Interior-Point Methods for Full-Information and Bandit Online Learning

Jacob D. Abernethy, Elad Hazan, and Alexander Rakhlin

***Abstract*—We study the problem of predicting individual sequences with linear loss with full and partial (or bandit) feedback. Our main contribution is the first *efficient* algorithm for the problem of online linear optimization in the bandit setting which achieves the optimal $\tilde{O}(\sqrt{T})$ regret. In addition, for the full-information setting, we give a novel regret minimization algorithm. These results are made possible by the introduction of interior-point methods for convex optimization to online learning.**

***Index Terms*—Bandit feedback, interior-point methods, online convex optimization, online learning.**

I. Introduction

In 1952, Robbins [1] introduced a problem of sequential decision making, later termed the "multiarmed bandit," in which a learner must repeatedly make choices in an uncertain environment. The name draws from the image of a gambler who, on each of a sequence of rounds, must pull the arm on one of several slot machines ("one-armed bandits") that each returns a reward chosen stochastically from some unknown fixed distribution. Of course, an ideal strategy would simply be to pull the arm of the machine which returns the highest expected reward. As the gambler does not know the best arm *a priori*, his goal is then to follow a strategy to maximize his rewards, averaged over time, relative to the (expected) reward he could have received had he known the identity of the best arm. The multiarmed bandit problem has received much attention over the years and many variants have been proposed.

We draw attention to one key attribute of such sequential decision frameworks: what *feedback* is received by the decision maker after an action is selected? For a gambler playing slot machines, the gambler receives exactly one value, the reward on the chosen arm. But a gambler placing a bet in a horse race can observe not only the outcome of the chosen bet but also that of any counterfactual. ("I knew I should have bet on Jolly Roger.") The former decision-making model has recently been called the *bandit setting*, whereas the latter is occasionally referred to as the *full-information setting*; we shall follow this terminology. We note that the bandit setting, where strictly less information is available, is clearly the more challenging for the decision maker.

Much of the literature in decision theory has relied on a key assumption, that our observations are stochastic objects drawn from a fixed distribution. This approach follows naturally from our intuitive belief that the past ought to predict the future. But what if we are unwilling to accept this belief, may we still construct decision strategies with reasonable performance guarantees? Such a problem of *universal* decision making (or prediction), where our observation sequences are nonstochastic, has received much attention in a range of communities—learning theory, information theory, game theory, optimization—and we refer the reader to the survey by Merhav and Feder [2] as well as to the excellent book of Cesa-Bianchi and Lugosi [3] for a thorough exposition.

A. Review of Universal Prediction and Regret Minimization

Before describing the results of this paper, we give a quick review of the sequential decision-making framework with deterministic sequences, also known as adversarial online learning. On each of a sequence of rounds, a *learner* (similarly, *player*) must select a potentially randomized *strategy* (similarly *prediction* or *action*) denoted by x_t from some set $\mathcal{K}$. Before the learning process begins, the *environment* (similarly *nature* or an *adversary*) has chosen a sequence of *outcomes* (similarly, *data points*) $f_1, f_2, \ldots$ drawn from some set $\mathcal{F}$. There is some real-valued loss function ℓ known to both parties, and for each pair $(x, f) \in \mathcal{K} \times \mathcal{F}$ the value $\ell(x, f)$ is the cost to the learner for playing x when the outcome was f. After the learner has selected x_t, he receives some "feedback" $\mathscr{F}_t$ which may depend on both the outcome f_t and his action x_t. We say the learner has *full information* when $\mathscr{F}_t := f_t$, whereas we say the learner has *bandit feedback* when $\mathscr{F}_t := \ell(x_t, f_t)$. After t rounds, the choice of x_t may then only depend on $\mathscr{F}_1, \ldots, \mathscr{F}_{t-1}$.

In a stochastic setting, under the assumption that each outcome f_t is drawn i.i.d. from a given (unknown) distribution D, a natural benchmark for the learner is the optimal expected cost $\inf_{x \in \mathcal{K}} \mathbb{E}_{f \sim D} \ell(x, f)$. In the deterministic setting, however, there is only one sequence $f_1, f_2, \ldots$, and the benchmark needs to be set as the best performance on this sequence. It is then natural to consider expected *regret*

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell(x_t, f_t) - \inf_{x^* \in \mathcal{K}} \sum_{t=1}^{T} \ell(x^*, f_t)\right]$$

Manuscript received May 04, 2011; revised November 17, 2011; accepted November 26, 2011. Date of publication April 27, 2012; date of current version June 12, 2012. This work was supported in part by the Defense Advanced Research Projects Agency under Grant FA8750-05-2-0249 and in part by the National Science Foundation under Grant DMS-0707060. This paper was presented at the 21st Annual Conference on Learning Theory, Helsinki, Finland, July 2008.

J. D. Abernethy is with the Department of Computer Science, University of California Berkeley, Berkeley, CA 94720 USA (e-mail: jake@cs.berkeley.edu).

E. Hazan is with the Department of Industrial Engineering and Operations Research, Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: ehazan@ie.technion.ac.il).

A. Rakhlin is with the Department of Statistics, University of Pennsylvania, Philadelphia, PA 19104-6376 USA (e-mail: rakhlin@wharton.upenn.edu).

Communicated by N. Cesa-Bianchi, Associate Editor for Pattern Recognition, Statistical Learning, and Inference.

Digital Object Identifier 10.1109/TIT.2012.2192096

as the performance measure for the learner. We note, importantly, that the expectation here is taken with respect to the randomness of the learner's choices and not those of Nature. Nature has selected the outcomes f_t in advance and hence are fixed and nonstochastic. The goal of the learner is to provide a bound on the regret for any such fixed outcome sequence.

Let us return our attention to the K-armed bandit problem, where on each round the learner repeatedly selects an arm $x_t \in \{1, \ldots, K\}$ and the environment selects a vector of payoffs (equivalently, losses) $f_t \in [0, B]^K$, where $B > 0$ is some bound on the payoff values. In this case, both the loss function and the feedback are defined as $\ell(x_t, f_t) = \mathscr{f}_t := f_t[x_t]$, the loss value of arm x_t. This problem has been thoroughly studied for decades under stochastic assumptions, where the f_t are drawn i.i.d. (see, e.g., [4]). The earliest work studying this problem within the deterministic framework was that of Auer *et al.* [5] in 2003, who dubbed it the "nonstochastic multiarmed bandit problem." One major result of their paper is a learning algorithm with a regret guarantee that scales[1] as $O^*(\sqrt{TK})$. This was rather surprising, as the easier full-information version of the problem, where $\mathscr{f}_t := f_t$, has an optimal regret of $O(\sqrt{T \log K})$, providing the same asymptotic dependence on T.

B. Online Linear Optimization in the Bandit Setting

Consider a natural extension of the K-armed bandit problem, which we shall call Online Shortest Path. On each of a sequence of rounds, our learner must choose a path between two nodes s and t in a graph with n edges, and let us imagine this path is a route between home and the learner's workplace, where the goal is to minimize travel time. Once a path is selected, the learner takes the route and observes the cost (duration) of the trip, which shall depend on the traffic on that day. The learner would clearly like to have low regret, regardless of the process generating the traffic, relative to the best route averaged over all time.

It is clear that the set of $s - t$ paths may be prohibitively large for even a reasonably sized graph, making the straightforward reduction to the K-armed bandit problem computationally infeasible. That is, K will represent the total number of paths which can easily be exponential in n. But here is one way around this difficulty: relax the set of $s-t$ paths to the set of $s-t$ *flows*, where a flow can be thought of as a randomized path. The set of flows, while infinitely large, can be described by a convex set known as the flow polytope $\mathcal{K} \subset [0, 1]^n$, where the vertices of $\mathcal{K}$ are precisely the set of paths. Any flow $\mathbf{x} \in \mathcal{K}$ can be decomposed into a distribution over polynomially many paths, in which the "cost" of the flow is the expected cost of the path. If the traffic is described by a vector $\mathbf{f} \in [0, 1]^n$, where the ith coordinate of $\mathbf{f}$ is the traffic on edge i, then the loss of flow $\mathbf{x}$ with traffic $\mathbf{f}$ is exactly $\ell(\mathbf{x}, \mathbf{f}) := \mathbf{f}^\top \mathbf{x}$.

Looking carefully at the problem of Online Shortest Path, we see a very general problem arise that shall be a central focus of this paper, and which we shall call online linear optimization with bandit feedback. More precisely, we shall assume that 1) the learner's actions $\mathbf{x}_t$ are drawn from a compact convex set $\mathcal{K} \subset \mathbb{R}^n$, 2) the environment's actions $\mathbf{f}_t$ are drawn from some bounded set $\mathcal{F} \subset \mathbb{R}^n$, 3) the loss is defined according to the inner product, $\ell(\mathbf{x}_t, \mathbf{f}_t) := \mathbf{f}_t^\top \mathbf{x}_t$, and 4) we are in the bandit feedback setting, $\mathscr{f}_t := \mathbf{f}_t^\top \mathbf{x}_t$. This generic problem, as well as the special case of online routing, has received attention in some form or another from several authors [6]–[14].

[1]The notation $O^*(\,)$ hides logarithmic factors.

C. Our Results

Despite this large body of work on the bandit linear optimization problem, for years one question remained elusive: does there exist an algorithm which achieves an $O^*(\sqrt{T})$ regret. While an algorithm achieving $O^*(\sqrt{T})$ regret was given by Auer *et al.* [5], this was only applicable to the simpler K-armed bandit problem. The early works on this problem were unable to prove a rate any faster than an $O(T^{2/3})$ [6]–[10]. The first breakthrough was from Dani *et al.* [12] whose algorithm did achieve a regret rate of $O^*(\sqrt{T})$. Their algorithm, which utilizes a clever reduction to the technique of Auer *et al.* [5], requires covering the set $\mathcal{K}$ with an ϵ^{-n}-sized grid and cannot be implemented efficiently in general.

The primary contributions of this paper are twofold.

1) *Result 1:* The first known *efficient* $O^*(\sqrt{T})$-regret algorithm, SCRiBLe, for the bandit setting with arbitrary convex decision sets.
2) *Result 2:* A novel efficient algorithm for full-information sequential prediction over arbitrary convex decision sets with new regret bounds.

In this paper we closely study a particular family of algorithms, called *Follow the Regularized Leader (FTRL)*, which at every step minimize an objective that is smoothed out by a "regularization" function. One of the key insights of our work is that the choice of regularization must be made very carefully. With this in mind, we shall consider interior-point methods for convex optimization, in particular looking at a class of functions known as *self-concordant barriers*. Such barrier functions will turn out to posses precisely the properties we need to achieve the optimal regret rate efficiently. The approach we take uncovers novel connections between interior-point methods and the study of universal prediction and decision making. We begin in Section II with a look at the theory of optimization and brief review of interior-point methods. We return to sequential prediction in Sections III and IV, where we prove the second main result in the full-information setting. This serves as the basis for our main result, proven in Section V.

II. Convex Optimization: Self-Concordant Barriers and the Dikin Ellipsoid

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

Interior-point methods were designed for precisely this problem and they are arguably one of the greatest achievements in the field of convex optimization in the past two decades.

These iterative polynomial-time algorithms for convex optimization find the solution by adding a barrier function to the objective such that the barrier diverges at the boundary of the set. We may now interpret the resulting optimization problem, on the modified objective function, as an unconstrained minimization problem which, as mentioned, can now be solved quickly. Roughly speaking, this approximate solution can be iteratively improved by gradually reducing the weight of the barrier function as one approaches the true optimum. In work pioneered by Karmarkar in the context of linear programming [15], and greatly generalized to constrained convex optimization by Nesterov and Nemirovskii, it has been shown that this technique admits a polynomial-time complexity as long as the barrier function is *self-concordant*, a property we soon define explicitly.

In this paper, we will borrow several tools from the interior-point literature, foremost among these is the use of self-concordant barrier functions. The utility of such functions is somewhat surprising, as our ultimate goal is not polynomial-time complexity but rather low-regret learning algorithms. While learning algorithms often involved adding "regularization" to a particular objective function, for the special case of learning with "bandit" feedback, as we shall see in Section V, the self-concordant regularizer provides the missing piece in obtaining a near-optimal regret guarantee.

The construction of barrier functions for general convex sets has been studied extensively, and we refer the reader to [16] and [17] for a thorough treatment on the subject. To be more precise, most of the results of this section can be found in [18, pp. 22--23], as well as in the aforementioned texts. We also refer the reader to the survey of Nemirovskii and Todd [19].

A. Definitions and Properties

In what follows, we list the relevant definitions and results on the theory of interior-point methods that will be used later in this paper. Let $\mathcal{K} \subset \mathbb{R}^n$ be a convex compact set with nonempty interior $\mathrm{int}(\mathcal{K})$.

1) Basic Properties of Self-Concordant Functions:

Definition 2.1: A self-concordant function $\mathcal{R} : \mathrm{int}(\mathcal{K}) \to \mathbb{R}$ is a C^3 convex function such that for all $\mathbf{h} \in \mathbb{R}^n$ and $\mathbf{x} \in \mathrm{int}(\mathcal{K})$

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

and $\mathcal{R}$ approaches infinity along any sequence of points approaching the boundary of $\mathcal{K}$.

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

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

Here, the third-order differential is defined as

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

We assume that $\mathcal{R}$ is nondegenerate, i.e., its Hessian exists and has full rank.

A central fact about interior-point methods is that they can be applied quite generally, as any arbitrary n-dimensional closed convex set admits an $O(n)$-self-concordant barrier [16]. Hence, throughout this paper, $\vartheta = O(n)$, but can even be independent of the dimension, as for the sphere.

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

Definition 2.2: Let $\mathcal{R}$ be a self-concordant function. For any $\mathbf{x} \in \mathrm{int}(\mathcal{K})$, we define an associated norm $\|\cdot\|_{\mathbf{x}}$ as

$$\|\mathbf{h}\|_{\mathbf{x}} = \sqrt{\mathbf{h}^{\intercal}\nabla^2\mathcal{R}(\mathbf{x})\mathbf{h}}.$$

We can also define $\|\cdot\|_{\mathbf{x}}^*$, the *dual norm* to $\|\cdot\|_{\mathbf{x}}$, as[2]

$$\|\mathbf{h}\|_{\mathbf{x}}^* = \sqrt{\mathbf{h}^{\intercal}\nabla^{-2}\mathcal{R}(\mathbf{x})\mathbf{h}},$$

where we denoted the inverse of the Hessian matrix by $\nabla^{-2}\mathcal{R}(\mathbf{x}) \equiv [\nabla^2\mathcal{R}(\mathbf{x})]^{-1}$. For any $\mathbf{x} \in \mathrm{int}(\mathcal{K})$ we define the *Dikin Ellipsoid* of radius r

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

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

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

When we use the term Dikin Ellipsoid it will be implied that the radius is 1 unless otherwise noted. This ellipsoid $W_1(\mathbf{x})$ is a key piece of our main result, in particular due to the following nontrivial fact (see [18, p. 23] for proof):

$$\forall\, \mathbf{x} \in \mathrm{int}(\mathcal{K}) \quad W_1(\mathbf{x}) \subset \mathcal{K}. \tag{1}$$

In other words, the inverse Hessian of the self-concordant function $\mathcal{R}$ stretches the space in such a way that the unit ball according to the norm defined by $\nabla^{-2}\mathcal{R}$ falls in the set $\mathcal{K}$.

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

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

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

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

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

While not necessarily clear at first glance, this iterative process converges *very* quickly. It is convenient to measure the

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

progress to the minimizer in terms of the Newton decrement, which leads us to the following Theorem.

Theorem 2.1 (see, e.g., [19]): For any self-concordant function $\mathcal{R}$, let $\mathbf{x}$ be any point in the interior of $\mathcal{K}$ and let $\mathbf{x}^* := \arg\min \mathcal{R}$. Then, $\mathrm{DN}(\mathbf{x}, \mathcal{R}) \in \mathcal{K}$ and whenever $\lambda(\mathbf{x}, \mathcal{R}) \leq 1/4$ we have

$$\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{x}} \leq 2\lambda(\mathbf{x}, \mathcal{R})$$
$$\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{x}^*} \leq 2\lambda(\mathbf{x}, \mathcal{R})$$
$$\lambda(\mathrm{DN}(\mathbf{x}, \mathcal{R}), \mathcal{R}) \leq 2\lambda(\mathbf{x}, \mathcal{R})^2.$$

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

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

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

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

Theorem 2.2: For any ϑ-self-concordant barrier on $\mathcal{K}$, and for any $\mathbf{x}, \mathbf{y} \in \mathrm{int}(\mathcal{K})$, we have that

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

A proof can be found in the lecture notes of Nemirovskii [18] and elsewhere.

It is important to notice that any linear perturbation $\mathcal{R}'(\mathbf{x}) := \mathcal{R}(\mathbf{x}) + \mathbf{h} \cdot \mathbf{x}$ of a self-concordant function $\mathcal{R}$ is again a self-concordant function. Indeed, the linear term disappears in the second and third derivatives in the first requirement of Definition 2.1. In the same vein, the norm induced by such $\mathcal{R}'$ is identical to that of $\mathcal{R}$.

B. Examples of Self-Concordant Functions

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

Lemma 2.1: Let $\mathcal{R}_1$ be a ϑ_1-self-concordant barrier function for the set $\mathcal{K}_1$ and let $\mathcal{R}_2$ be a ϑ_2-self-concordant barrier function for the set $\mathcal{K}_2$, then $\mathcal{R} := \mathcal{R}_1 + \mathcal{R}_2$ is a $(\vartheta_1 + \vartheta_2)$-self-concordant barrier function for the set $\mathcal{K}_1 \cap \mathcal{K}_2$.

The aforementioned Lemma is most useful for constructing self-concordant barriers on sets defined by the intersection of simpler sets. For example, on the set $[0, \infty]$ there exists a very simple barrier, namely $\mathcal{R}(x) = -\log x$. A quick check verifies that this function satisfies both the self-concordance and the barrier property with equality with $\vartheta = 1$. In addition, we can easily extend this to any half-space H in $\mathbb{R}^n$ by letting $\mathcal{R}(\mathbf{x}) = -\log \delta(\mathbf{x}, H)$, where $\delta(\cdot, H)$ is the Euclidean distance to the half-space. Finally, if the set $\mathcal{K}$ is a polytope in $\mathbb{R}^n$, then it is defined as the intersection of a number of halfspaces. Equivalently, it can be defined by linear inequalities $A\mathbf{x} \succeq \mathbf{b}$ for some $m \times n$ matrix A, which leads us immediately to the *log-barrier* function of this polytope, namely

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

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

For the n-dimensional ball

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

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

III. Low-Regret Learning With Full Information

The Interior-point literature, which we reviewed in the previous Section, is aimed at the problem of optimizing an objective particularly when the feasible set is constrained. Yet optimization is essentially an *offline* problem, as it presumed that the data to be optimized over are known in advance. Offline optimization is a very useful tool, but is only applicable when the objective is known in advance and can be queried. For the task of *learning*, however, the problem is precisely that the true objective is unknown *a priori* to the decision maker and is instead revealed one piece at a time. In this section, we turn our attention to the central focus of this paper, namely *online* convex optimization, which generalizes a number of well-known online-learning problems.

We begin by reviewing the setting of online convex optimization. We then discuss the main algorithmic template we employ, FTRL, and we prove several generic bounds for this class of algorithms. We then apply the tools developed in the previous section, and we prove an FTRL bound when a self-concordant barrier function is used for regularization.

A. Online Linear Optimization With Full Information

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

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

1) player chooses $\mathbf{x}_t \in \mathcal{K}$;
2) adversary independently chooses $\mathbf{f}_t \in \mathbb{R}^n$;
3) player suffers loss ${\mathbf{f}_t}^\top \mathbf{x}_t$ and observes feedback $\mathscr{F}_t$.

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

that chooses the predictions $\mathbf{x}_1, \mathbf{x}_2, \ldots$ and is presented with a sequence of functions $\mathbf{f}_{1:T} := (\mathbf{f}_1, \ldots, \mathbf{f}_T)$, then we define

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

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

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

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

For the remainder of this section, we will focus entirely on the *full-information* version of the problem. That is, we will assume that the player may observe the entire function $\mathbf{f}_t$ as his feedback $\mathscr{F}_t$ and can exploit this in making his decisions. We distinguish this version from the more challenging *bandit setting* in which the player may only observe the cost that he incurred, namely the scalar value $\mathbf{f}_t{}^{\top}\mathbf{x}_t$. The bandit problem was the motivation for this study, and we turn our attention in this direction in Section V.

B. FTRL and Associated Bounds

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

$$\mathbf{x}_{t+1} := \arg\min_{\mathbf{x} \in \mathcal{K}} \sum_{s=1}^{t} \mathbf{f}_s{}^{\top}\mathbf{x}. \tag{2}$$

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

A natural approach,[3] and more well-known within statistical learning, is to *regularize* the optimization problem (2) with an appropriate regularization function $\mathcal{R}(\mathbf{x})$, which is generally considered to be smooth and convex. The decision strategy is described in the following algorithm, which we refer to as FTRL.

We recall that this algorithm can only be applied in the full-information setting. That is, the choice of $\mathbf{x}_{t+1}$ requires observing $\mathbf{f}_1, \ldots, \mathbf{f}_t$ to evaluate the objective in (3).

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

[3]In the context of classification, this approach has been formulated and analyzed by Shalev-Shwartz and Singer [20].

Proposition 3.1: Given any sequence of cost vectors $\mathbf{f}_1, \ldots, \mathbf{f}_T$ and for any point $\mathbf{u} \in \mathcal{K}$, Algorithm 1 (FTRL) enjoys the guarantee

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

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

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

$$\mathbf{x}_{t+1} := \arg\min_{\mathbf{x} \in \mathcal{K}} \left[\eta \sum_{s=1}^{t} \mathbf{f}_s{}^{\top}\mathbf{x} + \mathcal{R}(\mathbf{x}) \right]. \tag{3}$$

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

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

$$\sum_{s=1}^{t} \mathbf{f}_s{}^{\top}\mathbf{x}_{s+1} \leq \min_{\mathbf{u} \in \mathcal{K}} \sum_{s=1}^{t} \mathbf{f}_s{}^{\top}\mathbf{u} + \frac{\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)}{\eta}. \tag{4}$$

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

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

Combining this with (4) gives the proof.

We now proceed to prove (4) by induction. The base case, for $t = 0$, holds by the choice of $\mathbf{x}_1$ as the minimizer over $\mathcal{K}$. Now assume the above bound holds for $t-1$. The crucial observation is that the point $\mathbf{x}_{t+1}$ is chosen as the minimizer of (4)

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

which completes the proof. ■

C. Vanilla Approach: Utilizing Strongly Convex $\mathcal{R}$

The bound stated in Proposition 3.1 is difficult to interpret for, at present, it tells us that the regret is bounded by the size of successive steps between $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$. Notice that the point $\mathbf{x}_{t+1}$ depends on both $\mathbf{f}_t$ and η as well as on the behavior of $\mathcal{R}$. Ultimately, we want a bound independent of the $\mathbf{x}_t$'s since these points are not under our control once we have fixed $\mathcal{R}$.

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

$$\mathbf{f}_t^{\mathsf{T}}(\mathbf{x}_t - \mathbf{x}_{t+1}) \leq \|\mathbf{f}_t\|^* \|\mathbf{x}_t - \mathbf{x}_{t+1}\| \tag{5}$$

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

Definition 3.1: Given any strictly convex and differentiable function $\mathcal{R}$, define the *Bregman divergence* with respect to $\mathcal{R}$ as

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

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

Definition 3.2: A function $\mathcal{R}(\mathbf{x})$ is *strongly convex* with respect to some norm $\|\cdot\|$ whenever the associated Bregman divergence satisfies $D_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \geq \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$ for all $\mathbf{x}, \mathbf{y}$.

While it might not be immediately obvious, the strong convexity of the regularization function in the FTRL algorithm is directly connected to the bound in Proposition 3.1. Specifically, the term $\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)$ increases with larger curvature of $\mathcal{R}$, whereas the terms $\mathbf{f}_t^{\mathsf{T}}(\mathbf{x}_t - \mathbf{x}_{t+1})$ shrink. Toward making the latter more precise, we give two lemmas regarding the "distance" between the pairs $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$.

Lemma 3.1: For the sequence $\{\mathbf{x}_t\}$ chosen according to $\mathrm{FTRL}(\mathcal{R}, \eta)$, we have that for any t:

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

Proof: Recalling that the divergence is always nonnegative, we obtain the first inequality by noting that for any $\mathbf{x}, \mathbf{y} \in \mathcal{K}$, $D_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) \leq D_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) + D_{\mathcal{R}}(\mathbf{y}, \mathbf{x}) = \langle \nabla \mathcal{R}(\mathbf{x}) - \nabla \mathcal{R}(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle$. For the second inequality, we observe that $\mathbf{x}_{t+1}$ is obtained in the optimization (3), and hence we have the first-order optimality condition

$$\left\langle \nabla \mathcal{R}(\mathbf{x}_{t+1}) + \eta \sum_{s=1}^{t} \mathbf{f}_s, \mathbf{y} - \mathbf{x}_{t+1} \right\rangle \geq 0 \quad \forall\, \mathbf{y} \in \mathcal{K}. \tag{6}$$

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

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

concluding the proof. ■

Lemma 3.2: For the sequence $\{\mathbf{x}_t\}$ chosen according to $\mathrm{FTRL}(\mathcal{R}, \eta)$, and $\mathcal{R}$ strongly convex with respect to the norm $\|\cdot\|$, we have that for any t:

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

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

Proof: Using the definition of strong convexity, we have

$$\begin{aligned} \|\mathbf{x}_t - \mathbf{x}_{t+1}\|^2 &\leq D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) + D_{\mathcal{R}}(\mathbf{x}_{t+1}, \mathbf{x}_t) \\ &= \langle \nabla \mathcal{R}(\mathbf{x}_t) - \nabla \mathcal{R}(\mathbf{x}_{t+1}), \mathbf{x}_t - \mathbf{x}_{t+1} \rangle \\ \text{(Lemma 3.1)} \quad &\leq \langle \eta \mathbf{f}_t, \mathbf{x}_t - \mathbf{x}_{t+1} \rangle \\ \text{(Hölder's Ineq.)} \quad &\leq \eta \|\mathbf{f}_t\|^* \|\mathbf{x}_t - \mathbf{x}_{t+1}\|. \end{aligned}$$

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

Applying (5) and Lemma 3.2 to Proposition 3.1, we arrive at the following.

Proposition 3.2: When $\mathcal{R}$ is strongly convex with respect to the norm $\|\cdot\|$, with $\min_{\mathbf{x} \in \mathcal{K}} \mathcal{R}(\mathbf{x}) = 0$, then for any $\mathbf{u} \in \mathcal{K}$

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

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

Corollary 3.1: When $\mathcal{R}$ is strongly convex with respect to the norm $\|\cdot\|$, and for constants $G, R > 0$ we have $\|\mathbf{f}_t\|^* \leq G$ for every t and $\mathcal{R}(\mathbf{x}) \leq R$ for every $\mathbf{x} \in \mathcal{K}$, then by setting $\eta = \sqrt{\frac{R}{G^2 T}}$ we have

$$\mathrm{Regret}_T(\mathrm{FTRL}(\mathcal{R}, \eta)) \leq 2G\sqrt{TR}.$$

IV. Improved Bounds Via Interior-Point Methods

In Section II, we presented a brief summary of known results from the literature on interior-point methods and self-concordant functions. In Section III, we switched gears and turned our attention to the study of online linear optimization and developed a technique for proving regret bounds within the regularization framework. In this section, we bring these seemingly dissimilar topics together and show that, by utilizing a self-concordant barrier as a regularization function, one can obtain much improved bounds for an array of problems. In particular, the introduction of these interior point techniques leads to a novel efficient algorithm for the Bandit setting with an essentially op-

timal regret guarantee, resolving what was an open question for several years.

A. Refined Regret Bound: Measuring $\mathbf{f}_t$ Locally

We return our attention to proving regret bounds as in Section III, but we now add a twist. The conclusions from that Section can be summarized as follows. For any FTRL algorithm, we achieve the fully general (yet unsatisfying) bound in Proposition 3.1. We can also apply Hölder's Inequality and, with the assumption that $\mathcal{R}$ is strongly convex, we arrive at Proposition 3.2.

The analysis of Proposition 3.2 is the typical approach, and indeed it can be shown that the above bound is tight (within a small constant factor from optimal), for instance, in the setting of prediction with expert advice [3]. On the other hand, there are times when we cannot make the assumption that $\mathbf{f}_t$ is bounded with respect to a fixed norm. This is particularly relevant in the bandit setting, when we will be estimating the functions $\mathbf{f}_t$ yet our estimates will blow up depending on the location of the point $\mathbf{x}_t$. In such cases, to obtain tighter bounds, it will be beneficial to measure the size of $\mathbf{f}_t$ with respect to a *changing norm*. While it may not be obvious at present, a useful way to measure $\mathbf{f}_t$ is with the quadratic form defined by the inverse Hessian of $\mathcal{R}$ at the point $\mathbf{x}_t$. Indeed, this is precisely the norm defined in Section II-A.

Theorem 4.1: Suppose for all t we have $\eta\|\mathbf{f}_t\|^*_{\mathbf{x}_t} \leq \frac{1}{4}$, and $\mathcal{R}$ is a self-concordant barrier with $\min_{\mathbf{x}\in\mathcal{K}} \mathcal{R}(\mathbf{x}) = 0$. Then for any $\mathbf{u} \in \mathcal{K}$

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

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

Proof of Theorem 4.1: Since $\mathcal{R}$ is a barrier, the minimization problem in (3) is unconstrained. As with Proposition 3.2, we can apply Hölders inequality to the term $\mathbf{f}_t{}^{\mathsf{T}}(\mathbf{x}_t - \mathbf{x}_{t+1})$. As the inequality holds for any primal-dual norm pair, we bound

$$\mathbf{f}_t{}^{\mathsf{T}}(\mathbf{x}_t - \mathbf{x}_{t+1}) \leq \|\mathbf{f}_t\|^*_{\mathbf{x}_t}\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_{\mathbf{x}_t}. \tag{7}$$

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

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

We can then bound

$$\begin{aligned} \|\mathbf{x}_t - \mathbf{x}_{t+1}\|_{\mathbf{x}_t} &= \|\mathbf{x}_t - \arg\min \Phi_t\|_{\mathbf{x}_t} \\ (\text{by Theorem 2.1}) \quad &\leq 2\lambda(\mathbf{x}_t,\Phi_t) = 2\|\nabla\Phi_t(\mathbf{x}_t)\|^*_{\mathbf{x}_t} \end{aligned}$$

Recall that Theorem 2.1 requires $\lambda(\mathbf{x}_t,\Phi_t) = \|\nabla\Phi_t(\mathbf{x}_t)\|^*_{\mathbf{x}_t} \leq 1/4$. However, since $\mathbf{x}_t$ minimizes Φ_{t-1}, and because $\Phi_t(\mathbf{x}) = \Phi_{t-1}(\mathbf{x}) + \eta\mathbf{f}_t{}^{\mathsf{T}}\mathbf{x}$, it follows that $\nabla\Phi_t(\mathbf{x}_t) = \eta\mathbf{f}_t$. By assumption, $\eta\|\mathbf{f}_t\|^*_{\mathbf{x}_t} \leq \frac{1}{4}$. Furthermore, we have now shown that

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

and when applied to (7) gives

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

Combining this inequality with Proposition 3.1 finishes the proof. ■

B. Improvement Compared to Previous Bounds

Assuming that $\mathcal{R}$ is strongly convex, modulo multiplicative $\log T$ terms, the bound obtained in Theorem 4.1 is never asymptotically worse than previous bounds, and at times is significantly tighter. We will briefly demonstrate this point next.

The key advantage is that, by measuring the loss functions $\mathbf{f}_t$ using $\|\mathbf{f}_t\|^{*2}_{\mathbf{x}_t} = \mathbf{f}_t{}^{\mathsf{T}}\nabla^{-2}\mathcal{R}(\mathbf{x}_t)\mathbf{f}_t$, the upper bound on regret depends on the position of $\mathbf{x}_t$ in the set $\mathcal{K}$. In particular, if a majority of the points $\mathbf{x}_t$ are close to the boundary, where the regularizer $\mathcal{R}$ has large curvature and the inverse Hessian $\nabla^{-2}\mathcal{R}(\mathbf{x}_t)$ is tiny, we can expect the terms $\|\mathbf{f}_t\|^{*2}_{\mathbf{x}_t}$ to be small.

As a simple example, consider an OLO problem in which the convex set is the real line segment $\mathcal{K} = [-1,1]$, and we shall use FTRL with the simple logarithmic barrier $\mathcal{R}(\mathbf{x}) = -\log(1-\mathbf{x}) - \log(1+\mathbf{x})$. Let us now imagine a natural scenario in which our sequence of cost vectors $\mathbf{f}_1, \mathbf{f}_2, \ldots \in \mathcal{F}$ has some positive or negative bias, and hence for some $c > 0$ we have $\|\mathbf{f}_1 + \cdots + \mathbf{f}_t\| \geq ct$ for large enough t, say $t > t_0$ for constant t_0. It is easily checked that the FTRL optimization for our chosen regularization will lead to $\|\mathbf{x}_t\| \geq 1 - \frac{1}{c\eta t}$ for $t > t_0$, which implies that $\nabla^2\mathcal{R}(\mathbf{x}_t) \geq \frac{4}{c^2\eta^2t^2}$. Pick a constant B so that $\sum_{t=1}^{t_0} 2\|\mathbf{f}_t\|^{*2}_{\mathbf{x}_t} \leq B$ (this constant depends on the constant t_0 and the diameter of $\mathcal{F}$, but not on t). For $t > t_0$, we can now bound

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

Depending on the sign of $\mathbf{f}_1 + \cdots + \mathbf{f}_T$, set the comparator according to $\mathbf{u} = \pm(1 - 1/T)$ so that $\mathcal{R}(\mathbf{u}) \leq \log T$. We arrive at the following bound on the regret via theorem 4.1:

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

where C hides the constant $\frac{8}{c^2}\sum_{t=t_0+1}^{T}\frac{1}{t^2}$. With an appropriately tuned[4] η, we obtain a bound on the order of $O(\sqrt{\log T})$.

We may now compare this to the previous "fixed-norm" regret bounds, such as those for online gradient descent of Zinkevich [21]. The corresponding bound for this algorithm would be $O(\eta T + \eta^{-1})$ which, even when optimally tuned, must grow at a rate at least $\Theta(\sqrt{T})$.

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

C. Computational Efficiency

Algorithm 1 requires a solution of a convex program at every iteration. Minimizing a convex function over a convex domain $\mathcal{K}$ can be carried out efficiently, i.e., in polynomial time, depending on the representation of $\mathcal{K}$. Algorithms for minimizing a convex function over a convex domain in polynomial time include the ellipsoid algorithm [22] and random-walk-based methods [23].

Most interesting convex sets $\mathcal{K}$ are known to admit efficient self-concordant barriers. This family includes polytopes, sets defined by conic constraints, and the semi-definite cone. In this case, interior-point methods give rise to much more efficient algorithms. In our case, the objective function is a linear function plus a self-concordant barrier (which is smooth and strongly convex). Hence, the optimization problem of Algorithm 1 with a ϑ-self-concordant barrier can be approximated up to ε precision in time $\tilde{O}(\sqrt{\vartheta}n^3 \log \frac{1}{\varepsilon})$ (see [18]).

This already gives an efficient algorithm as a basis for our bandit algorithm that we develop in Sections V and VI. We give an even more efficient algorithm next.

D. Iterative Interior Point Algorithm

Although Algorithm 1 runs in polynomial time, in this subsection we give an even more efficient iterative algorithm. Instead of optimizing the objective of Algorithm 1 anew at every step, the iterative algorithm makes small adjustments to the previously computed solution.

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

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

While Algorithm 2 may seem very different from the FTRL method, it can be viewed as an efficient implementation of the same algorithm, and hence it borrows the almost same regret bounds. Notice in the bound below that for $\eta = \tilde{O}(\frac{1}{\sqrt{T}})$, the optimal setting of parameter in Theorem 4.1, the additive term is a constant independent of the number of iterations.

Algorithm 2 Iterative FTRL (I-FTRL)

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

$$\mathbf{y}_{t+1} := \mathrm{DN}(\Phi_t, \mathbf{y}_t) = \mathbf{y}_t + \frac{1}{1+\lambda(\mathbf{y}_t, \Phi_t)} \mathbf{e}(\mathbf{y}_t, \Phi_t). \quad (8)$$

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

Theorem 4.2: Let $\mathcal{K}$ be a compact convex set and $\mathcal{R}$ be a ϑ-self-concordant barrier on $\mathcal{K}$. Assume $\|\mathbf{f}_t\|_{\mathbf{y}_t}^* \le C$ for all t and $\eta C \le \frac{1}{8}$. Then, for any $\mathbf{u} \in \mathcal{K}$

$$\begin{aligned}&\mathrm{Regret}^{\mathbf{u}}(\text{I-FTRL}(\mathcal{R},\eta); \mathbf{f}_{1:T}) \\ &\le \mathrm{Regret}^{\mathbf{u}}(\mathrm{FTRL}(\mathcal{R},\eta); \mathbf{f}_{1:T}) + 16C^3\eta^2 T. \end{aligned} \quad (9)$$

To prove this theorem, we show that the predictions generated by Algorithm 2 are very close to those generated by Algorithm 1. More formally, we prove the following lemma, where $\{\mathbf{x}_t\}$ denotes the sequence of vectors generated by the FTRL algorithm as defined in (3).

Lemma 4.1:

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

Before proving this lemma, let us show how it immediately implies Theorem 4.2

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

We can now proceed to prove Lemma 4.1.

Proof of Lemma 4.1: The proof is by induction on t. For $t = 1$ the result is true because $\mathbf{x}_1, \mathbf{y}_1$ are chosen to minimize $\mathcal{R}$. Suppose the statement holds for t, we prove for $t+1$. By definition

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

Note that

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

Using $(\mathbf{x}+\mathbf{y})^\top \mathbf{A}(\mathbf{x}+\mathbf{y}) \le 2\mathbf{x}^\top\mathbf{A}\mathbf{x} + 2\mathbf{y}^\top\mathbf{A}\mathbf{y}$ we obtain

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

The first term can be bounded by the induction hypothesis

$$\lambda^2(\mathbf{y}_t, \Phi_{t-1}) \le 64\eta^4 C^4. \quad (10)$$

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

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

Combining the results

$$\lambda^2(\mathbf{y}_t, \Phi_t) \le 2 \cdot (64\eta^4 C^4 + \eta^2 C^2) \le 4\eta^2 C^2 \quad (11)$$

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

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

according to Theorem 2.1. The induction step is completed by applying Theorem 2.1 again

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

■

V. Bandit Feedback

We now return our attention to the *bandit version* of the online linear optimization problem that we have discussed. The additional difficulty in the bandit setting rests in the feedback model. As before, an $\mathbf{x}_t$ is chosen at round t, an Adversary chooses $\mathbf{f}_t$, and the cost ${\mathbf{f}_t}^{\mathsf{T}}\mathbf{x}_t$ is paid. But, instead of receiving the entire vector $\mathbf{f}_t$, the learner may only observe the scalar value $({\mathbf{f}_t}^{\mathsf{T}}\mathbf{x}_t)$. Recall that, in our FTRL template, the point $\mathbf{x}_t$ is computed with access to $(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{t-1})$ whereas an algorithm in the bandit setting is given only $({\mathbf{f}_1}^{\mathsf{T}}\mathbf{x}_1, {\mathbf{f}_2}^{\mathsf{T}}\mathbf{x}_2, \ldots, {\mathbf{f}_{t-1}}^{\mathsf{T}}\mathbf{x}_{t-1})$ as input.

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

In this section, we make an additional assumption that the adversary is *oblivious*. That is, the sequence $\mathbf{f}_1, \ldots, \mathbf{f}_T$ is fixed ahead of the game. For results in bandit optimization against nonoblivious adversaries, we refer the reader to [24].

A. Constructing a Bandit Algorithm

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

1) A *full-information algorithm* $\mathcal{A}$ which takes as input a sequence of loss vectors $\mathbf{f}_t$ and returns points $\mathbf{x} \in \mathcal{K}$; that is
$$\mathbf{x}_t \leftarrow \mathcal{A}(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{t-1}).$$
2) A *sampling scheme* $\texttt{sampler}(\mathbf{x})$ for each $\mathbf{x}$ that defines a distribution on $\mathcal{K}$ with the property that
$$\mathop{\mathbb{E}}_{\mathbf{y}\sim\texttt{sampler}(\mathbf{x})} \mathbf{y} = \mathbf{x}. \tag{12}$$
3) A corresponding *estimation scheme* $\texttt{guesser}(\ell, \mathbf{y}, \mathbf{x})$ which uses the randomly chosen $\mathbf{y}$ and the observed value $\ell = \mathbf{f}^{\mathsf{T}}\mathbf{y}$ to produce a "guess" of $\mathbf{f}$. For every linear function $\mathbf{f}$, guesser must satisfy
$$\mathop{\mathbb{E}}_{\mathbf{y}\sim\texttt{sampler}(\mathbf{x})} [\texttt{guesser}(\mathbf{f}^{\mathsf{T}}\mathbf{y}, \mathbf{y}, \mathbf{x})] = \mathbf{f}. \tag{13}$$
For the remainder of this paper, we will use $\tilde{\mathbf{f}}_t$ to denote the random variable $\texttt{guesser}({\mathbf{f}_t}^{\mathsf{T}}\mathbf{y}_t, \mathbf{y}_t, \mathbf{x}_t)$ when the definition of sampler and guesser are clear.

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

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

$$\begin{aligned}
\mathbb{E}[\mathrm{Regret}^{\mathbf{u}}(\mathcal{A}'; \mathbf{f}_1, \ldots, \mathbf{f}_T)] &= \mathbb{E}\left[\sum_{t=1}^{T} {\mathbf{f}_t}^{\mathsf{T}}(\mathbf{y}_t - \mathbf{u})\right] \\
\text{(Law of total expectation)} \quad &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}_t[{\mathbf{f}_t}^{\mathsf{T}}(\mathbf{y}_t - \mathbf{u})]\right] \\
\text{(by (12))} \quad &= \mathbb{E}\left[\sum_{t=1}^{T} {\mathbf{f}_t}^{\mathsf{T}}(\mathbf{x}_t - \mathbf{u})\right] \\
\text{(by (13))} \quad &= \mathbb{E}\left[\sum_{t=1}^{T} \mathbb{E}_t[\tilde{\mathbf{f}}_t^{\mathsf{T}}(\mathbf{x}_t - \mathbf{u})]\right] \\
\text{(law of total expectation)} \quad &= \mathbb{E}\left[\sum_{t=1}^{T} \tilde{\mathbf{f}}_t^{\mathsf{T}}(\mathbf{x}_t - \mathbf{u})\right] \\
&= \mathbb{E}[\mathrm{Regret}^{\mathbf{u}}(\mathcal{A}; \tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_T)].
\end{aligned}$$

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

Lemma 5.1: Assume we are given any full-information algorithm $\mathcal{A}$ and any sampling and estimation schemes sampler, guesser. If we let the associated bandit algorithm be $\mathcal{A}' := \mathrm{BanditReduction}(\mathcal{A}, \texttt{sampler}, \texttt{guesser})$, then the expected regret of the (randomized) algorithm $\mathcal{A}'$ on the fixed sequence $\{\mathbf{f}_t\}$ is equal to the expected regret of the (deterministic[5]) algorithm $\mathcal{A}$ on the random sequence $\{\tilde{\mathbf{f}}_t\}$. That is,
$$\mathbb{E}[\mathrm{Regret}^{\mathbf{u}}(\mathcal{A}'; \mathbf{f}_1, \ldots, \mathbf{f}_T)] = \mathbb{E}[\mathrm{Regret}^{\mathbf{u}}(\mathcal{A}; \tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_T)].$$

This Lemma is quite powerful: it says that we can construct a bandit algorithm from a full-information one, achieve a bandit regret bound in terms of the full-information bound, and we need only construct sampling and estimation schemes which satisfy the properties in (12) and (13).

B. Dilemma of Bandit Optimization

At first glance, Lemma 5.1 may appear to be a slam dunk: as long as we have a full-information algorithm $\mathcal{A}$ with a low-regret guarantee, we can seemingly construct a Bandit version $\mathcal{A}'$ with an identical regret guarantee in expectation. The remaining difficulty, which may not be so obvious, is that the regret of $\mathcal{A}$ is taken with respect to the random estimates $\{\tilde{\mathbf{f}}_t\}$, and these estimates can unfortunately have very high variance! In general, the typical bound on $\mathrm{Regret}(\mathcal{A}; \mathbf{f}_1, \ldots, \mathbf{f}_T)$ will scale with the magnitude of the $\mathbf{f}_t$'s, and this can be quite bad if the $\mathbf{f}_t$'s can grow arbitrarily large.

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

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

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

It is easily checked that these sampling and estimation schemes satisfy the desired requirements (13) and (12). The downside that the magnitude of $\tilde{\mathbf{f}}$ can grow with $\max\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\}$ (assuming here that $\|\mathbf{f}\| = O(1)$). While the careful reader may notice that things are not so bad in expectation, as $\mathbb{E}\|\tilde{\mathbf{f}}\| = O(1)$, the typical regret bound generally depends on $\mathbb{E}\|\tilde{\mathbf{f}}\|^2$ which grows with $\max\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\}$. If we apply the strong-convexity result from Section III-C, and by correctly choosing η, we would have a regret bound scaling with the quantity $\mathbb{E}\left[\sqrt{\sum_{t=1}^T \|\tilde{\mathbf{f}}_t\|^{*2}}\right] \leq \sqrt{\sum_{t=1}^T \mathbb{E}\|\tilde{\mathbf{f}}_t\|^{*2}}$. To obtain a rate of roughly $O(\sqrt{T})$ it is necessary that we have $\mathbb{E}\|\tilde{\mathbf{f}}_t\|^{*2} = O(1)$.

Perhaps our sampling and estimation schemes could have been better designed? Unfortunately no: the variance of $\tilde{\mathbf{f}}$ cannot be untethered from $\max\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\}$. This example sheds light on a crucial issue of Bandit optimization: how does one handle estimation variance when $\mathbf{x}$ is close to the boundary? Note that the aforementioned example does not lead to difficulty when $\max\{\frac{1}{\alpha}, \frac{1}{1-\alpha}\} = O(1)$. A common approach, used in various forms throughout the literature [5]–[10], is simply to restrict $\mathbf{x} = \alpha\mathbf{e}_1 + (1-\alpha)\mathbf{e}_2$ away from the boundary, requiring that $\alpha \in [\gamma, 1-\gamma]$ for some appropriately chosen $\gamma \in (0, 1/2)$. This restriction does have the benefit of guaranteeing $\mathbb{E}\|\tilde{\mathbf{f}}\|^2 = O(1/\gamma)$, but comes at a price: a γ-perturbation means we can only compete with a suboptimal comparator, and this approximation shall give an additive $O(\gamma T)$ in the regret bound.

The solution, which we present in Section VI, is based on measuring the function $\tilde{\mathbf{f}}_t$ with a *local* norm. This was our original aim in developing the FTRL algorithms based on self-concordant barrier functions: they allow us to obtain a regret bound which measures each $\tilde{\mathbf{f}}_t$ in a way that depends on the current hypothesis $\mathbf{x}_t$. Indeed, the norm $\|\cdot\|^*_{\mathbf{x}_t}$, which *locally* measures $\tilde{\mathbf{f}}_t$ is precisely what we shall need. Ultimately, we will show that, with the correct choice of sampling scheme, we can always guarantee that $\|\tilde{\mathbf{f}}_t\|^{*2}_{\mathbf{x}_t} = O(1)$.

C. Main Result

We now describe the primary contribution of this paper, which is an efficient algorithm for Bandit linear optimization that achieves a $\sqrt{T}$-regret bound. We call this algorithm SCRiBLe, standing for self-concordant regularization in bandit learning.

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

1) A self-concordant barrier function $\mathcal{R}$ for the set $\mathcal{K}$ (see Section II-A1).
2) The full-information algorithm FTRL (Section III-B) using the barrier $\mathcal{R}$ as the regularization.
3) A sampling scheme $\texttt{sampler}(\mathbf{x})$ based on the Dikin ellpisoid $W_1(\mathbf{x})$ (see Section II-A1) chosen according to $\mathcal{R}$. Specifically, if we denote $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ and $\{\lambda_1, \ldots, \lambda_n\}$ as the eigenvalues and eigenvectors of $\nabla^2\mathcal{R}(\mathbf{x}_t)$, the algorithm will sample $\mathbf{y}_t \leftarrow \mathbf{x}_t \pm \lambda_i^{-1/2}\mathbf{e}_i$, one of the $2n$ poles of the Dikin ellipsoid, uniformly at random.
4) An estimation scheme $\texttt{guesser}(\cdot,\cdot,\cdot)$ which produces estimates aligned with eigenpoles of $W_1(\mathbf{x})$. Specifically, corresponding to the eigenpole chosen by $\texttt{sampler}$, $\texttt{guesser}$ outputs
$$\tilde{\mathbf{f}}_t \leftarrow \pm n\,(\mathbf{f}_t{}^\top \mathbf{y}_t)\,\lambda_i^{1/2} \cdot \mathbf{e}_i.$$
5) An improved regret bound for self-concordant functions using local norms (see Section IV-A)

We now state the main result of this paper.

Theorem 5.1: Let $\mathcal{K}$ be a compact convex set and $\mathcal{R}$ be a ϑ-self-concordant barrier on $\mathcal{K}$. Assume $|\mathbf{f}_t{}^\top \mathbf{x}| \leq L$ for any $\mathbf{x} \in \mathcal{K}$ and any t. Setting $\eta = \sqrt{\frac{\vartheta \log T}{2n^2L^2T}}$, the regret of SCRiBLe (Algorithm 4) is bounded as

$$\mathbb{E}[\mathrm{Regret}^{\mathbf{u}}(\texttt{SCRiBLe}; \mathbf{f}_1, \ldots, \mathbf{f}_T)] \leq nL\sqrt{8\vartheta T \log T} + 2L$$

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

Proof: SCRiBLe is exactly in the template of Algorithm 3, using the full-information algorithm $\mathrm{FTRL}_{\mathcal{R}}$ and with $\texttt{sampler}(\cdot)$ and $\texttt{guesser}(\cdot,\cdot,\cdot)$ that satisfy properties (12) and (13), respectively. By Lemma 5.1, we can write

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

We now apply Theorem 4.1. Notice that its conditions are satisfied since with probability one

$$\begin{aligned} \eta\|\mathbf{f}_t\|^*_{\mathbf{x}_t} &= \eta\sqrt{\tilde{\mathbf{f}}_t^\top \nabla^{-2}\mathcal{R}(\mathbf{x}_t)\tilde{\mathbf{f}}_t} \\ &= \eta n\,|\mathbf{f}_t{}^\top \mathbf{y}_t|\,\sqrt{\lambda_{i_t}\mathbf{e}_{i_t}{}^\top \nabla^{-2}\mathcal{R}(\mathbf{x}_t)\mathbf{e}_{i_t}} \\ &= \eta n\,|\mathbf{f}_t{}^\top \mathbf{y}_t| \leq \eta n L \\ &\leq nL\sqrt{\frac{\vartheta \log T}{2n^2L^2T}} \leq \frac{1}{4} \end{aligned}$$

where the last inequality is by the condition on $\frac{T}{\log T}$. From the above calculation $\|\mathbf{f}_t\|^{*2}_{\mathbf{x}_t} \leq n^2L^2$. Hence, we obtain for any $\mathbf{u} \in \mathcal{K}$

$$\begin{aligned} &\mathbb{E}[\mathrm{Regret}^{\mathbf{u}}(\mathrm{FTRL}_{\mathcal{R}}; \tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_T)] \\ &\quad \leq 2\eta\mathbb{E}\left[\sum_{t=1}^T \|\tilde{\mathbf{f}}_t\|^{*2}_{\mathbf{x}_t}\right] + \eta^{-1}\mathcal{R}(\mathbf{u}) \\ &\quad \leq 2\eta n^2L^2T + \eta^{-1}\mathcal{R}(\mathbf{u}). \end{aligned}$$

If $\mathbf{u}$ is such that $\pi_{\mathbf{x}_1}(\mathbf{u}) \leq 1 - \frac{1}{T}$, then by Theorem 2.2, we have that

$$\mathcal{R}(\mathbf{u}) \leq \vartheta \log T. \tag{14}$$

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

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

■

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

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

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

Computational efficiency: Algorithm 4 can be implemented to run in polynomial time: at every iteration the eigenvectors of the Hessian of $\mathcal{R}$ need be computed, and this can be carried out in time $\tilde{O}(n^3)$. Besides other elementary computations, the dominating operation is solving the mathematical program in stage 4. The latter can be performed efficiently, as detailed in Section IV-C. The previously known algorithms for bandit linear optimization that achieve optimal regret guarantees are based on discretizing the set $\mathcal{K}$, rendering the method computationally infeasible [12], [14]. Algorithm 4 is the first efficient algorithm for bandit linear optimization with optimal regret bound.

Algorithm 4 SCRiBLe

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

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

10: **end for**

We remark that it is also possible to analyze an iterative version of our bandit algorithm, based on Algorithm 2 and the ideas presented in this section, and obtain the same asymptotic regret bounds as Algorithm 4.

VI. Conclusion

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

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

Acknowledgment

We would like to thank Peter Bartlett for numerous illuminating discussions.

References

[1] H. Robbins, "Some aspects of the sequential design of experiments," *Bull. Amer. Math. Soc.*, vol. 58, no. 5, pp. 527–535, 1952.
[2] N. Merhav and M. Feder, "Universal prediction," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2124–2147, 1998.
[3] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge, U.K.: Cambridge Univ. Press, 2006.
[4] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. Appl. Math.*, vol. 6, pp. 4–22, 1985.
[5] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, 2003.
[6] H. B. McMahan and A. Blum, "Online geometric optimization in the bandit setting against an adaptive adversary," in *Proc. Annu. Conf. Learning Theory*, 2004, pp. 109–123.
[7] B. Awerbuch and R. D. Kleinberg, "Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches," in *Proc. 36th Annu. ACM Symp. Theory Comput.*, New York, 2004, pp. 45–53.
[8] A. D. Flaxman, A. T. Kalai, and H. B. McMahan, "Online convex optimization in the bandit setting: Gradient descent without a gradient," in *Proc. Annu. ACM-SIAM Symp. Discrete Algorithm*, 2005, pp. 385–394.
[9] V. Dani and T. P. Hayes, "Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary," in *Proc. 17th Annu. ACM-SIAM Symp. Discrete Algorithm*, New York, 2006, pp. 937–943.
[10] A. György, T. Linder, G. Lugosi, and G. Ottucsák, "The on-line shortest path problem under partial monitoring," *J. Mach. Learn. Res.*, vol. 8, pp. 2369–2403, 2007.
[11] A. Rakhlin, A. Tewari, and P. Bartlett, Closing the gap between bandit and full-information online optimization: High-probability regret bound EECS Dept., UC Berkeley, Berkeley, Tech. Rep. UCB/EECS-2007-109, Aug. 2007.
[12] V. Dani, T. Hayes, and S. Kakade, "The price of bandit information for online optimization," in *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Cambridge, MA: MIT Press, 2008.
[13] B. Awerbuch and R. Kleinberg, "Online linear optimization and adaptive routing," *J. Comput. Syst. Sci.*, vol. 74, no. 1, pp. 97–114, 2008.
[14] P. Bartlett, V. Dani, T. Hayes, S. Kakade, A. Rakhlin, and A. Tewari, "High-probability regret bounds for bandit online linear optimization," in *Proc. Annu. Conf. Learning Theory*, 2008, pp. 335–342.
[15] N. Karmarkar, "New polynomial-time algorithm for linear programming," *Combinatorica*, vol. 4, pp. 373–395, 1984.
[16] Y. E. Nesterov and A. S. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming*. Philadelphia, PA: SIAM, 1994.
[17] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, ser. MPS/SIAM Series on Optimization. Philadelphia, PA: SIAM, 2001, vol. 2.
[18] A. Nemirovskii, Interior point polynomial time methods in convex programming, ser. Lecture Notes, 2004.

[19] A. Nemirovski and M. Todd, "Interior-point methods for optimization," *Acta Numerica*, vol. 17, pp. 191–234, 2008.
[20] S. Shalev-Shwartz and Y. Singer, "A primal-dual perspective of online learning algorithms," *Mach. Learn.*, vol. 69, no. 2–3, pp. 115–142, 2007.
[21] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. Int. Conf. Machine Learning*, 2003, pp. 928–936.
[22] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*. New York: Springer-Verlag, 1988.
[23] D. Bertsimas and S. Vempala, "Solving convex programs by random walks," *J. ACM*, vol. 51, pp. 540–556, Jul. 2004.
[24] J. Abernethy and A. Rakhlin, "Beating the adaptive bandit with high probability," presented at the Annu. Conf. Learning Theory, 2009.

Jacob D. Abernethy received a B.Sc. from the Massachusetts Institute of Technology in 2002, a Masters in Computer Science from the Toyota Technological Institute at Chicago in 2006, and his Ph.D. in Computer Science from the University of California at Berkeley. He is currently a Simons Postdoctoral Fellow at the University of Pennsylvania in the Department of Computer Science.

Elad Hazan received the B.Sc. and M.Sc. degrees in computer science from Tel Aviv University in 2001 and 2002 respectively, and Ph.D. in computer science from Princeton University in 2006. From 2006 to 2010 he was a research staff member of the Theory Group at the IBM Almaden Research Center. Since 2010, he has been on the faculty in the Department of Operations Research, Faculty of Industrial Engineering and Management at the Technion—Israel Institute of Technology. His interests include machine learning, optimization, game theory and computational complexity.

Alexander Rakhlin received the B.A. degrees in Mathematics and Computer Science from Cornell University in 2000, and Ph.D. in Computational Neuroscience from MIT in 2006. From 2006 to 2009 he was a postdoctoral fellow at the Department of Electrical Engineering and Computer Sciences, UC Berkeley. Since 2009, he has been on the faculty in the Department of Statistics, University of Pennsylvania. He has been a co-director of Penn Research in Machine Learning (PRiML) since 2010. His interests include machine learning, statistics, optimization, and game theory.