An Efficient Algorithm for Bandit Linear Optimization

Jacob Duncan Abernethy
Elad Hazan
Alexander Rakhlin

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2008-18

http://www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-18.html

February 21, 2008

An Efficient Algorithm for Bandit Linear Optimization

Jacob Abernethy
Computer Science Division
UC Berkeley
jake@cs.berkeley.edu

Elad Hazan
IBM Almaden
hazan@us.ibm.com

Alexander Rakhlin
Computer Science Division
UC Berkeley
rakhlin@cs.berkeley.edu

February 21, 2008

Abstract

We introduce an *efficient* algorithm for the problem of online linear optimization in the bandit setting which achieves the optimal $O^*(\sqrt{T})$ regret. The setting is a natural generalization of the non-stochastic multi-armed bandit problem, and the existence of an efficient optimal algorithm has been posed as an open problem in a number of recent papers. We show how the difficulties encountered by previous approaches are overcome by the use of a self-concordant potential function. Our approach presents a novel connection between online learning and interior point methods.

1 Introduction

One's ability to learn and make decisions rests heavily on the availability of feedback. Indeed, an agent may only improve itself when it can reflect on the outcomes of its own taken actions. In many environments feedback is readily available: a gambler, for example, can observe entirely the outcome of a horse race regardless of where he placed his bet. But such perspective is not always available in hindsight. When the same gambler chooses his route to travel to the race track, perhaps at a busy hour, he will likely never learn the outcome of possible alternatives. When betting on horses, the gambler has thus the benefit (or perhaps the detriment) to muse *"I should have done..."*, yet when betting on traffic he can only think *"the result was..."*.

This problem of sequential decision making was stated by Robbins [19] in 1952 and was later termed "the multi-armed bandit problem". The name inherits from the model whereby, on each of a sequence of rounds, a gambler must pull the arm on one of several slot machines ("one-armed bandits") that each returns a reward chosen stochastically from a fixed distribution. Of course, an ideal strategy would simply be to pull the arm of the machine with the greatest rewards. However, as the gambler does not know the best arm a priori, his goal is then to maximize the reward of his strategy relative to reward he would receive had he known the optimal arm. This problem has gained nuch interest over the past 20 years in a number of fields, as it presents a very natural model of an agent seeking to simultaneously explore the world while exploiting high-reward actions.

As early as 1990 [8, 13] the sequential decision problem was studied under *adversarial* assumptions, where we assume the environment may even try to hurt the learner. The multi-armed bandit problem was brought into the adversarial learning model in 2002 by Auer et al [1], who showed that one may obtain nontrivial guarantees on the gambler's performance relative to the best arm *even when the arm values are chosen by adversary*! In particular, Auer et al [1] showed that the gambler's *regret*, i.e. the difference between the gain of the best arm minus the gain of the gambler, can be bounded by $O(\sqrt{NT})$ where N is the number of bandit arms, and T is the length of the game. In comparison to the game where the gambler is given full information about alternative arms (such as the horse racing example mentioned above), it is possible to obtain $O(\sqrt{T \log N})$, which scales better in N but identically in T.

One natural and well studied problem which escapes the Auer et al result, is online shortest path. In this problem the decision set is exponentially large (i.e. set of all paths in a given graph), and the straightforward reduction of modeling each path as an arm for the multi-armed bandit problem suffers from both efficiency issues as well as exponential regret. To cope with these issues, several authors [2, 9, 14] have recently proposed a very natural generalization of the multi-armed bandit problem to field of Convex Optimization, and we will call this "bandit linear optimization". In this setting we imagine that, on each round t, an adversary chooses some linear function $f_t(\cdot)$ which is not revealed to the player. The player then chooses a point $\mathbf{x}_t$ within some given convex set[1] $\mathcal{K} \subset \mathbb{R}^n$. The player then suffers $f_t(\mathbf{x}_t)$ and this quantity is revealed to him. This process continues for T rounds, and at the end the learner's payoff is his *regret*:

$$R_T = \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x}^* \in \mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}^*).$$

Online linear optimization has been often considered, yet primarily in the full-information setting where the learner sees all of $f_t(\cdot)$ rather than just $f_t(\mathbf{x}_t)$. In the full-information model, it has been known for some time that the optimal regret bound is $O(\sqrt{T})$, and it had been conjectured that the same should hold for the bandit setting as well. Nevertheless, several initially proposed algorithms were shown only to obtain bounds with $O(T^{3/4})$ (e.g. [14, 9]) or $O(T^{2/3})$ (e.g. [2, 7]). Only recently was this conjecture proven to be true by Dani et al. [6], who provided an algorithm with $O(poly(n)\sqrt{T})$ regret. However, their proposed method, which deploys a clever reduction to the multi-armed bandit algorithm of Auer et al [1], is not efficient.

We propose an algorithm for online linear bandit optimization that is the first, we believe, to be both computationally efficient and achieve a $O(poly(n)\sqrt{T})$ regret bound. Moreover, with a thorough analysis we aim to shed light on the difficulties in obtaining such an algorithm. Our technique provides a curious link between the notion of Bregman divergences, which have often been used for constructing and analyzing online learning algorithms, and self-concordant barriers, which are of great importance in the study of interior point methods in convex optimization. A rather surprising consequence is that divergence functions, which are widely used as a regularization tool in online learning, are also entirely necessary in our algorithm for the purpose of managing uncertainty. To our knowledge, this is the first time such connections have been made.

2 Notation and Motivation

Let $\mathcal{K} \subset \mathbb{R}^n$ be a compact closed convex set. For two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we denote their dot product as $\mathbf{x}^\mathsf{T}\mathbf{y}$. We write $A \succeq B$ if $(A - B)$ is positive semi-definite. Let

$$D_{\mathcal{R}}(\mathbf{x}, \mathbf{y}) := \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{y}) - \nabla \mathcal{R}(\mathbf{y})^\mathsf{T}(\mathbf{x} - \mathbf{y})$$

[1] In the case of online shortest path, the convex set can be represented as a set of vectors in $\mathbb{R}^{|E|}$. Hence, the dependence on number of paths in the graph can be circumvented.

be the *Bregman divergence* between $\mathbf{x}$ and $\mathbf{y}$ with respect to a convex differentiable $\mathcal{R}$.

Define the Minkowsky function (see page 34 of [16] for details) on $\mathcal{K}$, parametrized by a pole $\mathbf{y}_t$ as

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

We define a scaled version of $\mathcal{K}$ by

$$\mathcal{K}_\delta = \{\mathbf{u} : \pi_{\mathbf{x}_1}(\mathbf{u}) \leq (1+\delta)^{-1}\}$$

for $\delta > 0$. Here $\mathbf{x}_1$ is a "center" of $\mathcal{K}$ defined in the later sections. We assume that $\mathcal{K}$ is not "flat" and so $\mathbf{x}_1$ is a constant distance away from the boundary.

In the rest of the section we describe the rich body of previous work which led to our result. The reader familiar with online optimization in the full and partial information settings can skip directly to the next section.

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

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

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

The goal of the Player is not simply to minimize his total loss $\sum_{t=1}^{T} \mathbf{f}_t^\mathsf{T}\mathbf{x}_t$, 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* R_T defined as

$$R_T := \sum_{t=1}^{T} \mathbf{f}_t^\mathsf{T}\mathbf{x}_t - \min_{\mathbf{x}^* \in \mathcal{K}} \sum_{t=1}^{T} \mathbf{f}_t^\mathsf{T}\mathbf{x}^*.$$

When the objective is his regret, the Player is not competing against arbitrary strategies, he need only perform well relative to the total loss of the single best fixed point in $\mathcal{K}$.

We distinguish the *full-information* and *bandit* versions of the above problem. The full-information version, the Player may observe the entire function $\mathbf{f}_t$ as his feedback $\Im$ and can exploit this in making his decisions. In this paper we study the more challenging bandit setting, where the feedback $\Im$ provided to the player on round t is only the scalar value $\mathbf{f}_t^\mathsf{T}\mathbf{x}_t$. This is significantly less information for the Player: instead of observing the entire function $\mathbf{f}_t$, he may only witness the value of $\mathbf{f}_t$ *at a single point.*

2.1 Algorithms Based on Full Information

All previous work on bandit online learning, including the present one, relies heavily on techniques developed in the full-information setting and we now give a brief overview of some well-known approaches.

Follow The Leader (FTL) is perhaps the simplest online learning strategy one might think of: the player simply uses the heuristic "select the best choice thus far". For the online optimization task we study, this can be written as

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

For certain types of problems, applying FTL does guarantee low regret. Unfortunately, when the loss functions $\mathbf{f}_t$ are linear on the input space it can be shown that FTL will suffer regret that grows linearly in T. A natural approach, and more well-known within statistical learning, is to *regularize* the optimization problem (1). That is, an appropriate regularization function $\mathcal{R}(\mathbf{x})$ and a trade-off parameter λ are selected, and the prediction is obtained as

$$\mathbf{x}_{t+1} := \arg\min_{\mathbf{x}\in\mathcal{K}} \left[\sum_{s=1}^{t} \mathbf{f}_s^{\mathsf{T}}\mathbf{x} + \lambda\mathcal{R}(\mathbf{x})\right]. \tag{2}$$

We call the above approach Follow The Regularized Leader (FTRL). An alternative way to view this exact algorithm is by sequential updates, which capture the difference between consecutive solutions for FTRL. Given that $\mathcal{R}$ is convex and differentiable, the general form of this update is

$$\bar{\mathbf{x}}_{t+1} = \nabla\mathcal{R}^*(\nabla\mathcal{R}(\bar{\mathbf{x}}_t) - \eta\mathbf{f}_t), \tag{3}$$

followed by a projection onto $\mathcal{K}$ with respect to the divergence $D_{\mathcal{R}}$:

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{u}\in\mathcal{K}} D_{\mathcal{R}}(\mathbf{u}, \bar{\mathbf{x}}_{t+1}).$$

Here $\mathcal{R}^*$ is the Fenchel dual function and η is a parameter. This procedure is known as the mirror descent (e.g. [5]).

Applying the above rule we see that the well known Online Gradient Descent algorithm [21, 10] is derived[2] by choosing the regularizer to be the squared Euclidean norm. Similarly, the Exponentiated Gradient [12] algorithm is obtained with the entropy function as the regularizer.

This unified view of various well-known algorithms as solutions to regularization problems gives us an important degree of freedom of choosing the regularizer. Indeed, we will choose a regularizer for our problem that possesses key properties needed for the regret to scale as $O(\sqrt{T})$. In Section 4, we give a bound on the regret for (2) with any regularizer $\mathcal{R}$ and in Section 5 we will discuss the specific $\mathcal{R}$ used in this paper.

2.2 The Dilemma of Bandit Optimization

Effectively all previous algorithms for the Bandit setting have utilized a reduction to the full-information setting in one way or another. This is reasonable: any algorithm that aimed for low-regret in the bandit setting would necessarily have to achieve low regret given full information. Furthermore, as the full-information online learning setting is relatively well-understood, it is natural to exploit such techniques for this more challenging problem.

The crucial reduction that has been utilized by several authors [1, 2, 6, 7, 14] is the following. First choose some full-information online learning algorithm $\mathcal{A}$. $\mathcal{A}$ will receive input vectors $\mathbf{f}_1, \ldots, \mathbf{f}_t$, corresponding to previously observed functions, and will return some point $\mathbf{x}_{t+1} \in \mathcal{K}$ to predict. On every round t, do one *or both* of the following:

- Query $\mathcal{A}$ for its prediction $\mathbf{x}_t$ and either predict $\mathbf{x}_t$ exactly or in expectation.
- Construct some random estimate $\tilde{\mathbf{f}}_t$ in such a way that $\mathbb{E}\tilde{\mathbf{f}}_t = \mathbf{f}_t$, and input $\tilde{\mathbf{f}}_t$ into $\mathcal{A}$ as though it had been observed on this round

[2] Strictly speaking, this equivalence is true if the updates are applied to unprojected versions of $\mathbf{x}_t$.

The key idea here is simple: so long as we are roughly predicting $\mathbf{x}_t$ per advice of $\mathcal{A}$, and so long as we are "guessing" $\mathbf{f}_t$ (i.e. so that the estimates $\tilde{\mathbf{f}}_t$ is correct in expectation), then we can guarantee low regret. This approach is validated in Lemma 4.2 which shows that, as long as $\mathcal{A}$ performs well against the random estimates $\tilde{\mathbf{f}}_t$ in expectation, then we will also do well against the true functions $\mathbf{f}_1, \ldots, \mathbf{f}_T$.

This observation is quite reassuring yet unfortunately does not address a significant obstacle: *how can we simultaneously estimate* $\tilde{\mathbf{f}}_t$ *and predict* $\mathbf{x}_t$ *when only one query is allowed?* The algorithm faces an inherent dilemma: whether to follow the advice of $\mathcal{A}$ of predicting $\mathbf{x}_t$, or to try to estimate $\mathbf{f}_t$ by sampling in a wide region around $\mathcal{K}$, possibly hurting its performance on the given round. This exploration-exploitation trade-off is the primary source of difficulty in obtaining $O(\sqrt{T})$ guarantees on the regret.

Roughly two categories of approaches have been suggested to perform both exploration and exploitation:

1. **Alternating Explore/Exploit:** Flip an ϵ-biased coin to determine whether to explore or exploit. On explore rounds, sample uniformly on some wide region around $\mathcal{K}$ and estimate $\mathbf{f}_t$ accordingly, and input this into $\mathcal{A}$. On exploit rounds, query $\mathcal{A}$ for $\mathbf{x}_t$ and predict this.

2. **Simultaneous Explore/Exploit:** Query $\mathcal{A}$ for $\mathbf{x}_t$ and construct a random vector X_t such that $\mathbb{E}X_t = \mathbf{x}_t$. Construct $\tilde{\mathbf{f}}_t$ randomly based on the outcome of X_t and the learned value $\mathbf{f}_t^\top X_t$.

The methods of [14, 2, 7] fit within in the first category but, unfortunately, fail to obtain the desired $O(poly(n)\sqrt{T})$ regret. This is not surprising: it has been suggested by [7] that $\Omega(T^{2/3})$ regret is unavoidable by any algorithm in which the observation $\mathbf{f}_t^\top \mathbf{x}_t$ is ignored on rounds pledged for exploitation. Algorithms falling into the second category, such as those of [1, 7, 9], are more sophisticated and help to motivate our results. We review these methods below.

2.3 Methods For Simultaneous Exploration and Exploition

On first glance, it is rather surprising that one can perform the task of predicting some $\mathbf{x}_t$ (in expectation) while, simultaneously, finding an unbiased estimate of $\mathbf{f}_t$. To get a feel for how this can be done, we briefly review the methods of [1] and [9] below.

The work of **Auer et al [1]** is not, strictly speaking, a bandit optimization problem but instead the more simple "Multi-armed bandit" problem. The authors consider the problem of sequentially choosing one of N "arms" each of which contains a hidden loss where the learner may only see the loss of his chosen arm. The regret, in this case, is the learner's minus the smallest cumulative loss over all arms. This multi-armed bandit problem can indeed be cast as a bandit optimization problem: let $\mathcal{K}$ be the N-simplex (convex hull of $\{\mathbf{e}_1, \ldots, \mathbf{e}_N\}$), let $\mathbf{f}_t$ be identically the vector of hidden losses on the set of arms, and note that $\min_{\mathbf{x} \in \mathcal{K}} \sum \mathbf{f}_s^\top \mathbf{x} = \min_i \sum \mathbf{f}_s[i]$.

The algorithm of [1], EXP3, utilizes EG (mentioned earlier) as its black box full-information algorithm $\mathcal{A}$. First, a point $\mathbf{x}_t \in \mathcal{K}$ is returned by $\mathcal{A}$. The hypothesis $\mathbf{x}_t$ is then *biased* slightly:

$$\mathbf{x}_t \leftarrow (1-\gamma)\mathbf{x}_t + \gamma \left\langle \frac{1}{n}, \ldots, \frac{1}{n} \right\rangle .$$

We describe the need for this bias Section 2.4. EXP3 then randomly chooses one of the corners of $\mathcal{K}$ according to the distribution $\mathbf{x}_t$ and uses this as its prediction. More precisely, a basis vector

$\mathbf{e}_i$ is sampled with probability $\mathbf{x}_t[i]$ and clearly $\mathbb{E}_{I \sim \mathbf{x}_t} \mathbf{e}_I = \mathbf{x}_t$. Once we observe $\mathbf{f}_t^\mathsf{T} \mathbf{e}_i = \mathbf{f}_t[i]$, the estimate is constructed as follows:

$$\tilde{\mathbf{f}}_t := \left\langle \frac{\mathbf{f}_t[i]}{\mathbf{x}_t[i]} \mathbf{1}[j = i] \right\rangle_j .$$

It is very easy to check that $\mathbb{E}\tilde{\mathbf{f}}_t = \mathbf{f}_t$.

Flaxman et al [9] developed a bandit optimization algorithm that used OGD as the full-information subroutine $\mathcal{A}$. Their approach uses a quite different method of performing exploration and exploitation. On each round, the algorithm queries $\mathcal{A}$ for a hypothesis $\mathbf{x}_t$ and, as in [1], this hypothesis is biased slightly:

$$\mathbf{x}_t \leftarrow (1 - \gamma)\mathbf{x}_t + \gamma \mathbf{u}$$

where $\mathbf{u}$ is some "center" vector of the set $\mathcal{K}$. Similarly to EXP3, the algorithm doesn't actually predict $\mathbf{x}_t$. The algorithm determines the distance r to the boundary of the set, and a vector $r\mathbf{v}$ is sampled uniformly at random from a sphere of radius r. The prediction is $\mathbf{y}_t := \mathbf{x}_t + r\mathbf{v}$ and indeed $\mathbb{E}\mathbf{y}_t = \mathbf{x}_t + r\mathbb{E}\mathbf{v} = \mathbf{x}_t$ as desired. The algorithm predicts $\mathbf{y}_t$, receives feedback $\mathbf{f}_t^\mathsf{T} \mathbf{y}_t$, and function $\mathbf{f}_t$ is estimated as

$$\tilde{\mathbf{f}}_t := \frac{\mathbf{f}_t^\mathsf{T} \mathbf{y}_t}{r} \mathbf{v}.$$

It is, again, easy to check that this provides an unbiased estimate of $\mathbf{f}_t$.

2.4 The Curse of High Variance and the Blessing of Regularization

Upon inspecting the definitions of $\tilde{\mathbf{f}}_t$ in the method of Auer et al and Flaxman et al it becomes apparent that the estimates are inversely proportional to the distance of $\mathbf{x}_t$ to the boundary. This implies high variance of the estimated functions. At first glance, this seems to be a disaster. Indeed, most full-information algorithms scale linearly with the magnitude of the functions played by the environment. Let us take a closer look at how exactly this leads to the suboptimality of the algorithm of Flaxman et al.

The bound on the expected regret of OGD on $\tilde{\mathbf{f}}_t$'s involves terms $\mathbb{E}\|\tilde{\mathbf{f}}_t\|^2$ (see proof of Lemma 4.1), which scale as the inverse of the squared distance to the boundary. Biasing of $\mathbf{x}_t$ away from the boundary leads to an upper bound on this quantity of the order γ^{-2}. Unfortunately, γ cannot be taken to be large. Indeed, the optimal point $\mathbf{x}^*$, chosen in hindsight, lies on the boundary of the set, as the cost functions are linear. Thus, stepping away from the boundary comes at a cost of potentially losing $O(\gamma T)$ over the course of the game. Since the goal is to obtain an $O(\sqrt{T})$ bound on the regret, $\gamma = O(T^{-1/2})$ is the most that can be tolerated. Biasing away from the boundary does reduce the variance of the estimates somewhat; unfortunately, it is not the panacea. To terminate the discussion on the method of Flaxman et al, we state the dependence of the regret bound on the learning rate η and the biasing parameter γ:

$$R_T = O(\eta^{-1} + \gamma^{-2}\eta T + \gamma T).$$

The first term is due to the distance between the initial choice and the comparator; the second is the problematic $\mathbb{E}\|\tilde{\mathbf{f}}_t\|^2$ term summed over time; and the last term is due to stepping away from the boundary. The best choice of the parameters leads to the unsatisfying $O(T^{3/4})$ bound.

From the above discussion it is clear that the problematic term is $\mathbb{E}\|\tilde{\mathbf{f}}_t\|^2 = O(1/r^2)$, owing its high magnitude to its inverse dependence on the squared distance to the boundary. A similar dependence occurs in the estimate of Auer et al, though the non-uniform sampling from the basis implies an $O(1/\mathbf{x}_t[i])$ magnitude. One can ask whether this inverse dependence on the distance is an artifact of these algorithms and can be avoided. In fact, it is possible to prove that this is intrinsic to the problem if we require that $\tilde{\mathbf{f}}_t$ be unbiased and $\mathbf{x}_t$ be the center of the sampling distribution.

Does this result imply that no $O(\sqrt{T})$ bound on the regret is possible? Fortunately, no. If we restrict our search to a regularization algorithm of the type (2), the expected regret can be proved to be *equal* to an expression involving $\mathbb{E}D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1})$ terms. For $\mathcal{R}(\mathbf{x}) \propto \|\mathbf{x}\|^2$ we indeed recover (modulo projections) the method of Flaxman et al with its insurmountable hurdle of $\mathbb{E}\|\tilde{\mathbf{f}}_t\|^2$. Fortunately, other choices of $\mathcal{R}$ have better behavior. Here, the formulation of the regularized minimization (2) as a dual-space mirror descent comes to the rescue.

In the space of gradients (the dual space), the step-wise updates (3) for Follow The Regularized Leader are $\eta\tilde{\mathbf{f}}_t$ no matter what $\mathcal{R}$ we choose. It is a known fact (e.g. [5]) that the divergence in the original space between $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ is equal to the divergence between the corresponding gradients with respect to the dual potential $\mathcal{R}^*$. It is, therefore, not surprising that the dual divergence can be tuned to be small even if $\|\tilde{\mathbf{f}}_t\|$ is very large. Having small divergence corresponds to the requirement that $\mathcal{R}^*$ be "flat" whenever $\|\tilde{\mathbf{f}}_t\|$ is large, i.e. when $\mathbf{x}_t$ is close to the boundary. Flatness in the dual space corresponds to large curvature in the primal. This motivates the use of a potential function $\mathcal{R}$ which becomes more and more curved at the boundary of the set $\mathcal{K}$. In a nutshell, this is the Blessing of Regularization which allows us to obtain an efficient optimal algorithm which was escaping all previous attempts.

Recall that the method of Auer et al attains the optimal $O(\sqrt{T})$ rate *but only when $\mathcal{K}$ is the simplex.* If our intuition about the importance of regularization is sound, we should find that the method uses a potential which curves at the edges of the simplex. One can see that the exponential weights (more generally, EG) used by Auer et al corresponds to regularization with $\mathcal{R}$ being the entropy function $\mathcal{R}(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{x}[i] \log \mathbf{x}[i]$. Taking the second derivative, we see that, indeed, the curvature increases as $1/\mathbf{x}[i]$ as $\mathbf{x}$ gets closer to the boundary. For the present paper, we will actually choose a regularizer that curves as inverse *squared* distance to the boundary. The reader can probably guess that such a regularizer should be defined, roughly, as the log-distance to the boundary.

While for simple convex bodies, such as sphere, existence of a function behaving like log-distance to the boundary seems plausible, a similar statement for general convex sets $\mathcal{K}$ seems very complex. Luckily, this very question has been studied in the theory of Interior Point Methods, and existence and construction of such functions, called *self- concordant barriers*, is well-established.

3 Main Result

We first state our main result: an algorithm for online linear optimization in the bandit setting for an arbitrary compact convex set $\mathcal{K}$. The analysis of this algorithm has a number of facets and we discuss these individually throughout the remainder of this paper. In Section 4 we describe the regularization framework in detail and show how the regret can be computed in terms of Bregman divergences. In Section 5 we review the theory of self-concordant functions and state two important properties of such functions. In Section 6 we highlight several key elements of the proof of our regret bound. In Section 7 we show how this algorithm can be used for one interesting case, namely the bandit version of the Online Shortest Path problem. The precise analysis of our algorithm is given

Algorithm 1 Bandit Online Linear Optimization

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

in Section 8. Finally, in Section 9 we spell out how to implement the algorithm with only one iteration of the Damped Newton method per time step.

The following theorem is the main result of this paper (see Section 5 for the definition of ϑ-self-concordant barrier).

Theorem 3.1. *Let $\mathcal{K}$ be a convex set and $\mathcal{R}$ be a ϑ-self-concordant barrier on K. Let $\mathbf{u}$ be any vector in $\mathcal{K}' = \mathcal{K}_{1/\sqrt{T}}$. Suppose we have the property that $|\mathbf{f}_t^{\mathsf{T}}\mathbf{x}| \leq 1$ for any $\mathbf{x} \in \mathcal{K}$. Setting $\eta = \frac{\sqrt{\vartheta \log T}}{4n\sqrt{T}}$, the regret of Algorithm 1 is bounded as*

$$\mathbb{E}\sum_{t=1}^{T} \mathbf{f}_t^{\mathsf{T}} \mathbf{y}_t \leq \min_{\mathbf{u}\in\mathcal{K}'} \mathbb{E}\left(\sum_{t=1}^{T} \mathbf{f}_t^{\mathsf{T}} \mathbf{u}\right) + 16n\sqrt{\vartheta T \log T}$$

whenever $T > 8\vartheta \log T$.

The expected regret over the original set $\mathcal{K}$ is within an additive $O(\sqrt{nT})$ factor from the above guarantee, as implied by Lemma A.1 in the Appendix.

4 Regularization Algorithms and Bregman Divergences

As our algorithm is clearly based on a regularization framework, we now state a general result for the performance of any algorithm minimizing the regularized empirical loss. We call this method Follow the Regularized Leader, and we defer the proof of the regret bound to the Appendix. A similar analysis for convex loss functions can be found in [5], Chapter 11. We remark that the use of Bregman divergences in the context of online learning goes back at least to Kivinen and Warmuth [12].

Let $\tilde{\mathbf{f}}_1, \ldots, \tilde{\mathbf{f}}_T \in \mathbb{R}^n$ be any sequence of vectors. Suppose $\mathbf{x}_{t+1}$ is obtained as

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{K}} \underbrace{\left[\eta \sum_{s=1}^{t} \tilde{\mathbf{f}}_s^{\mathsf{T}} \mathbf{x} + \mathcal{R}(\mathbf{x})\right]}_{\Phi_t(\mathbf{x})} \tag{4}$$

for some strictly-convex differentiable function $\mathcal{R}$. We denote $\Phi_0(x) = \mathcal{R}(x)$ and $\Phi_t = \Phi_{t-1} + \eta\tilde{\mathbf{f}}_t$.

We will assume $\nabla\mathcal{R}$ approaches infinity at the boundary of $\mathcal{K}$ so that the unconstrained minimization problem will have a unique solution within $\mathcal{K}$. We have the following bound on the performance of such an algorithm.

Lemma 4.1. *For any $\mathbf{u} \in \mathcal{K}$, the algorithm defined by* (4) *enjoys the following regret guarantee*

$$\begin{aligned} \eta \sum_{t=1}^{T} \tilde{\mathbf{f}}_t^{\mathsf{T}}(\mathbf{x}_t - \mathbf{u}) &\leq D_{\mathcal{R}}(\mathbf{u}, \mathbf{x}_1) + \sum_{t=1}^{T} D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) \\ &\leq D_{\mathcal{R}}(\mathbf{u}, \mathbf{x}_1) + \eta \sum_{t=1}^{T} \tilde{\mathbf{f}}_t^{\mathsf{T}}(\mathbf{x}_t - \mathbf{x}_{t+1}) \end{aligned}$$

for any sequence $\{\tilde{\mathbf{f}}_t\}_{t=1}^T$.

In addition, we state a useful result that bounds the true regret based on the regret against the estimated functions $\tilde{\mathbf{f}}_t$.

Lemma 4.2. *Suppose that, for $t = 1, \ldots, T$, $\tilde{\mathbf{f}}_t$ is such that $\mathbb{E}\,\tilde{\mathbf{f}}_t = \mathbf{f}_t$ and $\mathbf{y}_t$ is such that $\mathbb{E}\,\mathbf{y}_t = \mathbf{x}_t$. Suppose that we have the following regret bound:*

$$\sum_{t=1}^{T} \tilde{\mathbf{f}}_t^{\mathsf{T}} \mathbf{x}_t \leq \min_{\mathbf{u} \in \mathcal{K}'} \sum_{t=1}^{T} \tilde{\mathbf{f}}_t^{\mathsf{T}} \mathbf{u} + C_T.$$

Then the expected regret satisfies

$$\mathbb{E}\left(\sum_{t=1}^{T} \mathbf{f}_t^{\mathsf{T}} \mathbf{y}_t\right) \leq \min_{\mathbf{u} \in \mathcal{K}'} \mathbb{E}\left(\sum_{t=1}^{T} \mathbf{f}_t^{\mathsf{T}} \mathbf{u}\right) + C_T.$$

5 Self-concordant Functions and the Dikin ellipsoid

Interior-point methods 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 and solving the unconstrained minimization problem. The rough idea is to gradually reduce the weight of the barrier function as one approaches the solution. The construction of barrier functions for general convex sets has been studied extensively, and we refer the reader to [16, 4] for a thorough treatment on the subject. To be more precise, most of the results of this section can be found in [15], page 22-23, as well as in the aforementioned texts.

5.1 Definitions and Properties

Definition 5.1. *A* self-concordant function $\mathcal{R}$: *int* $\mathcal{K} \to \mathbb{R}$ *is a C^3 convex function such that*

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

Here, the third-order differential is defined as

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

We will further assume that the function approaches infinity for any sequence of points approaching the boundary of $\mathcal{K}$. An additional requirement leads to the notion of a self-concordant *barrier*.

Definition 5.2. *A* ϑ-self-concordant barrier $\mathcal{R}$ *is a self-concordant function with*

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

The generality of interior-point methods comes from the fact that 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.

We note that some of the results of this paper, such as the Dikin ellipsoid, rely on $\mathcal{R}$ being a self-concordant function, while others necessarily require the barrier property. We therefore assume from the outset that $\mathcal{R}$ is a self-concordant barrier.

Since $\mathcal{K}$ is compact, we can assume that $\mathcal{R}$ is non-degenerate. For a given $\mathbf{x} \in \mathcal{K}$, define

$$\langle \mathbf{g},\mathbf{h}\rangle_{\mathbf{x}} = \mathbf{g}^{\mathsf{T}}\nabla^2\mathcal{R}(\mathbf{x})\mathbf{h} \quad \text{and} \quad \|\mathbf{h}\|_{\mathbf{x}} = (\langle \mathbf{h},\mathbf{h}\rangle_{\mathbf{x}})^{-1/2}.$$

This inner product defines the local Euclidean structure at $\mathbf{x}$. Nondegeneracy of $\mathcal{R}$ implies that the above norm is indeed a norm, not a seminorm.

It is natural to talk about a ball with respect to the above norm. Define the open *Dikin ellipsoid* of radius r centered at $\mathbf{x}$ as the set

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

The following facts about the Dikin ellipsoid are central to the results of this paper (we refer to [15], page 23 for proofs). The first non-trivial fact is that $W_1(\mathbf{x}) \subseteq \mathcal{K}$ for any $\mathbf{x} \in \mathcal{K}$. In other words, the inverse Hessian of the self-concordant function $\mathcal{R}$ stretches the space in such a way that the eigenvectors fall in the set $\mathcal{K}$. This is crucial for our sampling procedure. Indeed, our method (Algorithm 1) samples $\mathbf{y}_t$ from the Dikin ellipsoid centered at $\mathbf{x}_t$. Since $W_1(\mathbf{x}_t)$ is contained in $\mathcal{K}$, the sampling procedure is legal.

The second fact is that within the Dikin ellipsoid, that is for $\|\mathbf{h}\|_{\mathbf{x}} < 1$, the Hessians of $\mathcal{R}$ are "almost proportional" to the Hessian of $\mathcal{R}$ at the center of the ellipsoid :

$$\begin{aligned}(1-\|\mathbf{h}\|_{\mathbf{x}})^2\nabla^2\mathcal{R}(\mathbf{x}) &\preceq \nabla^2\mathcal{R}(\mathbf{x}+\mathbf{h}) \\ &\preceq (1-\|\mathbf{h}\|_{\mathbf{x}})^{-2}\nabla^2\mathcal{R}(\mathbf{x}).\end{aligned} \tag{5}$$

This gives us the crucial control of the Hessians for second-order approximations. Finally, if $\|\mathbf{h}\|_{\mathbf{x}} < 1$ (i.e. $\mathbf{x}+\mathbf{h}$ is in the unit Dikin ellipsoid), then for any $\mathbf{z}$,

$$|\mathbf{z}^{\mathsf{T}}(\nabla\mathcal{R}(\mathbf{x}+\mathbf{h}) - \nabla\mathcal{R}(\mathbf{x}))| \leq \frac{\|\mathbf{h}\|_{\mathbf{x}}}{1-\|\mathbf{h}\|_{\mathbf{x}}}\|\mathbf{z}\|_{\mathbf{x}}. \tag{6}$$

Assuming that $\mathcal{R}$ is a ϑ-self-concordant barrier, we have (see page 34 of [16])

$$\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1) \le \vartheta \ln \frac{1}{1-\pi_{\mathbf{x}_1}(\mathbf{u})}.$$

For any $\mathbf{u} \in \mathcal{K}_\delta$, $\pi_{\mathbf{x}_1}(\mathbf{u}) \le (1+\delta)^{-1}$ by definition, implying that $(1-\pi_{\mathbf{x}_1}(\mathbf{u}))^{-1} \le \frac{1+\delta}{\delta}$. We conclude that

$$\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1) \le \vartheta \ln(\sqrt{T}+1) \le 2\vartheta \log T \tag{7}$$

for $\mathbf{u} \in \mathcal{K}_{1/\sqrt{T}}$.

5.2 Examples of Self-Concordant Functions

A nice fact about self-concordant barriers is that $\mathcal{R}_1 + \mathcal{R}_2$ is $\vartheta_1 + \vartheta_2$-self-concordant for ϑ_1-self-concordant $\mathcal{R}_1$ and ϑ_2-self-concordant $\mathcal{R}_2$. For linear constraints $\mathbf{a}^\mathsf{T}\mathbf{x}_t \le b$, the barrier $-\ln(b - \mathbf{a}^\mathsf{T}\mathbf{x}_t)$ is 1-self-concordant. Hence, for a polyhedra defined by m constraints, the corresponding barrier is m-self-concordant. Thus, for the n-dimensional simplex or a cube, $\theta = n$, leading to $n^{3/2}$ dependence on the dimension in the main result.

For the n-dimensional ball,

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

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

6 Sketch of Proof

We have now presented all necessary tools to prove Theorem 3.1: regret in terms of Bregman divergences, self-concordant barriers and the Dikin ellipsoid. While we provide a complete proof in Section 8 here we sketch the key elements of the analysis of our algorithm.

As we tried to motivate in the end of Section 2, any method that can simultaneously (a) predict $\mathbf{x}_t$ in expectation and (b) obtain an unbiased one-sample estimate of $\tilde{\mathbf{f}}_t$ will necessarily suffer from high variance when $\mathbf{x}_t$ is close to the boundary of the set $\mathcal{K}$. As we have hinted previously, we would like our regularizer $\mathcal{R}$ to control the variance. Yet the problem is even more subtle than this: $\mathbf{x}_t$ may be close to the boundary in one dimension while have plenty of space in another, which in turn suggests that $\tilde{\mathbf{f}}_t$ need only have high variance in certain directions.

Quite amazingly, the self-concordant function $\mathcal{R}$ gives us a handle on two key issues. The Dikin ellipsoid, defined in terms $\nabla^2\mathcal{R}(\mathbf{x}_t)$, gives us exactly a rough approximation to the available "space" around $\mathbf{x}_t$. At the same time, $\nabla^2\mathcal{R}(\mathbf{x}_t)^{-1}$ annihilates $\tilde{\mathbf{f}}_t$ in exactly the directions in which it is large. This is absolutely necessary for bounding the regret, as we discuss next.

Lemma 4.1 implies that regret scales with the cumulative divergence $\eta^{-1}\sum_t D_\mathcal{R}(\mathbf{x}_t, \mathbf{x}_{t+1})$ and thus we must have that $\mathbb{E}\, D_\mathcal{R}(\mathbf{x}_t, \mathbf{x}_{t+1}) = O(\eta^2)$ on average to obtain a regret bound of $O(\sqrt{T})$. Analyzing the divergence requires some care and so we provide only a rough sketch here (with more in Section 8). If $\mathcal{R}$ were exactly quadratic then the divergence is

$$D_\mathcal{R}(\mathbf{x}_t, \mathbf{x}_{t+1}) := \eta^2 \tilde{\mathbf{f}}_t^\mathsf{T} (\nabla^2 \mathcal{R}(\mathbf{x}_t))^{-1} \tilde{\mathbf{f}}_t. \tag{8}$$

Even when $\mathcal{R}$ is not quadratic, however, (8) still provides a decent approximation to the divergence and, given certain regularity conditions on $\mathcal{R}$, it is enough to bound the quadratic form $\tilde{\mathbf{f}}_t^{\mathsf{T}}(\nabla^2\mathcal{R}(\mathbf{x}_t))^{-1}\tilde{\mathbf{f}}_t$.

The precise interaction between the Dikin ellipsoid, the estimates $\tilde{\mathbf{f}}_t$, and the divergence $D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1})$ is as follows. Assume we are at the point $\mathbf{x}_t$ and we have computed the unit eigenvectors $\mathbf{e}_1, \ldots, \mathbf{e}_n$ and corresponding eigenvalues $\lambda_1, \ldots, \lambda_n$ of $\nabla^2\mathcal{R}(\mathbf{x}_t)$. Properties of self-concordant functions ensure that the Dikin ellipsoid around $\mathbf{x}_t$ is contained within $\mathcal{K}$ and thus, in particular, so are the points $\mathbf{x}_t \pm \lambda_i^{-1/2}\mathbf{e}_i$ for each i. Assuming the point $\mathbf{y}_t := \mathbf{x}_t + \lambda_j^{-1/2}\mathbf{e}_j$ was sampled and we received the value $\mathbf{f}_t^{\mathsf{T}}\mathbf{y}_t$, we then construct the estimate

$$\tilde{\mathbf{f}}_t := n\sqrt{\lambda_j}(\mathbf{f}_t^{\mathsf{T}}\mathbf{y}_t)\mathbf{e}_j.$$

Notice it is crucial that we scale by $\sqrt{\lambda_j}$, the *inverse* ℓ_2 distance between $\mathbf{x}_t$ and $\mathbf{y}_t$, to ensure that $\mathbf{f}_t$ is unbiased.On the other hand, we see that the divergence is approximately computed as

$$\begin{aligned} D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) &\approx \eta^2\tilde{\mathbf{f}}_t^{\mathsf{T}}\nabla^2\mathcal{R}^{-1}\tilde{\mathbf{f}}_t \\ &= \eta^2 n^2(\mathbf{f}_t^{\mathsf{T}}\mathbf{y}_t)^2\lambda_j(\mathbf{e}_j^{\mathsf{T}}\nabla^2\mathcal{R}^{-1}\mathbf{e}_j) \\ &= \eta^2 n^2(\mathbf{f}_t^{\mathsf{T}}\mathbf{y}_t)^2. \end{aligned}$$

As an interesting and important aside, a *necessary* requirement of the above analysis is that we construct our estimates $\tilde{\mathbf{f}}_t$ from the eigendirections $\mathbf{e}_j$. To see this, imagine that one eigenvalue λ_1 is very large, while another, λ_2 small. This corresponds to a thin and long Dikin ellipsoid, which would occur near a flat boundary. Suppose that instead of eigen-directions, we sample at an angle between them. With the thin ellipsoid the sampled points are still close in ℓ_2 distance, implying that $\tilde{\mathbf{f}}_t$ will be large in both eigen-directions. However, the inverse Hessian will only annihilate one of these directions.

7 Application to the online shortest path problem

Because of its appealing structure, the online shortest path problem is one of the best studied problems in online optimization. Takimoto and Warmuth [20], and later Kalai and Vempala [11], gave efficient algorithms for the full information setting. Awerbuch and Kleinberg [2] were the first to give an efficient algorithm with $O(T^{2/3})$ regret in the partial information (bandit) setting. The recent work of Dani et al [6] implies a $O(m^{3/2}\sqrt{T})$-regret algorithm, where $m = |E|$ is the number of edges in the graph. However, it is not clear how to implement this algorithm efficiently. In this section we describe how the algorithm in the previous section implies an efficient $O(m^{3/2}\sqrt{T})$-regret algorithm.

Formally, the *bandit shortest path* problem is defined as the following repeated game:

Given a directed graph $G = (V, E)$ and a source-sink pair $s, t \in V$, at each time step $t = 1$ to T,

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

In order to model the problem as an bandit linear optimization problem, we recall the standard description of the set of all distributions over paths (flows) in graph as a convex set in $\mathbb{R}^m$, with

$O(m+|V|)$ constraints. Let σ be an arbitrary alignment of the edges, and define $\sigma_{e,v}$ to be one if the vertex v appears first in this alignment, and minus one otherwise. Then the convex set of all probability distributions can be described as a subset of the m-dimensional hypercube as follows:

$$\begin{aligned} x \in \mathbb{R}^m \ , \ \forall e \in E \ , \ 0 \leq x_e \leq 1 \\ \forall v \in V \setminus \{s,t\} \ , \ \sum_{v \in e} x_e \sigma_{e,v} = 0 \\ \sum_{s \in e} x_e \sigma_e = - \sum_{t \in e} x_e \sigma_e = 1 \end{aligned}$$

We now let the convex set given by the constraints above, denoted $\mathcal{K}$, be the underlying set for our bandit algorithm. That is, at each iteration the algorithm chooses a vector $\mathbf{y}_t \in \mathcal{K}$, the adversary chooses weights $\mathbf{f}_t \in \mathbb{R}^m$, and the loss suffered by the online player is $\mathbf{f}_t^\mathsf{T} \mathbf{y}_t$. With this model, our algorithm produces a flow, i.e. distribution over paths.

Theorem 3.1 implies (see argument for the simplex) that Algorithm 1 attains $O(m^{3/2}\sqrt{T})$ regret in this setting. However, strictly speaking this is not an online shortest path algorithm, as in each iteration the algorithm returns a flow (distribution over paths) rather a path.

However, it is easy to convert this flow algorithm into a randomized online shortest path algorithm: according to the standard flow decomposition theorem (see e.g. [18]), a given flow in the graph can be decomposed into a distribution over at most $O(m)$ paths in polynomial time. Hence, given a flow $\mathbf{y}_t \in \mathcal{K}$, one can obtain an unbiased estimator for $\mathbf{f}_t^\mathsf{T} \mathbf{y}_t$ by choosing a path according to the distribution of the decomposition, and estimating $\mathbf{f}_t^\mathsf{T} \mathbf{y}_t$ by the length of this path according to $\mathbf{f}_t$ (i.e. $\sum_{e \in p} \mathbf{f}_t(e)$). The variance of this unbiased estimator is $O(1)$ (assuming that the paths cost is bounded by one), and that the expected regret is unchanged.

In each iteration the algorithm requires to compute the Hessians of all $O(m+|V|)$ self-concordant barriers for the constraints, as well as the eigenvectors of the total Hessian and a flow decomposition computation. This results in somewhat slower algorithm (although certainly poly-time) than the previous approaches, in which the most expensive operation was a shortest path computation. We leave it as an open question to find a more efficient implementation.

8 Proof of the regret bound

8.1 Unbiasedness

First, we show that $\mathbb{E}\tilde{\mathbf{f}}_t = \mathbf{f}_t$. Condition on the choice i_t and average over the choice of ε_t:

$$\begin{aligned} \mathbb{E}_{\varepsilon_t} \tilde{\mathbf{f}}_t &= \frac{1}{2} n \left(\mathbf{f}_t \cdot (\mathbf{x}_t + \lambda_{i_t}^{-1/2} \mathbf{e}_{i_t}) \right) \lambda_{i_t}^{1/2} \cdot \mathbf{e}_{i_t} \\ &- \frac{1}{2} n \left(\mathbf{f}_t \cdot (\mathbf{x}_t - \lambda_{i_t}^{-1/2} \mathbf{e}_{i_t}) \right) \lambda_{i_t}^{1/2} \cdot \mathbf{e}_{i_t} \\ &= n(\mathbf{f}_t^\mathsf{T} \mathbf{e}_{i_t}) \mathbf{e}_{i_t}. \end{aligned}$$

Hence,

$$\mathbb{E}\tilde{\mathbf{f}}_t = n \left(\mathbb{E}_{i_t} \mathbf{e}_{i_t} \mathbf{e}_{i_t}^\mathsf{T} \right) \mathbf{f}_t = \mathbf{f}_t.$$

Furthermore, $\mathbb{E}\mathbf{y}_t = \mathbf{x}_t$.

8.2 Closeness of the next minima

We now use the properties of the Dikin ellipsoids mentioned in the previous section.

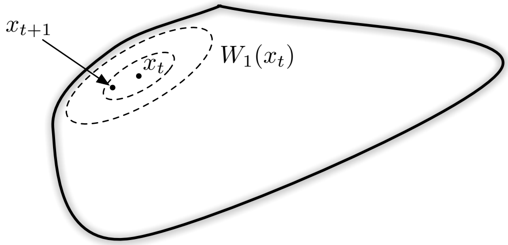

Figure 1: The Dikin ellipsoid $W_1(\mathbf{x}_t)$ at $\mathbf{x}_t$. The next minima is guaranteed to lie in its scaled version $W_{4n\eta}(\mathbf{x}_t)$.

Lemma 8.1. *The next minimizer* $\mathbf{x}_{t+1}$ *is "close" to* $\mathbf{x}_t$*:*

$$\mathbf{x}_{t+1} \in W_{4n\eta}(\mathbf{x}_t).$$

Proof. Recall that

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{K}} \Phi_t(\mathbf{x}) \quad \text{and} \quad \mathbf{x}_t = \arg\min_{\mathbf{x}\in\mathcal{K}} \Phi_{t-1}(\mathbf{x})$$

where $\Phi_t(\mathbf{x}) = \eta \sum_{s=1}^{t} \tilde{\mathbf{f}}_t^{\mathsf{T}}\mathbf{x} + \mathcal{R}(\mathbf{x})$. Since $\nabla\Phi_{t-1}(\mathbf{x}_t) = 0$, we conclude that $\nabla\Phi_t(\mathbf{x}_t) = \eta\tilde{\mathbf{f}}_t$.

Consider any point in $\mathbf{z} \in W_{\frac{1}{2}}(\mathbf{x}_t)$. It can be written as $\mathbf{z} = \mathbf{x}_t + \alpha\mathbf{u}$ for some vector $\mathbf{u}$ such that $\|\mathbf{u}\|_{\mathbf{x}_t} = 1$ and $\alpha \in (-\frac{1}{2}, \frac{1}{2})$. Expanding,

$$\begin{aligned}\Phi_t(\mathbf{z}) &= \Phi_t(\mathbf{x}_t + \alpha\mathbf{u}) \\ &= \Phi_t(\mathbf{x}_t) + \alpha\nabla\Phi_t(\mathbf{x}_t)^{\mathsf{T}}\mathbf{u} + \alpha^2\frac{1}{2}\mathbf{u}^{\mathsf{T}}\nabla^2\Phi_t(\xi)\mathbf{u} \\ &= \Phi_t(\mathbf{x}_t) + \alpha\eta\tilde{\mathbf{f}}_t^{\mathsf{T}}\mathbf{u} + \alpha^2\frac{1}{2}\mathbf{u}^{\mathsf{T}}\nabla^2\Phi_t(\xi)\mathbf{u}\end{aligned}$$

for some ξ on the path between $\mathbf{x}_t$ and $\mathbf{x}_t + \alpha\mathbf{u}$.

Let us check where the optimum of the RHS is obtained. Setting the derivative with respect to α to zero, we obtain

$$|\alpha^*| = \frac{\eta|\tilde{\mathbf{f}}_t^{\mathsf{T}}\mathbf{u}|}{\mathbf{u}^T\nabla^2\Phi_t(\xi)\mathbf{u}} = \frac{\eta|\tilde{\mathbf{f}}_t^{\mathsf{T}}\mathbf{u}|}{\mathbf{u}^T\nabla^2\mathcal{R}(\xi)\mathbf{u}}.$$

The fact that ξ is on the line $\mathbf{x}_t$ to $\mathbf{x}_t + \alpha\mathbf{u}$ implies that $\|\xi - \mathbf{x}_t\|_{\mathbf{x}_t} \leq \|\alpha\mathbf{u}\|_{\mathbf{x}_t} < \frac{1}{2}$. Hence, by Eq (5),

$$\nabla^2\mathcal{R}(\xi) \succeq (1 - \|\xi - \mathbf{x}_t\|_{\mathbf{x}_t})^2\nabla^2\mathcal{R}(\mathbf{x}_t) \succ \frac{1}{4}\nabla^2\mathcal{R}(\mathbf{x}_t).$$

Thus $\mathbf{u}^T\nabla^2\mathcal{R}(\xi)\mathbf{u} > \frac{1}{4}\|\mathbf{u}\|_{\mathbf{x}_t} = \frac{1}{4}$, and hence

$$\alpha^* < 4\eta|\tilde{\mathbf{f}}_t^{\mathsf{T}}\mathbf{u}|.$$

Recall that $\tilde{\mathbf{f}}_t = n\,(\mathbf{f}_t \cdot \mathbf{y}_t)\,\varepsilon_t \lambda_{i_t}^{1/2} \cdot \mathbf{e}_{i_t}$ and so $\tilde{\mathbf{f}}_t^\mathsf{T} \mathbf{u}$ is maximized/minimized when $\mathbf{u}$ is a unit (with respect to $\|\cdot\|_{\mathbf{x}_t}$) vector in the direction of $\mathbf{e}_{i_t}$, i.e. $\mathbf{u} = \pm \lambda_{i_t}^{-1/2} \mathbf{e}_{i_t}$. We conclude that

$$|\tilde{\mathbf{f}}_t^\mathsf{T} \mathbf{u}| \le n\,|\mathbf{f}_t \cdot \mathbf{y}_t| \le n$$

and

$$|\alpha^*| < 4n\eta < \frac{1}{2}$$

by our choice of η and T. We conclude that the local optimum $\arg\min_{\mathbf{z} \in W_{\frac{1}{2}}(\mathbf{x}_t)} \Phi_t(\mathbf{z})$ is strictly inside $W_{4n\eta}(\mathbf{x}_t)$, and since Φ_t is convex, the global optimum is

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{z} \in \mathcal{K}} \Phi_t(\mathbf{z}) \in W_{4n\eta}(\mathbf{x}_t).$$

□

8.3 Proof of Theorem 3.1

We are now ready to prove the regret bound for Algorithm 1. Since $\mathbf{x}_{t+1} \in W_{4n\eta}(\mathbf{x}_t)$, we invoke Eq (6) at $\mathbf{x} = \mathbf{x}_t$ and $\mathbf{z} = \mathbf{h} = \mathbf{x}_{t+1} - \mathbf{x}_t$:

$$|\mathbf{h}^\mathsf{T}(\nabla \mathcal{R}(\mathbf{x}_{t+1}) - \nabla \mathcal{R}(\mathbf{x}_t))| \le \frac{\|\mathbf{h}\|_{\mathbf{x}_t}^2}{1 - \|\mathbf{h}\|_{\mathbf{x}_t}}.$$

Observe that $\mathbf{x}_{t+1} \in W_{4n\eta}(\mathbf{x}_t)$ implies $\|\mathbf{h}\|_{\mathbf{x}_t} < 4n\eta$.

Proof of Lemma 4.1 (Equation (11) in the Appendix) reveals that

$$\nabla \mathcal{R}(\mathbf{x}_t) - \nabla \mathcal{R}(\mathbf{x}_{t+1}) = \eta \tilde{\mathbf{f}}_t.$$

We have

$$\begin{aligned} \tilde{\mathbf{f}}_t^\mathsf{T}(\mathbf{x}_t - \mathbf{x}_{t+1}) &= \eta^{-1} \mathbf{h}^\mathsf{T}(\nabla \mathcal{R}(\mathbf{x}_{t+1}) - \nabla \mathcal{R}(\mathbf{x}_t)) \\ &\le \eta^{-1} \frac{\|\mathbf{h}\|_{\mathbf{x}_t}^2}{1 - \|\mathbf{h}\|_{\mathbf{x}_t}} \\ &\le \frac{16 n^2 \eta}{1 - 4n\eta} \\ &\le 32 n^2 \eta. \end{aligned} \tag{9}$$

By Lemma 4.1, for any $\mathbf{u} \in \mathcal{K}_{1/\sqrt{T}}$

$$\begin{aligned} \sum_{t=1}^T \tilde{\mathbf{f}}_t^\mathsf{T}(\mathbf{x}_t - \mathbf{u}) &\le \eta^{-1} D_{\mathcal{R}}(\mathbf{u}, \mathbf{x}_1) + \sum_{t=1}^T \tilde{\mathbf{f}}_t^\mathsf{T}(\mathbf{x}_t - \mathbf{x}_{t+1}) \\ &\le \eta^{-1} D_{\mathcal{R}}(\mathbf{u}, \mathbf{x}_1) + 32 n^2 \eta T \\ &= \eta^{-1}(\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)) + 32 n^2 \eta T \\ &\le \frac{1}{\eta}(2\vartheta \log T) + 32 n^2 \eta T, \end{aligned}$$

where the first equality follows since $\nabla\mathcal{R}(\mathbf{x}_1) = 0$, by the choice of $\mathbf{x}_1$; the last inequality follows from Equation (7). Balancing with $\eta = \frac{\sqrt{\vartheta \log T}}{4n\sqrt{T}}$, we get

$$\sum_{t=1}^{T} \tilde{\mathbf{f}}_t^{\mathsf{T}}(\mathbf{x}_t - \mathbf{u}) \leq 16n\sqrt{\vartheta T \log T}.$$

for any $\mathbf{u}$ in the scaled set $\mathcal{K}'$. Using Lemma 4.2, which we prove below, we obtain the statement of Theorem 3.1.

8.4 Expected Regret

Note that it is not $\tilde{\mathbf{f}}_t^{\mathsf{T}}\mathbf{x}_t$ that the algorithm should be incurring, but rather $\mathbf{f}_t^{\mathsf{T}}\mathbf{y}_t$. However, it is easy to see that these are equal in expectation.

Lemma 4.2. Let $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot | i_1, \ldots, i_{t-1}, \varepsilon_1, \ldots, \varepsilon_{t-1}]$ denote the conditional expectation. Note that

$$\mathbb{E}_t \tilde{\mathbf{f}}_t^{\mathsf{T}}\mathbf{x}_t = \mathbf{f}_t^{\mathsf{T}}\mathbf{x}_t = \mathbb{E}_t \mathbf{f}_t^{\mathsf{T}}\mathbf{y}_t.$$

Taking expectations on both sides of the bound for $\tilde{\mathbf{f}}_t$'s,

$$\begin{aligned} \mathbb{E}\sum_{t=1}^{T} \tilde{\mathbf{f}}_t^{\mathsf{T}}\mathbf{x}_t &\leq \mathbb{E}\min_{\mathbf{u}\in\mathcal{K}'} \sum_{t=1}^{T} \tilde{\mathbf{f}}_t^{\mathsf{T}}\mathbf{u} + C_T \\ &\leq \min_{\mathbf{u}\in\mathcal{K}'} \mathbb{E}\left(\sum_{t=1}^{T} \tilde{\mathbf{f}}_t^{\mathsf{T}}\mathbf{u}\right) + C_T \\ &= \min_{\mathbf{u}\in\mathcal{K}'} \mathbb{E}\left(\sum_{t=1}^{T} \mathbf{f}_t^{\mathsf{T}}\mathbf{u}\right) + C_T. \end{aligned}$$

□

In the case of an oblivious adversary,

$$\min_{\mathbf{u}\in\mathcal{K}'} \mathbb{E}\left(\sum_{t=1}^{T} \mathbf{f}_t^{\mathsf{T}}\mathbf{u}\right) = \min_{\mathbf{u}\in\mathcal{K}'} \sum_{t=1}^{T} \mathbf{f}_t^{\mathsf{T}}\mathbf{u}.$$

However, if the adversary is not oblivious, $\mathbf{f}_t$ depends on the random choices at time steps $1, \ldots, t-1$. Of course, it is desirable to obtain a stronger bound on the regret

$$\mathbb{E}\left[\sum_{t=1}^{T} \mathbf{f}_t^{\mathsf{T}}\mathbf{y}_t - \min_{\mathbf{u}\in\mathcal{K}'} \sum_{t=1}^{T} \mathbf{f}_t^{\mathsf{T}}\mathbf{u}\right] = O(\sqrt{T}),$$

which allows the optimal $\mathbf{u}$ to depend on the randomness of the player[3]. Obtaining guarantees for adaptive adversaries is another dimension of the bandit optimization problem and is beyond the scope of the present paper.

[3] It is known that the optimal strategy for the adversary does not need any randomization beyond player's choices.

Auer et al [1] provides a clever modification of their EXP3 algorithm which leads to high-probability bounds on the regret, thus guaranteeing low regret against an adaptive adversary. The modification is based on the idea of adding confidence intervals to the losses. The same idea has been employed in the work of [3, 17] (note that [3] is submitted concurrently with this paper) for the bandit optimization over arbitrary convex sets. While the work of [3] does succeed in obtaining a high-probability bound, the algorithm is based on the inefficient method of Dani et al [6], which is a reduction to the algorithm of Auer et al.

9 Efficient Implementation

In this section we describe how to efficiently implement Algorithm 1. Recall that in each iteration our algorithm requires the eigen-decomposition of the Hessian in order to derive the unbiased estimator, which takes $O(n^3)$ time. This is coupled with a convex minimization problem in order to compute $\mathbf{x}_t$, which seems to be the most time consuming operation in the entire algorithm.

The message of this section is that the computation of $\mathbf{x}_t$ given the previous iterate $\mathbf{x}_{t-1}$ takes essentially only **one iteration of the Damped Newton method**. More precisely, instead of using $\mathbf{x}_t$ as defined in Algorithm 1, it suffices to maintain a sequence of points $\{\mathbf{z}_t\}$, such that $\mathbf{z}_t$ is obtained from $\mathbf{z}_{t-1}$ by only one iteration of the Damped Newton method. The sequence of points $\{\mathbf{z}_t\}$ are shown to be sufficiently close to $\{\hat{\mathbf{x}}_t\}$, which enjoy the same guarantee as the sequence of $\{\mathbf{x}_t\}$ defined by Algorithm 1.

A single iteration of the Damped Newton method requires matrix inversion. However, since we have the eigen-decomposition ready made, as it was required for the unbiased estimator, we can produce the inverse and the Newton direction in $O(n^2)$ time. Thus, the most time-consuming part of the algorithm is the eigen-decomposition of the Hessian, and the total running time is $O(n^3)$ per iteration.

Before we begin, we require a few more facts from the theory of interior point methods, taken from [15].

Let Ψ be a non-degenerate self-concordant barrier on domain $\mathcal{K}$, for any $\mathbf{x} \in \mathcal{K}$ define the Newton direction as

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

and let the Newton decrement be

$$\lambda(\Psi, x) = \sqrt{\nabla \Psi(x)^{\mathsf{T}} [\nabla^2 \Psi(x)]^{-1} \nabla \Psi(x)}.$$

The *Damped Newton iteration* for a given $\mathbf{x} \in \mathcal{K}$ is

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

The following facts can be found in [15]:

A: $DN(\Psi, \mathbf{x}) \in \mathcal{K}$. [4]

B: $\lambda(\Psi, DN(\Psi, \mathbf{x})) \leq 2\lambda(\Psi, \mathbf{x})^2$.

C: $\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{x}^*} \leq \frac{\lambda(\Psi,\mathbf{x})}{1-\lambda(\Psi,\mathbf{x})}$.

[4] This follows easily since the Newton increment is in the Dikin ellipsoid $\frac{1}{1+\lambda(\Psi,\mathbf{x})} e(\Psi, \mathbf{x}) \in W_1(\mathbf{x})$.

D: $\|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{x}} \leq \frac{\lambda(\Psi,\mathbf{x})}{1-2\lambda(\Psi,\mathbf{x})}$.

Here $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{K}} \Psi(\mathbf{x})$.

Algorithm 2 Efficient Implementation

```
Input: $\eta > 0$, $\vartheta$-self-concordant $\mathcal{R}$.
Let $\mathbf{z}_1 = \arg\min_{\mathbf{x}\in\mathcal{K}} \mathcal{R}(\mathbf{x})$.
for $t = 1$ to $T$ do
    Let $\{\mathbf{e}_1, \ldots, \mathbf{e}_n\}$ and $\{\lambda_1, \ldots, \lambda_n\}$ be the set of eigenvectors and eigenvalues of $\nabla^2\mathcal{R}(\mathbf{z}_t)$.
    Choose $i_t$ uniformly at random from $\{1, \ldots, n\}$ and $\varepsilon_t = \pm 1$ with probability 1/2.
    Predict $\mathbf{y}_t = \mathbf{z}_t + \varepsilon_t \lambda_{i_t}^{-1/2} \mathbf{e}_{i_t}$.
    Observe the gain $\mathbf{f}_t^\mathsf{T}\mathbf{y}_t \in \mathbb{R}$.
    Define $\hat{\mathbf{f}}_t := n\,(\mathbf{f}_t^\mathsf{T}\mathbf{y}_t)\,\varepsilon_t \lambda_{i_t}^{1/2} \cdot \mathbf{e}_{i_t}$.
    Update
        $$\mathbf{z}_{t+1} = \mathbf{z}_t - \frac{1}{1+\lambda(\Psi_t, \mathbf{z}_t)} e(\Psi_t, \mathbf{z}_t),$$
    where
        $$\Psi_t(\mathbf{z}) \equiv \eta \sum_{s=1}^{t} \hat{\mathbf{f}}_s^\mathsf{T}\mathbf{z} + \mathcal{R}(\mathbf{z}).$$
end for
```

The functions $\hat{\mathbf{f}}_t$ computed by the above algorithm are unbiased estimates of $\mathbf{f}_t$ constructed by sampling eigenvectors of $\nabla^2\mathcal{R}(\mathbf{z}_t)$. Define the Follow The Regularized Leader solutions

$$\hat{\mathbf{x}}_{t+1} \equiv \arg\min_{\mathbf{x}\in\mathcal{K}} \Psi_t(\mathbf{x}),$$

on the new functions $\hat{\mathbf{f}}_t$'s. The sequence $\{\hat{\mathbf{x}}_t, \hat{\mathbf{f}}_t\}$ is different from the sequence $\{\mathbf{x}_t, \tilde{\mathbf{f}}_t\}$ generated by Algoritm 1. However, the same regret bound can be proved for the new algorithm. The only difference from the proof for Algorithm 1 is in the fact that $\hat{\mathbf{f}}_t$'s are estimated using the Hessian at $\mathbf{z}_t$, not $\hat{\mathbf{x}}_t$. However, as we show next, $\mathbf{z}_t$ is very close to $\hat{\mathbf{x}}_t$, and therefore the Hessians are within a factor of 2 by Equation (5), leading to a slightly worse constant for the regret.

Lemma 9.1. *It holds that for all t,*

$$\lambda^2(\Psi_t, \mathbf{z}_t) \leq 4n^2\eta^2$$

Proof. The proof is by induction on t. For $t = 1$ the result is true because $\mathbf{x}_1$ is chosen to minimize $\mathcal{R}$. Suppose the statement holds for $t-1$. By definition,

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

Note that

$$\nabla\Psi_t(\mathbf{z}_t) = \nabla\Psi_{t-1}(\mathbf{z}_t) + \eta\hat{\mathbf{f}}_t^\mathsf{T}.$$

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

$$\begin{aligned}\frac{1}{2}\lambda^2(\Psi_t, \mathbf{z}_t) &\leq \nabla\Psi_{t-1}(\mathbf{z}_t)[\nabla^2\mathcal{R}(\mathbf{z}_t)]^{-1}\nabla\Psi_{t-1}(\mathbf{z}_t) \\ &\quad + \eta^2\hat{\mathbf{f}}_t^\mathsf{T}[\nabla^2\mathcal{R}(\mathbf{z}_t)]^{-1}\hat{\mathbf{f}}_t \\ &= \lambda^2(\Psi_{t-1}, \mathbf{z}_t) + \eta^2\hat{\mathbf{f}}_t^\mathsf{T}[\nabla^2\mathcal{R}(\mathbf{z}_t)]^{-1}\hat{\mathbf{f}}_t.\end{aligned}$$

The first term can be bounded by fact (B) and using the induction hypothesis,

$$\lambda^2(\Psi_{t-1}, \mathbf{z}_t) \leq 4\lambda^4(\Psi_{t-1}, \mathbf{z}_{t-1}) \leq 64n^4\eta^4. \tag{10}$$

As for the second term,

$$\hat{\mathbf{f}}_t[\nabla^2\mathcal{R}(\mathbf{z}_t)]^{-1}\hat{\mathbf{f}}_t \leq n^2$$

because of the way $\hat{\mathbf{f}}_t$ is defined and since $|\mathbf{f}_t^\mathsf{T}\mathbf{y}_t| \leq 1$ by assumption. Combining the results,

$$\lambda^2(\Psi_t, \mathbf{z}_t) \leq 128n^4\eta^4 + 2n^2\eta^2 \leq 4n^2\eta^2$$

using the definition of η of Theorem 3.1 and large enough T. This proves the induction step.

□

Note that Equation (10) with the choice of η and large enough T implies $\lambda^2(\Psi_{t-1}, \mathbf{z}_t) << \frac{1}{2}$. Using this together with the above Lemma and facts (B) and (C), we conclude that

$$\|\mathbf{z}_t - \hat{\mathbf{x}}_t\|_{\hat{\mathbf{x}}_t} \leq 2\lambda(\Psi_{t-1}, \mathbf{z}_t) \leq 4\lambda(\Psi_{t-1}, \mathbf{z}_{t-1})^2 \leq 16n^2\eta^2$$

We observe that $\hat{\mathbf{x}}_t$ and $\mathbf{z}_t$ are very close in the local distance. This implies closeness in L_2 distance as well. Indeed, square roots of inverse eigenvalues $\lambda_i^{-1/2}$, being the distances from $\hat{\mathbf{x}}_t$ to the corresponding radii of the Dikin ellipsoid, can be at most the D. Thus, $\nabla^2\mathcal{R} \geq D^2 I$ and thus $\|\mathbf{z}_t - \hat{\mathbf{x}}_t\|_2 \leq D^{-1}\|\mathbf{z}_t - \hat{\mathbf{x}}_t\|_{\hat{\mathbf{x}}_t} \leq 16D^{-1}n^2\eta^2$.

As we proved, it requires only one Damped Newton update to maintain the sequence $\mathbf{z}_t$, which are $O(1/T)$ close to $\hat{\mathbf{x}}_t$. Hence,

$$\sum_{t=1}^{T} |\mathbf{f}_t^\mathsf{T}(\mathbf{z}_t - \hat{\mathbf{x}}_t)| \leq \sum_{t=1}^{T} \|\mathbf{f}_t\|\|\mathbf{z}_t - \hat{\mathbf{x}}_t\| = O(1).$$

Therefore, for any $\mathbf{u} \in \mathcal{K}$

$$\begin{aligned}\mathbb{E}\sum_{t=1}^{T}\mathbf{f}_t^\mathsf{T}(\mathbf{y}_t - \mathbf{u}) &= \mathbb{E}\sum_{t=1}^{T}\hat{\mathbf{f}}_t^\mathsf{T}(\mathbf{z}_t - \mathbf{u}) \\ &= \mathbb{E}\sum_{t=1}^{T}\hat{\mathbf{f}}_t^\mathsf{T}(\hat{\mathbf{x}}_t - \mathbf{u}) + \mathbb{E}\sum_{t=1}^{T}\hat{\mathbf{f}}_t^\mathsf{T}(\mathbf{z}_t - \hat{\mathbf{x}}_t) \\ &= \mathbb{E}\sum_{t=1}^{T}\hat{\mathbf{f}}_t^\mathsf{T}(\hat{\mathbf{x}}_t - \mathbf{u}) + \mathbb{E}\sum_{t=1}^{T}\mathbf{f}_t^\mathsf{T}(\mathbf{z}_t - \hat{\mathbf{x}}_t) \\ &= \mathbb{E}\sum_{t=1}^{T}\hat{\mathbf{f}}_t^\mathsf{T}(\hat{\mathbf{x}}_t - \mathbf{u}) + O(1)\end{aligned}$$

A slight modification of the proofs of Section 8 leads to a $O(\sqrt{T})$ bound on the expected regret of the sequence $\{\hat{\mathbf{x}}_t\}$.

Acknowledgments.

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

References

[1] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2003.

[2] Baruch Awerbuch and Robert D. Kleinberg. Adaptive routing with end-to-end feedback: distributed learning and geometric approaches. In *STOC '04: Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 45–53, New York, NY, USA, 2004. ACM.

[3] P. Bartlett, V. Dani, T. Hayes, S. Kakade, A. Rakhlin, and A. Tewari. High-probability bounds for the regret of bandit online linear optimization, 2008. In submission to COLT 2008.

[4] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, volume 2 of *MPS/SIAM Series on Optimization*. SIAM, Philadelphia, 2001.

[5] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[6] Varsha Dani, Thomas Hayes, and Sham Kakade. The price of bandit information for online optimization. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2008.

[7] Varsha Dani and Thomas P. Hayes. Robbing the bandit: less regret in online geometric optimization against an adaptive adversary. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 937–943, New York, NY, USA, 2006. ACM.

[8] Meir Feder, Neri Merhav, and Michael Gutman. Correction to 'universal prediction of individual sequences' (jul 92 1258-1270). *IEEE Transactions on Information Theory*, 40(1):285, 1994.

[9] Abraham D. Flaxman, Adam Tauman Kalai, and H. Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 385–394, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.

[10] D. P. Helmbold, J. Kivinen, and M. K. Warmuth. Relative loss bounds for single neurons. *IEEE Transactions on Neural Networks*, 10(6):1291–1304, November 1999.

[11] Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[12] Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Inf. Comput.*, 132(1):1–63, 1997.

[13] Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.

[14] H. Brendan McMahan and Avrim Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *COLT*, pages 109–123, 2004.

[15] A.S. Nemirovskii. Interior point polynomial time methods in convex programming, 2004. Lecture Notes.

[16] Y. E. Nesterov and A. S. Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming.* SIAM, Philadelphia, 1994.

[17] Alexander Rakhlin, Ambuj Tewari, and Peter Bartlett. Closing the gap between bandit and full-information online optimization: High-probability regret bound. Technical Report UCB/EECS-2007-109, EECS Department, University of California, Berkeley, Aug 2007.

[18] Satish Rao. Lecure notes: Cs 270, graduate algorithms. 2006.

[19] Herbert Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527–535, 1952.

[20] Eiji Takimoto and Manfred K. Warmuth. Path kernels and multiplicative updates. *J. Mach. Learn. Res.*, 4:773–818, 2003.

[21] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pages 928–936, 2003.

A Proofs

Lemma 4.1. Since the argmin is in the set, $\nabla\Phi_{t-1}(\mathbf{x}_t) = 0$ and

$$D_{\Phi_{t-1}}(\mathbf{u}, \mathbf{x}_t) = \Phi_{t-1}(\mathbf{u}) - \Phi_{t-1}(\mathbf{x}_t).$$

Moreover,

$$\Phi_t(\mathbf{u}) = \Phi_{t-1}(\mathbf{u}) + \eta\tilde{\mathbf{f}}_t^{\mathsf{T}}\mathbf{u}.$$

Combining the above,

$$\eta\tilde{\mathbf{f}}_t^{\mathsf{T}}\mathbf{u} = D_{\Phi_t}(\mathbf{u}, \mathbf{x}_{t+1}) + \Phi_t(\mathbf{x}_{t+1}) - \Phi_{t-1}(\mathbf{u})$$

and

$$\eta\tilde{\mathbf{f}}_t^{\mathsf{T}}\mathbf{x}_t = D_{\Phi_t}(\mathbf{x}_t, \mathbf{x}_{t+1}) + \Phi_t(\mathbf{x}_{t+1}) - \Phi_{t-1}(\mathbf{x}_t).$$

Thus,

$$\eta\tilde{\mathbf{f}}_t^{\mathsf{T}}(\mathbf{x}_t - \mathbf{u}) = D_{\Phi_t}(\mathbf{x}_t, \mathbf{x}_{t+1}) + D_{\Phi_{t-1}}(\mathbf{u}, \mathbf{x}_t) - D_{\Phi_t}(\mathbf{u}, \mathbf{x}_{t+1}).$$

Summing over $t = 1 \dots T$,

$$\begin{aligned}
\eta \sum_{t=1}^{T} \tilde{\mathbf{f}}_t^{\mathsf{T}}(\mathbf{x}_t - \mathbf{u}) &= D_{\Phi_0}(\mathbf{u}, \mathbf{x}_1) - D_{\Phi_T}(\mathbf{u}, \mathbf{x}_{T+1}) \\
&\quad + \sum_{t=1}^{T} D_{\Phi_t}(\mathbf{x}_t, \mathbf{x}_{t+1}) \\
&\leq D_{\Phi_0}(\mathbf{u}, \mathbf{x}_1) + \sum_{t=1}^{T} D_{\Phi_t}(\mathbf{x}_t, \mathbf{x}_{t+1})
\end{aligned}$$

By definition, $\mathbf{x}_t$ satisfies $\sum_{s=1}^{t-1} \tilde{\mathbf{f}}_s + \nabla\mathcal{R}(\mathbf{x}_t) = 0$ and $\mathbf{x}_{t+1}$ satisfies $\sum_{s=1}^{t} \tilde{\mathbf{f}}_s + \nabla\mathcal{R}(\mathbf{x}_{t+1}) = 0$. Subtracting,

$$\nabla\mathcal{R}(\mathbf{x}_t) - \nabla\mathcal{R}(\mathbf{x}_{t+1}) = \eta\tilde{\mathbf{f}}_t. \tag{11}$$

Now we realize that

$$\begin{aligned}
D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) &\leq D_{\mathcal{R}}(\mathbf{x}_t, \mathbf{x}_{t+1}) + D_{\mathcal{R}}(\mathbf{x}_{t+1}, \mathbf{x}_t) \\
&= -\nabla\mathcal{R}(\mathbf{x}_{t+1})(\mathbf{x}_t - \mathbf{x}_{t+1}) \\
&\quad - \nabla\mathcal{R}(\mathbf{x}_t)(\mathbf{x}_{t+1} - \mathbf{x}_t) \\
&= \eta\tilde{\mathbf{f}}_t^{\mathsf{T}}(\mathbf{x}_t - \mathbf{x}_{t+1}).
\end{aligned}$$

□

Lemma A.1. *For any point $\mathbf{x} \in \mathcal{K}$, it holds that*

$$\min_{\mathbf{y} \in \mathcal{K}_\delta} \|\mathbf{x} - \mathbf{y}\| \leq \delta.$$

Proof. Consider the point on the segment $[\mathbf{x}_1, \mathbf{x}]$ which intersects the boundary of $\mathcal{K}_\delta$, denote it $\mathbf{z}$. By definition, we have

$$\frac{\|\mathbf{z} - \mathbf{x}_1\|}{\|\mathbf{x} - \mathbf{x}_1\|} = \frac{1}{1+\delta}.$$

As $\mathbf{x}, \mathbf{x}_1, \mathbf{z}_t$ are on the same line

$$\|\mathbf{z} - \mathbf{x}\| = \|\mathbf{x} - \mathbf{x}_1\| - \|\mathbf{z} - \mathbf{x}_1\| = \|\mathbf{x} - \mathbf{x}_1\| \cdot (1 - \frac{1}{1+\delta}) \leq \delta,$$

where the last inequality is by our assumption that the diameter of $\mathcal{K}$ is bounded by one. The lemma follows. □

B Ideas for High Probability Bounds

We now briefly sketch a possible direction for obtaining high-probability bounds. The general idea employed by [1, 3] is to add confidence intervals to the estimates. We observe that for our estimates the variance is

$$\begin{aligned}
\mathrm{var}\left(\sum_{t=1}^{T}(\tilde{\mathbf{f}}_t - \mathbf{f}_t)^{\mathsf{T}}(\mathbf{x}_t - \mathbf{u})\right) &\le \sum_{t=1}^{T}(\mathbf{x}_t - \mathbf{u})^{\mathsf{T}}\left(\mathbb{E}\tilde{\mathbf{f}}_t\tilde{\mathbf{f}}_t^{\mathsf{T}}\right)(\mathbf{x}_t - \mathbf{u}) \\
&\le \sum_{t=1}^{T}(\mathbf{x}_t - \mathbf{u})^{\mathsf{T}}\nabla^2\mathcal{R}(\mathbf{x}_t)(\mathbf{x}_t - \mathbf{u}) \\
&= \sum_{t=1}^{T}\|\mathbf{x}_t - \mathbf{u}\|_{\mathbf{x}_t}^2
\end{aligned}$$

for any $\mathbf{u} \in \mathcal{K}$. This matches our intuition: the uncertainty about the estimate $\tilde{\mathbf{f}}_t$ is increasing as $\mathbf{u}$ moves away from $\mathbf{x}_t$. Moreover, it is increasing faster in the directions in which $\tilde{\mathbf{f}}_t$ is large. However, subtracting this variance from the linear functions leads to concave functions and the methods of this paper are no longer valid. This issue did not arise in [1, 3] because geometry of the set is not crucial for exponential updates.

A possible approach to deal with concavity is to construct functions $\tilde{\mathbf{g}}_t$ such that $\|\mathbf{x}_t - \mathbf{u}\|_{\mathbf{x}_t}^2 \le \tilde{\mathbf{g}}_t^{\mathsf{T}}\mathbf{u}$ for any $\mathbf{u} \in \mathcal{K}$ and at each step find the minimum

$$\mathbf{x}_{t+1} = \arg\min_{\mathbf{x}\in\mathcal{K}} \eta^{-1}\sum_{s=1}^{t}(\tilde{\mathbf{f}}_t - T^{-1/2}\tilde{\mathbf{g}}_t)^{\mathsf{T}}\mathbf{x} + \mathcal{R}(\mathbf{x}).$$

This idea seems natural, as we would have

$$\begin{aligned}
\sum_{t=1}^{T}\tilde{\mathbf{f}}_t^{\mathsf{T}}(\mathbf{x}_t - \mathbf{u}) + 2\sqrt{\sum_{t=1}^{T}\|\mathbf{x}_t - \mathbf{u}\|_{\mathbf{x}_t}^2} &\le \sum_{t=1}^{T}\tilde{\mathbf{f}}_t^{\mathsf{T}}(\mathbf{x}_t - \mathbf{u}) + \sum_{t=1}^{T}\frac{\|\mathbf{x}_t - \mathbf{u}\|_{\mathbf{x}_t}^2}{\sqrt{T}} + \sqrt{T} \\
&\le \sum_{t=1}^{T}\tilde{\mathbf{f}}_t^{\mathsf{T}}(\mathbf{x}_t - \mathbf{u}) + \sum_{t=1}^{T}\frac{\tilde{\mathbf{g}}_t^{\mathsf{T}}\mathbf{u}}{\sqrt{T}} + \sqrt{T} \\
&\le \sum_{t=1}^{T}(\tilde{\mathbf{f}}_t - T^{-1/2}\tilde{\mathbf{g}}_t)^{\mathsf{T}}(\mathbf{x}_t - \mathbf{u}) + \frac{1}{\sqrt{T}}\sum_{t=1}^{T}\tilde{\mathbf{g}}_t^{\mathsf{T}}\mathbf{x}_t + \sqrt{T}
\end{aligned}$$

If the regret on the modified functions $\tilde{\mathbf{f}}_t - T^{-1/2}\tilde{\mathbf{g}}_t$ can be shown to be $O(\sqrt{T})$ and $\tilde{\mathbf{g}}_t$ can always be constructed in a way that keeps $\tilde{\mathbf{g}}_t^{\mathsf{T}}\mathbf{x}_t$ is small, the above bound would imply a high-probability guarantee of $O(\sqrt{T})$. While such an approach avoids the issue of concavity, constructing these linear upper bounds over general sets in a principled manner seems difficult.

We further conjecture that the Hessian of $\mathcal{R}$ needs to scale as inverse distance d^{-1} to the boundary, not inverse square distance to the boundary, in order to obtain high-probability guarantees. Such a regularizer would have the form $d\log d$ instead of $-\log d$.