, so it is deferred to Section 3.4.

High-level description of the proof

The proof of the lower bound on regret is based on the following intuition. If there is uncertainty about the demand curve, then no single price can achieve a low expected regret for all demand curves. The family of demand curves exhibited above is parametrized by a single parameter t, and we will see that if the uncertainty about t is on the order of ε then the regret per buyer is $\Omega(\varepsilon^2)$. (This statement will be made precise in Lemma 3.12 below.) So to avoid accumulating $\Omega(\sqrt{T})$ regret on the last $\Omega(T)$ buyers, the pricing strategy must ensure that it reduces the uncertainty to $O(T^{-1/4})$ during its interactions with the initial $O(T)$ buyers. However — and this is the crux of the proof — we will show that offering prices far from x^* is much more informative than offering prices near x^*, so there is a quantifiable cost to reducing the

uncertainty in t. In particular, reducing the uncertainty to $O(T^{-1/4})$ costs $\Omega(\sqrt{T})$ in terms of expected regret.

To make these ideas precise, we will introduce a notion of "knowledge" which quantifies the seller's ability to distinguish the actual demand curve from nearby ones based on the information obtained from past transactions, and a notion of "conditional regret" whose expectation is a lower bound on the pricing strategy's expected regret. We will show that the ratio of conditional regret to knowledge is bounded below, so that the strategy cannot accumulate $\Omega(\sqrt{T})$ knowledge without accumulating $\Omega(\sqrt{T})$ regret. Finally, we will show that when the expected knowledge is less than a small constant multiple of $\sqrt{T}$, there is so much uncertainty about the true demand curve that the expected regret is $\Omega(\sqrt{T})$ with high probability (taken over the probability measure on demand curves).

Definition of knowledge

In the following definitions, log denotes the natural logarithm function. $\mathbb{T}$ denotes a finite planar binary tree, labeled with a pricing strategy as explained in Section 3.2.1. When f is a function defined on leaves of $\mathbb{T}$, we will use the notation $\mathbf{E}_D f$ to denote the expectation of f with respect to the probability distribution p_D on leaves, i.e.

$$\mathbf{E}_D f = \sum_{\ell \in \mathbb{T}} p_D(\ell) f(\ell).$$

For a given demand curve $D = D_{t_0}$, we define the *infinitesimal relative entropy* of a leaf $\ell \in \mathbb{T}$ by

$$IRE_D(\ell) = \frac{d}{dt}(-\log p_{D_t}(\ell))|_{t=t_0},$$

and we define the *knowledge* of ℓ as the square of the infinitesimal relative entropy:

$$K_D(\ell) = IRE_D(\ell)^2.$$

Readers familiar with information theory may recognize $IRE_D(\ell)$ as the t-derivative of ℓ's contribution to the weighted sum defining the Kullback-Leibler divergence $KL(p_D \| p_{D_t})$, and $K_D(\ell)$ as a random variable whose expected value is a generalization of the notion of *Fisher information*.

An important feature of $IRE_D(\ell)$ is that it may be expressed as a sum of terms coming from the edges of $\mathbb{T}$ leading from the root to ℓ. For an edge $e = (a, b) \in \mathbb{T}$, let

$$\begin{aligned} ire_D(e) &= \begin{cases} \frac{d}{dt}(\log D(x_a)) & \text{if } e \in \mathbb{T}_r(a) \\ \frac{d}{dt}(\log(1 - D(x_a))) & \text{if } e \in \mathbb{T}_l(a) \end{cases} \\ &= \begin{cases} \dot{D}(x_a)/D(x_a) & \text{if } e \in \mathbb{T}_r(a) \\ -\dot{D}(x_a)/(1 - D(x_a)) & \text{if } e \in \mathbb{T}_l(a) \end{cases} \end{aligned}$$

Then

$$IRE_D(\ell) \;=\; \sum_{e \prec \ell} ire_D(e).$$

Definition of conditional regret

For a given D, the conditional regret $R_D(\ell)$ may be informally defined as follows. At the end of the auction, if the demand curve D were revealed to the seller and then she were required to offer the same sequence of prices $\{x_a : a \prec \ell\}$ to a new, independent random population of buyers whose valuations are distributed according to D, then $R_D(\ell)$ is the expected regret incurred by the seller during this second round of selling. Formally, $R_D(\ell)$ is defined as follows. Let

$$r_D(x) = x^* D(x^*) - xD(x),$$

where $x^* = \arg\max_{x \in [0,1]} \{xD(x)\}$ as always. Note that if two different sellers offer prices x^*, x, respectively, to a buyer whose valuation is distributed according to D, then $r_D(x)$ is the difference in their expected revenues. Now let

$$R_D(\ell) = \sum_{a \prec \ell} r_D(x_a).$$

Although $R_D(\ell)$ is *not* equal to the seller's actual regret conditional on outcome ℓ, it is a useful invariant because $\mathbf{E}_D R_D(\ell)$ is equal to the actual expected regret of $\mathbb{S}$ relative to $\mathbb{S}^*$. (It is also therefore a lower bound on the expected regret of $\mathbb{S}$ relative to $\mathbb{S}^{\text{opt}}$.) This fact is far from obvious, because the distribution of the actual buyers' valuations, conditioned on their responses to the prices they were offered, is very different from the distribution of T new independent buyers. In general the expected revenue of $\mathbb{S}$ or $\mathbb{S}^*$ on the hypothetical independent population of T buyers will not equal the expected revenue obtained from the actual population of T buyers, conditioned on those buyers' responses. Yet the expected difference between the two random variables, i.e. the regret, is the same for both populations of buyers. This fact is proved in the following lemma.

Lemma 3.7. *Let S be a strategy with decision tree $\mathbb{T}$, and let S^* be the fixed-price strategy which offers x^* to each buyer. If the buyers' valuations are independent random samples from the distribution specified by D, then the expected revenue of S^* exceeds that of S by exactly $\mathbf{E}_D R_D(\ell)$.*

Proof. Let

$$\chi_a^* = \begin{cases} 1 & \text{if } v_a \geq x^* \\ 0 & \text{if } v_a < x^* \end{cases} .$$

At a given point of the sample space, let ℓ denote the outcome leaf, and let the ancestors of ℓ be denoted by $a_1, a_2, \ldots, a_T$. Then the revenue of S^* is $\sum_{i=1}^T \chi^*_{a_i} x^*$, and the revenue of S is $\sum_{i=1}^T \chi_{a_i} x_{a_i}$. It follows that the expected difference between the two is

$$\begin{aligned}
\sum_{a\in\mathbb{T}} p_D(a)[\mathbf{E}_D(\chi^*_a x^*) - \mathbf{E}_D(\chi_a x_a)] &= \sum_{a\in\mathbb{T}} p_D(a)[x^* D(x^*) - x_a D(x_a)] \\
&= \sum_{a\in\mathbb{T}} \sum_{\ell\succ a} p_D(\ell) r_D(x_a) \\
&= \sum_{\ell\in\mathbb{T}} p_D(\ell) \left(\sum_{a\prec\ell} r_D(x_a) \right) \\
&= \mathbf{E}_D R_D(\ell).
\end{aligned}$$

□

Proof of the lower bound

In stating the upcoming lemmas, we will introduce constants $c_1, c_2, \ldots$. When we introduce such a constant we are implicitly asserting that there exists a constant $0 < c_i < \infty$ depending only on the demand curve family $\mathcal{D}$, and satisfying the property specified in the statement of the corresponding lemma.

We begin with a series of lemmas which establish that $\mathbf{E}_D K_D$ is bounded above by a constant multiple of $\mathbf{E}_D R_D$. Assume for now that D is fixed, so x^* is also fixed, and put

$$h_a = x_a - x_*.$$

Lemma 3.8. $\mathbf{E}_D R_D(\ell) \geq c_1 \sum_{a\in\mathbb{T}} p_D(a) h_a^2$.

Proof. Recall from Lemma 3.6 that

$$(x^* + h)D(x^* + h) < x^* D(x^*) - \beta h^2,$$

hence

$$r_D(x_a) = x^* D(x^*) - x_a D(x_a) > \beta h_a^2.$$

Now we see that

$$\begin{aligned}
\mathbf{E}_D R_D(\ell) &= \sum_{\ell \in \mathbb{T}} p_D(\ell) \left(\sum_{a \prec \ell} r_D(x_a) \right) \\
&= \sum_{a \in \mathbb{T}} \left(\sum_{\ell \succ a} p_D(\ell) \right) r_D(x_a) \\
&= \sum_{a \in \mathbb{T}} p_D(a) r_D(x_a) \\
&> \beta \sum_{a \in \mathbb{T}} p_D(a) h_a^2.
\end{aligned}$$

so the lemma holds with $c_1 = \beta$. □

Lemma 3.9. $\mathbf{E}_D K_D(\ell) \leq c_2 \sum_{a \in \mathbb{T}} p_D(a) h_a^2$.

Proof. As in the preceding lemma, the idea is to rewrite the sum over leaves as a sum over internal nodes and then bound the sum term-by-term. (In this case, actually it is a sum over internal edges of T.) A complication arises from the fact that the natural expression for $\mathbf{E}_D K_D(\ell)$ involves summing over pairs of ancestors of a leaf; however, we will see that all of the cross-terms cancel, leaving us with a manageable expression.

$$\begin{aligned}
\mathbf{E}_D K_D(\ell) &= \sum_{\ell} p_D(\ell) IRE_D(\ell)^2 \\
&= \sum_{\ell} p_D(\ell) \left(\sum_{e \prec \ell} ire_D(e) \right)^2 \\
&= \sum_{\ell} p_D(\ell) \left[\sum_{e \prec \ell} ire_D(e)^2 + 2 \sum_{e \prec e' \prec \ell} ire_D(e) ire_D(e') \right] \\
&= \left[\sum_{e} \sum_{\ell \succ e} p_D(\ell) ire_D(e)^2 \right] + 2 \left[\sum_{e} \sum_{e' \succ e} ire_D(e) \left(\sum_{\ell \succ e'} p_D(\ell) ire_D(e') \right) \right] \\
&= \left[\sum_{e} p_D(e) ire_D(e)^2 \right] + 2 \left[\sum_{e} ire_D(e) \left(\sum_{e' \succ e} p_D(e') ire_D(e') \right) \right]. \quad (3.1)
\end{aligned}$$

For any $e \in \mathbb{T}$, the sum $\sum_{e' \succ e} p_D(e') ire_D(e')$ vanishes because the terms may be grouped into pairs $p_D(e') ire_D(e') + p_D(e'') ire_D(e'')$ where e', e'' are the edges joining a node $a \in \mathbb{T}$ to its right and left children, respectively, and we have

$$\begin{aligned}
&p_D(e') ire_D(e') + p_D(e'') ire_D(e'') \\
&\quad = p_D(a) \left[D(x_a) \left(\frac{\dot{D}(x_a)}{D(x_a)} \right) + (1 - D(x_a)) \left(-\frac{\dot{D}(x_a)}{1 - D(x_a)} \right) \right] = 0.
\end{aligned}$$

Thus

$$\begin{aligned}
\mathbf{E}_D K_D(\ell) &= \sum_{e\in\mathbb{T}} p_D(e) ire_D(e)^2 \\
&= \sum_a p_D(a) \left[D(x_a) \left(\frac{\dot{D}(x_a)}{D(x_a)} \right)^2 + (1 - D(x_a)) \left(-\frac{\dot{D}(x_a)}{1 - D(x_a)} \right)^2 \right] \\
&\leq \sum_a p_D(a) \left[\left(\frac{\dot{D}(x_a)}{D(x_a)} \right)^2 + \left(-\frac{\dot{D}(x_a)}{1 - D(x_a)} \right)^2 \right] \\
&< \sum_a p_D(a) \left(\gamma^2 h_a^2 + \gamma^2 h_a^2 \right),
\end{aligned}$$

so the lemma holds with $c_2 = 2\gamma^2$. □

Corollary 3.10. $\mathbf{E}_D K_D(\ell) \leq c_3 \mathbf{E}_D R_D(\ell)$.

The relevance of Corollary 3.10 is that it means that when $\mathbf{E}_D R_D$ is small, then $p_{D_t}(\ell)$ cannot shrink very rapidly as a function of t, for most leaves ℓ. This is made precise by the following Lemma. Here and throughout the rest of this section, D refers to a demand curve $D_{t_0} \in \mathcal{D}$.

Lemma 3.11. *For all sufficiently large T, if $\mathbf{E}_D R_D < \sqrt{T}$ then there exists a set S of leaves such that $p_D(S) \geq 1/2$, and $p_{D_t}(\ell) > c_4 p_D(\ell)$ for all $\ell \in S$ and all $t \in \left[t_0, t_0 + T^{-1/4}\right]$.*

The proof is quite elaborate, so we have deferred it to Section 3.5.

We will also need a lemma establishing the growth rate of $R_{D_t}(\ell)$ for a fixed leaf ℓ, as t varies.

Lemma 3.12. *$R_D(\ell) + R_{D_t}(\ell) > c_5 (t - t_0)^2 T$ for all leaves $\ell \in \mathbb{T}_n$ and for all $D_t \in \mathcal{D}$.*

Proof. We know that

$$\begin{aligned}
R_D(\ell) &= \sum_{a \prec \ell} r_D(x_a) \\
R_{D_t}(\ell) &= \sum_{a \prec \ell} r_{D_t}(x_a)
\end{aligned}$$

so it suffices to prove that $r_D(x) + r_{D_t}(x) > c_4 (t - t_0)^2$ for all $x \in [0, 1]$. Assume without loss of generality that $t - t_0 > 0$. (Otherwise, we may reverse the roles of D and D_t.) Let x^* and x_t^* denote the optimal prices for D, D_t, respectively. (Recall that $D = D_{t_0}$.) Note that $x_t^* - x^* > \alpha(t - t_0)$, by property 1 of Lemma 3.6.

Let $h = x - x^*$, $h_t = x - x_t^*$, and note that $|h| + |h_t| > \alpha(t - t_0)$. Now

$$\begin{aligned} r_D(x) &> c_1|h|^2 \\ r_{D_t}(x) &> c_1|h_t|^2 \\ r_D(x) + r_{D_t}(x) &> c_1(|h|^2 + |h_t|^2) \\ &\geq \frac{1}{2}c_1(|h| + |h_t|)^2 \\ &> \frac{1}{2}c_1\alpha^2 t^2, \end{aligned}$$

so the lemma holds with $c_5 = \frac{1}{2}c_1\alpha^2$. □

We now exploit Lemmas 3.11 and 3.12 to prove that if $\mathbf{E}_D R_D$ is less than some small constant multiple of $\sqrt{T}$ when $D = D_{t_0}$, then $\mathbf{E}_{D_t} R_{D_t} = \Omega(\sqrt{T})$ on a large fraction of the interval $[t_0, t_0 + T^{-1/4}]$. The idea behind the proof is that Lemma 3.11 tells us there is a large set S of leaves whose measure does not vary by more than a constant factor as we move t across this interval, while Lemma 3.12 tells us that the regret contribution from leaves in S is $\Omega(\sqrt{n})$ for a large fraction of the t-values in this interval. In the following proposition, $c(M)$ denotes the function $\min\left\{1, \frac{1}{2}c_4c_5(1 + c_4)^{-1}M^{-2}\right\}$.

Proposition 3.13. *For all M and all sufficiently large T, if $\mathbf{E}_D R_D < c(M)\sqrt{T}$, then $\mathbf{E}_{D_t} R_{D_t} > c(M)\sqrt{T}$ for all $t \in \left[t_0 + (1/M)T^{-1/4},\, t_0 + T^{-1/4}\right]$.*

Proof. If $\mathbf{E}_D R_D < c(M)\sqrt{T}$, we may apply Lemma 3.11 to produce a set S of leaves such that $p_D(S) \geq 1/2$ and $p_{D_t}(\ell) > c_4 p_D(\ell)$ for all $\ell \in S$ and all $t \in [t_0, t_0 + T^{-1/4}]$. Now,

$$\mathbf{E}_D R_D \;\geq\; \sum_{\ell \in S} p_D(\ell) R_D(\ell)$$

and, for all $t \in [t_0 + (1/M)T^{-1/4}, t_0 + T^{-1/4}]$,

$$\begin{aligned} \mathbf{E}_{D_t} R_{D_t} &\geq \sum_{\ell \in S} p_{D_t}(\ell) R_{D_t}(\ell) \\ &\geq c_4 \sum_{\ell \in S} p_D(\ell) R_{D_t}(\ell) \\ &> c_4 \sum_{\ell \in S} p_D(\ell)(c_5(t - t_0)^2 T - R_D(\ell)) \\ &\geq c_4 p_D(S) c_5 \sqrt{T}/M^2 - c_4 \mathbf{E}_D R_D \\ &> (c_4c_5/2M^2)\sqrt{T} - c_4 c(M)\sqrt{T} \\ &\geq c(M)\sqrt{T} \end{aligned}$$

where the fourth line is derived from the third by applying Lemma 3.12. □

Theorem 3.14. *Let $\mathbb{S}$ be any randomized non-uniform strategy, and let $\mathcal{R}_D(\mathbb{S}, T)$ denote the expected ex ante regret of S on a population of T buyers whose valuations are independent random samples from the probability distribution specified by the demand curve D. Then*

$$\Pr_{D \leftarrow \mathcal{D}} \left(\limsup_{T \to \infty} \frac{\mathcal{R}_D(\mathbb{S}, T)}{\sqrt{T}} > 0 \right) = 1.$$

In other words, if D is drawn at random from $\mathcal{D}$, then almost surely $\mathcal{R}_D(\mathbb{S}, T)$ is not $o(\sqrt{T})$.

Proof. It suffices to prove the theorem for a deterministic strategy $\mathbb{S}$, since any randomized strategy is a probability distribution over such strategies. Now assume, to the contrary, that

$$\Pr_{D \leftarrow \mathcal{D}} \left(\limsup_{T \to \infty} \frac{\mathcal{R}_D(\mathbb{S}, T)}{\sqrt{T}} = 0 \right) > 0. \tag{3.2}$$

and choose M large enough that the left side of (3.2) is greater than $1/M$. Recall from Lemma 3.8 that $\mathbf{E}_D R_D = \mathcal{R}_D(\mathbb{S}, T)$. We know that for every $D = D_{t_0} \in \mathcal{D}$ such that $\mathbf{E}_D R_D < c(M)\sqrt{T}$,

$$\mathbf{E}_{D_t} R_{D_t} > c(M)\sqrt{T} \quad \forall t \in [t_0 + (1/M)T^{-1/4}, t_0 + T^{-1/4}]. \tag{3.3}$$

Now choose N large enough that the set

$$X_N = \left\{ D \in \mathcal{D} : \sup_{T > N} \frac{\mathcal{R}_D(\mathbb{S}, T)}{\sqrt{T}} < c(M) \right\}$$

has measure greater than $1/M$. Replacing X_N if necessary with a proper subset still having measure greater than $1/M$, we may assume that $\{t : D_t \in X_N\}$ is disjoint from $[0.4 - \varepsilon, 0.4]$ for some $\varepsilon > 0$. Choosing T large enough that $T > N$ and $T^{-1/4} < \varepsilon$, equation (3.3) ensures that the sets

$$X_N^k = \{D_s : s = t + (k/M)T^{-1/4}, D_t \in X_N\}$$

are disjoint for $k = 0, 1, \ldots, M - 1$. But each of the sets X_N^k, being a translate of X_N, has measure greater than $1/M$. Thus their total measure is greater than 1, contradicting the fact that $\mathcal{D}$ has measure 1. □

General demand-curve models

The methods of the preceding section extend to much more general families of demand curves. Here we will merely sketch the ideas underlying the extension. Suppose that $\mathcal{D}$ is a compact subset of the space $C^4([0, 1])$ of functions on $[0, 1]$ with continuous fourth derivative, and that the demand curves $D \in \mathcal{D}$ satisfy the following two additional hypotheses:

- **(Unique global max)** The function $f(x) = xD(x)$ has a unique global maximum $x^* \in [0, 1]$, and it lies in the interior of the interval.
- **(Non-degeneracy)** The second derivative of f is strictly negative at x^*.

Suppose $\mathcal{D}$ is also endowed with a probability measure, denoted μ. The proof of the lower bound relied heavily on the notion of being able to make a "one-parameter family of perturbations" to a demand curve. This notion may be encapsulated using a *flow* $\phi(D, t)$ mapping an open set $U \subseteq \mathcal{D} \times \mathbb{R}$ into $\mathcal{D}$, such that $\{D \in \mathcal{D} : (D, 0) \in U\}$ has measure 1, and $\phi(D, 0) = D$ when defined. We will use the shorthand D_t for $\phi(D, t)$. The flow must satisfy the following properties:

- **(Additivity)** $\phi(D, s + t) = \phi(\phi(D, s), t)$.
- **(Measure-preservation)** If $X \subseteq \mathcal{D}$ and $\phi(D, t)$ is defined for all $D \in X$, then $\mu(\phi(X, t)) = \mu(X)$.
- **(Smoothness)** The function $g(t, x) = D_t(x)$ is a C^4 function of t and x.
- **(Profit-preservation)** If x_t^* denotes the point at which the function $xD_t(x)$ achieves its global maximum, then $x_t^* D_t(x_t^*) = x_0^* D_0(x_0^*)$ for all t such that D_t is defined.
- **(Non-degeneracy)** $\frac{d}{dt}(x_t^*) \neq 0$.
- **(Rate dampening at 0 and 1)** For $k = 1, 2, 3, 4$, the functions $\left|\frac{D^{(k)}}{D}\right|$ *and* $\left|\frac{D^{(k)}}{1-D}\right|$ are uniformly bounded above, where $D^{(k)}$ denotes the k-th derivative of D with respect to t.

Provided that these axioms are satisfied, it is possible to establish all of the properties specified in Lemma 3.6. Property 1 follows from compactness of $\mathcal{D}$ and non-degeneracy of ϕ, property 2 follows from the compactness of $\mathcal{D}$ together with the non-degeneracy and "unique global max" axioms for $\mathcal{D}$, and property 4 is the rate-dampening axiom. Property 3 is the subtlest: it follows from the smoothness, profit-preservation, and rate-dampening properties of ϕ. The key observation is that profit-preservation implies that

$$x^* D_t(x^*) \leq x_t^* D_t(x_t^*) = x^* D(x^*),$$

so that $x^* D_t(x^*)$, as a function of t, is maximized at $t = 0$. This, coupled with smoothness of ϕ, proves that $\dot{D}(x^*) = 0$. Another application of smoothness yields the desired bounds.

The final steps of Theorem 1 used the translation-invariance of Lebesgue measure on the interval $[0.3, 0.4]$ to produce M sets whose disjointness yielded the desired contradiction. This argument generalizes, with the flow ϕ playing the role of the group of translations. It is for this reason that we require ϕ to satisfy the additivity and measure-preservation axioms.

3.2.3 Upper bound

The upper bound on regret in the random-valuation model is based on applying the multi-armed bandit algorithm UCB1 [3] which was presented and analyzed in Section 2.4 of this thesis. To do so, we discretize the set of possible actions by limiting the seller to use pricing strategies which only offer prices belonging to the set $\{1/K, 2/K, \ldots, 1 - 1/K, 1\}$, for suitably-chosen K. (It will turn out that $K = \theta((T/\log T)^{1/4})$ is the best choice.)

We are now in a setting where the seller must choose one of K possible actions on each of T trials, where each action yields a reward which is a random variable taking values in $[0, 1]$, whose distribution depends on the action chosen, but the rewards for a given action are i.i.d. across the T trials. This is the scenario studied in Section 2.4. There, we defined $\bar{c}(i)$ to be the expected reward of action i, $i^* = \arg\max_{1 \le i \le K} \bar{c}(i)$, and

$$\Delta_i = \bar{c}(i^*) - \bar{c}(i). \tag{3.4}$$

We also defined the notion of a "(ζ, s_0)-bounded adversary"; in the present context, the random payoff distribution represents a $(1, 1)$-bounded adversary according to Lemma 2.7, since all payoffs are between 0 and 1. Applying Theorem 2.8 and using the fact that $\zeta = s_0 = 1$, we find that the regret of UCB1 satisfies

$$\textit{Regret}(\mathsf{UCB1}) \le \varepsilon T + \left[\sum_{i \,:\, \Delta_i > \varepsilon} \left(32\Delta_i + \frac{32}{\Delta_i} \right) \right] \log T + \left(1 + \frac{\pi^2}{3} \right) \sum_{i \in \mathbb{S}} \Delta_i \tag{3.5}$$

for any $\varepsilon \ge 0$.

To apply this bound, we need to know something about the values of $\Delta_1, \ldots, \Delta_K$ in the special case of interest to us. When the buyer's valuation is v, the payoff of action i/K is

$$X_i = \begin{cases} i/K & \text{if } v \ge i/K \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

Hence

$$\bar{c}(i) = \mathbf{E}(X_i) = (i/K) D(i/K). \tag{3.7}$$

Recall that we are making the following hypothesis on the demand curve D: the function $f(x) = xD(x)$ has a unique global maximum at $x^* \in (0, 1)$, and $f''(x^*)$

is defined and strictly negative. This hypothesis is useful because it enables us to establish the following lemma, which translates directly into bounds on Δ_i.

Lemma 3.15. *There exist constants C_1, C_2 such that*

$$C_1(x^* - x)^2 < f(x^*) - f(x) < C_2(x^* - x)^2$$

for all $x \in [0, 1]$.

Proof. The existence and strict negativity of $f''(x^*)$ guarantee that there are constants $A_1, A_2, \varepsilon > 0$ such that $A_1(x^* - x)^2 < f(x^*) - f(x) < A_2(x^* - x)^2$ for all $x \in (x^* - \varepsilon, x^* + \varepsilon)$. The compactness of $X = \{x \in [0, 1] : |x^* - x| \geq \varepsilon\}$, together with the fact that $f(x^*) - f(x)$ is strictly positive for all $x \in X$, guarantees that there are constants B_1, B_2 such that $B_1(x^* - x)^2 < f(x^*) - f(x) < B_2(x^* - x)^2$ for all $x \in X$. Now put $C_1 = \min\{A_1, B_1\}$ and $C_2 = \max\{A_2, B_2\}$ to obtain the lemma. □

Corollary 3.16. *If $\tilde{\Delta}_0 \leq \tilde{\Delta}_1 \leq \ldots \leq \tilde{\Delta}_{K-1}$ are the elements of the set $\{\Delta_1, \ldots, \Delta_K\}$ sorted in ascending order, then $\tilde{\Delta}_j \geq C_1(j/2K)^2 - C_2/K^2$.*

Proof. We have

$$\Delta_i = \bar{c}(i^*) - \bar{c}(i) = f(i^*/K) - f(i/K) = (f(x^*) - f(i/K)) - (f(x^*) - f(i^*/K))$$

We know that $f(x^*) - f(i/K) > C_1(x^* - i/K)^2$. To put an upper bound on $f(x^*) - f(i^*/K)$, let $i_0 = \lfloor Kx^* \rfloor$ and observe that $x^* - i_0/K < 1/K$ and $f(x^*) - f(i_0/K) < C_2/K^2$. Thus

$$\Delta_i > C_1(x^* - i/K)^2 - C_2/K^2.$$

The lower bound on $\tilde{\Delta}_j$ follows upon observing that at most j elements of the set $\{1/K, 2/K, \ldots, 1\}$ lie within a distance less than $j/2K$ of x^*. □

Corollary 3.17. $\mu^* > x^* D(x^*) - C_2/K^2$.

Proof. At least one of the numbers $\{1/K, 2/K, \ldots, 1\}$ lies within $1/K$ of x^*; now apply the upper bound on $f(x^*) - f(x)$ stated in Lemma 3.15. □

Putting all of this together, we have derived the following upper bound.

Theorem 3.18. *Assuming that the function $f(x) = xD(x)$ has a unique global maximum $x^* \in (0, 1)$, and that $f''(x^*)$ is defined and strictly negative, the strategy* UCB1 *with $K = \lceil (T/\log T)^{1/4} \rceil$ achieves expected regret $O(\sqrt{T \log T})$.*

Proof. Consider the following four strategies:

- UCB1, the strategy defined in [3].

- $\mathbb{S}^{\text{opt}}$, the optimal fixed-price strategy.
- $\mathbb{S}^*$, the fixed-price strategy which offers x^* to every buyer.
- $\mathbb{S}^*_K$, the fixed-price strategy which offers i^*/K to every buyer, where i^*/K is the element of $\{1/K, 2/K, \ldots, 1\}$ closest to x^*.

As usual, we will use $\rho(\cdot)$ to denote the expected revenue obtained by a strategy. We will prove a $O(\sqrt{T \log T})$ upper bound on each of $\rho(\mathbb{S}^*_K) - \rho(\mathsf{UCB1})$, $\rho(\mathbb{S}^*) - \rho(\mathbb{S}^*_K)$, and $\rho(\mathbb{S}^{\text{opt}}) - \rho(\mathbb{S}^*)$, from which the theorem follows immediately.

We first show, using (3.5), that $\rho(\mathbb{S}^*_K) - \rho(\mathsf{UCB1}) = O(\sqrt{T \log T})$. Let $\varepsilon = 2C_2/K^2 = O(\sqrt{\log(T)/T})$, and let $\tilde{\Delta}_0 \leq \ldots \leq \tilde{\Delta}_{K-1}$ be as in Corollary 3.16. Let $j_0 = \sqrt{12C_2/C_1}$, so that $\tilde{\Delta}_j > \varepsilon$ when $j > j_0$. By Corollary 3.16,

$$\begin{aligned}
\sum_{\tilde{\Delta}_j > \varepsilon} \left(\frac{1}{\tilde{\Delta}_j} \right) &< \frac{j_0}{\varepsilon} + \sum_{j=j_0+1}^{K} C_1^{-1} (j/2K)^{-2} \\
&< \frac{K^2}{2C_2} \sqrt{\frac{12C_2}{C_1}} + \frac{4K^2}{C_1} \sum_{j=1}^{\infty} j^{-2} \\
&= \left(\sqrt{\frac{3}{C_1 C_2}} + \frac{2\pi^2}{3C_1} \right) K^2 \\
&= O((T/\log T)^{1/2}).
\end{aligned}$$

Also,

$$\varepsilon T = \frac{2C_2 T}{K^2} = O(\sqrt{T \log T})$$

and

$$\begin{aligned}
\sum_{j=1}^{K} \Delta_j \log T &< K \log T \\
&= O(T^{1/4} \log T)^{3/4}).
\end{aligned}$$

Plugging these estimates into 3.5, we see that the regret of $\mathsf{UCB1}$ relative to $\mathbb{S}^*_K$ is $O((T \log T)^{1/2})$, as claimed.

Next we bound the difference $\rho(\mathbb{S}^*) - \rho(\mathbb{S}^*_K)$. The expected revenues of $\mathbb{S}^*$ and $\mathbb{S}^*_K$ are $Tx^*D(x^*)$ and $T\mu^*$, respectively. Applying Corollary 3.17, the regret of $\mathbb{S}^*_K$ relative to $\mathbb{S}^*$ is bounded above by

$$\frac{C_2 T}{K^2} \leq \frac{C_2 T}{(T/\log T)^{1/2}} = O((T \log T)^{1/2}).$$

Finally, we must bound $\rho(\mathbb{S}^{\text{opt}}) - \rho(\mathbb{S}^*)$. For any $x \in [0,1]$, let $\rho(x)$ denote the revenue obtained by the fixed-price strategy which offers price x, and let $x^{\text{opt}} = \arg\max_{x\in[0,1]} \rho(x)$. We begin by observing that for all $x < x^{\text{opt}}$,

$$\rho(x) \geq \rho(x^{\text{opt}}) - T(x^{\text{opt}} - x).$$

This is simply because every buyer that accepts price x^{opt} would also accept x, and the amount of revenue lost by setting the lower price is $x^{\text{opt}} - x$ per buyer. Now

$$\begin{aligned}\int_0^1 \Pr(\rho(x) - \rho(x^*) > \lambda)dx &\geq \int_0^1 \Pr\left(\rho(x^{\text{opt}}) - \rho(x^*) > 2\lambda \ \wedge \ x^{\text{opt}} - x < \frac{\lambda}{T}\right) dx \\ &= \frac{\lambda}{T}\Pr(\rho(x^{\text{opt}}) - \rho(x^*) > 2\lambda),\end{aligned}$$

so a bound on $\Pr(\rho(x)-\rho(x^*) > \lambda)$ for fixed x translates into a bound on $\Pr(\rho(x^{\text{opt}})-\rho(x^*) > \lambda)$. But for fixed x, the probability in question is the probability that a sum of T i.i.d. random variables, each supported in $[-1,1]$ and with negative expectation, exceeds λ. The Chernoff-Hoeffding bound tells us that

$$\Pr(\rho(x) - \rho(x^*) > \lambda) < e^{-\lambda^2/2T},$$

so

$$\Pr(\rho(x^{\text{opt}}) - \rho(x^*) > 2\lambda) < \min\{1, \frac{T}{\lambda}e^{-\lambda^2/2T}\}.$$

Finally,

$$\begin{aligned}\mathbf{E}(\rho(x^{\text{opt}}) - \rho(x^*)) &< \int_0^\infty \Pr(\rho(x^{\text{opt}}) - \rho(x^*) > y)dy \\ &< \int_0^\infty \min\{1, \frac{2T}{y}e^{-y^2/2T}\}dy \\ &< \int_0^{\sqrt{4T\log T}} dy + \frac{2T}{\sqrt{4T\log T}}\int_{\sqrt{4T\log T}}^\infty e^{-y^2/2T}dy \\ &= O(\sqrt{T\log T}).\end{aligned}$$

□

Remark 3.2. If the seller does not have foreknowledge of T, it is still possible to achieve regret $O(\sqrt{T\log T})$ using the doubling technique introduced in Section 2.6.

3.3 Worst-case valuations

In the worst-case valuation model, we assume that the buyers' valuations are chosen by an adversary who has knowledge of T and of the pricing strategy, but is oblivious

to the algorithm's random choices. This means that the pricing problem, in the worst-case valuation model, is a special case of the adversarial multi-armed bandit problem [4], and we may apply the algorithm Exp3 which was presented and analyzed in Section 2.5. This algorithm was first applied in the setting of on-line auctions by Blum *et. al.* [21], who normalize the buyers' valuations to lie in an interval $[1, h]$ and then prove the following theorem:

Theorem 3.19 ([21], Theorem 5.). *For all $\varepsilon > 0$, there exists a pricing strategy* Exp3 *and a constant $c(\varepsilon)$ such that for all valuation sequences, if the optimal fixed-price revenue $\rho(\mathbb{S}^{\text{opt}})$ satisfies $\rho(\mathbb{S}^{\text{opt}}) > c(\varepsilon)h \log h \log\log h$, then* Exp3 *is $(1+\varepsilon)$-competitive relative to $\rho(\mathbb{S}^{\text{opt}})$.*

Our upper and lower bounds for regret in the worst-case valuation model are based on the techniques employed in [4] and [21]. The upper bound in Theorem 1.10 is virtually a restatement of Blum *et. al.*'s theorem, though the change in emphasis from competitive ratio to additive regret necessitates a minor change in technical details. Our worst-case lower bound (Theorem 3.20) is influenced by Auer *et. al.*'s proof of the corresponding lower bound for the adversarial multi-armed bandit problem in [4].

3.3.1 Upper bound

Following [21], as well as the technique used in Section 3.2.3 above, we specify a finite set of offer prices $X = \{1/K, 2/K, \ldots, 1\}$ and constrain the seller to select prices from this set only. This reduces the online pricing problem to an instance of the multi-armed bandit problem, to which the algorithm Exp3 of Section 2.5 may be applied. Denote this pricing strategy by $\mathbb{S}$. If $\mathbb{S}_X^{\text{opt}}$ denotes the fixed-price strategy which chooses the best offer price i^*/K from X, and $\mathbb{S}^{\text{opt}}$ denotes the fixed-price strategy which chooses the best offer price x^* from $[0, 1]$, we have the following inequalities:

$$\begin{aligned} \rho(\mathbb{S}_K^{\text{opt}}) - \rho(\mathbb{S}) &= O(\sqrt{TK \log K}) \\ \rho(\mathbb{S}^*) - \rho(\mathbb{S}_K^{\text{opt}}) &< T(1/K) = T/K \end{aligned}$$

where the first inequality is derived from Theorem 2.12 and the second inequality follows from the fact that $\mathbb{S}_K^{\text{opt}}$ is no worse than the strategy which offers $\frac{1}{K}\lfloor Kx^* \rfloor$ to each buyer.

Setting $K = \lceil T/\log T \rceil^{1/3}$, both $\sqrt{TK \log K}$ and T/K are $O(T^{2/3}(\log T)^{1/3})$. We have thus expressed the regret of **Exp3** as a sum of two terms, each of which is $O(T^{2/3}(\log T)^{1/3})$, establishing the upper bound asserted in Theorem 1.10.

Readers familiar with [21] will recognize that the only difference between this argument and their Theorem 5 is that they choose the prices in X to form a geometric

progression (so as to optimize the competitive ratio) while we choose them to form an arithmetic progression (so as to optimize the additive regret).

3.3.2 Lower bound

In [4], the authors present a lower bound of $\sqrt{TK}$ for the multi-armed bandit problem with payoffs selected by an oblivious adversary. Ironically, the power of the adversary in this lower bound comes not from adapting to the on-line algorithm ALG, but from adapting to the number of trials T. In fact, the authors define a model of random payoffs (depending on T but not on the algorithm) such that the expected regret of *any* algorithm on a random sample from this distribution is $\Omega(\sqrt{TK})$. The idea is select one of the K actions uniformly at random and designate it as the "good" action. For all other actions, the payoff in each round is a uniform random sample from $\{0, 1\}$, but for the good action the payoff is a biased sample from $\{0, 1\}$, which is equal to 1 with probability $1/2 + \varepsilon$, where $\varepsilon = \theta(\sqrt{K/T})$. A strategy which knows the good action will achieve expected payoff $(1/2 + \varepsilon)T = 1/2 + \theta(\sqrt{TK})$. It can be shown, for information-theoretic reasons, that no strategy can learn the good action rapidly and reliably enough to play it more than $T/K + \theta(\varepsilon\sqrt{T^3/K})$ times in expectation, from which the lower bound on regret follows. Our version of this lower bound proof — adapted from the proof in [4] — was presented earlier, in Section 2.8. There we proved a slightly weaker result, in that the payoff distribution was allowed to depend on the algorithm in addition to depending on T.

A similar counterexample can be constructed in the context of our online pricing problem, i.e. given any algorithm, one can construct a probability distribution on buyers' valuations such that the expected regret of the algorithm on a sequence of independent random samples from this distribution is $\Omega(T^{2/3})$. The idea is roughly the same as above: one chooses a subinterval of $[0, 1]$ of length $1/K$ to be the interval of "good prices", and chooses the distribution of buyers' valuations so that the expected revenue per buyer is a constant independent of the offer price outside the interval of good prices, and is ε higher than this constant inside the interval of good prices. As above, there is a trade-off between choosing ε too large (which makes it too easy for strategies to learn which prices belong to the good interval) or too small (which leads to a negligible difference in revenue between the best strategy and all others), and the optimal trade-off is achieved when $\varepsilon = \theta(\sqrt{K/T})$. However, in our setting there is an additional constraint that $\varepsilon \leq 1/K$, since the expected payoff can grow by no more than $1/K$ on an interval of length $1/K$. This leads to the values $K = \theta(T^{1/3}), \varepsilon = \theta(T^{-1/3})$ and yields the stated lower bound of $\Omega(T^{2/3})$.

There are two complications which come up along the way. One is that the seller's algorithm has a continuum of alternatives at every step, rather than a finite set of

K alternatives as in Section 2.8. This can be dealt with by restricting the buyers' valuations to lie in a finite set $V = \{v_1, v_2, \ldots, v_K\}$. Then the seller can gain no advantage from offering a price which lies outside of V, so we may assume the seller is constrained to offer prices in V and prove lower bounds for this restricted class of strategies.

The second complication which arises is that the adversary in Section 2.8 was more powerful: it could specify the reward for each action independently, whereas our adversary can only set a valuation v, and this v determines the rewards for all actions simultaneously. While this entails choosing a more complicated reward distribution, the complication only makes the computations messier without introducing any new ideas into the proof.

Theorem 3.20. *For any $T > 0$ and any pricing strategy $\mathbb{S}$, there exists a probability distribution on $[0, 1]$ such that if the valuations of T buyers are sampled independently at random from this distribution, the expected regret of $\mathbb{S}$ on this population of buyers is $\Omega(T^{2/3})$.*

Proof sketch. For simplicity, assume $T = \frac{1}{2}K^3$, and put $\varepsilon = \frac{1}{4K}$. The valuations will be independent random samples from the set $V = \{0, \frac{3}{4}, \frac{3}{4} + \varepsilon, \frac{3}{4} + 2\varepsilon, \ldots, 1 - \varepsilon, 1\}$. A "baseline probability distribution" p_{base} on V is defined so that

$$p_{\text{base}}(\{v \geq 1 - i\varepsilon\}) = \frac{1}{2}(1 - i\varepsilon)^{-1} \quad (0 \leq i \leq K).$$

A finite family of probability distributions $\{p_j^T\}_{j=1}^K$ is defined as follows: to generate a random sample from p_j^T, one samples $v \in V$ at random from the distribution p_{base}, and then one adds ε to it if and only if $v = 1 - j\varepsilon$.

If v_1, v_2 are consecutive elements of V and $\mathbb{S}$ offers a price x such that $v_1 < x < v_2$, then $\mathbb{S}$ could obtain at least as much revenue (against buyers with valuations in V) by offering price v_2 instead of x. Thus we may assume, without loss of generality, that $\mathbb{S}$ never offers a price outside of V.

Our proof now parallels the proof of Theorem 2.25 given in Section 2.8. One defines a Bernoulli random variable $\chi_j(x_t)$ which is equal to 1 if $x_t = 1 - (j-1)\varepsilon$, and 0 otherwise, and one defines

$$Q_j(\mathbb{S}, p; T) = \mathbf{E}\left[\sum_{t=1}^{T} \chi_j(x_t)\right]$$

to be the expected number of times $\mathbb{S}$ offers price $1 - (j-1)\varepsilon$ to a sequence of buyers whose valuations are independent samples from distribution p. Note that the expected revenue per transaction at this price is at least $\frac{1}{2}(1+\varepsilon)$, whereas the expected

revenue at any other price in V is at most $\frac{1}{2}$. Consequently, the regret of $\mathbb{S}$ is at least $\frac{1}{2}\varepsilon[T - Q_j(\mathbb{S}, p_j^T; T)]$. Recalling that $\varepsilon = \Omega(T^{-1/3})$, we see now that it suffices to prove that there exists a j such that $Q_j(\mathbb{S}, p_j^T; T) \leq \frac{T}{2} + \frac{T}{K}$. This is accomplished using exactly the same steps as in the proof of Theorem 2.25. First one finds j such that $Q_j(\mathbb{S}, p_{\text{base}}; T) \leq T/K$, then one proves that

$$Q_j(\mathbb{S}, p_j^T; T) - Q_j(\mathbb{S}, p_{\text{base}}; T) \leq T/2$$

with the aid of Theorem 2.20 and an upper bound on the KL-divergence of two distributions on transcripts: one defined by using strategy $\mathbb{S}$ against T samples coming from distribution p_{base}, and the other defined by using $\mathbb{S}$ against T samples from p_j^T. The key observation is that if one offers a price $x \in V \setminus \{1 - (j-1)\varepsilon\}$ to a buyer whose value is randomly sampled from v, the probability that the buyer accepts price x does not depend on whether the buyer's value was sampled according to p_{base} or p_j^T. On the other hand, if one offers price $1 - (j-1)\varepsilon$ to a buyer whose value is randomly sampled according to p_{base} or p_j^T, these define two different distributions on single-transaction outcomes, and the KL-divergence of these two distributions is at most $16\varepsilon^2$. Summing over all T transactions — and recalling that at most T/K of these transactions, in expectation, take place at price $1 - (j-1)\varepsilon$ — we conclude that the KL-divergence of the two transcripts is at most $16\varepsilon^2(T/K) = T/K^3 \leq \frac{1}{2}$. As in the proof of Theorem 2.25, this upper bound on the KL-divergence is enough to finish the argument. □

3.4 Proof of Lemma 3.6

In this section we restate and prove Lemma 3.6.

Lemma 3.21. *There exist constants $\alpha, \beta > 0$ and $\gamma < \infty$ such that for all $D = D_{t_0} \in D$ and $x \in [0,1]$:*

1. $\frac{d}{dt}(x_t^*)|_{t=t_0} > \alpha$;
2. $x^* D(x^*) - xD(x) > \beta(x^* - x)^2$;
3. $|\dot{D}(x)/D(x)| < \gamma|x^* - x|$ *and* $|\dot{D}(x)/(1 - D(x))| < \gamma|x^* - x|$;
4. $|D^{(k)}(x)/D(x)| < \gamma$ *and* $|D^{(k)}(x)/(1 - D(x))| < \gamma$, *for* $k = 2, 3, 4$.

Here x^* denotes $x_{t_0}^*$, $D^{(k)}(x)$ denotes the k-th t-derivative of $D_t(x)$ at $t = t_0$, and $\dot{D}(x)$ denotes $D^{(1)}(x)$.

Proof. We begin with some useful observations about the relation between $\tilde{D}_t$ and D_t. The function $\tilde{D}_t$ is piecewise-linear, and linear functions are preserved under convolution with an even function whose integral is 1. Recall that the bump function b is an even function supported in $[-0.01, 0.01]$ and satisfying $\int_{-0.01}^{0.01} b(x)dx = 1$; hence $D_t(x) = \tilde{D}_t(x)$ unless x is within 0.01 of one of the two points where the derivative of $\tilde{D}_t$ is discontinuous. The x-coordinates of these two points are given by

$$\begin{aligned} x_0 &= \frac{7t^2 - 2t}{14t^2 - 1} \\ x_1 &= \frac{7t^2 - 4t}{7t^2 - 2}. \end{aligned}$$

For t in the range $[0.3, 0.4]$ this means that $x_0 \in (0.115, 0.259)$, $x_1 \in (0.416, 0.546)$. Recalling that b is a C^∞ function, we find that $t \mapsto \tilde{D}_t$ is a continuous mapping from $[0.3, 0.4]$ to $C^\infty([0, 1])$. Hence $\{\tilde{D}_t : 0.3 \leq t \leq 0.4\}$ is a compact subset of $C^\infty([0, 1])$, and consequently for $1 \leq k < \infty$, the k-th derivative of $\tilde{D}_t$ is bounded uniformly in t.

We now proceed to prove each of the properties stated in the Lemma.

1. First we verify that $x_t^* = t$, as stated in Section 3.2.2. If x lies in the interval $I_t = [x_0 + 0.01, x_1 - 0.01]$ where $D_t(x) = 2/7t - x/7t^2$, then $xD_t(x) = 2x/7t - x^2/7t^2 = \frac{1}{7}[1 - (1 - x/t)^2]$, which is uniquely maximized when $x = t$ and $xD_t(x) = 1/7$. Note that the estimates given above for x_0 and x_1 ensure that $[x_0 + 0.01, x_1 - 0.01]$ always contains $[0.3, 0.4]$, so t always lies in this interval. If x lies in the interval where $D_t(x) = 1 - 2x$ or $D_t(x) = (1 - x)/2$, then $xD_t(x)$ is equal to $2(1/16 - (x - 1/4)^2)$ or $(1/2)(1/4 - (x - 1/2)^2)$, and in either case $xD_t(x)$ can not exceed $1/8$. It is straightforward but tedious to verify that $xD_t(x)$ is bounded away from $1/7$ when $|x - x_0| \leq 0.01$ or $|x - x_1| \leq 0.01$; this confirms that $x_t^* = t$ is the unique global maximum of the function $xD_t(x)$. Having verified this fact, it follows immediately that $d/dt(x_t^*)|_{t=t_0} = 1$.

2. On the interval I_t where $D_t(x) = 2/7t - x/7t^2$, we have

$$xD_t(x) = 1/7 - \frac{1}{7t^2}(t - x)^2 = x_t^* D_t(x_t^*) - \frac{1}{7t^2}(x_t^* - x)^2$$

$$x_t^* D_t(x_t^*) - xD_t(x) = \frac{1}{7t^2}(x_t^* - x)^2. \tag{3.8}$$

   We have seen that for $t \in [0.3, 0, 4]$, $xD_t(x)$ attains its maximum value of $1/7$ at a point $x_t^* \in I_t$ and is strictly less than $1/7$ at all other points of $[0, 1]$. By compactness it follows that there exist $\varepsilon, \delta > 0$ such that

$$|x - x_t^*| > \delta \implies x_t^* D_t(x_t^*) - xD_t(x) > \varepsilon, \tag{3.9}$$

for all $x \in [0,1], t \in [0.3, 0.4]$. Combining (3.8), which holds when $x \in I_t$, with (3.9), which holds when $x \notin (x_t^* - \delta, x_t^* + \delta)$, we obtain

$$x_t^* D_t(x_t^*) - x D_t(x) \geq \min\{1/7t^2, \varepsilon/\delta^2\}(x - x_t^*)^2$$

for all $x \in [0,1]$.

3. If $x < 0.1$ or $x > 0.6$, then $D_t(x)$ is independent of t, so $\dot{D}(x) = 0$, which establishes the desired inequality. If $x \in [0.1, 0.6]$, then $D(x)$ and $1 - D(x)$ are both bounded below by 0.2, so it remains to verify that $\sup\{|\dot{D}_t(x)/(x_t^* - x)|\} < \infty$. The function $|\dot{D}_t(x)|$ is a continuous function of t and x, so by compactness it is bounded above by a constant. It follows that for any constant $\varepsilon > 0$, $\sup\{|\dot{D}_t(x)/(x_t^* - x)| : \varepsilon < |x_t^* - x|\} < \infty$. Choose ε small enough that $[x_t^* - \varepsilon, x_t^* + \varepsilon]$ is contained in the interval I_t where $D_t(x) = 2/7t - x/7t^2$ for all $t \in [0.3, 0.4]$. Then for $|x_t^* - x| \leq \varepsilon$,

$$\dot{D}_t(x) = -2/7t^2 + 2x/7t^3 = 2(x-t)/7t^3 = -\frac{2}{7t^3}(x_t^* - x),$$

so $\sup\{|\dot{D}_t(x)/(x_t^* - x)|\} < \infty$ as claimed.

4. As before, if $x \notin [0.1, 0.6]$ then $D^{(k)}(x) = 0$ so there is nothing to prove. If $x \in [0.1, 0.6]$ then $D(x)$ and $1 - D(x)$ are both bounded below by 0.2, and $|D^{(k)}(x)|$ is uniformly bounded above, by compactness.

□

3.5 Proof of Lemma 3.11

In this section we restate and prove Lemma 3.11.

Lemma 3.22. *For all sufficiently large T, if $\mathbf{E}_D R_D < \sqrt{T}$ then there exists a set S of leaves such that $p_D(S) \geq 1/2$, and $p_{D_t}(\ell) > c_4 p_D(\ell)$ for all $\ell \in S$ and all $t \in \left[t_0, t_0 + T^{-1/4}\right]$.*

Proof. It suffices to prove that there exists a set S of leaves such that $p_D(S) \geq 1/2$ and $|\log(p_{D_t}(\ell)/p_D(\ell))|$ is bounded above by a constant for $\ell \in S$. Let $F(t, \ell) = \log(p_{D_t}(\ell))$. By Taylor's Theorem, we have

$$\begin{aligned} F(t,\ell) - F(t_0,\ell) &= F'(t_0,\ell)(t-t_0) + \frac{1}{2}F''(t_0,\ell)(t-t_0)^2 + \\ &\quad \frac{1}{6}F'''(t_0,\ell)(t-t_0)^3 + \frac{1}{24}F''''(t_1,\ell)(t-t_0)^4, \end{aligned}$$

for some $t_1 \in [t_0, t]$. (Here F', F'', F''', F'''' refer to the t-derivatives of F. Throughout this section, we will adopt the same notational convention when referring to the t-derivatives of other functions, in contrast to the "dot" notation used in other sections of this chapter.) This means that

$$\begin{aligned}\left|\log\left(\frac{p_{D_t}(\ell)}{p_D(\ell)}\right)\right| &\leq |F'(t_0,\ell)|T^{-1/4} + \frac{1}{2}|F''(t_0,\ell)|T^{-1/2} + \\ &\quad \frac{1}{6}|F'''(t_0,\ell)|T^{-3/4} + \frac{1}{24}|F''''(t_1,\ell)|T^{-1}. \end{aligned} \tag{3.10}$$

We will prove that, when ℓ is randomly sampled according to p_D, the expected value of each term on the right side of (3.10) is bounded above by a constant. By Markov's Inequality, it will follow that right side is bounded above by a constant for a set S of leaves satisfying $p_D(S) \geq 1/2$, thus finishing the proof of the Lemma.

Unfortunately, bounding the expected value of the right side of (3.10) requires a separate computation for each of the four terms. For the first term, we observe that $|F'(t_0,\ell)|^2$ is precisely $K_D(\ell)$, so $\mathbf{E}_D\left(|F'(t_0,\ell)|^2\right) \leq c_3\sqrt{T}$ by Corollary 3.10. It follows, using the Cauchy-Schwarz Inequality, that $\mathbf{E}_D\left(|F'(t_0,\ell)|T^{-1/4}\right) \leq \sqrt{c_3}$.

To bound the remaining three terms, let $a_0, a_1, \ldots, a_n = \ell$ be the nodes on the path in T from the root a_0 down to the leaf ℓ. Let

$$q(a_i) = \begin{cases} D(x_{a_i}) & \text{if } \chi(a_i) = 1 \\ 1 - D(x_{a_i}) & \text{if } \chi(a_i) = 0. \end{cases}$$

We have

$$p_D(\ell) = \prod_{i=0}^{T-1} q(a_i),$$

so

$$F(t_0,\ell) = \sum_{i=0}^{T-1} \log q(a_i) \tag{3.11}$$

$$F'(t_0,\ell) = \sum_{i=0}^{T-1} \frac{q'(a_i)}{q(a_i)} \tag{3.12}$$

$$F''(t_0,\ell) = \sum_{i=0}^{T-1} \frac{q''(a_i)}{q(a_i)} - \left(\frac{q'(a_i)}{q(a_i)}\right)^2 \tag{3.13}$$

$$F'''(t_0,\ell) = \sum_{i=0}^{T-1} \frac{q'''(a_i)}{q(a_i)} - 3\left(\frac{q'(a_i)q''(a_i)}{q(a_i)^2}\right) + 2\left(\frac{q'(a_i)}{q(a_i)}\right)^3. \tag{3.14}$$

To prove that $\mathbf{E}_D(|F''(t_0,\ell)|) = O(\sqrt{T})$, we use the fact that the random variable $F''(t_0,\ell)$ is a sum of two random variables $\sum_{i=0}^{T-1} \frac{q''(a_i)}{q(a_i)}$ and $-\sum_{i=0}^{T-1}\left(\frac{q'(a_i)}{q(a_i)}\right)^2$. We

bound the expected absolute value of each of these two terms separately. For the second term, we use the fact that $|q'(a_i)/q(a_i)| = O(h_{a_i})$, which is property 3 from Lemma 3.6. Thus

$$\mathbf{E}\left(\left|\sum_{i=0}^{T-1}\left(\frac{q'(a_i)}{q(a_i)}\right)^2\right|\right) = O\left(\sum_{a\in\mathbb{T}} p_D(a)h_a^2\right),$$

and the right side is $O(\sqrt{T})$ using Lemma 3.8 and our hypothesis that $\mathbf{E}_D R_D \leq \sqrt{T}$. To bound the first term, $\sum_{i=0}^{T-1}\frac{q''(a_i)}{q(a_i)}$, we start by observing that, conditional on the value of a_i, the random variable $\frac{q''(a_i)}{q(a_i)}$ has mean zero and variance $O(1)$. The bound on the conditional variance follows from property 4 in Lemma 3.6. The mean-zero assertion follows from the computation

$$\mathbf{E}_D\left(\frac{q''(a_i)}{q(a_i)}\middle\| a_i\right) = D(x_{a_i})\left(\frac{D''(x_{a_i})}{D(x_{a_i})}\right) + (1-D(x_{a_i}))\left(\frac{-D''(x_{a_i})}{1-D(x_{a_i})}\right) = 0.$$

This means that the random variables $q''(a_i)/q(a_i)$ form a martingale difference sequence, hence

$$\mathbf{E}_D\left[\left(\sum_{i=0}^{T-1}\frac{q''(a_i)}{q(a_i)}\right)^2\right] = \sum_{i=0}^{T-1}\mathbf{E}_D\left[\left(\frac{q''(a_i)}{q(a_i)}\right)^2\right] = O(T).$$

The bound $\mathbf{E}_D\left(\left|\sum_{i=0}^{T-1}\frac{q''(a_i)}{q(a_i)}\right|\right) = O(\sqrt{T})$ follows using the Cauchy-Schwarz Inequality, as before.

We turn now to proving that $\mathbf{E}_D(|F'''(t_0,\ell)|) = O(n^{3/4})$. As before, the first step is to use (3.14) to express $F'''(t_0,\ell)$ as a sum of three terms

$$\begin{aligned} X &= \sum_{t=0}^{T-1}\frac{q'''(a_i)}{q(a_i)} \\ Y &= -3\sum_{t=0}^{T-1}\frac{q'(a_i)q''(a_i)}{q(a_i)} \\ Z &= 2\sum_{t=0}^{T-1}\left(\frac{q'(a_i)}{q(a_i)}\right)^2, \end{aligned}$$

and then to bound the expected absolute value of each of these terms separately. Exactly as above, one proves that the random variables $q'''(a_i)/q(a_i)$ form a martingale difference sequence and have bounded variance, and consequently $\mathbf{E}_D(|X|) = O(\sqrt{T})$. Recalling that $|q'(a_i)/q(a_i)| = O(h_{a_i})$ and $|q''(a_i)/q(a_i)| = O(1)$ (properties 3 and 4

from Lemma 3.6, respectively) we find that

$$\begin{aligned}
\frac{1}{3}\mathbf{E}_D\left(|Y|\right) &\leq \mathbf{E}_D\left(\sum_{i=0}^{T-1}\left|\frac{q'(a_i)q''(a_i)}{q(a_i)}\right|\right)\\
&\leq \mathbf{E}_D\left(\sum_{i=0}^{T-1}\left(\frac{q'(a_i)}{q(a_i)}\right)^2\right)^{1/2}\mathbf{E}_D\left(\sum_{i=0}^{T-1}\left(\frac{q''(a_i)}{q(a_i)}\right)^2\right)^{1/2}\\
&= \mathbf{E}_D\left(\sum_{i=0}^{T-1}O(h_{a_i}^2)\right)^{1/2}\mathbf{E}_D\left(\sum_{i=0}^{T-1}O(1)\right)^{1/2}\\
&= \mathbf{E}_D\left(\sum_{a\in\mathbb{T}}p_D(a)h_a^2\right)^{1/2}\cdot O(\sqrt{T})\\
&= O(T^{3/4}),
\end{aligned}$$

where the last line follows from Lemma 3.8. Finally, we have

$$\begin{aligned}
\frac{1}{2}\mathbf{E}_D(|Z|) &\leq \mathbf{E}_D\left(\sum_{i=0}^{T-1}\left|\frac{q'(a_i)}{q(a_i)}\right|^3\right)\\
&= \mathbf{E}_D\left(\sum_{i=0}^{T-1}O(h_{a_i}^3)\right)\\
&= \mathbf{E}_D\left(\sum_{a\in\mathbb{T}}p_D(a)h_a^3\right)\\
&\leq \mathbf{E}_D\left(\sum_{a\in\mathbb{T}}p_D(a)h_a^2\right)\\
&= O(\sqrt{T}).
\end{aligned}$$

Combining the estimates for $\mathbf{E}_D(|X|), \mathbf{E}_D(|Y|), \mathbf{E}_D(|Z|)$, we obtain the bound

$$\mathbf{E}_D(|F'''(t_0,\ell)|) = O(T^{3/4})$$

as desired.

Finally, to prove $|F''''(t_1,\ell)| = O(T)$, we use the formula

$$\begin{aligned}
F''''(t_1,\ell) &= \sum_{i=0}^{T-1}\frac{q''''(a_i)}{q(a_i)} - 4\left(\frac{q'(a_i)}{q(a_i)}\right)\left(\frac{q'''(a_i)}{q(a_i)}\right) - 3\left(\frac{q''(a_i)}{q(a_i)}\right)^2 +\\
&\qquad 12\left(\frac{q'(a_i)}{q(a_i)}\right)^2\left(\frac{q''(a_i)}{q(a_i)}\right) - 6\left(\frac{q'(a_i)}{q(a_i)}\right)^4. \qquad (3.15)
\end{aligned}$$

Each of the random variables $q^{(k)}(a_i)/q(a_i)$ for $k = 1, 2, 3, 4$ is $O(1)$, hence each summand on the right side of (3.15) is $O(1)$. Summing all n terms, we obtain $|F''''(t_1, \ell)| = O(n)$ as desired. $\square$

Chapter 4

Online optimization in one-parameter spaces

A generalized bandit problem whose strategy set is $[0, 1]$ is called a *continuum-armed bandit problem* [2]. The online pricing problems considered in the preceding chapter were continuum-armed bandit problems with a cost function class Γ whose elements were all functions of the form

$$c(x) = \begin{cases} x & \text{if } x \leq v \\ 0 & \text{otherwise} \end{cases}$$

for $v \in [0, 1]$. This chapter further pursues the study of continuum-armed bandit problems, deriving nearly matching upper and lower bounds on the regret of the optimal algorithms against i.i.d. adversaries as well as adaptive adversaries, when the cost function class Γ is a set of uniformly locally Lipschitz functions on $\mathcal{S}$. (See Definition 4.1 below.) The upper bounds are derived using a straightforward reduction to the finite-armed bandit algorithms presented in Chapter 2. The lower bounds are also inspired by lower bounds for the finite-armed bandit problem (e.g. [4]) but require more sophisticated analytical machinery.

One of the surprising consequences of this analysis is that the optimal regret bounds against an i.i.d. adversary are *identical* (up to a factor no greater than $o(\log T)$, and possibly up to a constant factor) with the optimal regret bounds against an adaptive adversary. This contrasts sharply with the K-armed bandit problem, in which the optimal algorithms have regret $\theta(\log T)$ against i.i.d. adversaries and $\theta(\sqrt{T})$ against adaptive adversaries. This qualitative difference between the finite-armed and continuum-armed bandit problems may be explained conceptually as follows. In the finite-armed bandit problem, the instance which establishes the $\Omega(\sqrt{T})$ lower bound for regret against adaptive adversaries is actually based on an i.i.d. adversary whose distribution depends on T. In other words, the power of adaptive adversaries in the

finite-armed case comes not from their adaptivity or their nonstationarity, but only from their foreknowledge of T. In the continuum-armed case, we can construct an i.i.d. adversary based on a cost function distribution which "embeds" the worst-case distribution for the K-armed bandit problem (for progressively larger values of K and T) into the continuous strategy space at progressively finer distance scales.

4.1 Terminology and Conventions

Throughout this chapter, $\mathcal{S}$ denotes the set $[0, 1]$.

Definition 4.1. A function f is *uniformly locally Lipschitz* with constant L ($0 \leq L < \infty$), exponent α ($0 < \alpha \leq 1$), and restriction δ ($\delta > 0$) if it is the case that for all $u, u' \in \mathcal{S}$ with $\|u - u'\| \leq \delta$,

$$|f(u) - f(u')| \leq L|u - u'|^{\alpha}.$$

The class of all such functions f will be denoted by $ulL(\alpha, L, \delta)$.

We will consider two sets of adversaries. The set $\mathcal{A}_{\text{adpt}}$ is the set of adaptive adversaries for $(\mathcal{S}, ulL(\alpha, L, \delta))$, for some specified values of α, L, δ which are known to the algorithm designer. The set $\mathcal{A}_{\text{iid}}$ is the set of i.i.d. adversaries defined as follows: an i.i.d. adversary ADV governed by a distribution P on functions $c : \mathcal{S} \to \mathbb{R}$ is in $\mathcal{A}_{\text{iid}}$ if:

- P is a (ζ, s_0)-bounded distribution.
- The function $\bar{c}(x) = \mathbf{E}_P[c(x)]$ is in $ulL(\alpha, L, \delta)$. Here $\mathbf{E}_P[\cdot]$ denotes the expectation operator defined by the distribution P.

Note that neither of the sets $\mathcal{A}_{\text{adpt}}, \mathcal{A}_{\text{iid}}$ is contained in the other: our i.i.d. adversaries (unlike our adaptive adversaries) are not required to choose continuous cost functions nor are these functions required to take values between 0 and 1.

4.2 Continuum-armed bandit algorithms

There is a trivial reduction from the continuum-armed bandit problem to the K-armed bandit problem: one limits one's attention to the strategy set

$$\mathcal{S}_K = \{1/K, 2/K, \ldots, (K-1)/K, 1\} \subset \mathcal{S}$$

and runs a K-armed bandit algorithm with strategy set $\mathcal{S}_K$. In this section we analyze algorithms based on this reduction.

Theorem 4.1. *Let $\mathcal{A}$ denote either of the adversary sets $\mathcal{A}_{\text{adpt}}$ or $\mathcal{A}_{\text{iid}}$. There exists an algorithm* **CAB** *satisfying*

$$R(\mathbf{CAB}, \mathcal{A}; T) = O(T^{\frac{\alpha+1}{2\alpha+1}} \log^{\frac{\alpha}{2\alpha+1}}(T)).$$

Proof. Let $K = \left\lceil (T/\log T)^{\frac{1}{2\alpha+1}} \right\rceil$, and let $\mathcal{S}_K = \{1/K, 2/K, \ldots, 1\}$. Let **MAB** denote a multi-armed bandit algorithm with strategy set $\mathcal{S}_K$; **MAB** is either the algorithm UCB1 if $\mathcal{A} = \mathcal{A}_{\text{iid}}$, or it is the algorithm Exp3 if $\mathcal{A} = \mathcal{A}_{\text{adpt}}$.

For any adversary $\mathsf{ADV} \in \mathcal{A}$, there is a well-defined restriction $\mathsf{ADV}|_{\mathcal{S}_K}$ whose cost functions c_t are the cost functions selected by ADV, restricted to $\mathcal{S}_K$. We claim that $R(\mathbf{MAB}, \mathsf{ADV}|_{\mathcal{S}_K}; T) = O(\sqrt{TK \log K})$. If $\mathcal{A} = \mathcal{A}_{\text{adpt}}$ this follows directly from Theorem 2.12. If $\mathcal{A} = \mathcal{A}_{\text{iid}}$ then we have $R(\mathbf{MAB}, \mathsf{ADV}|_{\mathcal{S}_K}; T) = O(K + \sqrt{TK \log K})$ by Corollary 2.9. Also $R(\mathbf{MAB}, \mathsf{ADV}|_{\mathcal{S}_K}; T) = O(T)$ because $|\bar{c}(x) - \bar{c}(y)| = O(1)$ for $\mathsf{ADV} \in \mathcal{A}_{\text{iid}}$ and $x, y \in \mathcal{S}$. The bound $R(\mathbf{MAB}, \mathsf{ADV}|_{\mathcal{S}_K}; T) = O(\sqrt{TK \log K})$ now follows from the fact that

$$\min\{T, K + \sqrt{TK \log K}\} < 2\sqrt{TK \log K}.$$

Let $\bar{c}$ denote the function $\bar{c}(x) = \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x)\right]$. (If $\mathsf{ADV} \in \mathcal{A}_{\text{iid}}$ this coincides with the usage of $\bar{c}$ defined earlier.) We have $\bar{c} \in ulL(\alpha, L, \delta)$. For any $x \in \mathcal{S}$, there exists a strategy $y \in \mathcal{S}_K$ such that $|x - y| \leq 1/K$, and consequently,

$$\bar{c}(y) - \bar{c}(x) \leq LK^{-\alpha} \leq LT^{-\frac{\alpha}{2\alpha+1}} \log^{\frac{\alpha}{2\alpha+1}}(T).$$

We have

$$\begin{aligned}
\mathbf{E}\left[\sum_{t=1}^{T} c_t(x_t) - c_t(x)\right] &\leq \mathbf{E}\left[\sum_{t=1}^{T} c_t(x_t) - c_t(y)\right] + T(\bar{c}(y) - \bar{c}(x)) \\
&\leq R(\mathbf{MAB}, \mathsf{ADV}|_{\mathcal{S}_K}; T) + LT^{1-\frac{\alpha}{2\alpha+1}} \log^{\frac{\alpha}{2\alpha+1}}(T) \\
&= O\left(\sqrt{TK \log K} + T^{\frac{\alpha+1}{2\alpha+1}} \log^{\frac{\alpha}{2\alpha+1}}(T)\right).
\end{aligned}$$

The theorem now follows from the fact that

$$\sqrt{TK \log K} = O\left(\sqrt{T \cdot \left(\frac{T}{\log T}\right)^{\frac{1}{2\alpha+1}} \cdot \log T}\right) = O\left(T^{\frac{\alpha+1}{2\alpha+1}} \log^{\frac{\alpha}{2\alpha+1}}(T)\right).$$

□

4.3 Lower bounds for the one-parameter case

There are many reasons to expect that Algorithm **CAB** is an inefficient algorithm for the continuum-armed bandit problem. Chief among these is that fact that it treats the strategies $\{1/K, 2/K, \ldots, 1\}$ as an unordered set, ignoring the fact that experiments which sample the cost of one strategy j/K are (at least weakly) predictive of the costs of nearby strategies. In this section we prove that, contrary to this intuition, **CAB** is in fact quite close to the optimal algorithm. Specifically, in the regret bound of Theorem 4.1, the exponent $\frac{\alpha+1}{2\alpha+1}$ is the best possible: for any $\beta < \frac{\alpha+1}{2\alpha+1}$, no algorithm can achieve regret $O(n^\beta)$. This lower bound applies to both $\mathcal{A}_{\text{adpt}}$ and $\mathcal{A}_{\text{iid}}$.

The lower bound relies on a function $f : [0,1] \to [0,1]$ defined as the sum of a nested family of "bump functions." The details of the construction are specified as follows.

Construction 4.1. Let B be a C^∞ bump function defined on the real line, satisfying $0 \le B(x) \le 1$ for all x, $B(x) = 0$ if $x \le 0$ or $x \ge 1$, and $B(x) = 1$ if $x \in [1/3, 2/3]$. (For a construction of such functions, see e.g. [22], Theorem II.5.1.) For an interval $[a,b]$, let $B_{[a,b]}$ denote the bump function $B(\frac{x-a}{b-a})$, i.e. the function B rescaled and shifted so that its support is $[a,b]$ instead of $[0,1]$. Define a random nested sequence of intervals $[0,1] = [a_0, b_0] \supset [a_1, b_1] \supset \ldots$ as follows: for $k > 0$, the middle third of $[a_{k-1}, b_{k-1}]$ is subdivided into intervals of width

$$w_k = 3^{-k!},$$

and $[a_k, b_k]$ is one of these subintervals chosen uniformly at random. Now let

$$f(x) = \frac{1}{3} + \left(\frac{3^\alpha - 1}{3}\right) \sum_{k=1}^{\infty} w_k^\alpha B_{[a_k,b_k]}(x).$$

Finally, define a probability distribution on functions $c : [0,1] \to [0,1]$ by the following rule: sample λ uniformly at random from the open interval $(0,1)$ and put $c(x) = \lambda^{f(x)}$. Observe that the expected value of $c(x)$ is

$$\bar{c}(x) = \int_0^1 \lambda^{f(x)}\, dx = \frac{1}{1 + f(x)}.$$

The relevant technical properties of this construction are summarized in the following lemma.

Lemma 4.2. *Let $\{u^*\} = \bigcap_{k=1}^{\infty} [a_k, b_k]$. The function $f(x)$ belongs to $ulL(\alpha, L, \delta)$ for some constants L, δ, it takes values in $[1/3, 2/3]$, and it is uniquely maximized at u^*. For each $\lambda \in (0,1)$, the function $c(x) = \lambda^{f(x)}$ belongs to $ulL(\alpha, L, \delta)$ for some constants L, δ, and is uniquely minimized at u^*. The same two properties are satisfied by the function $\bar{c}(x) = (1 + f(x))^{-1}$.*

Proof. Put $s_k(x) = \left(3^{\alpha-1} - \frac{1}{3}\right) w_k^\alpha B_{[a_k,b_k]}(x)$, so that $f(x) = \frac{1}{3} + \sum_k s_k(x)$. Let

$$\beta = \sup_{x,y\in[0,1]} \frac{|B(x) - B(y)|}{|x-y|^\alpha}.$$

Note that $\beta < \infty$ because B is differentiable and $[0,1]$ is compact. We claim that for any interval $[a,b] \subseteq [0,1]$ of width $w = b - a$, the function $w^\alpha B_{[a,b]}$ is in $ulL(\alpha, \beta, 1)$. This follows from a direct calculation:

$$\frac{\left|w^\alpha B_{[a,b]}(x) - w^\alpha B_{[a,b]}(y)\right|}{|x-y|^\alpha} \;=\; \frac{\left|B\left(\frac{x-a}{w}\right) - B\left(\frac{y-a}{w}\right)\right|}{\left|\left(\frac{x-a}{w}\right) - \left(\frac{y-a}{w}\right)\right|^\alpha} \;\le\; \beta.$$

For any $x \in [0,1]$, let $n(x) = \sup\{k : x \in [a_k, b_k]\}$. Given any two points $x, y \in [0,1]$, if $n(x) = n(y) = n$, then $B_{[a_k,b_k]}(x) = B_{[a_k,b_k]}(y)$ for all $k \neq n$. (Both sides are equal to 1 when $k < n$ and 0 when $k > n$.) Consequently,

$$\frac{|f(x) - f(y)|}{|x-y|^\alpha} = \frac{|s_n(x) - s_n(y)|}{|x-y|^\alpha} \le \left(\frac{3^\alpha - 1}{3}\right)\beta$$

since $B_{[a_n,b_n]} \in ulL(\alpha, \beta, 1)$. Now suppose without loss of generality that $n(x) > n(y) = n - 1$. Then $s_k(y) = 0$ for all $k > n$, so

$$\begin{aligned}\frac{|f(x) - f(y)|}{|x-y|^\alpha} &\le \frac{|s_{n-1}(x) - s_{n-1}(y)|}{|x-y|^\alpha} + \frac{|s_n(x) - s_n(y)|}{|x-y|^\alpha} + \sum_{k>n} \frac{s_k(x)}{|x-y|^\alpha} \\ &\le \left(\frac{3^\alpha - 1}{3}\right)\left[2\beta + \sum_{n<k\le n(x)} \frac{w_k^\alpha}{|x-y|^\alpha}\right]. \qquad (4.1)\end{aligned}$$

If $n(x) = n$ then the right side of (4.1) is $\frac{2(3^\alpha-1)\beta}{3}$, so we are done. If $n(x) > n$ then x belongs to the middle third of $[a_n, b_n]$ while $y \notin [a_n, b_n]$, which implies that $|x - y| \ge w_n/3$. Therefore

$$\begin{aligned}\sum_{n<k\le n(x)} \frac{w_k^\alpha}{|x-y|^\alpha} &\le 3^\alpha \sum_{n<k<\infty} \left(\frac{w_k}{w_n}\right)^\alpha \\ &< 3^\alpha \sum_{k=1}^{\infty} 3^{-\alpha} = \frac{3^\alpha}{3^\alpha - 1},\end{aligned}$$

which completes the proof that $f \in ulL(\alpha, L, \delta)$ with $L = \frac{2(3^\alpha-1)\beta+3^\alpha}{3}$ and $\delta = 1$.

The verification that f takes values in $[1/3, 2/3]$ and is uniquely maximized at u^* is routine. The first derivatives of the functions $g(y) = \lambda^y$ and $h(y) = (1 + f(y))^{-1}$ are uniformly bounded above, for $y \in [1/3, 2/3]$, by a constant independent of λ; hence the Lipschitz regularity of $c(x) = g(f(x))$ and of $\bar{c}(x) = h(f(x))$ follow from the

Lipschitz regularity of f itself. The functions $c(x)$ and $\bar{c}(x)$ are uniquely minimized at $x = u^*$ because f is uniquely maximized at u^* and g, h are decreasing functions of y. $\square$

The proof of our lower bound theorem will rely heavily on the properties of KL-divergence proven in Section 2.7. Before embarking on the proof itself, which is quite technically detailed, we offer the following proof sketch. In Lemma 4.3 we will demonstrate that for any $u, u' \in [a_{k-1}, b_{k-1}]$, the KL-divergence $KL(c(u)\|c(u'))$ between the cost distributions at u and u' is $O(w_k^{2\alpha})$, and that it is equal to zero unless at least one of u, u' lies in $[a_k, b_k]$. This means, roughly speaking, that the algorithm must sample strategies in $[a_k, b_k]$ at least $w_k^{-2\alpha}$ times before being able to identify the interval $[a_k, b_k]$ with constant probability. But $[a_k, b_k]$ could be any one of $w_{k-1}/3w_k$ possible subintervals, and we don't have enough time to play $w_k^{-2\alpha}$ trials in more than a small constant fraction of these subintervals before reaching time T_k. Therefore, with constant probability, a constant fraction of the strategies chosen up to time T_k are not located in $[a_k, b_k]$, and each of them contributes $\Omega(w_k^{\alpha})$ to the regret. This means the expected regret at time T_k is $\Omega(T_k w_k^{\alpha})$. From this, we obtain the stated lower bound using the fact that

$$T_k w_k^{\alpha} = T_k^{\frac{\alpha+1}{2\alpha+1} - o(1)}.$$

Lemma 4.3. *For a given $y \in (0,1)$, let $G : [0,1] \to [0,1]$ denote the function $G(\lambda) = \lambda^y$, let m denote Lebesgue measure on $[0,1]$, and let σ_y denote the probability measure $G_* m$, i.e. the distribution of the random variable λ^y when λ is sampled from the uniform distribution on $[0,1]$. For all $x \in (0,1)$ and all positive $\varepsilon < 1/2$,*

$$KL(\sigma_{(1-\varepsilon)x} \,\|\, \sigma_x) = O(\varepsilon^2).$$

Proof. Let $y = \frac{1}{(1-\varepsilon)x}$ and $z = \frac{1}{x}$. For $0 \le r \le 1$ we have

$$\sigma_{(1-\varepsilon)x}([0,r]) = m(\{q \,:\, q^{(1-\varepsilon)x} \in [0,r]\}) = m([0, r^y]) = r^y,$$

hence the probability measure $\sigma_{(1-\varepsilon)x}$ has density function yr^{y-1}. Similarly σ_x has density function zr^{z-1}. Thus the Radon-Nikodym derivative of σ_x with respect to $\sigma_{(1-\varepsilon)x}$ is given by

$$\rho = \frac{zr^{z-1}}{yr^{y-1}} = \frac{z}{y}\, r^{z-y}.$$

By Theorem 2.19,

$$\begin{aligned} KL(\sigma_{(1-\varepsilon)x} \,\|\, \sigma_x) &= -\int_0^1 \log(\rho) d\sigma_{(1-\varepsilon)x} \\ &= -\int_0^1 \left[\log\left(\frac{z}{y}\right) + (z-y)\log r\right] yr^{y-1} dr. \end{aligned}$$

Substituting $s = r^y$, so that $ds = yr^{y-1}dr$ and $\log s = y \log r$, the integral above is transformed into

$$\begin{aligned} -\int_0^1 \left[\log\left(\frac{z}{y}\right) + \frac{z-y}{y}\log s\right] ds &= -\log\left(\frac{z}{y}\right) - \frac{z-y}{y}\int_0^1 \log s\, ds \\ &= -\log\left(\frac{z}{y}\right) + \frac{z}{y} - 1 \\ &= -\log(1-\varepsilon) - \varepsilon. \end{aligned}$$

The lemma now follows from the power series expansion

$$-\log(1-\varepsilon) - \varepsilon = \sum_{n=2}^{\infty} \frac{\varepsilon^n}{n}.$$

□

Theorem 4.4. *For any randomized multi-armed bandit algorithm* ALG*, there exists an adversary* $\mathsf{ADV} \in \mathcal{A}_{\text{iid}} \cap \mathcal{A}_{\text{adpt}}$ *such that for all* $\beta < \frac{\alpha+1}{2\alpha+1}$*, the algorithm's regret satisfies*

$$\limsup_{T\to\infty} \frac{R(\mathsf{ALG}, \mathsf{ADV}; T)}{T^\beta} = \infty.$$

Proof. We will prove that there exists a nested sequence of intervals $[0,1] = [a_0, b_0] \supset [a_1, b_1] \supset \ldots$, defining a probability distribution on cost functions $c(x)$ according to Construction 4.1, such that the i.i.d. adversary ADV specified by this distribution satisfies $R(\mathsf{ALG}, \mathsf{ADV}; T_k)/T_k^\beta \to \infty$ when the sequence $T_1, T_2, T_3, \ldots$ is defined by $T_k = \lceil C(w_{k-1}/3w_k)w_k^{-2\alpha}\rceil$ for a suitable constant C. Parts of the proof are very similar to the proof of Theorem 2.25. We have deliberately copied the notation — and, in some cases, the actual wording — from the proof of Theorem 2.25 to highlight the similarity in these places.

For a set $S \subseteq \mathcal{S}$ let χ_S denote the characteristic function

$$\chi_S(x) = \begin{cases} 1 & \text{if } x \in S \\ 0 & \text{if } x \notin S \end{cases}.$$

and for an adversary ADV, let

$$Q(\mathsf{ALG}, \mathsf{ADV}; T, S) = \mathbf{E}\left[\sum_{t=1}^{T} \chi_S(x_t)\right]$$

denote the expected number of times the algorithm ALG selects a strategy x_t in S, among the first T trials, when playing against ADV. (Here the random variables

$x_1, x_2, \ldots$ are defined as in Definition 1.7.) Suppose an i.i.d. adversary ADV is defined according to Construction 4.1 using a nested sequence of intervals $[a_0, b_0] \supset [a_1, b_1] \supset \ldots$. Then for all $x \notin (a_k, b_k)$ we have $\bar{c}(x) - \bar{c}(u^*) = \Omega(w_k^\alpha)$. It follows that

$$\begin{aligned} R(\mathsf{ALG}, \mathsf{ADV}; T_k) &= \mathbf{E}\left[\sum_{t=1}^{T_k} c_t(x_t) - c_t(u^*)\right] \\ &= \mathbf{E}\left[\sum_{t=1}^{T_k} \bar{c}(x_t) - \bar{c}(u^*)\right] \\ &= \Omega\left(w_k^\alpha\, Q(\mathsf{ALG}, \mathsf{ADV}; T_k, \mathcal{S} \setminus (a_k, b_k))\right) \\ &= \Omega\left(w_k^\alpha\, (T_k - Q(\mathsf{ALG}, \mathsf{ADV}; T_k, (a_k, b_k)))\right). \end{aligned}$$

Assume for now that we have an adversary such that

$$Q(\mathsf{ALG}, \mathsf{ADV}; T_k, (a_k, b_k)) \leq T_k \left(\frac{1}{2} + o(1)\right) \tag{4.2}$$

for all $k \geq 1$. Then

$$R(\mathsf{ALG}, \mathsf{ADV}; T_k) = \Omega(w_k^\alpha\, T_k) = \Omega\left(w_{k-1} w_k^{-1-\alpha}\right) = \Omega\left(w_k^{1/k-1-\alpha}\right).$$

In light of the fact that $T_k = C w_k^{1/k-1-2\alpha}$ this implies

$$R(\mathsf{ALG}, \mathsf{ADV}; T_k) = \Omega\left(T_k^{\frac{\alpha+1}{2\alpha+1}-o(1)}\right) = \omega(T_k^\beta)$$

as desired. Thus the theorem will follow if we construct an adversary satisfying (4.2) for all sufficiently large k (say, for $k \geq k_0$). We will accomplish this inductively, using the following induction hypothesis on n: there exist intervals $[0,1] = [a_0, b_0] \supset \ldots \supset [a_n, b_n]$ (where $[a_j, b_j]$ is an interval of width w_j in the middle third of $[a_{j-1}, b_{j-1}]$ for $j = 1, \ldots, n$) such that whenever one applies Construction 4.1 using an infinite nested sequence of intervals which extends the sequence $[a_0, b_0] \supset \ldots \supset [a_n, b_n]$, the resulting adversary ADV satisfies (4.2) for $k_0 \leq k \leq n$. This induction hypothesis is vacuously true when $n = 0$.

Assume now that the induction hypothesis is true for $n-1$, and let $[a_0, b_0] \supset \ldots \supset [a_{n-1}, b_{n-1}]$ be a sequence of intervals satisfying the induction hypothesis. Let

$$\begin{aligned} s_k &= \frac{3^\alpha - 1}{3} w_k^\alpha B_{[a_k, b_k]} \\ f_n &= \frac{1}{3} + \sum_{k=1}^{n-1} s_k. \end{aligned}$$

Let ADV_n be the i.i.d. adversary determined by a probability distribution P_n on the set $\Gamma = [0,1]^{\mathcal{S}}$ which is defined as follows: to sample a random $c \in \Gamma$ from distribution P_n, one samples λ from the uniform distribution on the open interval $(0,1)$ and puts $c = \lambda^{f_n(x)}$. Partition the middle third of the interval $[a_{n-1}, b_{n-1}]$ into $N = \frac{w_{n-1}}{3w_n}$ subintervals, and let $I_1, \ldots, I_N$ denote the interiors of these subintervals. Since $I_1, \ldots, I_N$ are disjoint, we have

$$\sum_{i=1}^{N} Q(\mathsf{ALG}, \mathsf{ADV}_n; T_n, I_i) \leq T_n$$

so there must be at least one value of i such that $Q(\mathsf{ALG}, \mathsf{ADV}_n; T_n, I_i) \leq T_n/N$. Let $[a_n, b_n]$ be the closure of I_i. Our objective is to confirm the induction hypothesis for the sequence $[a_0, b_0] \supset \ldots \supset [a_n, b_n]$. To do so, it suffices to consider an arbitrary extension of this sequence to an infinite sequence $[a_0, b_0] \supset [a_1, b_1] \supset \ldots$ with $b_j - a_j = w_j$ for all j, apply Construction 4.1, and prove that the resulting adversary ADV satisfies (4.2) with $k = n$. In other words, we wish to prove that $Q(\mathsf{ALG}, \mathsf{ADV}_n; T_n, I_i) \leq T_n/N$ implies $Q(\mathsf{ALG}, \mathsf{ADV}; T_n, I_i) \leq T_n/2$ provided n is sufficiently large.

The transcripts of play for ALG against $\mathsf{ADV}_n, \mathsf{ADV}$ define two different probability distributions μ, ν on sequences $(x_1, y_1, \ldots, x_{T_n}, y_{T_n})$, where $x_1, \ldots, x_{T_n}$ denote the algorithm's choices in trials $1, \ldots, T_n$ and $y_1, \ldots, y_{T_n}$ denote the feedback values received. Recalling the definition of the L_1-distance between two measures from Section 2.7, i.e.

$$\|\nu - \mu\|_1 = 2 \sup_A (\nu(A) - \mu(A)),$$

we see that

$$\begin{aligned} Q(\mathsf{ALG}, \mathsf{ADV}; I_i, T_n) - Q(\mathsf{ALG}, \mathsf{ADV}_n; I_i, T_n) &= \sum_{t=1}^{T_n} \nu(\{x_t \in I_i\}) - \mu(\{x_t \in I_i\}) \\ &\leq T_n \left(\frac{\|\nu - \mu\|_1}{2} \right). \end{aligned}$$

From Theorem 2.20 we know that

$$\frac{\|\nu - \mu\|_1}{2} \leq \sqrt{\frac{KL(\mu \,\|\, \nu)}{2}}$$

so we are left with proving $KL(\mu \,\|\, \nu) \leq \frac{1}{2}$.

Using the chain rule for Kullback-Leibler divergence (Theorem 2.24) we have

$$KL(\mu \,\|\, \nu) = \sum_{i=0}^{2T_n - 1} \int_{\mathbb{R}^i} KL(\mu_{i+1} \,\|\, \nu_{i+1}) dp_{1..i*}\mu.$$

Now if $i = 2j$ is an even number, $\mu_{i+1}(x_1, y_1, \ldots, x_j, y_j) = \nu_{i+1}(x_1, y_1, \ldots, x_j, y_j)$ because both distributions are equal to the conditional distribution of the algorithm's choice x_{j+1}, conditioned on the strategies and feedbacks revealed in trials $1, \ldots, j$. If $i = 2j - 1$ is an odd number, then $\mu_{i+1}(x_1, y_1, \ldots, x_{j-1}, y_{j-1}, x_j)$ is the distribution of $y_j = \lambda^{f_n(x_j)}$, so $\mu_{i+1}(x_1, y_1, \ldots, x_j) = \sigma_{f_n(x_j)}$. Similarly $\nu_{i+1}(x_1, y_1, \ldots, x_j) = \sigma_{f(x_j)}$. Therefore, letting $\mathbf{E}_\mu$ denote the expectation operator defined by the measure μ,

$$KL(\mu \,\|\, \nu) = \sum_{j=1}^{T_n} \mathbf{E}_\mu[KL(\sigma_{f_n(x_j)} \,\|\, \sigma_{f(x_j)})].$$

If $x_j \notin I_i = (a_n, b_n)$ then $f_n(x_j) = f(x_j)$ and $KL(\sigma_{f_n(x_j)} \,\|\, \sigma_{f(x_j)}) = 0$. If $x_j \in I_i$ then

$$1/3 \le f_n(x_j) \le f(x_j) \le f_n(x_j) + O(w_n^\alpha) \le 2/3$$

which proves that $f_n(x_j)/f(x_j) = 1 - O(w_n^\alpha)$. By Lemma 4.3, this implies

$$KL(\sigma_{f_n(x_j)} \,\|\, \sigma_{f(x_j)}) = O(w_n^{2\alpha}).$$

Hence for some constant $C' < \infty$,

$$\begin{aligned} KL(\mu \,\|\, \nu) &< C' w_n^{2\alpha} \sum_{j=1}^{T_n} \mathbf{E}_\mu[\chi_{I_i}(x_j)] \\ &= C' w_n^{2\alpha} Q(\mathsf{ALG}, \mathsf{ADV}_n; I_i, T_n) \\ &\le C' w_n^{2\alpha} \frac{T_n}{N} \\ &\le C' w_n^{2\alpha} \frac{1 + C(w_{n-1}/w_n) w_n^{-2\alpha}}{w_{n-1}/3w_n} \\ &\le \frac{C' w_n^{2\alpha}}{N} + \frac{CC'}{3}. \end{aligned}$$

Upon choosing a positive constant $C < 1/C'$, we obtain

$$KL(\mu \,\|\, \nu) < (C' w_n^{2\alpha}/N) + 1/3 < 1/2$$

for sufficiently large n, which completes the proof. □

Chapter 5

Online optimization in vector spaces

5.1 Introduction

In the preceding chapter we studied online decision problems with a one-parameter strategy set. In this chapter we develop algorithms for online decision problems with a d-parameter strategy set, i.e. a strategy set $\mathcal{S} \subseteq \mathbb{R}^d$, for $d > 1$. Assuming the cost functions satisfy a Lipschitz condition, it is of course possible to design algorithms using an approach parallel to that taken in the one-parameter case: we choose a finite subset $X \subseteq \mathcal{S}$ such that every point of $\mathcal{S}$ is with distance ε of a point of X (for some sufficiently small ε) and we reduce to a finite-armed bandit problem whose strategy set is X. The problem with this approach, for large d, is that the cardinality of X will generally be exponential in d, leading to algorithms whose convergence time is exponential in d. In fact, this exponential convergence time is inherent: if Γ is a class of cost functions containing all C^∞ functions mapping $\mathcal{S}$ to $[0, 1]$, then any algorithm for the generalized bandit problem on $(\mathcal{S}, \Gamma)$ has exponential convergence time. (This is demonstrated by a simple counterexample in which the cost function is identically equal to 1 in all but one orthant of $\mathbb{R}^d$, takes a value less than 1 somewhere in that orthant, and does not vary over time.) This thesis proposes two ways of surmounting the exponential convergence time inherent in d-dimensional online decision problems with opaque feedback: one may consider more restrictive classes of cost functions, or one may relax the definitions of regret and convergence time. In this chapter we adopt the former approach; the latter approach is studied in Chapter 6.

We will present two algorithms in this chapter which achieve polynomial convergence time against an oblivious adversary. The first assumes $\mathcal{S}$ is a compact subset of $\mathbb{R}^d$ and Γ is the set of linear functions on $\mathcal{S}$ taking values in a bounded interval.

Apart from compactness, we assume no special structure on the set $\mathcal{S}$; we need only assume that the algorithm has access to an oracle for minimizing a linear function on $\mathcal{S}$. Our algorithm may thus be interpreted as a general-purpose reduction from offline to online linear optimization.

The second algorithm in this chapter assumes $\mathcal{S}$ is a compact convex subset of $\mathbb{R}^d$ and that Γ is a set of convex functions on $\mathcal{S}$ taking values in a bounded interval. (It is also necessary to assume that the first and second derivatives of the functions are bounded.)

As an application of the online linear optimization algorithm developed in this chapter, we consider the *online shortest path* problem. In this problem the strategy set $\mathcal{S}$ consists of (not necessarily simple) paths of at most H hops from a sender s to a receiver r in a directed graph $G = (V, E)$. A cost function $c \in \Gamma$ is specified by assigning lengths in $[0, 1]$ to the edges of G. The cost assigned to a path π by such a function is the sum of its edge lengths. We consider here the generalized bandit problem for $(\mathcal{S}, \Gamma)$. The online shortest path problem may be applied to overlay network routing, by interpreting edge costs as link delays. In formulating the problem as a generalized bandit problem, we are assuming that the feedback from each trial is limited to exposing the end-to-end delay from sender to receiver; this models the notion that no feedback is obtained from intermediate routers in the network.

For the online linear optimization problem, a novel idea in our work is to compute a special basis for the vector space spanned by the strategy set. This basis, which is called a *barycentric spanner*, has the property that all other strategies can be expressed as linear combinations *with bounded coefficients* of the basis elements. We prove that barycentric spanners exist whenever the strategy set is compact, and we provide a polynomial-time algorithm to compute a barycentric spanner given access to a linear optimization oracle for $\mathcal{S}$. We further demonstrate the usefulness of barycentric spanners by illustrating how to use them in a simple algorithm for computing approximate closest uniform approximations of functions.

5.2 Online linear optimization

5.2.1 Overview of algorithm

This section presents a randomized algorithm for online linear optimization, in which the strategy set $\mathcal{S}$ is a compact subset of $\mathbb{R}^d$ and the cost functions are linear functions mapping $\mathcal{S}$ to $[-M, M]$ for some predefined constant M. As stated in Section 1.3, the full-feedback version of this problem has been solved by Kalai and Vempala in [44]. We will use their algorithm as a black box (the *K-V black box*), reducing from the

opaque-feedback case to the full-feedback case by dividing the timeline into phases and using each phase to simulate one round of the full-feedback problem. We randomly subdivide the time steps in a phase into a small number of "exploration" steps which are used for explicitly sampling the costs of certain strategies, and a much larger number of "exploitation" steps in which we choose our strategy according to the output of the black box, with the aim of minimizing cost. The feedback to the black box at the end of a phase is an unbiased estimate of the average of the cost vectors in that phase, generated by averaging the data from the sampling steps. (Ideally, we would also use the data from the exploitation steps, since it is wasteful to throw this data away. However, we do not know how to incorporate this data without biasing our estimate of the average cost function. This shortcoming of the analysis partly explains why we are limited, in this chapter, to considering the oblivious adversary model.)

We now address the question of how to plan the sampling steps so as to generate a reasonably accurate and unbiased estimate of the average cost vector in a phase. One's instinct, based on the multi-armed bandit algorithm Exp3 of Section 2.5, might be to try sampling each strategy a small percentage of the time, and to ascribe to each strategy a simulated cost which is the average of the samples. The problem with this approach in our context is that there may be exponentially many, or even infinitely many, strategies to sample. So instead we take advantage of the fact that the cost functions are linear, to sample a small subset $X \subseteq \mathcal{S}$ of the strategies — a basis for the vector space spanned by $\mathcal{S}$ — and extend the simulated cost function from X to $\mathcal{S}$ by linear interpolation. In taking this approach, a subtlety arises which accounts for the main technical contribution of this section. The problem is that the average of the sampled costs at a point of X will generally differ from the true average cost by a small sampling error; if the point set X is badly chosen, these sampling errors will be amplified by an arbitrarily large factor when we extend the simulated cost function to all of $\mathcal{S}$. (See Figure 5-1. In that example, $\mathcal{S}$ is a triangle in $\mathbb{R}^2$. The point set on the left is bad choice for X, since small sampling errors can lead to large errors at the upper left and lower right corners. The point set on the right does not suffer from this problem.)

To avoid this pitfall, we must choose X to be as "well-spaced" inside $\mathcal{S}$ as possible. We formulate this notion of "well-spaced subsets" precisely in Section 5.2.2; such a subset will be called a *barycentric spanner*. Using barycentric spanners, we give a precise description and analysis of the online linear optimization algorithm sketched above. We then illustrate how these techniques may be applied to the online shortest path problem.

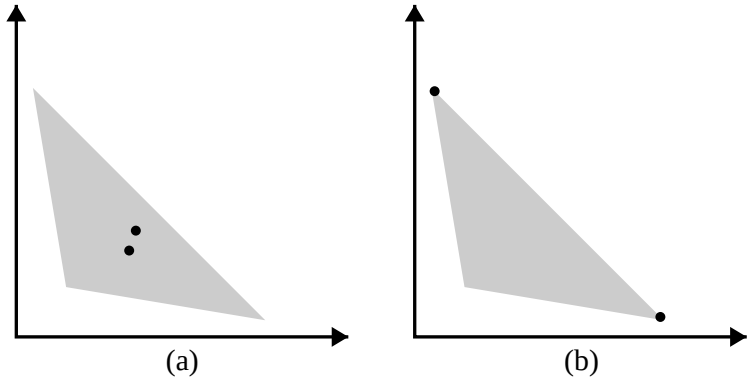

Figure 5-1: (a) A bad sampling set (b) A barycentric spanner.

5.2.2 Barycentric spanners

Definition 5.1. Let V be a vector space over the real numbers, and $\mathcal{S} \subseteq V$ a subset whose linear span is a d-dimensional subspace of V. A set $X = \{x_1, \ldots, x_d\} \subseteq \mathcal{S}$ is a *barycentric spanner* for $\mathcal{S}$ if every $x \in \mathcal{S}$ may be expressed as a linear combination of elements of X using coefficients in $[-1, 1]$. X is a C-approximate barycentric spanner if every $x \in \mathcal{S}$ may be expressed as a linear combination of elements of X using coefficients in $[-C, C]$.

Proposition 5.1. *If $\mathcal{S}$ is a compact subset of V, then $\mathcal{S}$ has a barycentric spanner.*

Proof. Assume without loss of generality that $span(\mathcal{S}) = V = \mathbb{R}^d$. Choose a subset $X = \{x_1, \ldots, x_d\} \subseteq \mathcal{S}$ maximizing $|\det(x_1, \ldots, x_d)|$. (The maximum is attained by at least one subset of $\mathcal{S}$, by compactness.) We claim X is a barycentric spanner of $\mathcal{S}$. For any $x \in \mathcal{S}$, write $x = \sum_{i=1}^{d} a_i x_i$. Then

$$\begin{aligned} |\det(x, x_2, x_3, \ldots, x_d)| &= \left| \det\left(\sum_{i=1}^{d} a_i x_i, x_2, x_3, \ldots, x_d \right) \right| \\ &= \left| \sum_{i=1}^{d} a_i \det\left(x_i, x_2, x_3, \ldots, x_d\right) \right| \\ &= |a_1| \left|\det\left(x_1, \ldots, x_d\right)\right| \end{aligned}$$

from which it follows that $|a_1| \leq 1$, by the maximality of $|\det(x_1, \ldots, x_d)|$. By symmetry, we see that $|a_i| \leq 1$ for all i, and we conclude (since x was arbitrary) that X is a barycentric spanner as claimed. □

Observation 5.1. Given a subset $X = \{x_1, \ldots, x_d\} \subseteq \mathcal{S}$ and an index $i \in \{1, \ldots, d\}$, let X_{-i} denote the $(d-1)$-tuple of vectors $(x_1, x_2, \ldots, x_{i-1}, x_{i+1}, \ldots, x_d)$. The proof of Proposition 5.1 actually establishes the following stronger fact. If $X = \{x_1, \ldots, x_d\}$ is a subset of $\mathcal{S}$ with the property that for any x in $\mathcal{S}$ and any $i \in \{1, \ldots, d\}$,

$$|\det(x, X_{-i})| \leq C|\det(x_1, x_2, \ldots, x_d)|,$$

then X is a C-approximate barycentric spanner for $\mathcal{S}$.

A consequence of Proposition 5.1 is the following matrix factorization theorem, which was independently proven by Barnett and Srebro [15]. For a matrix $M = (m_{ij})$, let $\|M\|_\infty = \max_{i,j} |m_{ij}|$.

Proposition 5.2. *If M is an m-by-n matrix satisfying $\|M\|_\infty \leq 1$ and $\operatorname{rank}(M) = k$, then we may write M as a product $M = AB$ where A, B are m-by-k and k-by-n matrices, respectively, satisfying $\|A\|_\infty \leq 1$ and $\|B\|_\infty \leq 1$.*

Interestingly, Barnett and Srebro's proof of Proposition 5.2 is non-constructive, making use of Kakutani's fixed point theorem, just as our proof of Proposition 5.1 is non-constructive, relying on minimizing the function $|\det(x_1, \ldots, x_n)|$ on the compact set $\mathcal{S}^d$. In fact, it is an open question whether barycentric spanners can be computed in polynomial time, given an oracle for optimizing linear functions over $\mathcal{S}$. However, the following proposition shows that C-approximate barycentric spanners (for any $C > 1$) may be computed in polynomial time given access to such an oracle.

Proposition 5.3. *Suppose $\mathcal{S} \subseteq \mathbb{R}^d$ is a compact set not contained in any proper linear subspace. Given an oracle for optimizing linear functions over $\mathcal{S}$, for any $C > 1$ we may compute a C-approximate barycentric spanner for $\mathcal{S}$ in polynomial time, using $O(d^2 \log_C(d))$ calls to the optimization oracle.*

Proof. The algorithm is shown in Figure 5-2. Here, as elsewhere in this paper, we sometimes follow the convention of writing a matrix as a d-tuple of column vectors. The matrix $(\mathbf{e_1}, \ldots, \mathbf{e_d})$ appearing in the first step of the algorithm is the identity matrix. The "for" loop in the first half of the algorithm transforms this into a basis $(x_1, x_2, \ldots, x_d)$ contained in $\mathcal{S}$, by replacing the original basis vectors $(\mathbf{e_1}, \ldots, \mathbf{e_d})$ one-by-one with elements of $\mathcal{S}$. Each iteration of the loop requires two calls to the optimization oracle, to compute $x^* := \arg\max_{x \in \mathcal{S}} |\det(x, X_{-i})|$ by comparing the maxima of the linear functions

$$\ell_i(x) = \det(x, X_{-i}), \qquad -\ell_i(x) = -\det(x, X_{-i}).$$

/* First, compute a basis of $\mathbb{R}^d$ contained in $\mathcal{S}$. */
$(x_1, \ldots, x_d) \leftarrow (\mathbf{e}_1, \ldots, \mathbf{e}_d)$;
for $i = 1, 2, \ldots, d$ **do**
    /* Replace x_i with an element of $\mathcal{S}$ which is linearly independent from X_{-i}. */
    $x_i \leftarrow \arg\max_{x \in \mathcal{S}} |\det(x, X_{-i})|$;
end

/* Transform basis into approximate barycentric spanner. */
while $\exists x \in \mathcal{S}$, $i \in \{1, \ldots, d\}$ satisfying $|\det(x, X_{-i})| > C|\det(x_i, X_{-i})|$
    $x_i \leftarrow x$;
end
return $(x_1, x_2, \ldots, x_d)$

Figure 5-2: Algorithm for computing a C-approximate barycentric spanner.

This x^* is guaranteed to be linearly independent of the vectors in X_{-i} because ℓ_i evaluates to zero on X_{-i}, and is nonzero on x^*. (ℓ_i is non-zero on at least one point $x \in \mathcal{S}$ because $\mathcal{S}$ is not contained in a proper subspace of $\mathbb{R}^d$.)

Lemma 5.4 below proves that the number of iterations of the "while" loop in the second half of the algorithm is $O(d \log_C(d))$. Each such iteration requires at most $2d$ calls to the optimization oracle, i.e. two to test the conditional for each index $i \in \{1, \ldots, d\}$. At termination, $(x_1, \ldots, x_d)$ is a C-approximate barycentric spanner, by Observation 5.1. □

Lemma 5.4. *The total number of iterations of the "while" loop is $O(d \log_C(d))$.*

Proof. Let $M_i = (x_1, x_2, \ldots, x_i, \mathbf{e_{i+1}}, \ldots, \mathbf{e_d})$ be the matrix whose columns are the basis vectors at the end of the i-th iteration of the "for" loop. (Columns $i+1$ through d are unchanged at this point in the algorithm.) Let $M = M_d$ be the matrix at the end of the "for" loop, and let M' be the matrix at the end of the algorithm. Henceforth in this proof, $(x_1, \ldots, x_d)$ will refer to the columns of M, not M'.

It suffices to prove that $|\det(M')/\det(M)| \leq d^{d/2}$, because the determinant of the matrix increases by a factor of at least C on each iteration of the "while" loop. Let U be the matrix whose i-th row is $u_i := \mathbf{e_i}^\mathsf{T} M_i^{-1}$, i.e. the i-th row of M_i^{-1}. Recalling the linear function $\ell_i(x) = \det(x, X_{-i})$, one may verify that

$$u_i x = \frac{\ell_i(x)}{\ell_i(x_i)} \quad \forall x \in \mathbb{R}^d, \tag{5.1}$$

by observing that both sides are linear functions of x and that the equation holds when x is any of the columns of M_i. It follows that $|u_i x| \leq 1$ for all $x \in \mathcal{S}$, since

$x_i = \arg\max_{x \in \mathcal{S}} |\ell_i(x)|$. Each entry of the matrix UM' is equal to $u_i x$ for some $i \in \{1, \ldots, d\}$, $x \in \mathcal{S}$, so the entries of UM' lie between -1 and 1. Hence $|\det(UM')| \leq d^{d/2}$. (The determinant of a matrix cannot exceed the product of the L^2-norms of its columns.) Again using equation (5.1), observe that $u_i x_j$ is equal to 0 if $j < i$, and is equal to 1 if $j = i$. In other words UM is an upper triangular matrix with 1's on the diagonal. Hence $\det(UM) = 1$. Now

$$\left|\frac{\det(M')}{\det(M)}\right| = \left|\frac{\det(UM')}{\det(UM)}\right| \leq d^{d/2},$$

as desired. □

5.2.3 The online linear optimization algorithm

Our algorithm will employ a subroutine known as the "Kalai-Vempala algorithm with parameter ε." (Henceforth the "K-V black box.") The K-V black box is initialized with a parameter $\varepsilon > 0$ and a set $\mathcal{S} \subseteq \mathbb{R}^d$ of strategies. It receives as input a sequence of linear cost functions $c_j : \mathcal{S} \to \mathbb{R}$, $(1 \leq j \leq t)$, taking values in $[-M, M]$. Given a linear optimization oracle for $\mathcal{S}$, it computes a sequence of probability distributions p_j on $\mathcal{S}$, such that p_j depends only on $c_1, c_2, \ldots, c_{j-1}$. The K-V black box meets the following performance guarantee: if $x^{(1)}, \ldots, x^{(t)}$ are random samples from $p_1, \ldots, p_t$, respectively, and x is any point in $\mathcal{S}$, then

$$\mathbf{E}\left[\frac{1}{t}\sum_{j=1}^{t} c_j(x^{(j)})\right] \leq O(\varepsilon M d^2 + M d^2/\varepsilon t) + \frac{1}{t}\sum_{j=1}^{t} c_j(x). \tag{5.2}$$

See Section 2.2 for a description and analysis of the Kalai-Vempala algorithm with parameter ε. The assumptions in that section differ from those made here: Theorem 2.4 assumes the cost vectors satisfy $\|c_j\|_1 \leq 1$ and expresses the regret bound as

$$\mathbf{E}\left[\frac{1}{t}\sum_{j=1}^{t} c_j(x^{(j)})\right] \leq D\left(\frac{\varepsilon}{+}\frac{1}{\varepsilon t}\right) + \frac{1}{t}\sum_{j=1}^{t} c_j(x), \tag{5.3}$$

where D is the L^1-diameter of $\mathcal{S}$. To derive (5.2) from this, let $\{x_1, \ldots, x_d\}$ be a 2-approximate barycentric spanner for $\mathcal{S}$, and transform the coordinate system by mapping x_i to $(Md)\mathbf{e_i}$, for $i = 1, \ldots, d$. This maps $\mathcal{S}$ to a set whose L^1-diameter satisfies $D \leq 4Md^2$, by the definition of a 2-approximate barycentric spanner. The cost vectors in the transformed coordinate system have no component whose absolute value is greater than $1/d$, hence they satisfy the required bound on their L^1-norms.

Our algorithm precomputes a 2-approximate barycentric spanner $X \subseteq \mathcal{S}$, and initializes an instance of the K-V black box with parameter ε, where $\varepsilon = (dT)^{-1/3}$.

Assume, for simplicity, that T is divisible by d^2 and that T/d^2 is a perfect cube.[1] Divide the timeline $1, 2, \ldots, T$ into phases of length $\tau = d/\delta$, where $\delta = \varepsilon d^2$; note that τ is an integer by our assumption on T. The time steps in phase ϕ are numbered $\tau(\phi-1)+1, \tau(\phi-1)+2, \ldots, \tau\phi$. Call this set of time steps $\mathcal{T}_\phi$. Within each phase, the algorithm selects a subset of d time steps uniformly at random, and chooses a random one-to-one correspondence between these time steps and the elements of X. The step in phase ϕ corresponding to $x_i \in X$ will be called the "sampling step for x_i in phase ϕ;" all other time steps will be called "exploitation steps." In a sampling step for x_i, the algorithm chooses strategy x_i; in an exploitation step it samples its strategy randomly using the probability distribution computed by the K-V black box. At the end of each phase, the algorithm updates its K-V black box algorithm by feeding in the unique cost vector c_ϕ such that, for all $i \in \{1, \ldots, d\}$, $c_\phi \cdot x_i$ is equal to the cost observed in the sampling step for x_i.

Theorem 5.5. *The algorithm achieves regret of $O(Md^{5/3}T^{2/3})$ against an oblivious adversary, where d is the dimension of the problem space.*

Proof. Note that the cost vector c_ϕ satisfies $|c_\phi \cdot x_i| \leq M$ for all $x_i \in X$, and that its expectation is

$$\bar{c}_\phi = \mathbf{E}\left[c_\phi\right] = \frac{1}{\tau}\sum_{j \in \mathcal{T}_\phi} c_j.$$

Let $t = T/\tau$; note that t is an integer by our assumption on T. The performance guarantee for the K-V algorithm ensures that for all $x \in \mathcal{S}$,

$$\mathbf{E}\left[\frac{1}{t}\sum_{\phi=1}^{t} c_\phi \cdot x_\phi\right] \leq O\left(\varepsilon M d^2 + \frac{Md^2}{\varepsilon t}\right) + \frac{1}{t}\sum_{\phi=1}^{t} c_\phi \cdot x, \tag{5.4}$$

where x_ϕ is a random sample from the probability distribution specified by the black box in phase ϕ. Henceforth we will denote the term $O(\varepsilon M d^2 + Md^2/\varepsilon t)$ on the right side by R. Now let's take the expectation of both sides with respect to the algorithm's random choices. The key observation is that $\mathbf{E}[c_\phi \cdot x_j] = \mathbf{E}[\bar{c}_\phi \cdot x_j]$. This is because c_ϕ and x_j are independent random variables: c_ϕ depends only on sampling decisions made by the algorithm in phase ϕ, while x_j depends only on data fed to the K-V black box before phase ϕ, and random choices made by the K-V box during phase ϕ. Hence

$$\mathbf{E}[c_\phi \cdot x_j] = \mathbf{E}[c_\phi] \cdot \mathbf{E}[x_j] = \bar{c}_\phi \cdot \mathbf{E}[x_j] = \mathbf{E}[\bar{c}_\phi \cdot x_j].$$

[1] If T does not satisfy these properties, we may replace T with another integer $T' = O(T + d^2)$ without affecting the stated bounds by more than a constant factor.

Now taking the expectation of both sides of (5.4) with respect to the random choices of both the algorithm and the black box, we find that for all $x \in \mathcal{S}$,

$$\begin{aligned}
\mathbf{E}\left[\frac{1}{t}\sum_{\phi=1}^{t} \bar{c}_\phi \cdot x_\phi\right] &\leq R + \frac{1}{t}\sum_{\phi=1}^{t} \bar{c}_\phi \cdot x \\
\mathbf{E}\left[\frac{1}{t\tau}\sum_{\phi=1}^{t}\sum_{j\in\mathcal{T}_\phi} c_j \cdot x_j\right] &\leq R + \frac{1}{t\tau}\sum_{\phi=1}^{t}\sum_{j\in\mathcal{T}_\phi} c_j \cdot x \\
\mathbf{E}\left[\frac{1}{T}\sum_{j=1}^{T} c_j \cdot x_j\right] &\leq R + \frac{1}{T}\sum_{j=1}^{T} c_j \cdot x \\
\mathbf{E}\left[\sum_{j=1}^{T} c_j \cdot x_j\right] &\leq RT + \sum_{j=1}^{T} c_j \cdot x.
\end{aligned}$$

The left side is an upper bound on the total expected cost of all exploitation steps. The total cost of all sampling steps is at most $Mdt = \delta MT$. Thus the algorithm's expected regret satisfies

$$\begin{aligned}
\textit{Regret} &\leq RT + \delta MT \\
&= O\left(\varepsilon M d^2 T + \frac{M d^2 T}{\varepsilon t} + \delta MT\right) \\
&= O\left((\delta + \varepsilon d^2)MT + \frac{Md^3}{\varepsilon\delta}\right).
\end{aligned}$$

Recalling that $\varepsilon = (dT)^{-1/3}$, $\delta = d^{5/3}T^{-1/3}$, we obtain

$$\textit{Regret} = O(T^{2/3} M d^{5/3}).$$

□

5.2.4 Application to the online shortest path problem

Recall the online shortest path problem defined in Section 1.6 of the introduction. One is given a directed graph G with n vertices and m edges, and with a designated pair of vertices s, r. The strategy set $\mathcal{S}$ consists of all (not necessarily simple) paths from s to r of length at most H. Given an assignment of a cost between 0 and 1 to each edge of G, one obtains a function on $\mathcal{S}$ which assigns a cost to each path equal to the sum of its edge costs. Let Γ be the set of all such functions. The online shortest path problem is the generalized bandit problem for $(\mathcal{S}, \Gamma)$.

To apply our online linear optimization algorithm to the online shortest path problem, we take the vector space $\mathbb{R}^d$ to be the space of all flows from s to r in G, i.e.

the linear subspace of $\mathbb{R}^m$ satisfying the flow conservation equations at every vertex except s, r. (Thus $d = m - n + 2$.) The set $\mathcal{S}$ of all paths of length at most H from s to r is embedded in $\mathbb{R}^d$ by associating each path with the corresponding unit flow. Specifying a set of edge lengths defines a linear cost function on $\mathbb{R}^d$, namely the function which assigns to each flow the weighted sum of the lengths of all edges used by that flow, weighted by the amount of flow traversing the edge. The linear optimization oracle over $\mathcal{S}$ may be implemented using a suitable shortest-path algorithm, such as Bellman-Ford. The algorithm in Figure 5-2 describes how to compute a set of paths which form a 2-approximate barycentric spanner for $\mathcal{S}$. Applying the bound on regret from section 5.2.3, we obtain

$$\textit{Regret} \;\; = \;\; O(T^{2/3} H m^{5/3}).$$

Remark 5.1. In a graph G with two specified vertices s, r, a maximal linearly independent set of $s - r$ paths is not necessarily an approximate barycentric spanner. In fact, it is possible to construct a graph of size $O(n)$ having a maximal linearly independent set of $s - r$ paths which is not a C-approximate barycentric spanner for any $C = 2^{o(n)}$. For instance, let G be a graph with $n + 1$ vertices $v_0, v_1, \ldots, v_n$, and with each pair of consecutive vertices v_{i-1}, v_i $(1 \leq i \leq n)$ connected by two parallel edges e_i, e_i'. Given a vector $\vec{\mathbf{x}} = (x_1, x_2, \ldots, x_n)$ of length n, one can obtain a unit flow $f = F(\vec{\mathbf{x}})$ from $s = v_0$ to $r = v_n$ by specifying the flow values $f(e_i) = x_i$ and $f(e_i') = 1 - x_i$ for $i = 1, 2, \ldots, n$. (Here we allow the flow value on an edge to be negative.) If every component of $\vec{\mathbf{x}}$ is either 0 or 1, then $F(\vec{\mathbf{x}})$ is a path from s to r. If $\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_n$ are the columns of a nonsingular matrix X with $\{0, 1\}$-valued entries, then the paths $F(0), F(\vec{\mathbf{x}}_1), F(\vec{\mathbf{x}}_2), \ldots, F(\vec{\mathbf{x}}_n)$ are a maximal linearly independent set of $s - r$ paths in G. Let $A = (a_{ij})$ be the inverse of the matrix X, and observe that for any j,

$$\sum_{i=1}^{n} a_{ij} \vec{\mathbf{x}}_i = \vec{\mathbf{e}}_j,$$

where $\vec{\mathbf{e}}_j$ is the j-th column of the identity matrix. Now using the fact that the function $L(\vec{\mathbf{x}}) = F(\vec{\mathbf{x}}) - F(0)$ is a linear mapping, we find that

$$\sum_{i=1}^{n} a_{ij} L(\vec{\mathbf{x}}_i) = L(\sum_{i=1}^{n} a_{ij} \vec{\mathbf{x}}_i) = L(\vec{\mathbf{e}}_j)$$

which implies that the path $F(\vec{\mathbf{e}}_j)$ can be expressed as a linear combination

$$F(\vec{\mathbf{e}}_j) = \left(1 - \sum_{i=1}^{n} a_{ij}\right) F(0) + \sum_{i=1}^{n} a_{ij} F(\vec{\mathbf{x}}_i).$$

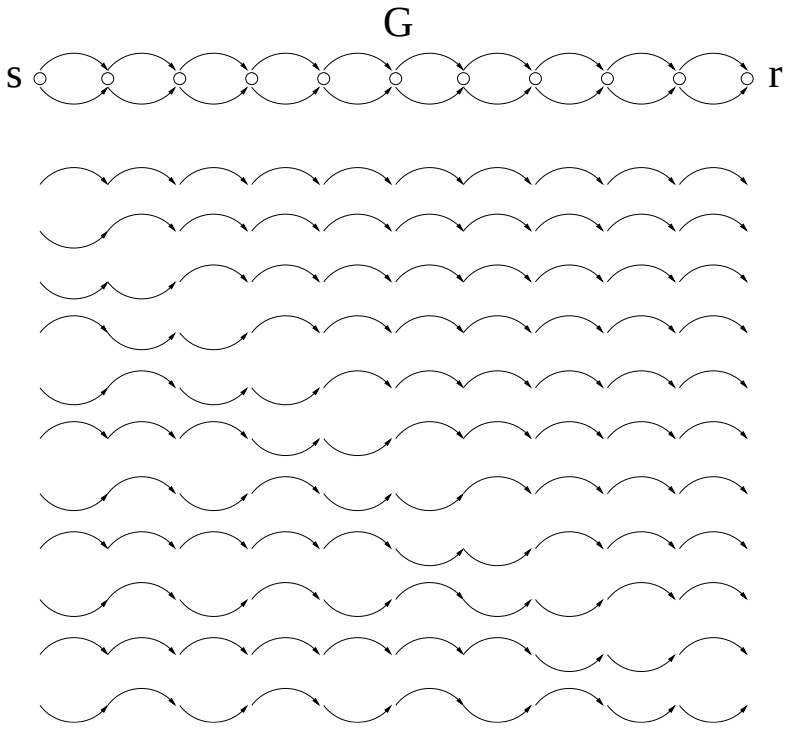

Figure 5-3: A maximal linearly independent set of paths which is not an approximate barycentric spanner.

To find a set of paths which are not a C-approximate barycentric spanner for any $C = 2^{o(n)}$, it therefore suffices to find an n-by-n matrix with $\{0, 1\}$-valued entries such that the inverse matrix contains entries whose absolute value is exponential in n. An example is the matrix

$$X_{ij} = \begin{cases} 1 & \text{if } i = j \\ 1 & \text{if } i = j + 1 \\ 1 & \text{if } i \text{ is even, } j \text{ is odd, and } i > j \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

whose inverse is

$$A_{ij} = \begin{cases} 1 & \text{if } i = j \\ -1 & \text{if } i = j + 1 \\ (-1)^{i-j} 2^{\lfloor i/2 \rfloor - \lfloor j/2 \rfloor - 1} & \text{if } i > j + 1 \\ 0 & \text{otherwise.} \end{cases} \tag{5.6}$$

See Figure 5-3 for an illustration of the graph G and the linearly independent set of paths defined by the matrix X in (5.5).

5.3 Further applications of barycentric spanners

Above, we have presented barycentric spanners as an ingredient in an algorithm for reducing online linear optimization to offline linear optimization. Our aim in this section is to demonstrate their usefulness in problems which involve uniform approximation of functions.

Suppose we are given a compact topological space $\mathcal{S}$ and a set V of continuous functions $\mathcal{S} \to \mathbb{R}$ which form a k-dimensional vector space under pointwise addition and scalar multiplication. For example, $\mathcal{S}$ could be a closed, bounded subset of $\mathbb{R}^d$ and V could be the space of all polynomial functions of degree at most n. (In this case $k \leq \binom{n+d}{d}$, with equality if and only if there is no non-zero polynomial of degree at most n which vanishes on $\mathcal{S}$.) For a continuous function $f : \mathcal{S} \to \mathbb{R}$, the L_∞-norm of f, denoted by $\|f\|_\infty$, is defined by

$$\|f\|_\infty = \min_{x \in \mathcal{S}} |f(x)|.$$

(Note that $\|f\|_\infty$ is finite for any continuous $f : \mathcal{S} \to \mathbb{R}$, since $\mathcal{S}$ is compact.) For any continuous function $f : \mathcal{S} \to \mathbb{R}$, we let $d_\infty(f, V)$ denote the minimum of $\|f - g\|_\infty$ as g ranges over V. If $g \in V$ and $\|f - g\|_\infty = d_\infty(f, V)$, we say that g is a *closest uniform approximation to f in V*. (See [62], Theorem I.1, for a proof that every continuous function f has at least one closest uniform approximation in V.)

Theorem 5.6. *Suppose $\mathcal{S}$ is a compact topological space and V is a set of continuous functions $\mathcal{S} \to \mathbb{R}$ which form a k-dimensional vector space under pointwise addition and scalar multiplication. Then there exist points $x_1, x_2, \ldots, x_k$ such that for any continuous function $f : \mathcal{S} \to \mathbb{R}$ there is a unique $g \in V$ satisfying $g(x_i) = f(x_i)$ for $i = 1, \ldots, k$, and $\|f - g\|_\infty \leq (k+1) \cdot d_\infty(f, V)$.*

In other words, we may compute a $(k+1)$-approximation of the closest uniform approximation to f in V, simply by evaluating f at a suitable point set X and interpolating an element of V through these function values. The set X does not depend on the function f, only on $\mathcal{S}$ and V.

Proof. Let $g_1, g_2, \ldots, g_k$ denote a basis for V. Let $G : \mathcal{S} \to R^k$ denote the mapping $G(x) = (g_1(x), g_2(x), \ldots, g_k(x))$. Since G is continuous and $\mathcal{S}$ is compact, the set $G(\mathcal{S})$ is a compact subset of $\mathbb{R}^k$. Note that $G(\mathcal{S})$ is not contained in any linear subspace of $\mathbb{R}^k$ since $g_1, \ldots, g_k$ are linearly independent. Let $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_k$ be a barycentric spanner for $G(\mathcal{S})$ and let $x_1, x_2, \ldots, x_k$ be elements of $\mathcal{S}$ such that $G(x_i) = \hat{x}_i$ for $i = 1, 2, \ldots, k$.

Since $\{G(x_i)\}_{i=1}^k$ is a basis for $\mathbb{R}^k$, the matrix M given by $M_{ij} = g_j(x_i)$ is non-singular. For any real numbers $y_1, y_2, \ldots, y_k$ there is a unique $g \in V$ such that $g(x_i) = y_i$ for $i = 1, 2, \ldots, k$. In fact, $g = \sum_{i=1}^k z_i g_i$, where $\vec{z} = M^{-1}\vec{y}$.

Now suppose f is any continuous function $\mathcal{S} \to \mathbb{R}$. Let g_0 be a closest uniform approximation to f in V, and let g be the unique element of V satisfying $g(x_i) = f(x_i)$ for $i = 1, 2, \ldots, k$. We have

$$|g_0(x_i) - g(x_i)| = |g_0(x_i) - f(x_i)| \leq d_\infty(f, V)$$

for $i = 1, 2, \ldots, k$. (The second inequality follows from the fact that g_0 is a closest uniform approximation to f in V.) Now for any $x \in \mathcal{S}$ we may write

$$G(x) = \sum_{i=1}^{k} z_i G(x_i)$$

for some coefficients $z_i \in [-1, 1]$ because $\{G(x_i)\}_{i=1}^k$ is a barycentric spanner for $G(\mathcal{S})$. Note that

$$h(x) = \sum_{i=1}^{k} z_i h(x_i)$$

for all $h \in V$, as may be verified by checking that both sides are linear functions of h and that they are equal when h is any element of the basis $\{g_1, \ldots, g_k\}$. In particular, taking $h = g_0 - g$, we find that

$$|g_0(x) - g(x)| = |h(x)| \leq \sum_{i=1}^{k} |z_i||h(x_i)| \leq k \cdot d_\infty(f, V),$$

since $|z_i| \leq 1$ and $|h(x_i)| \leq d_\infty(f, V)$ for $i = 1, \ldots, k$. Since x was an arbitrary point of $\mathcal{S}$, we conclude that $\|g_0 - g\|_\infty \leq k \cdot d_\infty(f, V)$, and

$$\|f - g\|_\infty \leq \|f - g_0\|_\infty + \|g_0 - g\|_\infty \leq (k + 1) \cdot d_\infty(f, V),$$

as claimed. □

A related theorem is the following generalization of Theorem 1.5.

Theorem 5.7. *Let $\mathcal{S}, V$ satisfy the hypotheses of Theorem 5.6 and suppose that $\Gamma \subset V$ is a subset of V such that $|g(x)| \leq M$ for all $x \in \mathcal{S}, g \in \Gamma$. Let $\mathcal{A}$ denote the set of oblivious adversaries for online decision domain $(\mathcal{S}, \Gamma)$. There is an algorithm* ALG *satisfying*

$$R(\mathsf{ALG}, \mathcal{A}; T) = O(T^{2/3} k^{5/3} M).$$

The algorithm ALG *may be requires only polynomial computation time, given an oracle to minimize functions in V over the space $\mathcal{S}$.*

Proof. Let $\{g_1, \ldots, g_k\}$ be a basis for V. Map $\mathcal{S}$ to $\mathbb{R}^k$ via the mapping $G(x) = (g_1(x), \ldots, g_k(x))$ and apply the algorithm of Section 5.2.3 with strategy set $G(\mathcal{S})$. □

Thus, for example, given a compact set $\mathcal{S} \subset \mathbb{R}^d$ and a cost function class Γ consisting of bounded-degree polynomials mapping $\mathcal{S}$ to $[-M, M]$, and given an oracle for minimizing bounded-degree polynomials on $\mathcal{S}$, there is an efficient online algorithm for the generalized bandit problem with strategy set $\mathcal{S}$ and cost function class Γ.

5.4 Online convex optimization

An *online convex programming problem* is an online decision problem in which the strategy set $\mathcal{S}$ is a convex subset of $\mathbb{R}^d$ (for some $d \geq 0$) and the cost function class Γ is a subset of the set of real-valued convex functions on $\mathcal{S}$. In Section 2.3 we considered the generalized best-expert problem for online convex programming problems with full feedback, when $(\mathcal{S}, \Gamma)$ is an online decision domain satisfying:

- $\mathcal{S}$ is a compact convex subset of $\mathbb{R}^d$.
- The elements of Γ are differentiable, and their gradients are uniformly bounded.

We presented an algorithm called Greedy Projection, due to Zinkevich, which achieves polynomial convergence time for such problems. In this section we assume opaque feedback — i.e. the generalized bandit problem for $(\mathcal{S}, \Gamma)$ — and we will design an online convex programming algorithm, based on Zinkevich's algorithm, achieving polynomial convergence time in this more limited feedback model. To achieve this, we must make some slightly more restrictive assumptions about the cost functions in Γ. We now explain these assumptions.

Define $\|x\| = \sqrt{x \cdot x}$ and $d(x, y) = \|x - y\|$. We will make the following standing assumptions about the strategy set $\mathcal{S}$ and the convex functions c_t.

1. The diameter of $\mathcal{S}$ is bounded. Let $\|\mathcal{S}\| = \max_{x,y \in \mathcal{S}} d(x, y)$.
2. $\mathcal{S}$ is a closed subset of $\mathbb{R}^d$.
3. The cost functions c_t are twice continuously differentiable.
4. $0 \leq c_t(x) \leq 1$ for all $x \in \mathcal{S}$.
5. The gradient ∇c_t is bounded. Let
$$\|\nabla c\| = \max\{\nabla c_t(x) \mid 1 \leq t \leq n,\, x \in \mathcal{S}\}.$$
6. The Hessian matrix $H(c_t) = \left(\frac{\partial^2 c_t}{\partial x_i \partial x_j}\right)_{i,j=1}^{n}$ has bounded L^2 operator norm. Let
$$\|H(c)\| = \max\{u^{\mathsf{T}} H(c_t) u \mid 1 \leq t \leq n,\, u \in \mathbb{R}^d, u \cdot u = 1\}.$$

7. For all $y \in \mathbb{R}^d$, there is an algorithm which can produce the vector

$$P(y) := \arg\min_{x \in \mathcal{S}} d(x, y).$$

In comparison with Zinkevich's assumptions, we are making additional boundedness assumptions about the cost functions. Specifically, we are assuming upper bounds on the size of c_t, its gradient, and its Hessian matrix, whereas Zinkevich requires only an upper bound on the size of the gradient.

Recall the Greedy Projection algorithm from Section 2.3. Observe that this algorithm is nearly implementable in the opaque feedback model: if the feedback revealed $\nabla c_t(x_t)$, we would have sufficient information to run the algorithm. Since we instead learn only $c_t(x_t)$, our solution will be to spend a sequence of $d+1$ consecutive time steps $t, t+1, \ldots, t+d$ sampling random vectors near x_t and then interpolate a linear function through the sampled values. We will use the gradient of this linear function as a substitute for $\nabla c_t(x_t) + \nabla c_{t+1}(x_t) + \ldots + \nabla c_{t+d}(x_t)$. The resulting algorithm will be called *simulated greedy projection*, or SGP.

Before specifying the algorithm precisely, we must make a few assumptions, definitions, and observations. First, we may assume without loss of generality that $\mathcal{S}$ is not contained in a proper linear subspace of $\mathbb{R}^d$; if it were contained in a proper linear subspace, we would replace $\mathbb{R}^d$ with the lower-dimensional vector space $\text{span}(\mathcal{S})$ and conduct the algorithm and proof in that space. Second, we may assume without loss of generality that the centroid of $\mathcal{S}$ is 0; if not, shift the coordinate system by parallel translation to move the centroid of $\mathcal{S}$ to 0. Now, for a bounded measurable set $X \subseteq \mathbb{R}^d$ of positive volume, the *moment of inertia tensor* is the matrix

$$M(X) = \frac{1}{\text{vol}(X)} \int_{x \in X} x x^{\mathsf{T}} dx.$$

This is a symmetric positive definite matrix, hence there exists a rotation matrix Q such that $M(QX) = QM(X)Q^{\mathsf{T}}$ is a diagonal matrix with eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d > 0$. Note that all of our assumptions about the strategy set $\mathcal{S}$ and the cost functions c_t are unchanged if we rotate the coordinate system using an orthogonal matrix Q, replacing $\mathcal{S}$ with $Q\mathcal{S}$ and c_t with the function $x \mapsto c_t(Q^{\mathsf{T}}x)$. We may therefore assume, without loss of generality, that $M(\mathcal{S})$ is a diagonal matrix.

We will apply one more coordinate transformation to transform $M(\mathcal{S})$ into a scalar multiple of the identity matrix. Let P be the diagonal matrix whose diagonal entries are $1, \sqrt{\lambda_1/\lambda_2}, \ldots, \sqrt{\lambda_1/\lambda_d}$, so that $M(P\mathcal{S}) = PM(\mathcal{S})P^{\mathsf{T}} = \lambda_1 \text{Id}_d$. (Here Id_d denotes the d-by-d identity matrix.) Note that each diagonal entry of P is greater than or equal to 1, so that the linear transformation $x \mapsto Px$ scales up each coordinate of x by a factor of at least 1. If we change coordinates by replacing $\mathcal{S}$ with $P\mathcal{S}$ and

replacing c_t with the function $x \mapsto c_t(P^{-1}x)$, this may increase the diameter of $\mathcal{S}$ but it does not increase any of the derivatives of c_t. The increase in diameter of $\mathcal{S}$ is bounded above by a factor which may be determined as follows. First, we have $P\mathcal{S} \subseteq B(0, (d+1)\sqrt{\lambda_1})$ (see [46]) which implies $\|P\mathcal{S}\| \leq 2(d+1)\sqrt{\lambda_1}$. Second, we have $(1/\mathrm{vol}(\mathcal{S})) \int_{x\in\mathcal{S}} x_1^2 dx = \lambda_1$, which implies that $\|\mathcal{S}\| \geq \sqrt{\lambda_1}$. Putting these two estimates together, we get $\|P\mathcal{S}\|/\|\mathcal{S}\| \leq 2(d+1)$. From now on, we will replace $\mathcal{S}$ with $P\mathcal{S}$, keeping in mind that the increase in diameter must be accounted for at the end of the analysis of the algorithm.

Our algorithm SGP uses a sequence of *learning rates* $\eta_1, \eta_2, \ldots$ and *frame sizes* $\nu_1, \nu_2, \ldots$, defined by

$$\begin{aligned} \eta_k &= k^{-3/4} \\ \nu_k &= k^{-1/4}\lambda_1^{-1/2}. \end{aligned}$$

Let $q = d+1$. The timeline $1, 2, \ldots, T$ is divided into $\tau = \lceil T/q \rceil$ phases of length q. The phases are numbered $0, 1, \ldots, \tau - 1$, with the time steps in phase ϕ being numbered $q\phi + 1, q\phi + 2, \ldots, q\phi + q$. The algorithm computes a sequence of vectors $u_0, u_1, \ldots, u_{\tau-1}$, beginning with an arbitrary vector $u_0 \in \mathbb{R}^d$. At the start of a phase ϕ, the vector u_ϕ being already defined, the algorithm selects a barycentric spanner $\{y_1, \ldots, y_d\}$ for the set $\mathcal{S} - u_\phi$. During phase ϕ, the algorithm samples the q vectors in the set $\{u_\phi\} \cup \{u_\phi + \nu_\phi y_k \,|\, 1 \leq k \leq d\}$ in a random order $x_{q\phi+1}, \ldots, x_{q\phi+q}$. After receiving the feedback values $c_t(x_t)$ for $q\phi + 1 \leq t \leq q\phi + q$, it computes the unique affine function $\tilde{\Lambda}_\phi(x) = a_\phi^{\mathsf{T}} x + b_\phi$ satisfying $\tilde{\Lambda}_\phi(x_t) = c_t(x_t)$ for $q\phi + 1 \leq t \leq q\phi + q$. The next vector $u_{\phi+1}$ is determined according to the Greedy Projection rule:

$$u_{\phi+1} = P(u_\phi - \eta_\phi \nabla \tilde{\Lambda}_\phi(u_\phi)).$$

This completes the description of the algorithm SGP. The analysis is as follows.

Theorem 5.8. *The expected regret of the simulated greedy projection algorithm satisfies the bound*

$$\begin{aligned} R(\mathsf{SGP}, \mathcal{A}; T) &= O\left(d^{9/4}\|\mathcal{S}\|^2 T^{3/4} + d^{9/4}T^{3/4} + d^{17/4}\|\mathcal{S}\|\,\|H(c)\|T^{3/4}\right. \\ &\qquad \left. + d^{5/4}\|\nabla c\|T^{3/4}\right). \end{aligned}$$

Proof. Let $C_\phi = \sum_{t=q\phi+1}^{q\phi+q} c_t$. We will consider the online convex programming program defined by the cost functions $C_0, C_1, \ldots, C_{\tau-1}$, and we will prove that the vectors $u_0, u_1, \ldots, u_{\tau-1}$ are a low-regret solution to this problem with the help of Theorem 2.6. This implies that if we use the sequence of vectors $u_0, \ldots, u_{\tau-1}$ each repeated q times, we obtain a low-regret solution to the original convex programming problem defined by $c_1, \ldots, c_T$. Finally, we will show that the algorithm's regret is not much greater

when using the vectors $x_1, \ldots, x_T$ rather than using $u_0, \ldots, u_{\tau-1}$ each repeated q times.

Theorem 2.6 guarantees that

$$\sum_{\phi=0}^{\tau-1} \tilde{\Lambda}_\phi(u_\phi) \leq \sum_{\phi=0}^{\tau-1} \tilde{\Lambda}_\phi(x) + O\left(\|\mathcal{S}\|^2 (T/q)^{3/4} + \|\nabla\tilde{\Lambda}\|^2 (T/q)^{1/4}\right). \tag{5.7}$$

To apply this bound, we need to get an upper bound for $\|\nabla\tilde{\Lambda}_\phi\|^2$. This is accomplished via the following calculation:

$$\begin{aligned}
\|\nabla\tilde{\Lambda}_\phi\|^2 &= (\nabla\tilde{\Lambda}_\phi)^\mathsf{T} \nabla\tilde{\Lambda}_\phi \\
&= \frac{1}{\lambda_1} (\nabla\tilde{\Lambda}_\phi)^\mathsf{T} M(\mathcal{S}) \nabla\tilde{\Lambda}_\phi \\
&= \frac{1}{\lambda_1 \mathrm{vol}(\mathcal{S})} \int_{x\in\mathcal{S}} (\nabla\tilde{\Lambda}_\phi)^\mathsf{T} x x^\mathsf{T} \nabla\tilde{\Lambda}_\phi dx \\
&= \frac{1}{\lambda_1 \mathrm{vol}(\mathcal{S})} \int_{x\in\mathcal{S}} (\nabla\tilde{\Lambda}_\phi \cdot x)^2 dx \\
&= \frac{1}{\lambda_1 \mathrm{vol}(\mathcal{S})} \int_{x\in\mathcal{S}} (\tilde{\Lambda}_\phi(x) - \tilde{\Lambda}_\phi(0))^2 dx
\end{aligned}$$

Now recall that each $x \in \mathcal{S}$ may be expressed in the form $u_\phi + a_1 y_1 + \ldots + a_d y_d$ where $a_i \in [-2, 2]$, and therefore $|\tilde{\Lambda}_\phi(x)| \leq |\tilde{\Lambda}_\phi(u_\phi)| + 2|\tilde{\Lambda}_\phi(y_1)| + \ldots + 2|\tilde{\Lambda}_\phi(y_d)|$ for each $x \in \mathcal{S}$. We have $\tilde{\Lambda}_\phi(u_\phi) \in [0, 1]$ and $\tilde{\Lambda}_\phi(u_\phi + \nu_\phi y_k) \in [0, 1]$ by the construction of $\tilde{\Lambda}_\phi$. It follows that $|\tilde{\Lambda}_\phi(u_\phi)| \leq 1$ and $|\tilde{\Lambda}_\phi(y_k)| \leq 1/\nu_\phi \leq \sqrt{\lambda_1}(T/q)^{1/4}$. Putting this all together, we have $|\tilde{\Lambda}_\phi(x)| \leq 2\sqrt{\lambda_1} q (T/q)^{1/4}$ for all $x \in \mathcal{S}$ (including the case $x = 0$) hence

$$\|\nabla\tilde{\Lambda}_\phi\|^2 \leq \frac{1}{\lambda_1 \mathrm{vol}(\mathcal{S})} \int_{x\in\mathcal{S}} 16\lambda_1 q^2 (T/q)^{1/2} dx = 16 q^{3/2}\sqrt{T}.$$

Substituting this back into (5.7), we obtain

$$\sum_{\phi=0}^{\tau-1} \tilde{\Lambda}_\phi(u_\phi) \leq \sum_{\phi=0}^{\tau-1} \tilde{\Lambda}_\phi(x) + O\left(\|\mathcal{S}\|^2 (T/q)^{3/4} + q^{5/4} T^{3/4}\right). \tag{5.8}$$

Let Λ_ϕ be the unique affine function satisfying $\Lambda_\phi(x_t) = C_\phi(x_t)/q$ for $q\phi + 1 \leq t \leq q\phi + q$. Note that $\Lambda_\phi = \mathbf{E}(\tilde{\Lambda}_\phi \| u_\phi)$, as may be verified by evaluating both sides at the vectors x_t ($q\phi + 1 \leq t \leq q\phi + q$). Taking the expectation of both sides of (5.8) we obtain

$$\frac{1}{q} \sum_{\phi=0}^{\tau-1} \mathbf{E}\left[C_\phi(u_\phi)\right] \leq \sum_{\phi=0}^{\tau-1} \mathbf{E}\left[\Lambda_\phi(x)\right] + O\left(\|\mathcal{S}\|^2 (T/q)^{3/4} + q^{5/4} T^{3/4}\right). \tag{5.9}$$

Our next goal is to transform this bound so that the sum on the right side is $\frac{1}{q}\sum_{\phi=0}^{\tau-1} C_\phi(x)$ rather than $\sum_{\phi=0}^{\tau-1}\mathbf{E}\left[\Lambda_\phi(x)\right]$. This requires estimating an upper bound for the difference $\Lambda_\phi - C_\phi/q$. To do so, let L_ϕ be the linearization of C_ϕ/q at u_ϕ, i.e. the function $L_\phi(x) = \frac{1}{q}[C_\phi(u_\phi) + \nabla C_\phi(u_\phi)\cdot(x-u_\phi)]$. By convexity of C_ϕ, we have $L_\phi \le C_\phi/q$ and therefore $\Lambda_\phi - C_\phi/q \le \Lambda_\phi - L_\phi$. To bound $\Lambda_\phi(x) - L_\phi(x)$ for $x \in \mathcal{S}$, we write $x = u_\phi + a_1 y_1 + \ldots + a_d y_d$ with $a_k \in [-2,2]$ and use the fact that $\Lambda_\phi(u_\phi) = L_\phi(u_\phi)$ to obtain:

$$\Lambda_\phi(x) - L_\phi(x) = \sum_{k=1}^{d} a_k[\Lambda_\phi(y_k) - L_\phi(y_k)] \le 2\sum_{k=1}^{d}\left|\left(\nabla\Lambda_\phi - \frac{1}{q}\nabla C_\phi(u_\phi)\right)\cdot y_k\right|.$$

For $s \in [0,1]$, let $f(s) = \Lambda_\phi(u_\phi + s y_k) - \frac{1}{q}C_\phi(u_\phi + s y_k)$. We have

$$\begin{aligned} f'(s) &= \left[\nabla\Lambda_\phi - \frac{1}{q}\nabla C_\phi(u_\phi + s y_k)\right]\cdot y_k \\ |f''(s)| &= \frac{1}{q} y_k^{\mathsf{T}} H(C_\phi) y_k \\ &\le \frac{1}{q}\|y_k\|^2 \|H(C_\phi)\| \\ &\le \|\mathcal{S}\|^2 \|H(c)\|. \end{aligned}$$

We also know that $f(0) = f(\nu_\phi) = 0$, so by the mean value theorem $f'(t) = 0$ for some $t \in [0, \nu_\phi]$. Now

$$\begin{aligned} \left|\left(\nabla\Lambda_\phi - \frac{1}{q}\nabla C_\phi(u_\phi)\right)\cdot y_k\right| &= |f'(0)| \\ &= \left|-\int_0^t f''(s)ds\right| \\ &\le qt\|\mathcal{S}\|^2\|H(c)\| \\ &\le q\phi^{-1/4}\lambda_1^{-1/2}\|\mathcal{S}\|^2\|H(c)\| \end{aligned}$$

$$\Lambda_\phi(x) - \frac{1}{q}C_\phi(x) \le 2\sum_{k=1}^{d}\left|\left(\nabla\Lambda_\phi - \frac{1}{q}\nabla C_\phi(u_\phi)\right)\cdot y_k\right| \le 4q^3\|\mathcal{S}\|\,\|H(c)\|\phi^{-1/4},$$

where the last inequality used the fact that $\|\mathcal{S}\| \le 2q\sqrt{\lambda_1}$ as was established above. Now summing over $\phi = 0, 1, \ldots, \tau-1$, we obtain:

$$\sum_{\phi=0}^{\tau-1}\Lambda_\phi(x) \le \frac{1}{q}\sum_{\phi=0}^{\tau-1} C_\phi(x) + O(q^3\|\mathcal{S}\|\,\|H(c)\|(T/q)^{3/4}). \tag{5.10}$$

Putting together (5.9) and (5.10) and taking unconditional expectations, we have

$$\sum_{t=1}^{T} \mathbf{E}\left[c_t(u_\phi)\right] \leq \sum_{t=1}^{T} c_t(x) + O\left(q^{1/4}\|\mathcal{S}\|^2 T^{3/4} + q^{9/4}T^{3/4} + q^{13/4}\|\mathcal{S}\|\,\|H(c)\|T^{3/4}\right). \tag{5.11}$$

Finally, note that

$$\begin{aligned}
\sum_{\phi=0}^{\tau-1}\sum_{t=q\phi+1}^{q\phi+q} c_t(x_t) - c_t(u_\phi) &\leq \sum_{\phi=0}^{\tau-1}\sum_{t=q\phi+1}^{q\phi+q} \|\nabla c\|\,\|x_t - u_\phi\| \\
&\leq \|\nabla c\|\,\|\mathcal{S}\| q \sum_{\phi=0}^{\tau-1} \nu_\phi \\
&= O(q\,\|\nabla c\|\,\|\mathcal{S}\|(T/q)^{3/4}\lambda_1^{-1/2}) \\
&= O(q^{5/4}\|\nabla c\|\,T^{3/4}).
\end{aligned}$$

Combining this with (5.11) we obtain

$$\begin{aligned}
\sum_{t=1}^{T} c_t(x_t) &\leq \sum_{t=1}^{T} c_t(x) + O\left(q^{1/4}\|\mathcal{S}\|^2 T^{3/4} + q^{9/4}T^{3/4} + q^{13/4}\|\mathcal{S}\|\,\|H(c)\|T^{3/4}\right. \\
&\qquad \left. + q^{5/4}\|\nabla c\|\,T^{3/4}\right) \\
R(\mathsf{ALG}, \mathcal{A}; T) &= O\left(q^{1/4}\|\mathcal{S}\|^2 T^{3/4} + q^{9/4}T^{3/4} + q^{9/4}\|\mathcal{S}\|\,\|H(c)\|T^{3/4}\right. \\
&\qquad \left. + q^{5/4}\|\nabla c\|T^{3/4}\right).
\end{aligned} \tag{5.12}$$

Recalling that we initially applied a coordinate transformation which potentially increased the diameter of $\mathcal{S}$ by a factor of $2q$, we replace each factor of $\|\mathcal{S}\|$ in (5.12) by $q\|\mathcal{S}\|$ to obtain the bound specified in the statement of the theorem. □

5.4.1 The algorithm of Flaxman, Kalai, and McMahan

A different algorithm for online convex programming with opaque feedback was discovered by Flaxman, Kalai, and McMahan [32] independently and simultaneously with our discovery of the algorithm presented above. In this section we will present this alternative algorithm, denoted by BGD. We will present two theorems from [32] specifying upper bounds on the regret of BGD under different cost function classes, and we will compare BGD with our algorithm SGP.

The algorithm BGD is shown in Figure 5-4. It depends on three real-valued parameters α, δ, ν which are specified when the algorithm is invoked. As in Section 2.3, for a convex body S, we use the notation $P_S(\cdot)$ to denote the projection function

$$P_S(z) = \arg\min_{x \in S} d(x, z)$$

Algorithm BGD(α, δ, ν)
/* Initialization */
$u_1 \leftarrow 0$

/* Main loop */
for $t = 1, 2, \ldots, T$
    Select unit vector w_t uniformly at random.
    $x_t \leftarrow u_t + \delta w_t$
    Play strategy x_t.
    Observe feedback y_t.
    $u_{t+1} \leftarrow P_{(1-\alpha)\mathcal{S}}(u_t - \nu y_t w_t)$
end

Figure 5-4: The algorithm BGD

where $d(\cdot, \cdot)$ denotes Euclidean distance.

As with our algorithm SGP, the key idea in the algorithm BGD is to reduce from the opaque feedback model to the full feedback model, and then to use Zinkevich's Greedy Projection algorithm. The reduction from opaque feedback to full feedback requires estimating the gradient of a cost function at a point $u \in \mathcal{S}$, given the limitation that the algorithm, in trial t, can evaluate c_t at only one point $x_t \in \mathcal{S}$. The two algorithms differ in their approach to obtaining this estimate of the gradient. The SGP algorithm chooses to estimate the gradient of the average of $d+1$ *consecutive* cost functions (i.e. all of the cost functions in phase ϕ) by evaluating them at a set of $d+1$ points near u whose affine span contains $\mathcal{S}$. The BGD algorithm estimates the gradient of a *single* cost function while evaluating the cost function at only one point. This is achieved by moving a small distance ν away from u in a uniformly-distributed random direction w. The correctness of the algorithm hinges on the observation (Lemma 2.1 of [32]) that for any integrable function c, the vector-valued function $\mathbf{E}[(d/\nu)c(u + \nu w)w]$ is equal to the gradient of a "smoothed" version of c in which one replaces the value of c, at each point, with the average value of c over a ball of radius ν centered at that point.

To state upper bounds on the performance of algorithm BGD, we must make some definitions and assumptions. Let $\mathbb{B}$ denote the closed unit ball centered at the origin in $\mathbb{R}^d$, and assume that

$$r\mathbb{B} \subseteq \mathcal{S} \subseteq R\mathbb{B}.$$

Let Γ denote the set of convex functions from $\mathcal{S}$ to $[-M, M]$, for some constant M.

Let $\Gamma_L \subseteq \Gamma$ denote the subset of Γ consisting of functions which satisfy a Lipschitz condition with exponent 1 and constant L, i.e. Γ_L is the set of all $c \in \Gamma$ satisfying

$$|c(x) - c(y)| \leq Ld(x, y)$$

for all $x, y \in \mathcal{S}$. Let $\mathcal{A}(\Gamma)$ (resp. $\mathcal{A}(\Gamma_L)$) denote the set of adaptive adversaries for Γ (resp. Γ_L).

The following two theorems are stated as Theorems 3.2 and 3.3 in [32].

Theorem 5.9. *For any $T \geq \left(\frac{3Rd}{2r}\right)^2$ and for $\nu = \frac{R}{M\sqrt{T}}$, $\delta = \sqrt[3]{\frac{rR^2d^2}{12T}}$, and $\alpha = \sqrt[3]{\frac{3Rd}{2r\sqrt{T}}}$,*

$$R(\mathsf{BGD}(\alpha, \delta, \nu), \mathcal{A}(\Gamma); T) \leq 3CT^{5/6}\sqrt[3]{dR/r}.$$

Theorem 5.10. *Given any $L < \infty$, for sufficiently large T and for $\nu = \frac{R}{c\sqrt{T}}$, $\delta = T^{-1/4}\sqrt{\frac{RdCr}{3(Lr+M)}}$, $\alpha = \frac{\delta}{r}$,*

$$R(\mathsf{BGD}(\alpha, \delta, \nu), \mathcal{A}(\Gamma_L); T) \leq 2T^{3/4}\sqrt{3RdM(L + M/r)}.$$

The bounds in the preceding theorems depend on the radii of the balls $R\mathbb{B}, r\mathbb{B}$, one containing $\mathcal{S}$ and the other contained in $\mathcal{S}$. To eliminate this dependence, we may first apply a linear transformation to the coordinate system which puts $\mathcal{S}$ in isotropic position, as above, and then we may run BGD in the transformed coordinate system. Let us refer to the resulting algorithm as $\mathsf{BGD}'(\alpha, \delta, \nu)$. This leads to the following bounds, stated as Corollary 3.1 in [32].

Corollary 5.11. *For α, δ, ν as in Theorem 5.9, and for T sufficiently large,*

$$R(\mathsf{BGD}'(\alpha, \delta, \nu), \Gamma; T) \leq 6T^{5/6}dM.$$

For any $L > 0$, if one sets α, δ, ν as in Theorem 5.10 and if T is sufficiently large, then

$$R(\mathsf{BGD}'(\alpha, \delta, \nu), \Gamma; T) \leq 6T^{3/4}d(\sqrt{ML} + M).$$

Comparing Corollary 5.11 with our Theorem 5.8, we see that the analysis of BGD is stronger in the sense that:

- It achieves a stronger upper bound on regret — $O(T^{3/4}d)$ for Lipschitz cost functions, as opposed to $O(T^{3/4}d^{17/4})$.
- It applies against a stronger class of adversaries — adaptive as opposed to oblivious.

- It requires weaker continuity and smoothness assumptions about the cost functions — L-Lipschitz as opposed to C^2 with uniformly bounded first and second derivatives.

One could also argue that the BGD algorithm is easier to implement, since it does not require computing approximate barycentric spanners.

Chapter 6

Online optimization in measure spaces

In earlier chapters we have studied bandit problems with constrained cost functions. What if the strategy set is exponential or infinite, but the set of cost functions is unconstrained? Can one prove non-trivial bounds on the regret of generalized bandit algorithms, if the sequence of trials has only polynomial length? Trivially, one can not devise algorithms whose regret is $o(T)$ when T is much smaller than the cardinality of the strategy set. (For instance, there might be one strategy with cost 0, while all other strategies have cost 1. Given significantly fewer than $|\mathcal{S}|$ trials, an algorithm is unlikely to find the unique strategy with cost 0.) But what if we only require that the algorithm should have small regret relative to *most* strategies at time T? For example, suppose the strategy set $\mathcal{S}$ is a finite set of size K. After a constant number of trials, one might hope that the algorithm's expected cost is not much worse than the *median* cost of a strategy in $\mathcal{S}$, and after a larger — but still constant — number of trials one might hope that the algorithm's expected cost nearly outperforms all but the best $\frac{K}{100}$ strategies in $\mathcal{S}$. More generally one might require that for all $\delta > 0$, the fraction of strategies in $\mathcal{S}$ which outperform the algorithm's expected cost by more than δ converges to zero as $T \to \infty$, at a rate which does not depend on K. We call algorithms with this property *anytime bandit algorithms* because they have the property that, if stopped at any time $T > 0$, they satisfy a non-trivial performance guarantee which improves as $T \to \infty$, eventually converging to optimality. In this section we formulate two precise definitions of "anytime bandit algorithm," prove these two notions are equivalent, and present algorithms satisfying either of the equivalent definitions. We also formulate a stronger notion which we call a "perfect anytime bandit algorithm," and we prove that no such algorithm exists.

6.1 Definitions

Definition 6.1. Suppose we are given a strategy set $\mathcal{S}$, a time horizon T, an algorithm ALG, and a set of adversaries $\mathcal{A}$. For a subset $U \subseteq \mathcal{S}$, the *normalized U-regret* of ALG against $\mathcal{A}$ is defined by:

$$\overline{R}(\mathsf{ALG}, \mathcal{A}; U, T) = \max_{\mathsf{ADV} \in \mathcal{A}} \max_{x \in U} \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) - c_t(x)\right].$$

If U is a singleton set $\{x\}$ or $\mathcal{A}$ is a singleton set $\{\mathsf{ADV}\}$, we will use notations such as $\overline{R}(\mathsf{ALG}, \{\mathsf{ADV}\}; \{x\}, T)$ and $\overline{R}(\mathsf{ALG}, \mathsf{ADV}; x, T)$ interchangeably.

Definition 6.2. Given a probability space $(\mathcal{S}, \mu)$, an algorithm ALG is called an *anytime bandit algorithm for* $(\mathcal{S}, \mu)$ if there exists a function $\tau(\varepsilon, \delta)$, defined for all $\varepsilon, \delta > 0$ and taking values in $\mathbb{N}$, such that for all oblivious adversaries ADV there exists a subset $U \subseteq \mathcal{S}$ such that $\mu(\mathcal{S} \setminus U) \leq \varepsilon$ and $\overline{R}(\mathsf{ALG}, \mathsf{ADV}; U, T) < \delta$ for all $T > \tau(\varepsilon, \delta)$. It is a *perfect anytime bandit algorithm* if $\tau(\varepsilon, \delta) \leq (1/\varepsilon)\mathrm{poly}(\log(1/\varepsilon), 1/\delta)$.

To gain an intuition for Definition 6.2, it is helpful to consider the case in which $\mathcal{S}$ is a finite set of size K and μ is the uniform measure on $\mathcal{S}$. Then the definition states that for all $T > \tau(\varepsilon, \delta)$, there are at most εK strategies $x \in \mathcal{S}$ satisfying $\overline{R}(\mathsf{ALG}, \mathsf{ADV}; x, T) \geq \delta$. Generalizing this to an arbitrary measure space $(\mathcal{S}, \mu)$, Definition 6.2 says that ALG is an anytime bandit algorithm for $(\mathcal{S}, \mu)$ if the set of strategies which outperform ALG by more than δ shrinks to have measure zero as $T \to \infty$, and has measure less than ε whenever $T > \tau(\varepsilon, \delta)$.

A useful alternative definition of "anytime bandit algorithm" assumes that $\mathcal{S}$ is a countable set whose elements are arranged in an infinite sequence $x_1, x_2, \ldots$. (Equivalently, we may simply assume that $\mathcal{S} = \mathbb{N}$.) We think of an element's position in this sequence as indicating its "priority" for the algorithm, and the algorithm's objective at time T is to perform nearly as well as all of the highest-priority strategies in the sequence, i.e. those belonging to an initial segment $x_1, x_2, \ldots, x_j$ whose length tends to infinity with T.

Definition 6.3. An algorithm ALG with strategy set $\mathbb{N}$ is called an *anytime bandit algorithm for* $\mathbb{N}$ if there exists a function $\tau(j, \delta)$, defined for all $j \in \mathbb{N}$, $\delta > 0$ and taking values in $\mathbb{N}$, such that $\overline{R}(\mathsf{ALG}, \mathcal{A}; \{1, \ldots, j\}, T) < \delta$ for all $T > \tau(j, \delta)$. It is a *perfect anytime bandit algorithm* if $\tau(j, \delta) \leq j\,\mathrm{poly}(\log(j), 1/\delta)$.

In both cases, the function τ is called the *convergence time* of the algorithm. Observe that the $\Omega(\sqrt{KT})$ lower bound for the regret of K-armed bandit algorithms against an oblivious adversary implies a lower bound $\tau(j, \delta) = \Omega(j/\delta^2)$ for the convergence time of anytime bandit algorithms for $\mathbb{N}$; similarly it implies a lower bound

$\tau(\varepsilon, \delta) = \Omega(1/\varepsilon\delta^2)$ for the convergence time of anytime bandit algorithms for a probability space $(\mathcal{S}, \mu)$. Hence the definition of "perfect anytime bandit algorithm" ensures that the convergence time of such an algorithm is optimal up to a factor of $\mathrm{poly}(\log(j), 1/\delta)$ or $\mathrm{poly}(\log(1/\varepsilon), 1/\delta))$.

6.2 Equivalence of the definitions

Theorem 6.1. *The following are equivalent:*

1. *There is an anytime bandit algorithm for* $\mathbb{N}$.
2. *For all probability spaces* $(\mathcal{S}, \mu)$*, there is an anytime bandit algorithm for* $(\mathcal{S}, \mu)$.

Moreover, the two conclusions remain equivalent with "perfect anytime bandit algorithm" in place of "anytime bandit algorithm".

Proof. **(1) ⇒ (2):** Assume that there is an anytime bandit algorithm $\mathsf{ALG}_{\mathbb{N}}$ for $\mathbb{N}$ with convergence time $\tau(j, \delta)$. Given a probability space $(\mathcal{S}, \mu)$, we implement an anytime bandit algorithm ALG_μ for $(\mathcal{S}, \mu)$ as follows. At initialization time, the algorithm samples an infinite sequence $x_1, x_2, x_3, \ldots$ of elements of $\mathcal{S}$ by drawing independent samples from the distribution μ. Next, ALG_μ simulates algorithm $\mathsf{ALG}_{\mathbb{N}}$, choosing strategy x_j every time $\mathsf{ALG}_{\mathbb{N}}$ chooses a strategy $j \in \mathbb{N}$. (Of course, in an actual implementation of ALG_μ, one need not perform an infinite amount of computation at initialization time. Instead, the samples $x_1, x_2, \ldots$ can be determined by "lazy evaluation": whenever $\mathsf{ALG}_{\mathbb{N}}$ decides to a choose a strategy $j \in \mathbb{N}$ which has not been chosen before, ALG_μ draws a new sample $x_j \in \mathcal{S}$ from distribution μ.)

If $\mathsf{ALG}_{\mathbb{N}}$ has convergence time $\tau(j, \delta)$, we claim that ALG_μ has convergence time

$$\tau^*(\varepsilon, \delta) = \tau\left(\left\lceil \frac{1}{\varepsilon} \log\left(\frac{2}{\delta}\right)\right\rceil,\ \frac{\delta}{2}\right).$$

To see this, let T be any integer greater than $\tau^*(\varepsilon, \delta)$, and for $\theta \in [0, 1]$ let

$$U_\theta = \left\{ x \in \mathcal{S} \ :\ \frac{1}{T}\sum_{t=1}^{T} c_t(x) > \theta \right\}$$

denote the set of strategies whose average cost exceeds θ. This is a measurable subset of $\mathcal{S}$, so we may define

$$\begin{aligned} \theta^* &= \inf\{\theta \ :\ \mu(U_\theta) < 1 - \varepsilon\} \\ U &= \bigcap_{\theta < \theta^*} U_\theta \\ V &= U_{\theta^*} = \bigcup_{\theta > \theta^*} U_\theta. \end{aligned}$$

Note that $V \subseteq U$ and

$$\mu(V) \leq 1 - \varepsilon \leq \mu(U).$$

Now let $j = \lceil (1/\varepsilon) \log(2/\delta) \rceil$, and let $\mathcal{E}$ denote the event that $\{x_1, x_2, \ldots, x_j\}$ is a subset of V. If $\{x_1, x_2, \ldots, x_j\}$ is *not* a subset of V then for some $i \in \{1, 2, \ldots, j\}$ we have $\frac{1}{T}\sum_{t=1}^{T} c_t(x_i) \leq \theta^*$ and consequently, for all $x \in U$, $\sum_{t=1}^{T} c_t(x_i) \leq \sum_{t=1}^{T} c_t(x)$. Hence, for any $x \in U$,

$$\begin{aligned}
\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) - c_t(x)\right] &= \Pr(\mathcal{E})\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) - c_t(x) \;\middle\|\; \mathcal{E}\right] \\
&\quad + (1 - \Pr(\mathcal{E}))\,\mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) - c_t(x) \;\middle\|\; \overline{\mathcal{E}}\right] \\
&\leq \Pr(\mathcal{E}) + \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) - c_t(x) \;\middle\|\; \overline{\mathcal{E}}\right] \\
&\leq \Pr(\mathcal{E}) + \max_i \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) - c_t(x_i) \;\middle\|\; \overline{\mathcal{E}}\right] \qquad (6.1)
\end{aligned}$$

We claim each term on the right side of (6.1) is less than $\delta/2$, from which it follows that ALG_μ is an anytime bandit algorithm for $(\mathcal{S}, \mu)$ with convergence time $\tau^*(\varepsilon, \delta)$. The fact that $\Pr(\mathcal{E}) < \delta/2$ rests on straightforward calculation:

$$\Pr(\mathcal{E}) \leq (1 - \varepsilon)^j < e^{-\varepsilon j} \leq \delta/2.$$

To see that the second term on the right side of (6.1) is at most $\delta/2$, note that by the definition of $\tau(j, \delta/2)$ we have

$$\max_i \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) - c_t(x_i) \;\middle\|\; x_1, x_2, \ldots, x_j\right] < \delta/2$$

for any values of $x_1, x_2, \ldots, x_j$. Since the event $\mathcal{E}$ depends only on the values of $x_1, x_2, \ldots, x_j$, we conclude that

$$\max_i \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c_t(x_t) - c_t(x_i) \;\middle\|\; \overline{\mathcal{E}}\right] < \delta/2,$$

as desired. Finally, note that

$$\tau(j, \delta) \leq j\,\mathrm{poly}(\log(j), 1/\delta) \implies \tau^*(\varepsilon, \delta) \leq (1/\varepsilon)\mathrm{poly}(\log(1/\varepsilon), 1/\delta),$$

which confirms that (1) $\Rightarrow$ (2) with "perfect anytime" in place of "anytime."

(2) $\Rightarrow$ (1): Define a probability distribution μ on $\mathbb{N}$ by assigning to each singleton set $\{n\}$ a probability proportional to $(n \log^2 n)^{-1}$. (This is a well-defined probability distribution because $\sum_{n=1}^{\infty}(n \log^2 n)^{-1} = C < \infty$ for some constant C.) Now let ALG be an anytime bandit algorithm for $(\mathbb{N}, \mu)$ with convergence time $\tau(\varepsilon, \delta)$. We claim ALG is also an anytime bandit algorithm for $\mathbb{N}$ with convergence time $\tau^*(j, \delta) = \tau((2Cj \log^2 j)^{-1}, \delta)$. To see this, let $\varepsilon = (2Cj \log^2 j)^{-1}$ and observe that observe that $\mu(\{x\}) > \varepsilon$ for all $x \in \{1, 2, \ldots, j\}$. By the definition of an anytime bandit algorithm for $(\mathbb{N}, \mu)$, $\overline{R}(\mathsf{ALG}, \mathcal{A}; x, T) < \delta$ whenever $\mu(\{x\}) > \varepsilon$ and $T > \tau(\varepsilon, \delta)$. Thus $\overline{R}(\mathsf{ALG}, \mathcal{A}; \{1, 2, \ldots, j\}) < \delta$ for any $T > \tau^*(j, \delta)$, as claimed. Finally, note that

$$\tau(\varepsilon, \delta) \leq (1/\varepsilon)\mathrm{poly}(\log(1/\varepsilon), 1/\delta) \implies \tau^*(j, \delta) \leq j\,\mathrm{poly}(\log j, 1/\delta),$$

which confirms that (2) $\Rightarrow$ (1) with "perfect anytime" in place of "anytime." □

6.3 Construction of anytime bandit algorithms

In this section we specify an anytime bandit algorithm satisfying Definition 6.3. In fact, the definition may be strengthened by enlarging $\mathcal{A}$ to be the set of all adaptive adversaries for $\mathbb{N}$. The algorithm uses, as a subroutine, the adversarial multi-armed bandit algorithm $\mathsf{Exp3}$ [4] which was analyzed in Section 2.5. This algorithm achieves regret $O(\sqrt{TK \log(K)})$ with strategy set $\{1, 2, \ldots, K\}$ against an adaptive adversary.

Definition 6.4 ($\mathsf{ABA}(F)$). For any increasing function $F : \mathbb{N} \to \mathbb{N}$, we define an algorithm $\mathsf{ABA}(F)$ as follows. For each $k \geq 0$, at time $F(k)$ the algorithm initializes an instance of $\mathsf{Exp3}$ with strategy set $\{1, 2, \ldots, 2^k\}$. From time $F(k)$ to $F(k+1) - 1$ it uses this instance of $\mathsf{Exp3}$ to select strategies in $\mathbb{N}$, and at the end of each trial it feeds the cost of the chosen strategy back to $\mathsf{Exp3}$.

Theorem 6.2. *Let $\mathcal{A}$ denote the set of all adaptive adversaries for strategy set $\mathbb{N}$ and cost function class $\Gamma = [0, 1]^{\mathbb{N}}$. For any $k > 0$ and any $T < F(k)$, the regret of $\mathsf{ABA}(F)$ satisfies*

$$\overline{R}(\mathsf{ABA}(F), \mathcal{A}; \{1, 2, \ldots, j\}, T) = O(F(\lceil \log_2 j \rceil)/T + \sqrt{k2^k/T}).$$

Proof. For $x \in \{1, 2, \ldots, j\}$ and $\mathsf{ADV} \in \mathcal{A}$, we will prove that

$$\mathbf{E}\left[\sum_{t=1}^{T} c_t(x_t) - c_t(x)\right] \leq F(\lceil \log_2 j \rceil) + O(\sqrt{k2^k T}).$$

To do so, we make use of the fact that for $i \geq \lceil \log_2 j \rceil$, strategy x belongs to the strategy set of the $\mathsf{Exp3}$ subroutine operating from time $t_0 = F(i)$ to time $t_1 - 1 =$

$\min(T, F(i+1)-1)$. This strategy set has cardinality $K = 2^i$, so the regret bound for Exp3 guarantees that

$$\mathbf{E}\left[\sum_{t=t_0}^{t_1-1} c_t(x_t) - c_t(x)\right] = O\left(\sqrt{K\log(K)(t_1-t_0)}\right) = O\left(\sqrt{i2^iT}\right).$$

$$\begin{aligned}
\mathbf{E}\left[\sum_{t=1}^{T} c_t(x_t) - c_t(x)\right] &= \sum_{i=1}^{k-1} \mathbf{E}\left[\sum_{t=F(i)}^{\min(T,F(i+1)-1)} c_t(x_t) - c_t(x)\right] \\
&\leq \sum_{i<\lceil\log_2 j\rceil} \sum_{t=F(i)}^{F(i+1)-1} 1 \\
&\quad + \sum_{\lceil\log_2 j\rceil \leq i < k} \mathbf{E}\left[\sum_{t=F(i)}^{\min(T,F(i+1)-1)} c_t(x_t) - c_t(x)\right] \\
&\leq F(\lceil\log_2 j\rceil) + \sum_{i=1}^{k-1} O\left(\sqrt{i2^iT}\right) \\
&= F(\lceil\log_2 j\rceil) + O\left(\sqrt{k2^kT}\right).
\end{aligned}$$

□

Corollary 6.3. *For any $\alpha > 0$, there exists an algorithm* ABA *which is anytime bandit algorithm for $\mathbb{N}$, with regret and convergence time satisfying*

$$\begin{aligned}
\overline{R}(\mathsf{ABA}, \mathcal{A}; \{1,2,\ldots,j\}, T) &= O\left(\frac{j^{1+\alpha}}{T} + \sqrt{T^{-\frac{\alpha}{1+\alpha}}\log(T)}\right) \\
\tau(j,\delta) &= O\left(\frac{j^{1+\alpha}}{\delta} + \left(\frac{\log(j/\delta)}{\delta^2}\right)^{1+1/\alpha}\right)
\end{aligned}$$

Proof. Let $F(k) = \lceil 2^{(1+\alpha)k}\rceil$, let $\mathsf{ABA} = \mathsf{ABA}(F)$, and apply Theorem 6.2. □

Corollary 6.4. *There exists an algorithm* ABA *which is an anytime bandit algorithm $\mathbb{N}$, with regret satisfying*

$$\overline{R}(\mathsf{ABA}, \mathcal{A}; \{1,2,\ldots,j\}, T) = O\left(j\log(T)/T^{1/3}\right).$$

Proof. Setting $\alpha = 2$ in the preceding corollary, we obtain an algorithm whose regret satisfies

$$\overline{R}(\mathsf{ABA}, \mathcal{A}; \{1,2,\ldots,j\}, T) = O\left(j^3/T + \log(T)/T^{1/3}\right).$$

Trivially, the regret also satisfies

$$\overline{R}(\mathsf{ABA}, \mathcal{A}; \{1, 2, \ldots, j\}, T) \leq 1.$$

To prove the corollary, it suffices to prove that for all sufficiently large j, T,

$$j \log(T)/T^{1/3} \geq \min\{1, j^3/T + \log(T)/T^{1/3}\}.$$

Assume, to the contrary, that $j \log(T)/T^{1/3} < 1$ and that $j \log(T)/T^{1/3} < j^3/T + \log(T)/T^{1/3}$. Rearranging terms in the second inequality, we obtain

$$T^{2/3} \log(T) < \frac{j^3}{j-1}$$

while the first inequality implies

$$\frac{\log^2(T)}{T^{2/3}} < \frac{1}{j^2}.$$

Multiplying these two together, we obtain

$$\log^3(T) < \frac{j}{j-1},$$

which is not possible for sufficiently large j, T. □

Corollary 6.5. *There exists an algorithm* ABA *which is an anytime bandit algorithm for* $\mathbb{N}$*, with regret and convergence time satisfying*

$$\begin{aligned} \overline{R}(\mathsf{ABA}, \mathcal{A}; \{1, 2, \ldots, j\}, T) &= O(j \log^3(j)/T + 1/\log(T)) \\ \tau(j, \delta) &= O\left(j \log^3(j)/\delta + 2^{O(1/\delta)}\right). \end{aligned}$$

Proof. Let $F(k) = k^3 2^k$, let $\mathsf{ABA} = \mathsf{ABA}(F)$, and apply Theorem 6.2. □

6.4 Non-existence of perfect anytime bandit algorithms

In the preceding section we saw that anytime bandit algorithms for $\mathbb{N}$ can achieve convergence time $O(j^{1+\alpha} \operatorname{poly}(1/\delta))$ for arbitrarily small positive constants α, and that they can also achieve convergence time $O(j \operatorname{polylog}(j) 2^{O(1/\delta)})$. Given these positive results, it is natural to wonder whether one can achieve convergence time $O(j \operatorname{polylog}(j) \operatorname{poly}(1/\delta))$, i.e. whether a perfect anytime bandit algorithm exists. This question is answered negatively by the following theorem.

Theorem 6.6. *Let d be any positive integer. There does not exist an anytime bandit algorithm for $\mathbb{N}$ achieving convergence time $\tau(j,\delta) = O(j\,\log^d(j)\,\delta^{-d})$.*

Proof. Assume, by way of contradiction, that ALG is an algorithm with convergence time $\tau(j,\delta) < Cj\log^d(j)\delta^{-d}$. We will consider the algorithm's regret against an oblivious adversary ADV who supplies an input instance in which all cost functions c_t are equal to a single random cost function c, defined as follows. Let r_k $(1 \le k < \infty)$ be independent random variables, where r_k is uniformly distributed over the set $\{2^{2^{k-1}}+1, 2^{2^{k-1}}+2, \ldots, 2^{2^k}\} \times \{0,1\}$. Let $c(1) = 1/4$, and define $c(j)$ for $j \ge 2$ as follows: let $k = \lceil \log_2(\log_2(j)) \rceil$ and put

$$c(j) = \begin{cases} 2^{-k} & \text{if } r_k = (j,1) \\ 1 & \text{otherwise.} \end{cases}$$

In other words, with probability $1/2$ the cost of every element in the set $\{2^{2^{k-1}}+1, \ldots, 2^{2^k}\}$ is equal to 1, and with probability $1/2$ there is a uniformly distributed random element of this set with cost 2^{-k}, and all others have cost 1.

Presented with this input, the algorithm ALG will select a random sequence of strategies $x_1, x_2, \ldots, x_T$. Let us say that the algorithm performs a *probe* at time t if this is the first time that it samples x_t, i.e. $x_t \notin \{x_1, x_2, \ldots, x_{t-1}\}$. Let q_t be the Bernoulli random variable

$$q_t = \begin{cases} 1 & \text{if ALG performs a probe at time } t \\ 0 & \text{otherwise} \end{cases}$$

and let $Q = \sum_{t=1}^{T} q_t$ denote the random variable which counts the number of probes up to time T. We will frequently need to use the following fact.

Claim 6.7. $\Pr(\min_{1\le t\le T} c(x_t) \le 2^{-k} \,\|\, Q) \;\le\; Q \Big/ \left(2^{2^k} - 2^{2^{k-1}}\right)$.

Proof. For $x > 0$, let $\mathrm{loog}(x) = \lceil \log_2(\log_2(x)) \rceil$ and let $r(x) = 2^{2^{\mathrm{loog}(x)}} - 2^{2^{\mathrm{loog}(x)-1}}$ denote the number of strategies in the set $\{2^{2^{\mathrm{loog}(x)-1}}+1, \ldots, 2^{2^{\mathrm{loog}(x)}}\}$. Let $\tau_1 < \tau_2 < \ldots < \tau_Q$ denote the numbers of the trials in which the algorithm performs its Q probes. For $0 \le s < Q$,

$$\begin{aligned} \Pr\left(c(x_{\tau_{s+1}}) > 2^{-k} \,\|\, c(x_{\tau_1}), c(x_{\tau_2}), \ldots, c(x_{\tau_s})\right) &\ge \begin{cases} 1 - \frac{1}{r(x)-s} & \text{if } x > 2^{2^{k-1}} \\ 1 & \text{otherwise} \end{cases} \\ &\ge 1 - \frac{1}{2^{2^k} - 2^{2^{k-1}} - s}. \end{aligned}$$

Hence

$$\Pr\left(\min_{1\le t\le T} c(x_t) > 2^{-k}\right) \ge \prod_{s=0}^{Q-1} \left(1 - \frac{1}{2^{2^k} - 2^{2^{k-1}} - s}\right) = 1 - \frac{Q}{2^{2^k} - 2^{2^{k-1}}},$$

which establishes the claim. □

Resuming the proof of Theorem 6.6, put $T = 2^{2^k+3dk}$. We distinguish two cases.

Case 1: $\Pr\left(Q > \frac{1}{2}\left(2^{2^k} - 2^{2^{k-1}}\right)\right) < 3/4$.

Case 2: $\Pr\left(Q > \frac{1}{2}\left(2^{2^k} - 2^{2^{k-1}}\right)\right) \geq 3/4$.

In Case 1, let $j = 2^{2^k}$, $\delta = 2^{-k-5}$. For sufficiently large k,

$$\tau(j,\delta) = Cj\log^d(j)\delta^{-d} = 2^{2^k+2dk+5d+\log_2(C)} < T.$$

We will prove that $\overline{R}(\mathsf{ALG},\mathsf{ADV};\{1,2,\ldots,j\},T) > \delta$, thus obtaining a contradiction. Consider the following three events.

$$\begin{aligned}
\mathcal{E}_1 &= \left\{Q \leq \frac{1}{2}\left(2^{2^k} - 2^{2^{k-1}}\right)\right\} \\
\mathcal{E}_2 &= \left\{\min_{1\leq t\leq T} c(x_t) \geq 2^{-(k-1)}\right\} \\
\mathcal{E}_3 &= \left\{\min_{1\leq x\leq 2^{2^k}} c(x) = 2^{-k}\right\}
\end{aligned}$$

By assumption, $\Pr(\mathcal{E}_1) > 1/4$. Claim 6.7 establishes that $\Pr(\mathcal{E}_2 \,\|\, \mathcal{E}_1) \geq 1/2$. Next we argue that $\Pr(\mathcal{E}_3 \,\|\, \mathcal{E}_1 \wedge \mathcal{E}_2) \geq 1/3$. Let $U = \{2^{2^{k-1}}+1,\ldots,2^{2^k}\}$, and let V denote the intersection of U with $\{x_1,x_2,\ldots,x_T\}$. Conditional on $\mathcal{E}_1$, $|V| \leq |U|/2$, and conditional on $\mathcal{E}_2$, r_k is uniformly distributed in the set $U\times\{0,1\}\setminus V\times\{1\}$. Hence the probability is at least $1/3$ that $r_k \in (U\setminus V)\times\{1\}$, which implies $\mathcal{E}_3$.

Putting this all together,

$$\Pr(\mathcal{E}_1\wedge\mathcal{E}_2\wedge\mathcal{E}_3) > 1/24$$

Assuming $\mathcal{E}_2$ and $\mathcal{E}_3$, there exists a strategy $x \in \{1,2,\ldots,j\}$ such that

$$\frac{1}{T}\sum_{t=1}^{T} c(x_t) - c(x) \geq 2^{-k} = 32\delta.$$

Thus,

$$\overline{R}(\mathsf{ALG},\mathsf{ADV};\{1,2,\ldots,j\},T) = \mathbf{E}\left[\frac{1}{T}\sum_{t=1}^{T} c(x_t) - c(x)\right] \geq 32\delta\Pr(\mathcal{E}_2\wedge\mathcal{E}_3) > \delta,$$

as claimed.

In Case 2, let $j = 2^{2^{k-1}}$ and $\delta = 2^{-2^{k-1}/d}$. Note that $\tau(j,\delta) < T$ provided that k is sufficiently large. Letting $\overline{\mathcal{E}}$ denote the complement of an event $\mathcal{E}$, we have $\Pr(\overline{\mathcal{E}_1}) \geq 3/4$ by assumption, and we have

$$\Pr(\overline{\mathcal{E}_3}) = \Pr\left(r_k \in \{2^{2^{k-1}}+1, \ldots, 2^{2^k}\} \times \{0\}\right) = \frac{1}{2};$$

hence $\Pr(\overline{\mathcal{E}_1} \wedge \overline{\mathcal{E}_3}) \geq 1/4$. Let

$$\mathcal{E}_4 = \left\{\min_{1\leq t\leq T} c(x_t) \geq 2^{-k}\right\}.$$

By Claim 6.7,

$$\Pr(\overline{\mathcal{E}_4}) \;\leq\; T \Big/ \left(2^{2^{k+1}} - 2^{2^k}\right) = o(1),$$

so for k sufficiently large

$$\Pr(\overline{\mathcal{E}_1} \wedge \overline{\mathcal{E}_3} \wedge \mathcal{E}_4) > 1/8.$$

Let $x = \arg\min_{1\leq i\leq j} c(i)$. Assuming $\overline{\mathcal{E}_3}$ and $\mathcal{E}_4$, we have $c(x_t) \geq c(x)$ for $t = 1, 2, \ldots, T$. Moreover, assuming $\mathcal{E}_4$, there are at least $Q - k - 1$ probes with cost 1, so

$$\frac{1}{T}\sum_{t=1}^{T} c(x_t) - c(x) \geq \frac{1}{T}(Q-k-1)(1-c(x)) \geq \frac{Q-k-1}{2T}.$$

Assuming $\overline{\mathcal{E}_1}$ and assuming k is sufficiently large,

$$\frac{Q-k-1}{2T} > 2^{-4dk} > 2^{3-2^{k-1}/d} = 8\delta,$$

hence

$$\overline{R}(\mathsf{ALG}, \mathsf{ADV}; \{1,2,\ldots,j\}, T) \geq \frac{Q-k-1}{2T}\Pr(\overline{\mathcal{E}_1} \wedge \overline{\mathcal{E}_3} \wedge \mathcal{E}_4) > \delta,$$

contradicting the assumption that ALG is an anytime bandit algorithm with convergence time $\tau(j,\delta)$. □

Chapter 7

Collaborative learning

7.1 Introduction

This chapter proposes and analyzes a multiple-agent version of the multi-armed bandit problem which aims to model the challenges facing users of collaborative decision-making systems such as reputation systems in e-commerce, collaborative filtering systems, and resource location systems for peer-to-peer networks. As explained in Section 1.7, our approach is motivated by consideration of the following issues which are common to the applications cited above:

Malicious users. Since the Internet is open for anybody to join, the above systems are vulnerable to fraudulent manipulation by dishonest ("Byzantine") participants.

Distinguishing tastes. Agents' tastes may differ, so that the advice of one honest agent may not be helpful to another.

Temporal fluctuation. The quality of resources varies of time, so past experience is not necessarily predictive of future performance.

We model these problems using a theoretical framework which generalizes the multi-armed bandit problem by considering a set X of n *agents*, some of which (possibly a majority) may be dishonest, and a set Y of m *resources* which agents are allowed to choose. In each trial, each of the n agents chooses a resource, and the adversary chooses a cost for each resource. Each agent then learns the cost of the resource it selected, and this cost is charged to the agent. (The classical multi-armed bandit problem corresponds to the special case $n = 1$.) We assume that the honest agents belong to k *coalitions*, such that agents in the same coalition who choose the same resource at the same time will perceive the same expected cost. All agents may

communicate with each other between trials, to exchange information (or possibly disinformation, in the case of dishonest agents) about the costs of resources they have sampled. However, agents are unaware of which coalitions exist and which ones they belong to.

If an agent chooses to ignore the feedback from other agents, and simply runs the multi-armed bandit algorithm by itself, then the algorithm's convergence time is $\Omega(m \log m)$, i.e. for any constant $\delta > 0$, if $T = \Omega(m \log m)$, then the expected average cost of the resources chosen by that agent will exceed the average cost of the best resource in hindsight by no more than δ. However, it is possible that the honest agents may require much fewer than $\Omega(m \log m)$ trials to achieve this goal, if they can find a way to pool their information without being fooled by the bad advice from dishonest agents and agents from other coalitions. Here we show that this is in fact possible, by presenting an algorithm whose convergence time is polynomial in $k \log(n)$, assuming that a constant fraction of the agents are honest and that $m = O(n)$.

Briefly, our algorithm works by having each agent select a resource in each trial by taking a random walk on a "reputation network" whose vertex set is the set of all agents and resources. Resources are absorbing states of this random walk, while the transition probabilities at an agent x may be interpreted as the probability that x would select a given resource y, or would ask a given other agent x' for advice. When an agent learns the cost of the resource chosen in a given trial, it uses this feedback to update its transition probabilities according to the multi-armed bandit algorithm. In this way, agents will tend to raise the probability of asking for advice from other agents who have given good advice in the past. In particular, though the initial transition probabilities do not reflect the partition of the honest agents into coalitions, over time the honest agents will tend to place greater weight on edges leading to other agents in the same coalition, since the advice they receive from such agents is generally better, on average, than the advice they receive from agents in other coalitions or from dishonest agents.

The rest of this chapter is organized as follows. In Section 7.2 we specify our precise models and results. The collaborative learning algorithm, TrustFilter, appears in Section 7.3. In Section 7.4, we analyze the algorithm, modulo a random graph lemma which is proved in Section 7.5.

7.2 Statement of the Problem and the Results

We study a collaborative learning problem involving a set X of n agents and a set Y of m resources. A subset $H \subseteq X$ of the agents are *honest*, and the rest are dishonest. Honest agents are assumed to obey the distributed protocol to be specified, and to

report their observations truthfully, while dishonest agents may behave in a Byzantine manner, disobeying the protocol or reporting fictitious observations as they wish. We will assume throughout that the number of honest agents is at least αn, where $\alpha > 0$ is a parameter which may be arbitrarily small. The agents do not initially know which ones are honest and which are dishonest, nor are they assumed to know the value of α.

In each of T consecutive rounds, a cost function $c_t : X \times Y \to [0, 1]$ is given. We think of the cost $c_t(x, y)$ as agent x's perception of the cost of resource y. The costs may be set by an adaptive adversary who is allowed to choose c_t based on the agents' actions in rounds $1, \ldots, t-1$ but not on their random decisions in the present or future rounds; the adversary may also use randomization in determining c_t. Define two agents x_1, x_2 to be *consistent* if the costs $c_t(x_1, y), c_t(x_2, y)$ are random variables with the same expected value (conditional on the choices of all agents in all rounds preceding t), for all $y \in Y, 1 \leq t \leq T$. [1] We will assume that the honest agents may be partitioned into k *coalitions*, such that two agents belonging to the same coalition are consistent; the honest agents do not initially know which coalitions the other honest agents belong to.

At the beginning of each round, each agent $x \in X$ must choose a resource $y = y_t(x) \in Y$. Any agent is allowed to choose any resource in any round. The cost of the choice is $c_t(x, y)$, and this cost (but not the cost of any other resource) is revealed to x. The agents may communicate with each other between rounds, and this communication may influence their decisions in future rounds. To simplify the exposition we will assume all messages are exchanged using a shared, synchronous, public channel. In any round t all agents (including the Byzantine dishonest agents) must commit to their message on this channel before being able to read the messages posted by other agents in round t.

The goal of the algorithm is to minimize the total cost incurred by honest agents. Generalizing the definition of regret from Section 1.1, here the *regret* R is defined as the expected difference between this cost and the cost of best fixed strategy in S, i.e.

$$R = \mathbf{E}\left[\sum_{x \in H}\sum_{t=1}^{T} c_t(x, y_t(x)) - \min_{y:H\to Y}\sum_{x \in H}\sum_{t=1}^{T} c_t(x, y(x))\right]. \tag{7.1}$$

The following two parameters, closely related to R, are also of interest:

- The *normalized individual regret* $\overline{R} = R/\alpha n T$ is the regret per unit time of the average honest agent. For all of the algorithms we will consider, $\overline{R}$ converges to zero as $T \to \infty$.

[1] The randomness of the variables $c_t(x_1, y), c_t(x_2, y)$ is due to the adversary's potential use of randomness in determining c_t.

- The δ-*convergence time* of such an algorithm, denoted by $\tau(\delta)$, is defined as the minimum value of T necessary to guarantee that $\overline{R} = O(\delta)$. Here, δ is a positive constant which may be arbitrarily close to zero.

7.2.1 Our results

We present a distributed algorithm, named TrustFilter, in Section 7.3. Let $\beta = 1 + m/n$. We will typically be interested in the case where α, β, δ are all positive constants. For ease of exposition, we will adhere to this assumption when stating the theorems in this section, absorbing such constants into the $O(\cdot)$ notation. See equations (7.8),(7.9),(7.10), (7.11) in Section 7.4 for bounds which explicitly indicate the dependence on α, β, and δ; in all cases, this dependence is polynomial.

Theorem 7.1. *Suppose the set of honest agents may be partitioned into k subsets $S_1, S_2, \ldots, S_k$, such that the agents in each subset are mutually consistent. Then the normalized regret $\overline{R}$ and δ-convergence time $\tau(\delta)$ of* TrustFilter *satisfy*

$$\overline{R} = O\left(k \cdot \frac{\log n \log T}{T^{1/3}}\right) \tag{7.2}$$

$$\tau(\delta) = O(k^3 \log^3 n \log^3(k \log n)). \tag{7.3}$$

The δ-convergence time bound follows from the regret bound. Typically we are interested in the case where α, β, δ, k are constants, hence we will summarize this result by saying that the algorithm has *polylogarithmic convergence time.*

7.3 The Algorithm TrustFilter

7.3.1 Intuition

As stated in the introduction, our algorithm is based on a Markov chain representing a random walk in a directed graph, whose vertices represent the set of resources and agents. We refer to this directed graph as the "reputation network." At each time, each agent picks an outgoing edge in the reputation network with appropriate probability, and then traverses this edge. If the edge leads to an agent, "advice" is sought from that agent. Else, if the edge leads to a resource, this resource is selected for sampling. Depending on the observed cost of the sampled resource, the agent updates its transition probabilities.

As an aid in developing intuition, consider the special case when the costs of resources are $\{0, 1\}$-valued and do not change over time. In this case, one may use a simpler algorithm in which the Markov chain is based on a random graph. Specifically,

each agent picks at random a small subset of other agents and a small subset of the resources, takes their union, and sets equal transition probabilities on all outgoing edges leading to members of this set. All other outgoing edge probabilities are zero. Assume that agents adopt the following simple rule for updating their transition probabilities: if an agent chooses an outgoing edge and ends up selecting a resource with cost 0, it assigns probability 1 permanently to that resource and probability 0 to all other edges; otherwise it leaves the transition probabilities unchanged. This algorithm can be viewed as an alternative to the Random Advice Random Sample algorithm in [10]. It is easy to prove that it achieves logarithmic convergence time. The invariant used in the proof is the fact that the set of agents who recommend a resource with cost 0 grows exponentially with time, assuming there exists at least one resource with cost 0. This invariant is proved by induction on time. Indeed, with high probability there is an edge in the reputation network from some honest agent to a resource with cost 0, and in constant time that neighboring agent will either directly sample this resource, or will stumble on an equivalent resource following the advice of others. Consider the set S of honest agents who discovered a resource with cost 0. Note that the set N of neighbors of S (namely, nodes with outgoing edges leading into S) satisfies $|N| \geq |S| \cdot \rho$ where ρ is the expansion ratio of the underlying random graph. Note that within constant time in expectation, a constant fraction of agents in N will also discover a resource with cost 0 by sampling nodes in S or following advice to other equivalent resources. Thus, within logarithmic time in expectation, all the agents discover a resource with cost 0.

Our algorithm for the case of dynamic costs looks quite different from the algorithm for static costs presented in the preceding paragraph, but it is based on the same intuition: by structuring the reputation network as a random graph, the set of honest agents who are selecting an optimal or near-optimal resource will grow exponentially over time. The main technical difference is that agents must update their transition probabilities using the multi-armed bandit algorithm, rather than shifting all of their probability mass to one outgoing edge as soon as they discover a resource with zero cost. This modification is necessary in order to deal with the fact that a resource which has zero cost at one time may not have zero cost at future times. More subtly, when agents are using the multi-armed bandit algorithm to update their transition probabilities, they must use an *anytime bandit algorithm* as defined in Chapter 6. This is because the agents do not know how many other honest agents belong to their coalition, so they must consider all βn other vertices of the reputation network as potential neighbors. (Recall from Section 7.2 that $\beta = (m+n)/n$, so that βn is the cardinality $X \cup Y$, the vertex set of the reputation network.) Classical multi-armed bandit algorithms, e.g. Exp3 [4], will have a convergence time of $\Omega(n \log(n))$ in such

a scenario, whereas we seek a polylogarithmic convergence time. Accordingly, we use an anytime bandit algorithm ABA whose salient feature is that it satisfies a significantly better bound on regret when stopped at times $T < n\log(n)$. The algorithm ABA which we will use in this chapter is the one identified in Corollary 6.4.

7.3.2 The algorithm

Here we present an algorithm TrustFilter which solves the collaborative learning problem, establishing Theorem 7.1. We use, as a subroutine, the algorithm ABA whose existence is asserted by Corollary 6.4. The internal workings of the algorithm are unimportant for present purposes; the reader may consider it as a black box (instantiated separately by each agent x) which outputs, at each time t, a probability distribution $\pi_t(x)$ on the set of all agents and resources. We will use the notation $\pi_t(x, y)$ to denote the probability that $\pi_t(x)$ assigns to the element $y \in X \cup Y$.

At initialization time, each agent x initializes an instance of ABA, mapping the elements of $X \cup Y$ to the first βn elements of $\mathbb{N}$ using a random permutation, and associating an arbitrary element of Y to each remaining element of $\mathbb{N}$. At the beginning of each round t, each agent x queries its local bandit algorithm $\mathsf{ABA}(x)$ to obtain a probability distribution $\pi_t(x)$ on the set of agents and resources, and posts this distribution on the public channel. This enables each agent to construct an $(m+n)$-by-$(m+n)$ matrix M_t whose rows and columns are indexed by the elements of $X \cup Y$, and whose entries are given by:

$$(M_t)_{ij} = \begin{cases} \pi_t(i,j) & \text{if } i \in X \\ 1 & \text{if } i \in Y \text{ and } j = i \\ 0 & \text{if } i \in Y \text{ and } j \neq i. \end{cases}$$

We may think of M_t as the transition matrix for a Markov chain with state space $X \cup Y$, in which elements of Y are absorbing states, and the transition probabilities at an element x of X are determined by the bandit algorithm $\mathsf{ABA}(x)$. This Markov chain corresponds to the intuitive notion of "taking a random walk by following the advice of the bandit algorithm at each node."

The random walk starting from $x \in X$ will be absorbed, with probability 1, by some state $y \in Y$; this enables us to define a matrix A_t by

$$(A_t)_{ij} = \Pr(\text{absorbing state is } j \,|\, \text{starting state is } i).$$

Algebraically, A_t satisfies the equations $M_t A_t = A_t$ and $A_t \mathbf{1} = \mathbf{1}$, where $\mathbf{1}$ represents a column vector whose components are all equal to 1.

To select a resource $y = y_t(x) \in Y$, x uses $\mathsf{ABA}(x)$ to choose a strategy $s = s_t(x) \in X \cup Y$. It then samples y randomly using the probability distribution in the row of A_t

corresponding to s, learns the cost $c_t(y)$, and returns this feedback score to $\mathsf{ABA}(x)$. The probability distribution from which y is drawn can be determined either by computing A_t algebraically, or by simulating the random walk with transition matrix M_t starting from state s until it hits an absorbing state. We call this probability distribution on Y *harmonic measure relative to* x, by analogy with the harmonic measure defined on the boundary of a bounded domain $U \subset \mathbb{R}^d$ according the hitting probability of Brownian motion starting from a point $x \in U$.

7.4 Analysis of Algorithm TrustFilter

In this section we analyze algorithm TrustFilter by proving Theorem 7.1. Before proving this theorem, it is necessary to extend the analysis of ABA to a more general feedback model which we call the "noisy feedback model." This generalization is described as follows. In each round t, instead of specifying one random cost function $c_t \in \Gamma$, the adversary specifies *two* random cost functions $c_t, c'_t \in \Gamma$ satisfying

$$\forall x \in \mathcal{S}\ \mathbf{E}[c'_t(x) - c_t(x) \,\|\, \mathcal{F}_{<t}] = 0,$$

where $\mathcal{F}_{<t}$ denotes the σ-field generated by all random variables revealed by the algorithm and adversary prior to time t. Rather than receiving $c_t(x_t)$ as feedback, the algorithm's feedback is $c'_t(x_t)$. However, the cost charged to the algorithm as well as its regret are still defined in terms of the cost functions c_t rather than c'_t. The following easy proposition demonstrates that the regret of algorithm ABA is unaffected by the noisy feedback.

Proposition 7.2. *In the noisy feedback model, the regret R experienced by algorithm* ABA *relative to strategy j still satisfies*

$$\overline{R}(\mathsf{ABA}, \mathcal{A}; \{1, 2, \ldots, j\}, T) \;=\; O\left(j \log(T)/T^{1/3}\right).$$

Proof. Applying Corollary 6.4 to the sequence of cost functions $c'_1, c'_2, \ldots, c'_T$, gives that

$$\frac{1}{T}\mathbf{E}\left(\sum_{t=1}^{T} c'_t(x_t) - \sum_{t=1}^{T} c'_t(x)\right) = O(j \log(T)/T^{1/3})$$

for all $x \in \{1, 2, \ldots, j\}$. To finish proving the proposition, it suffices to prove that

$$\mathbf{E}\left(\sum_{t=1}^{T} c'_t(x) - \sum_{t=1}^{T} c_t(x)\right) = 0$$

and

$$\mathbf{E}\left(\sum_{t=1}^{T} c'_t(x_t) - \sum_{t=1}^{T} c_t(x_t)\right) = 0.$$

These follow from the fact that $\mathbf{E}(c'_t(x)) = \mathbf{E}(c_t(x))$ and $\mathbf{E}(c'_t(x_t)) = \mathbf{E}(c_t(x_t))$, both of which are consequences of the equation $\mathbf{E}(c_t(x') - c'_t(x') \| x_t, \mathcal{F}_{<t}) = c_t(x')$, which holds for all $x' \in S$. □

Proof of Theorem 7.1. For $x \in X, s \in X \cup Y$, let

$$\tilde{c}_t(x,s) = \begin{cases} c_t(x,s) & \text{if } s \in Y \\ \mathbf{E}[c_t(x, y_t(s))] & \text{if } s \in X. \end{cases}$$

From the standpoint of agent x, the bandit algorithm $\mathsf{ABA}(x)$ is running in the noisy feedback model with cost functions $\tilde{c}_t(x,\cdot)$ and random feedback variables $c'_t(s)$ distributed according to the cost $(c_t(x,y))$ of a random resource $y \in Y$ sampled according to the harmonic measure relative to s. For $u \in H, v \in X \cup Y$, define $\ell(u,v)$ to be the position of v in the random permutation selected by u when initializing its bandit algorithm $\mathsf{ABA}(u)$. It follows from the Proposition 7.2 that for each $u \in H$ and $v \in X \cup Y$,

$$\frac{1}{T}\mathbf{E}\left(\sum_{t=1}^{T}(\tilde{c}_t(u,u) - \tilde{c}_t(u,v))\right) = O\left(\ell(u,v)\log(T)/T^{1/3}\right). \tag{7.4}$$

Using the fact that $\tilde{c}(u,v) = \tilde{c}(v,v)$ when u, v are consistent, we may rewrite (7.4) as

$$\frac{1}{T}\mathbf{E}\left[\left(\sum_{t=1}^{T}\tilde{c}_t(u,u)\right) - \left(\sum_{t=1}^{T}\tilde{c}_t(v,v)\right)\right] = O\left(\ell(u,v)\log(T)/T^{1/3}\right), \tag{7.5}$$

provided that u and v are consistent. For $u \in H$, let

$$\bar{c}(u) = \mathbf{E}\left(\frac{1}{T}\sum_{t=1}^{T}\tilde{c}_t(u,u)\right).$$

Then (7.5) may be rewritten as

$$\bar{c}(u) - \bar{c}(v) = \ell(u,v) \cdot O(\log(T)/T^{1/3}). \tag{7.6}$$

Note that for a resource $y \in Y$, $\bar{c}(y)$ is simply the average cost of y, and for an agent $x \in H$, $\bar{c}(x)$ is the average cost of the resources sampled by x. Let S be a consistent cluster containing x, and let $\alpha(S) = |S|/n$. Letting y^* denote a resource with minimum average cost for members of S, and letting P denote a shortest path from x to y^* in the directed graph with vertex set $S \cup Y$ and edge lengths given by $\ell(\cdot,\cdot)$, we may sum up the bounds (7.6) over the edges of P to obtain

$$\bar{c}(x) - \bar{c}(y^*) = O(\mathit{length}(P) \cdot \log(T)/T^{1/3}) \tag{7.7}$$

Observe that the left side is the expected normalized regret of agent x. The random edge lengths $\ell(u,v)$ on the $m+n$ outgoing edges of u are simply the numbers $\{1,2,\ldots,m+n\}$ in a random permutation. For graphs with random edge lengths specified according to this distribution, we analyze the expected distance between two given vertices in Section 7.5. Applying Proposition 7.3 from that section, we may conclude that the expectation of the right side of (7.7) is $O((\beta/\alpha(S))\log(n)\log(T)/T^{1/3})$. It follows that the normalized regret and δ-convergence time for agents in the cluster S satisfy

$$\overline{R} = O\left(\left(\frac{\beta}{\alpha(S)}\right)\frac{\log(n)\log(T)}{T^{1/3}}\right) \tag{7.8}$$

$$\tau(\delta) = \widetilde{O}\left(\left(\frac{\beta}{\alpha(S)\delta}\right)^3 \log^3(n)\log^3\left(\frac{\beta\log n}{\alpha(S)\delta}\right)\right). \tag{7.9}$$

Note that (7.9) can be interpreted as saying that the large consistent clusters learn to approximate the cost of the best resource much more rapidly than do the small clusters, which accords with one's intuition about collaborative learning. To obtain Theorem 7.1, we must average over the k consistent clusters $S_1,\ldots,S_k$. We may multiply the regret bound for a cluster S in (7.8) by the size of S, to obtain an upper bound of $O(\beta n\log n\log T/T^{1/3})$ on the sum of the normalized regret of all users in S. Summing over k such clusters, the cumulative normalized regret of all honest users is $O(k\beta n\log n\log T/T^{1/3})$, so the normalized regret and convergence time satisfy:

$$\overline{R} = O\left(k\cdot\left(\frac{\beta}{\alpha}\right)\frac{\log(n)\log(T)}{T^{1/3}}\right) \tag{7.10}$$

$$\tau(\delta) = \widetilde{O}\left(k^3\cdot\left(\frac{\beta}{\alpha\delta}\right)^3 \log^3(n)\log^3\left(\frac{\beta k\log n}{\alpha\delta}\right)\right). \tag{7.11}$$

□

7.5 A random graph lemma

Let $G=(V,E)$ denote a directed graph with vertex set $V=X\cup Y$, in which each $x\in X$ has outgoing edges to every other element of V, and each $y\in Y$ has no outgoing edges. Assign random lengths to the edges of G as follows: for each $x\in X$, the $|X|+|Y|-1$ outgoing edges from x are assigned the lengths $\{1,2,\ldots,|X|+|Y|-1\}$ in a random permutation. Let $n=|X|$, $\alpha=|X_0|/|X|$, $\beta=|X\cup Y|/|X|$.

Proposition 7.3. *For any $x\in X_0, y\in Y$, the expected length of the shortest path from x to y in $X_0\cup Y$ is $O((\beta/\alpha)\log(n))$.*

Proof. As is common in random graph theory, we'll prove that the expected shortest length is logarithmic in βn by demonstrating that the size of a ball of radius r about x grows exponentially in r. To do so, it is useful to replace the original edge lengths $\ell(u,v)$ with new random lengths $f(u,v)$ defined as follows. For each vertex $u \in X$, choose an infinite sequence $u_1, u_2, \ldots$ of i.i.d. samples from the uniform distribution on $(X \setminus \{u\}) \cup Y$, and put $f(u,v) = \min\{j \,:\, u_j = v\}$. These lengths stochastically dominate the original edge lengths in the following sense. Let $g(u,v)$ be the number of distinct elements in the set $\{u_1, u_2, \ldots, u_{f(u,v)}\}$, i.e. the number of distinct elements which appear in the sequence $u_1, u_2, \ldots$ up to and including the first occurrence of v. Then $g(u,v) \leq f(u,v)$ for all (u,v), and the lengths $g(u,v)$ ($u \in X$, $v \in (X \setminus \{u\}) \cup Y$) have the same joint distribution as $\ell(u,v)$: for a fixed u, as v ranges over $(X \setminus \{u\}) \cup Y$ the values of $g(u,v)$ compose a random permutation of $\{1, 2, \ldots, |X| + |Y| - 1\}$ and these permutations are independent for different values of u.

Let $B(x,r)$ denote the set of all elements of $X_0 \cup Y$ reachable from x by a directed path of length at most r in $X_0 \cup Y$, and let $B_0(x,r) = B(x,r) \cap X_0$. (Here, edge lengths are defined according to f rather than ℓ. Using f rather than ℓ can only decrease the expected size of $B_0(x,r)$ since f stochastically dominates ℓ.) Now define:

$$\begin{aligned} r_0 &= 1 \\ r_i &= \min\{r \geq r_{i-1} \,:\, |B_0(x,r)| \geq \min(2|B_0(x,r_{i-1})|, |X_0|/3)\} \quad (i = 1, 2, \ldots, \lg(n)) \\ s &= \min\{r \geq r_{\lceil \lg(n) \rceil} \,:\, y \in B(x,r)\}. \end{aligned}$$

Here, $\lg(\cdot)$ denotes the base-2 logarithm function. The expected length of the shortest path from x to y in $X_0 \cup Y$ is bounded above by $\mathbf{E}[s]$. We'll prove the lemma by establishing that each of the random variables $r_{i+1} - r_i$ ($0 \leq i \leq \lceil \lg(n) \rceil - 1$), as well as $s - r_{\lceil \lg(n) \rceil}$, has expectation $O(\beta/\alpha)$.

To bound the expectation of $r_{i+1} - r_i$, note first that $r_{i+1} - r_i = 0$ if $|B_0(x,r)| \geq |X_0|/3$, so assume from now on that $|B_0(x,r)| < |X_0|/3$. Let $r = r_i + \lceil \beta/\alpha \rceil$, and consider the size of the set $A = B_0(x,r) \setminus B_0(x,r_i)$. This set may be described as follows: for each $u \in B_0(x,r)$, whose distance from x in the shortest-path metric is $r_i - k$ for some k, the set A contains each element of $\{u_{k+1}, u_{k+2}, \ldots, u_{k+\lceil \beta/\alpha \rceil}\} \cap X_0$ which is not already in $B_0(x,r_i)$. We claim that there exists a constant c such that $|A| > c|B_0(x,r_i)|$ with probability at least $1/2$. To prove this, consider taking an arbitrary ordering of the vertices $u \in B_0(x,r_i)$ and associating to each such u a set $A_u \subseteq A$ of cardinality either 0 or 1, as follows. Assume A_v is already defined for each $v < u$, and once again denote the distance from u to x in the shortest-path metric by $r_i - k$. If the set

$$S_u = \{u_{k+1}, \ldots, u_{k+\lceil \beta/\alpha \rceil}\} \cap \left(X_0 \setminus \left(B_0(x,r_i) \cup \bigcup_{v<u} A_v \right) \right)$$

is non-empty then A_u is defined to be an arbitrary one-element subset of S_u, else $A_u = \emptyset$. Observe that $|A| \geq \sum_u |A_u|$ since the sets A_u are disjoint subsets of A, so it now suffices to bound from below the expected number of u satisfying $|A_u| = 1$. For each $u \in B_0(x, r_i)$, let $T_u = X_0 \setminus \left(B_0(x, r_i) \cup \bigcup_{v<u} A_v\right)$. Observe that

$$|T_u| \geq |X_0| - |X_0|/3 - |X_0|/3 \geq \alpha n/3.$$

Each of the elements $u_{k+1}, \ldots, u_{k+\lceil \beta/\alpha \rceil}$ is an independent random sample from a set of size $\beta n - 1$, so the probability that at least one of them belongs to T_u is at least

$$1 - \left(1 - \frac{\alpha n/3}{\beta n - 1}\right)^{\lceil \beta/\alpha \rceil} > 1 - \left(1 - \frac{\alpha}{3\beta}\right)^{\beta/\alpha} > 1 - e^{-1/3}.$$

Thus

$$\mathbf{E}[\, |A_u| \;\parallel\; B_0(x, r_i),\, A_v\, (v < u)] > 1 - e^{-1/3}. \tag{7.12}$$

It follows that

$$\mathbf{E}\left[\sum_u |A_u| \;\parallel\; B_0(x, r_i)\right] > (1 - e^{-1/3})|B_0(x, r_i)|$$

and that with probability at least $1/2$, the random variable $\sum_u |A_u|$ is within a constant factor of its expected value. (The latter statement follows from an exponential tail inequality for $\sum_u |A_u|$ which is justified by equation (7.12).) Thus there exists a constant c such that $|A| > c|B_0(x, r_i)|$ with probability at least $1/2$, as claimed.

If we consider initially setting $r = r_i$ and then raising r in increments of $\lceil \beta/\alpha \rceil$ until $|B_0(x, r)| \geq \min\{2|B_0(x, r_i)|, |X_0|/3\}$, we have seen that on each such increment, the probability of $|B_0(x, r)|$ increasing by a factor of at least $(1 + c)$ is at least $1/2$. Hence the expected number of increments necessary to increase $|B_0(x, r)|$ by a factor of at least $(1 + c)$ is at most 2, and the expected number of increments necessary before $|B_0(x, r)|$ reaches $\min\{2|B_0(x, r_i)|, |X_0|/3\}$ is at most $2 \log_{1+c}(2) = O(1)$. Thus $\mathbf{E}[r_{i+1} - r_i] = O(\beta/\alpha)$ as claimed.

The claim about $\mathbf{E}[s - r_{\lceil \lg(n) \rceil}]$ is even easier to prove. By construction, the set $B_0(x, r_{\lceil \lg(n) \rceil})$ contains at least $\alpha n/3$ elements. Each time we increment r above $r_{\lceil \lg(n) \rceil}$, the set $B(x, r)$ gains at least $\alpha n/3$ new independent random samples from $X \cup Y$. For a specified element $y \in Y$, the probability that at least one of the new random samples is equal to y is bounded below by

$$1 - \left(1 - \frac{1}{\beta n}\right)^{\alpha n/3} > 1 - e^{-\alpha/3\beta} \geq \frac{\alpha}{3e\beta}.$$

Thus the expected number of times we must increment r before hitting y is $O(\beta/\alpha)$. □

Chapter 8

Conclusion

We have seen that online decision problems with large strategy sets arise naturally in many application areas, including economic theory, networking, and collaborative decision systems, yet the theoretical tools which existed prior to this thesis did not imply the existence of rapidly converging algorithms in many cases of interest. In this thesis, we have proposed new algorithms and lower bound techniques for dealing with such problems, specifically generalizations of the multi-armed bandit problem. Trivially, any multi-armed bandit problem with a strategy set of size K must have convergence time $\Omega(K)$; when K is exponential in the problem size or is infinite, this seems to preclude the existence of rapidly converging algorithms. We have suggested three approaches to circumventing this negative result, while giving sample applications to illustrate the utility of each approach.

1. When the strategy set is a bounded one-parameter interval and the cost functions (or their expected values) are Lipschitz continuous, there is an efficient algorithm for the generalized bandit problem. Consequently, there are efficient online pricing algorithms.

2. When the strategy set is a compact, convex subset of a low-dimensional vector space, and the cost functions are linear or convex, there is an efficient algorithm for the generalized bandit problem. Consequently, there are efficient online shortest path algorithms.

3. When the strategy set is a measure space and the goal is to perform nearly as well as all but a small fraction of the strategies, there is an efficient algorithm for the generalized bandit problem. Consequently, there are efficient collaborative learning algorithms.

We have also introduced new lower bound techniques for generalized multi-armed bandit problems with large strategy sets, based on notions from statistics and information

theory such as Kullback-Leibler divergence and Fisher information. In particular, we introduced a definition of *knowledge* in Chapter 3 which generalizes Fisher information and supplies a quantitative measure of the trade-off between exploration and exploitation.

These results represent first steps toward understanding the capabilities and limitations of algorithms for generalized bandit problems with large strategy sets, while also suggesting many interesting directions for future work. In the following sections, we survey some of the research problems raised by the work presented in this thesis.

8.1 Open questions

8.1.1 Theoretical aspects of online decision problems

We have seen that for many interesting online decision domains $(\mathcal{S}, \Gamma)$, if there exists an efficient offline algorithm to maximize or minimize functions in Γ over $\mathcal{S}$, then this algorithm can be transformed into an efficient algorithm for the generalized bandit problem for $(\mathcal{S}, \Gamma)$. (Examples were the online linear and convex optimization algorithms of Chapter 5.) It would be interesting to identify other online decision domains $(\mathcal{S}, \Gamma)$ where such a reduction from offline to online linear optimization is possible. As a concrete example, let X be a finite set, let $\mathcal{S} = 2^X$ be the set of all subsets of X, and let Γ be the set of submodular functions on $\mathcal{S}$. It is known that there are efficient offline algorithms to minimize submodular functions [27, 38, 42]. Is there an algorithm for the generalized bandit problem for $(\mathcal{S}, \Gamma)$ which achieves convergence time $O(\text{poly}(|X|))$? Even in the full feedback model there open questions concerning submodular function minimization. It is known that there is an algorithm with convergence time $O(|X|)$ (one simply uses the best-expert algorithm Hedge with one expert for each subset of X), but can one achieve convergence time $O(|X|)$ or even $O(\text{poly}(|X|))$ using an algorithm that performs only a polynomial amount of computation per trial?

A related question concerns online decision domains $(\mathcal{S}, \Gamma)$ for which one has an approximation algorithm to optimize functions in Γ over the set $\mathcal{S}$, but no exact algorithm for this problem. Given an offline algorithm for minimizing cost functions which achieves an approximation factor α, can one asymptotically achieve approximation factor α in the corresponding generalized best-expert problem? In other words, can one construct an online algorithm achieving the guarantee

$$\lim_{T \to \infty} \max_{x \in \mathcal{S}} \left[\frac{1}{T} \sum_{t=1}^{T} c_t(x_t) - \alpha c_t(x) \right] = 0 \tag{8.1}$$

for all $x \in \mathcal{S}$ and all cost function sequences $c_1, c_2, \ldots \in \Gamma$? As a concrete example, consider the *online metric traveling salesman problem*, in which X is a finite set and $\mathcal{S}$ consists of all permutations of X. A cost function $c \in \Gamma$ is specified as follows: one defines a metric on X, and assigns a cost to each permutation of X by computing the length of the traveling salesman tour defined by that permutation. Is there an algorithm for the generalized best-expert problem for $(\mathcal{S}, \Gamma)$ which achieves a guarantee of the form (8.1) for any constant α?

In two of our lower bound proofs (Theorems 4.4 and 6.6), the proof depended on constructing an input instance such that the algorithm fails to outperform the stated lower bound at an infinite sequence of times $T_1, T_2, \ldots$, where T_k grows extremely fast (doubly exponential or faster) as a function of k. One need not consider such proofs to be strong negative results, since they leave open the possibility that there may exist algorithms which outperform the stated lower bound at all but an extremely sparse set of time steps. This is related to the fact that our lower bounds are stated in terms of the lim sup rather than the lim inf of the regret sequence. Consider Theorem 4.4, for example. The theorem says that for any continuum-armed bandit algorithm ALG and any $\beta < \frac{\alpha+1}{2\alpha+1}$ (where α is the exponent of Lipschitz continuity of the cost functions),

$$\limsup_{T \to \infty} \frac{R(\mathsf{ALG}, \mathcal{A}; T)}{T^\beta} = \infty.$$

Because the theorem gives a lower bound on $\limsup_{T\to\infty} R(\mathsf{ALG}, \mathsf{ADV}; T)/T^\beta$ rather than $\liminf_{T\to\infty} R(\mathsf{ALG}, \mathsf{ADV}; T)/T^\beta$, we can say that the regret of algorithm ALG is not $o(T^\beta)$, but we can not say that it is $\Omega(T^\beta)$. In fact, it is not necessarily true that the regret of ALG is $\Omega(T^\beta)$. Counterintuitively, it is the case that for *any* function $f(T)$ satisfying $\liminf_{T\to\infty} f(T) = \infty$, there is an algorithm ALG satisfying

$$\liminf_{T \to \infty} \frac{R(\mathsf{ALG}, \mathcal{A}; T)}{f(T)} = 0$$

where $\mathcal{A}$ is either of the sets $\mathcal{A}_{\text{adpt}}$ or $\mathcal{A}_{\text{iid}}$ defined in Chapter 4. This illustrates that the lim inf is also an inadequate way of characterizing the performance of online decision algorithms for "typical" values of T. It would be desirable to find theorems which express the performance of continuum-armed bandit algorithms at typical values of T in a way which is both mathematically precise and conceptually satisfying. For example, given a pair of functions $f(T), g(T)$, let us say that a continuum-armed bandit algorithm satisfies a *bicriterion regret bound* of type (f, g) if

$$\limsup_{T \to \infty} \frac{R(\mathsf{ALG}, \mathsf{ADV}; T)}{f(T)} < \infty$$

and

$$\liminf_{T \to \infty} \frac{R(\mathsf{ALG}, \mathsf{ADV}; T)}{g(T)} < \infty.$$

What bicriterion regret bounds are achievable by continuum-armed bandit algorithms? Can one achieve a bound of type $(o(T), O(\sqrt{T}\,\mathrm{polylog}(T)))$?

8.1.2 Adaptive pricing

In Chapter 3 we presented nearly matching upper and lower bounds for the regret of the optimal adaptive pricing strategy, when a monopolist sells an unlimited supply of identical goods to a population of buyers whose valuations lie in a bounded interval. One can imagine many relaxations of these assumptions leading to interesting open questions:

1. What if the seller has a limited supply?
2. What if there is more than one seller?
3. What if there is more than one type of good? What if buyers request bundles of goods rather than singletons?
4. What if the buyers' valuations come from a heavy-tailed distribution?

One can interpret our Theorem 1.9 as identifying the value of knowing the demand curve in a particular type of market mechanism. It would be desirable to investigate other economic scenarios in which one can quantify the value of information in this way. Can the definition of "knowledge" given in Chapter 3 be broadened to apply in such scenarios?

8.1.3 Online routing

In Section 5.2.4 we demonstrated that our online linear optimization algorithm could be used as an adaptive routing algorithm with the objective of minimizing delay. But our algorithm is far from being useful in practical contexts, e.g. as a routing protocol for overlay networks. Can one use online decision problems as the basis for practical overlay routing protocols? This broad question suggests several specific questions for future research.

1. Can one devise an online shortest path algorithm which is more distributed than the one given in Chapter 5, i.e. an algorithm in which there is an agent at each node of the graph, and these agents select the routing paths in each trial in a distributed manner using only local information?
2. Can one devise an algorithm which deals gracefully with nodes and edges joining and leaving the network, i.e. an algorithm which adjusts to such events without re-initializing its entire state?

3. Instead of assuming a fixed sender s and receiver r, suppose the sender s_t and receiver r_t are specified at the start of each trial t, and that they may vary from one trial to the next. Can one design an algorithm which converges more rapidly than the naive algorithm which instantiates n^2 independent instances of the single-sender-single-receiver algorithm?

It is also intriguing that the complexity of computing a barycentric spanner is still unresolved. Proposition 5.3 indicates that a C-approximate barycentric spanner for a compact set $\mathcal{S}$ can be computed in polynomial time for any $C > 1$, but what about an actual barycentric spanner? If $\mathcal{S}$ is finite then we can compute a barycentric spanner of $\mathcal{S}$ (possibly in superpolynomial time) using the algorithm in Figure 5-2 with $C = 1$. Since this is a local search algorithm and each iteration requires only polynomial computation time, the barycentric spanner problem is in the complexity class PLS defined in [43]. Is the barycentric spanner problem PLS-complete?

8.1.4 Collaborative learning

In Chapter 7 we introduced and analyzed an algorithm for a simple model of collaborative learning. A key feature of our model is the presence of a large number of dishonest agents who are assumed to behave in an arbitrary Byzantine manner. However, other aspects of our model are quite idealized, and there are some very natural extensions of the model which more closely reflect the reality of collaborative learning systems such as eBay's reputation system and peer-to-peer resource discovery systems. It would be desirable to identify algorithms for some of the following extensions.

1. Chapter 7 was concerned with a *synchronous* decision-making problem in which each agent must choose one resource in each decision round. Study the *asynchronous* case, in which only a subset of the agents act as decision-makers in each round and the rest are inactive.

2. We assumed that any agent could choose any resource at any time. Study cases in which an agent x is restricted to a choose from a subset $S(x,t) \subseteq Y$ at time t. Useful special cases include the case in which $S(x,t)$ does not depend on t and the case in which it does not depend on x. (In the latter case, it is not even clear how to formulate the proper notion of "regret.")

3. We assumed a very strict consistency condition for two agents x_1, x_2 in the same cluster: at *every* time t, for *every* resource y, the random variables $c_t(x_1, y), c_t(x_2, y)$ should have the same expected value, conditioned on past history. Consider relaxations of this criterion, for instance:

- $|c_t(x_1, y) - c_t(x_2, y)| < \varepsilon$.
- $\mathbf{E}(c_t(x_1, y^*)) = \mathbf{E}(c_t(x_2, y^*))$, where y^* is the best resource for both x_1 and x_2. No such equation is required to hold for other resources y.
- **[The mixture model]** For each t, the functions $f_x(y) = c_t(x, y)$ belong to a k-dimensional linear subspace of the vector of functions $Y \to \mathbb{R}$, as x ranges over X.

4. Study more structured collaborative decision-making problems, e.g. selecting routing paths in a network, some of whose nodes are identified with the agents.

Finally, it would be desirable to discover non-trivial lower bounds for the convergence time of collaborative learning algorithms. At present the trivial lower bound of $\Omega(m/\alpha n)$ — the minimum number of rounds needed to ensure that the best resource is sampled by at least one honest agent with constant probability — is essentially the only known lower bound.

Bibliography

[1] Philippe Aghion, Patrick Bolton, Christopher Harris, and Bruno Jullien. Optimal learning by experimentation. *Review of Economic Studies*, 58(4):621–654, 1991.

[2] Rajeev Agrawal. The continuum-armed bandit problem. *SIAM J. Control and Optimization*, 33:1926–1951, 1995.

[3] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine Learning*, 47:235–256, 2002.

[4] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Computing*, 32(1):48–77, 2002.

[5] Baruch Awerbuch, David Holmer, Herbert Rubens, and Robert Kleinberg. Provably competitive adaptive routing. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, 2005.

[6] Baruch Awerbuch and Robert Kleinberg. Adaptive routing with end-to-end feedback: distributed learning and geometric approaches. In *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC)*, pages 45–53, 2004.

[7] Baruch Awerbuch and Robert Kleinberg. Competitive collaborative learning. In *Proceedings of the 18th Annual Conference on Learning Theory (COLT)*, 2005. To appear.

[8] Baruch Awerbuch and Yishay Mansour. Online learning of reliable network paths. In *Proceedings of the 22nd SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, pages 360–367, 2003.

[9] Baruch Awerbuch, Boaz Patt-Shamir, David Peleg, and Mark Tuttle. Collaboration of untrusting peers with changing interests. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 112–119, 2004.

[10] Baruch Awerbuch, Boaz Patt-Shamir, David Peleg, and Mark Tuttle. Improved recommendation systems. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1174–1183, 2005.

[11] Yossi Azar, Amos Fiat, Anna Karlin, Frank McSherry, and Jared Saia. Spectral analysis of data. In *Proceedings of the 33rd ACM Symposium on Theory of Computing (STOC)*, pages 619–626, 2001.

[12] Amitabha Bagchi, Amitabh Chaudhary, Rahul Garg, Michael T. Goodrich, and Vijay Kumar. Seller-focused algorithms for online auctioning. In *Proc. 7th Internation Workshop on Algorithms and Data Structures (WADS 2001)*, volume 2125 of *Lecture Notes in Computer Science*, pages 135–147. Springer Verlag, 2001.

[13] Jeffrey S. Banks and Rangarajan K. Sundaram. Denumerable-armed bandits. *Econometrica*, 60(5):1071–1096, 1992.

[14] Ziv Bar-Yossef, Kirsten Hildrum, and Felix Wu. Incentive-compatible online auctions for digital goods. In *Proceedings of the 13th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 964–970, 2002.

[15] John Barnett and Nathan Srebro, 2005. Personal communication.

[16] Donald A. Berry, Robert W. Chen, Alan Zame, David C. Heath, and Larry A. Shepp. Bandit problems with infinitely many arms. *Annals of Statistics*, 25(5):2103–2116, 1997.

[17] Donald A. Berry and Bert Fristedt. *Bandit problems: sequential allocation of experiments*. Chapman and Hall, 1985.

[18] Donald A. Berry and Larry M. Pearson. Optimal designs for two-stage clinical trials with dichotomous responses. *Statistics in Medicine*, 4:487–508, 1985.

[19] Patrick Billingsley. *Probability and Measure*. John Wiley, 1995.

[20] Avrim Blum. On-line algorithms in machine learning. In *Developments from a June 1996 seminar on Online algorithms*, pages 306–325, London, UK, 1998. Springer-Verlag.

[21] Avrim Blum, Vijay Kumar, Atri Rudra, and Felix Wu. Online learning in online auctions. *Theoretical Computer Science*, 324(2–3):137–146, 2004.

[22] William M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 1986.

[23] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the 7th International World Wide Web Conference (WWW)*, pages 107–117, 1998.

[24] Yan Chen, David Bindel, Hanhee Song, and Randy H. Katz. An algebraic approach to practical and scalable overlay network monitoring. In *SIGCOMM '04: Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 55–66, New York, NY, USA, 2004. ACM Press.

[25] Eric Cope. Regret and convergence bounds for immediate-reward reinforcement learning with continuous action spaces, 2004. Unpublished manuscript.

[26] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley, 1991.

[27] William H. Cunningham. On submodular function minimization. *Combinatorica*, 5:185–192, 1985.

[28] Ioana Dimitriou, Prasad Tetali, and Peter Winkler. On playing golf with two balls. *SIAM J. on Discrete Math.*, 16:604–615, 2003.

[29] Petros Drineas, Iordanis Kerenidis, and Prabhakar Raghavan. Competitive recommendation systems. In *Proceedings of the 34th ACM Symposium on Theory of Computing (STOC)*, pages 82–90, 2002.

[30] David Easley and Nicholas M. Kiefer. Controlling a stochastic process with unknown parameters. *Econometrica*, 56(5):1045–1064, 1988.

[31] Amos Fiat, Andrew V. Goldberg, Jason D. Hartline, and Anna R. Karlin. Competitive generalized auctions. In *Proceedings of the 34th ACM Symposium on Theory of Computing (STOC)*, pages 72–81. ACM Press, 2002.

[32] Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: Gradient descent without a gradient. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 385–394, 2005.

[33] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[34] J. C. Gittins. Bandit processes and dynamic allocation indices (with discussion). *J. Roy. Statist. Soc. Ser. B*, 41:148–177, 1979.

[35] J. C. Gittins and D. M. Jones. A dynamic allocation index for the sequential design of experiments. In J. Gani *et al.*, editor, *Progress in Statistics*, pages 241–266. North-Holland, 1974.

[36] Andrew V. Goldberg, Jason D. Hartline, and Andrew Wright. Competitive auctions and digital goods. In *Proc. 12th Symp. on Discrete Algorithms*, pages 735–744, 2001.

[37] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, December 1992.

[38] Martin Grotschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1:169–197, 1981.

[39] James Hannan. Approximation to bayes risk in repeated plays. In M. Dresher, A. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games*, volume 3, pages 97–139. Princeton University Press, 1957.

[40] Jason D. Hartline. Dynamic posted price mechanisms, 2002. Working draft.

[41] Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. In *Proceedings of the International Joint Conference in Artificial Intelligence (IJCAI)*, pages 688–693, 1999.

[42] Satoru Iwata, Lisa Fleischer, and Satoru Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *J. ACM*, 48(4):761–777, 2001.

[43] David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37:79–100, 1988.

[44] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. In *Proceedings of the 16th Annual Conference on Learning Theory (COLT)*, pages 26–40, 2003.

[45] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th International World Wide Web Conference (WWW)*, pages 640–651, 2003.

[46] Ravi Kannan, László Lovász, and Miklós Simonovits. Isoperimteric problems for convex bodies and a localization lemma. *Disc. Comput. Geometry*, 13:541–559, 1995.

[47] Michael N. Katehakis and Jr. Arthur F. Veinott. The multi-armed bandit problem: Decomposition and computation. *Math. of Operations Research*, 12(2):262–268, 1987.

[48] Jack Kiefer and Jacob Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 23:462–466, 1952.

[49] Jon Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[50] Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 697–704. MIT Press, Cambridge, MA, 2005.

[51] Robert Kleinberg and Tom Leighton. The value of knowing a demand curve: Bounds on regret for on-line posted-price auctions. In *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 594–605, 2003.

[52] T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocations rules. *Adv. in Appl. Math.*, 6:4–22, 1985.

[53] Ron Lavi and Noam Nisan. Competitive analysis of incentive compatible on-line auctions. In *Proceedings of the 2nd ACM Conference on Electronic Commerce (EC-00)*, pages 233–241, 2000.

[54] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–260, 1994.

[55] Pattie Maes, Robert H. Guttman, and Alexandros G. Moukas. Agents that buy and sell. *Communications of the ACM*, 42(3):81–91, 1999.

[56] Shie Mannor and John N. Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5:623–648, 2004.

[57] Andrew McLennan. Price dispersion and incomplete learning in the long run. *Journal of Economic Dynamics and Control*, 7:331–347, 1984.

[58] H. Brendan McMahan and Avrim Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *Proceedings of the 17th Annual Conference on Learning Theory (COLT)*, volume 3120 of *Lecture Notes in Computer Science*, pages 109–123. Springer Verlag, 2004.

[59] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[60] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[61] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 175 – 186, 1994.

[62] Theodore J. Rivlin. *An Introduction to the Approximation of Functions*. Dover, 1981.

[63] Michael Rothschild. A two-armed bandit theory of market pricing. *Journal of Economic Theory*, 9:185–202, 1974.

[64] Walter Rudin. *Real and Complex Analysis*. McGraw Hill, 1966.

[65] Ilya Segal. Optimal pricing mechanisms with unknown demand. *American Economic Review*, 93(3):509–529, 2003.

[66] Yuval Shavitt, Xiaodong Sun, Avishai Wool, and Bulent Yener. Computing the unmeasured: An algebraic approach to internet mapping. *IEEE J. on Selected Areas in Communications*, 22(1):67–78, 2004.

[67] Rangarajan K. Sundaram. Generalized bandit problems, 2003. Working paper, Stern School of Business, New York University. To appear in *Social and Strategic Behavior: Essays in Honor of Jeffrey S. Banks*, D. Austen-Smith and J. Duggan, Eds.

[68] Eiji Takimoto and Manfred K. Warmuth. Path kernels and multiplicative updates. *Journal of Machine Learning Research*, 4:773–818, 2003.

[69] Pravin P. Varaiya, Jean C. Walrand, and Cagatay Buyukkoc. Extensions of the multiarmed bandit problem: The discounted case. *IEEE Trans. Auto. Control*, AC-30:426–439, 1985.

[70] Peter Whittle. Multi-armed bandits and the Gittins index. *J. Roy. Statist. Soc. Ser. B*, 42:143–144, 1980.

[71] Peter Whittle. *Optimization over Time: Dynamic Programming and Stochastic Control*, volume I, pages 210–220. John Wiley, 1982.

[72] Bin Yu and Munindar P. Singh. A social mechanism of reputation management in electronic communities. In *Cooperative Information Agents*, pages 154–165, 2000.

[73] Giorgos Zacharia, Alexandros Moukas, and Pattie Maes. Collaborative reputation mechanisms in electronic marketplaces. In *HICSS '99: Proceedings of the Thirty-second Annual Hawaii International Conference on System Sciences-Volume 8*, page 8026, Washington, DC, USA, 1999. IEEE Computer Society.

[74] Oren Zamir and Oren Etzioni. Web document clustering: A feasibility demonstration. In *Research and Development in Information Retrieval*, pages 46–54, 1998.

[75] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pages 928–936, 2003.