

# Better Algorithms for Benign Bandits

Elad Hazan  
IBM Almaden

650 Harry Rd, San Jose, CA 95120  
ehazan@cs.princeton.edu

Satyen Kale

Microsoft Research

One Microsoft Way, Redmond, WA 98052  
satyen.kale@microsoft.com

## Abstract

The online multi-armed bandit problem and its generalizations are repeated decision making problems, where the goal is to select one of several possible decisions in every round, and incur a cost associated with the decision, in such a way that the total cost incurred over all iterations is close to the cost of the best fixed decision in hindsight. The difference in these costs is known as the *regret* of the algorithm. The term *bandit* refers to the setting where one only obtains the cost of the decision used in a given iteration and no other information.

Perhaps the most general form of this problem is the non-stochastic bandit linear optimization problem, where the set of decisions is a convex set in some Euclidean space, and the cost functions are linear. Only recently an efficient algorithm attaining  $\tilde{O}(\sqrt{T})$  regret was discovered in this setting.

In this paper we propose a new algorithm for the bandit linear optimization problem which obtains a regret bound of  $\tilde{O}(\sqrt{Q})$ , where  $Q$  is the total variation in the cost functions. This regret bound, previously conjectured to hold in the full information case, shows that it is possible to incur much less regret in a slowly changing environment even in the bandit setting. Our algorithm is efficient and applies several new ideas to bandit optimization such as reservoir sampling.

## 1 Introduction

Consider a person who commutes to work every day. Each morning, she has a choice of routes to her office. She chooses one route every day based on her past experience. When she reaches her office, she records the time it took her on that route that day, and uses this information to choose routes in the future. She doesn't obtain any information on the other routes she could have chosen to work. She would like to minimize her total time spent commuting in the long run; however, knowing nothing of how traffic patterns might change, she opts for the more pragmatic goal of trying to minimize the total time spent commuting in comparison with the time she would have spent had she

full knowledge of the future traffic patterns but had to choose the same fixed route every day. This difference in cost (using time as a metric of cost) measures how much she regrets not knowing traffic patterns and avoiding the hassle of choosing a new path every day.

This scenario, and many more like it, are modeled by the multi-armed bandit problem and its generalizations. It can be succinctly described as follows: iteratively an online learner has to choose an action from a set of  $n$  available actions. She then suffers a cost (or receives a reward) corresponding to the action she took and no other information as to the merit of other available actions. Her goal is to minimize her *regret*, which is defined the difference between her total cost and the total cost of the best single action knowing the costs of all actions in advance.

Various models of the “unknown” cost functions have been considered in the last half-a-century. Robbins [17] pioneered the study of various stochastic cost functions, followed by Hannan [10], Lai and Robbins [13] and others. It is hard to do justice to the numerous contributions and studies and we refer the reader to the book of [6] for references. Auer *et al* [2] considered an adversarial non-stochastic model of costs. In their influential paper, [2] gave an efficient algorithm that attains the optimal regret in terms of the number of iterations,  $T$ , a bound of  $O(\sqrt{nT})$ .<sup>1</sup> The sublinear (in  $T$ ) regret bound implies that on average, the algorithm's cost converges to that of the best fixed action in hindsight.

This paper was followed by a long line of work [3, 14, 9, 7] which considered the more general case of bandit online linear optimization over a convex domain. In this problem, the learner has to choose a sequence of points from the convex domain and obtains their cost from an unknown linear cost function. The objective, again, is to minimize the regret, i.e. the difference between the total cost of the algorithm and that of

<sup>1</sup>Strictly speaking, here and henceforth we talk about expected regret, as all algorithms that attain non-trivial guarantees must use randomization.

the best fixed point in hindsight. This generality is crucial to allow for *efficient* algorithms for problems with a large decision space, such as online shortest path problem considered at the beginning. This line of work finally culminated in the work of Abernethy, Hazan and Rakhlin [1], who obtained the first algorithm to give  $\tilde{O}(\text{poly}(n)\sqrt{T})$  regret with polynomial running time.

Even though the  $\tilde{O}(\sqrt{T})$  dependence on  $T$  was a great achievement, this regret bound is weak from the point of view of real-world bandit scenarios. Rarely would we encounter a case where the cost functions are truly adversarial. Indeed, the first work on this problem assumed a stochastic model of cost functions, which is a very restrictive assumption in many cases. One reasonable way to retain the appeal of worst-case bounds while approximating the steady-state nature of the stochastic setting is to consider the *variation* in the cost vectors.

For example, our office commuter doesn't expect the traffic gods to conspire against her every day. She might expect a certain predictability in traffic patterns. Most days the traffic pattern is about the same, except for some fluctuations depending on the day of the week, time of the day, etc. Coming up with a stochastic model for traffic patterns would be simply too onerous. An algorithm that quickly learns the dominant pattern of the traffic, and achieves regret bounded by the (typically small) variability in day-to-day traffic, would be much more desirable. Such regret bounds naturally interpolate between the stochastic models of Robbins and the worst case models of Auer *et al.*

In this paper we present the first such bandit optimization algorithm in the worst-case adversarial setting, with regret bounded by  $\tilde{O}(\sqrt{Q})$ <sup>2</sup>, where  $Q$  is the total observed variation in observed costs, defined as the sum of squared deviations of the cost vectors from their mean. This regret degrades gracefully with increasing  $Q$ , and in the worst case, we recover the regret bound  $\tilde{O}(\sqrt{T})$  of [1]. Our algorithm is efficient, running in polynomial time per iteration.

The conjecture that the regret of online learning algorithms should be bounded in terms of the total variation was put forth by Cesa-Bianchi, Mansour and Stoltz [5] in the full information model (where the online player is allowed to observe the costs of actions she did not choose). This conjecture was recently resolved on the affirmative in [11], in two important online learning scenarios, viz. online linear optimization and expert prediction. In addition, in [12], we give algorithms with

regret bounds of  $O(\log(Q))$  for the Universal Portfolio Selection problem and its generalizations. In this paper, we prove the surprising fact that such a regret bound of  $\tilde{O}(\sqrt{Q})$  is possible to obtain even when the only information available to the player is the cost she incurred (in particular, we may not even be able to estimate  $Q$  accurately in this model).

To prove our result we need to overcome the following difficulty: all previous approaches for the non-stochastic multi-armed bandit problem relied on the main tool of “unbiased gradient estimator”, i.e. the use of randomization to extrapolate the missing information (cost function). The variation in these unbiased estimators is unacceptably large even when the underlying cost function sequence has little or no variation.

To overcome this problem we introduce two new tools: first, we use historical costs to construct our gradient estimators. Next, in order to construct these estimators, we need an accurate method of accumulating historical data. For this we employ a method widely used in the data streaming community known as “reservoir sampling”. This method allows us to maintain an accurate “sketch” of history with very little overhead.

An additional difficulty which arises in the designing the algorithm is the fact that a learning rate parameter  $\eta$  needs to be set based on the total variation  $Q$  to obtain the  $\tilde{O}(\sqrt{Q})$  regret bound. Typically, in other scenarios where square root regret bound in some parameter is desired, a simple  $\eta$ -halving trick works, but requires the algorithm to be able to compute the relevant parameter after every iteration. However, as remarked earlier, even estimating  $Q$  is non-trivial problem. We do manage to bypass this problem by using a novel approach that implicitly mimics the  $\eta$ -halving procedure.

## 2 Preliminaries

We consider the online linear optimization model in which iteratively the online player chooses a point  $\mathbf{x}_t \in \mathcal{K}$ , where  $\mathcal{K}$  is a convex compact set called the *decision set*. After her choice, an adversary supplies a linear cost function  $\mathbf{f}_t$ , and the player incurs a cost of  $\mathbf{f}_t(\mathbf{x}_t)$ . The adversary does not have access to the random bits used by the player, and in this extended abstract we assume an *oblivious* adversary, see [8] for more details on various models of adversaries.

With some abuse of notation, we use  $\mathbf{f}_t$  to also denote the cost vector such that  $\mathbf{f}_t(\mathbf{x}) = \mathbf{f}_t^\top \mathbf{x}$ . The *only* information available to the player is the cost incurred, i.e. the scalar  $\mathbf{f}_t(\mathbf{x}_t)$ . Denote the total number of game iterations by  $T$ . The standard game-theoretic measure

<sup>2</sup>Here and henceforth we use the standard  $\tilde{O}$  notation to hide both constant and poly-logarithmic terms. Our precise bound for the regret will include a  $\log(T)$  term also.

of performance is regret, defined as

$$\text{Regret}_T = \sum_{t=1}^T \mathbf{f}_t(\mathbf{x}_t) - \min_{\mathbf{x} \in \mathcal{K}} \sum_{t=1}^T \mathbf{f}_t(\mathbf{x})$$

We assume that the cost functions are bounded, i.e.  $\|\mathbf{f}_t\| \leq 1$  and that for all  $\mathbf{x} \in \mathcal{K}$ , we have  $\|\mathbf{x}\| \leq 1$ . We also assume that all standard basis vectors  $\{\mathbf{e}_1, \dots, \mathbf{e}_n\} \in \mathcal{K}$  are in the decision set. This assumption is made only for the sake of simplicity of exposition, in general, all we need is a set of  $n$  linearly independent points in  $\mathcal{K}$  that are “well-conditioned”. We defer the details to the full version of the paper.

We denote by  $Q_T$  the total quadratic variation in cost functions, i.e.

$$Q_T := \sum_{t=1}^T \|\mathbf{f}_t - \mu\|^2,$$

where  $\mu = \frac{1}{T} \sum_{t=1}^T \mathbf{f}_t$  is the mean of all cost functions.

For a positive definite matrix  $A$  we denote its induced norm by  $\|\mathbf{x}\|_A = \sqrt{\mathbf{x}^\top A \mathbf{x}}$ . We make use of the following simple generalization of the Cauchy-Schwarz inequality:

$$(2.1) \quad \mathbf{x}^\top \mathbf{y} \leq \|\mathbf{x}\|_A \cdot \|\mathbf{y}\|_{A^{-1}}.$$

For symmetric matrices we denote by  $A \preceq B$  the fact that the matrix  $B - A \succeq 0$  is positive semidefinite, i.e. all its eigenvalues are non-negative.

**2.1 Reservoir Sampling.** A crucial ingredient in our algorithm is a sampling procedure ubiquitously used in streaming algorithms known as “reservoir sampling” [18]. In a streaming problem the algorithm gets to see a stream of data in one pass, and not allowed to revisit previous data. Suppose the stream consists of real numbers  $\mathbf{f}_1, \mathbf{f}_2, \dots$  and our goal is to maintain random samples from the data seen so far so that any time  $t$ , we can compute an estimate of the mean,  $\mu_t := \frac{1}{t} \sum_{\tau=1}^t \mathbf{f}_\tau$ . The simple solution is to maintain a uniform sample  $S$  from the stream. This is implemented by starting with  $S = \mathbf{f}_1$  and for every  $t$ , replacing  $S$  by  $\mathbf{f}_t$  with probability  $1/t$ . At time  $t$ , we set our estimator for the mean  $\mu_t$  to be  $\tilde{\mu}_t = S$ . The following fact is standard (see [18]) and easily proven inductively:

LEMMA 1. *For every  $t$ , the random variable  $S$  (and therefore, the estimator  $\tilde{\mu}_t$ ) is uniformly distributed on the values  $f_\tau(i)$  for  $1 \leq \tau \leq t$ .*

The following lemma follows immediately:

LEMMA 2.  $\mathbf{E}[\tilde{\mu}_t] = \mu_t$  and  $\text{VAR}[\tilde{\mu}_t] = \frac{1}{t} \sum_{\tau=1}^t (\mathbf{f}_\tau - \mu_t)^2 = \frac{1}{t} Q_t$ .

A simple extension of the above estimator to also obtain low variance is to keep independent  $k$  such estimators  $S_1, S_2, \dots, S_k$  and let  $\tilde{\mu}_t = \frac{1}{k} \sum_{i=1}^k S_i$  be their average. These  $k$  estimators might have repetitions. In order to reduce the variance further, it is better to keep a random subset of size  $k$  without repetitions.

The standard reservoir sampling idea to accomplish this is to start by selecting the first  $k$  data points. For any iteration  $t > k$ , we sample the new data point with probability  $\frac{k}{t}$  and then uniformly at random choosing one of the previously chosen samples  $S_1, S_2, \dots, S_k$  to replace with the new data point. Since we have  $k$  samples, the variance of the average  $\tilde{\mu}_t = \frac{1}{k} \sum_{i=1}^k S_i$  drops by a factor of at least  $k$ . The following lemma is straightforward:

LEMMA 3.  $\mathbf{E}[\tilde{\mu}_t] = \mu_t$  and  $\text{VAR}[\tilde{\mu}_t] \leq \frac{1}{tk} Q_t$ .

**2.2 Self-concordant Functions and the Dikin ellipsoid.** In this section we give a few definition and properties of self-concordant barriers that we will crucially need in the analysis. Our treatment of this subject directly follows [1], and is less detailed. Self-concordance in convex optimization is a beautiful and deep topic, and we refer the reader to [16, 4] for a thorough treatment on the subject.

We skip the definition of a self-concordant barrier and function, see references above. For our purposes it suffices to say that any  $n$ -dimensional closed convex set admits an  $\vartheta$ -self-concordant barrier (which may not necessarily be efficiently computable), where the self-concordance parameter is  $\vartheta = O(n)$ .

More concretely, the standard logarithmic barrier for a half-space  $\mathbf{u}^\top \mathbf{x} \leq b$  is given by  $\mathcal{R}(\mathbf{x}) = -\log(b - \mathbf{u}^\top \mathbf{x})$ , and is 1-self-concordant. For polytopes defined by  $m$  halfspaces, the standard logarithmic barrier (which is just the sum of all barriers for the defining half-spaces) has the self-concordance parameter as  $\vartheta = m$ . It suffices for the unfamiliar reader to think only of these examples.

For a given  $\mathbf{x} \in \mathcal{K}$ , define

$$\begin{aligned} \|\mathbf{h}\|_{\mathbf{x}} &:= \sqrt{\mathbf{h}^\top [\nabla^2 \mathcal{R}(\mathbf{x})] \mathbf{h}}, \quad \text{and} \\ \|\mathbf{h}\|_{\mathbf{x}}^* &:= \sqrt{\mathbf{h}^\top [\nabla^2 \mathcal{R}(\mathbf{x})]^{-1} \mathbf{h}}. \end{aligned}$$

Here,  $\nabla^2 \mathcal{R}(\mathbf{x})$  denotes the Hessian of  $\mathcal{R}$  at point  $\mathbf{x}$ .

Define the *Dikin ellipsoid* of radius  $r$  centered at  $\mathbf{x}$  as the set

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

The following facts about the Dikin ellipsoid and self concordant functions will be used in the sequel (we refer to [15] for proofs):

1.  $W_1(\mathbf{x}) \subseteq \mathcal{K}$  for any  $\mathbf{x} \in \mathcal{K}$ . This is crucial for most of our sampling steps (the “ellipsoidal sampling”), we sample 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.
2. Within the Dikin ellipsoid, that is for  $\|\mathbf{h}\|_{\mathbf{x}} < 1$ , the Hessians of  $\mathcal{R}$  are “almost constant”:

$$(2.2) \quad \begin{aligned} (1 - \|\mathbf{h}\|_{\mathbf{x}})^2 \nabla^2 \mathcal{R}(\mathbf{x}) &\preceq \nabla^2 \mathcal{R}(\mathbf{x} + \mathbf{h}) \\ &\preceq (1 - \|\mathbf{h}\|_{\mathbf{x}})^{-2} \nabla^2 \mathcal{R}(\mathbf{x}). \end{aligned}$$

This property, together with the fact that the convex set  $\mathcal{K}$  is contained in the unit ball, implies the following property relating the  $\|\cdot\|_{\mathbf{x}}^*$  norm to the standard  $\ell_2$  norm  $\|\cdot\|$ , for any vector  $\mathbf{h}$ ,

$$(2.3) \quad \|\mathbf{h}\|_{\mathbf{x}}^* \leq \|\mathbf{h}\|.$$

3. For any  $\delta > 0$ , we can define using the Minkowski function (see [15]) a convex body  $\mathcal{K}_\delta \subseteq \mathcal{K}$  with the following properties. First, for any  $\mathbf{x} \in \mathcal{K}$ , there exists a  $\mathbf{u} \in \mathcal{K}_\delta$  such that  $\|\mathbf{x} - \mathbf{u}\| \leq \delta$ . In addition, assuming that  $\mathcal{R}$  is a  $\vartheta$ -self-concordant barrier, we have for all  $\mathbf{x} \in \mathcal{K}$ ,  $\mathbf{u} \in \mathcal{K}_\delta$

$$\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}) \leq \vartheta \ln \frac{1 + \delta}{\delta}.$$

Of particular interest will be the parameter  $\delta = \frac{1}{T}$ , for which if  $\mathbf{u} \in \mathcal{K}_{1/T}$  then

$$(2.4) \quad \mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_t) \leq \vartheta \log(T + 1) \leq 2\vartheta \log(T).$$

### 3 The main theorem and algorithm

**Main result.** Before describing the algorithm, let us state the main result of this paper formally.

**THEOREM 4.** *There exists an efficient algorithm for the online linear optimization problem (Algorithm 1 below coupled with the halving procedure of Appendix A), whose running time is  $O(n^3)$  per iteration, and whose expected regret is bounded as follows. Let  $Q_T$  be the total variation of a cost function sequence in an online linear optimization instance over a convex set  $\mathcal{K}$  which admits an efficiently computable  $\vartheta$ -self-concordant barrier. Then*

$$\mathbf{E}[\text{Regret}_T] = O\left(n\sqrt{\vartheta Q_T \log(T)} + \sqrt{\vartheta} n^{1.5} \log^2(T)\right).$$

This theorem can be used with the well known logarithmic barrier to derive regret bounds for the online-shortest-paths problem and other linearly constrained

problems, and of course applicable much more generally.

**The multi-armed bandit problem.** A case of particular interest, which has been studied most extensively, is the “basic” multi-armed bandit (MAB) problem where in each iteration the learner pulls the arm of one out of  $n$  slot machines and obtains an associated reward, assumed to be in the range  $[0, 1]$ . The learner’s objective is to minimize his regret, i.e. the difference between his cumulative reward and that of the best fixed slot machine in hindsight.

This is a special case of the more general problem considered earlier and corresponds to taking the convex set  $\mathcal{K}$  to be the  $n$ -dimensional simplex of probability distributions over the arms. Since the  $n$ -dimensional simplex admits a simple  $n$ -self-concordant barrier, an immediate corollary of our main theorem is:

**COROLLARY 5.** *There exists an efficient algorithm for the multi-armed-bandit problem whose expected regret is bounded by*

$$\mathbf{E}[\text{Regret}_T] = O\left(n^2 \sqrt{Q \log(T)} + n^{2.5} \log^2(T)\right).$$

The additional factor of  $\sqrt{n}$  factor is because our results assume that  $\|\mathbf{f}_t\| \leq 1$ , and so we need to scale the costs down by  $\sqrt{n}$  to apply our bounds. In comparison, the best previously known bounds for this problem is  $O(\sqrt{nT \log(nT)})$  in [2]. Even though our bound is worse in the dependence on  $n$ , the dependence on the parameter which grows, viz.  $T$ , is much better.

It seems that even stronger guarantees should be attainable, i.e. instead of basing the regret bound on the variation in all arms, only the variation with respect to best arm may suffice. This issue is analogous to the one in [11], where tighter bounds are derived for the problem of prediction from expert advice, compared to the bounds for general online linear optimization. We intend to explore this issue in the full version of this paper.

**Overview of the algorithm.** The underlying scheme of our algorithm follows the recent approach of [1], who use the Follow-The-Regularized-Leader (FTRL) methodology with self-concordant barrier functions as a regularization (see also exposition in [11]). At the top level, at every iteration this algorithm simply chooses the point that would have minimized the total cost so far, plus an additional regularization cost function  $\mathcal{R}(\mathbf{x})$ , i.e. we predict with the point

$$\mathbf{x}_t = \arg \min_{\mathcal{K}} \left[ \eta \sum_{\tau=1}^{t-1} \tilde{\mathbf{f}}_\tau^T(\mathbf{x}) + \mathcal{R}(\mathbf{x}) \right],$$

where  $\eta$  is a learning rate parameter.

Here,  $\tilde{\mathbf{f}}_t$  is an estimator for the vector  $\mathbf{f}_t$ , which is carefully constructed to have low variance. In the full-information setting, when we can simply set  $\tilde{\mathbf{f}}_t = \mathbf{f}_t$ , such an algorithm can be shown to achieve low regret (see exposition in [1] and references therein). In the bandit setting, a variety of “one-point-gradient-estimators” are used [9, 1] which produce an unbiased estimator  $\tilde{\mathbf{f}}_t$  of  $\mathbf{f}_t$  by evaluating  $\mathbf{f}_t$  at just one point.

In order to exploit the variation in the loss function, we modify the unbiased estimators of previous approaches by incorporating our experience with previous loss functions as a “prior belief” on the upcoming loss function. Essentially, we produce an unbiased estimator of the *difference between the past and the current loss function*.

This brings the issue of the past loss functions, which are unfortunately also unknown. However, since we had many opportunities to learn about the past and it is an aggregate of many functions, our knowledge about the past cumulative loss function is much better than the knowledge of any one loss function in particular. We denote by  $\tilde{\mu}_t$  our estimator of  $\frac{1}{t} \sum_{\tau=1}^t \mathbf{f}_\tau$ . The straightforward way of maintaining this estimator would be to average all previous estimators  $\tilde{\mathbf{f}}_\tau$ . However, this estimator is far from being sufficiently accurate for our purposes.

Instead, we use the reservoir sampling idea of Section 2.1 to construct this  $\tilde{\mu}_t$ . For each coordinate  $i \in [n]$ , we maintain a reservoir of size  $k$ ,  $S_{i,1}, S_{i,2}, \dots, S_{i,k}$ . The estimator for  $\mu_t(i)$  is then  $\tilde{\mu}_t(i) = \frac{1}{k} \sum_{j=1}^k S_{i,j}$ . Our current approach is to use separate exploration steps in order to construct  $\tilde{\mu}_t$ . While it is conceivable that there are more efficient methods of integrating exploration and exploitation, as done by the algorithm in the other iterations, reservoir sampling turns out to be extremely efficient and incur only a logarithmic penalty in regret.

The general scheme is given in Algorithm 1. It is composed of exploration steps, called `SIMPLEXSAMPLE` steps, and exploration-exploitation steps, called `ELLIPSOIDSAMPLE` steps. Note that we use the notation  $\mathbf{y}_t$  for the actual point in  $\mathcal{K}$  chosen by the algorithm in either of these steps. In the `ELLIPSOIDSAMPLE` step,  $\mathbf{y}_t$  is chosen from a distribution with mean  $\mathbf{x}_t$ .

It remains to precisely state the `SIMPLEXSAMPLE` and `ELLIPSOIDSAMPLE` procedures. The `SIMPLEXSAMPLE` procedure is the simpler of the two. It essentially performs reservoir sampling on all the coordinates with a reservoir of size  $k$ . All the initial samples in the reservoir are initialized to 0. This is equivalent to assuming that we have  $nk$  fictitious initial time periods indexed by  $t = -(nk - 1), -(nk - 2), \dots, 0$  where the cost functions are  $\mathbf{f}_t = 0$ . This can be assumed without loss of generality since it doesn’t affect the regret bound at all.

---

**Algorithm 1** Bandit Online Linear Optimization

---

- 1: Input:  $\eta > 0$ ,  $\vartheta$ -self-concordant  $\mathcal{R}$ , reservoir size parameter  $k$
  - 2: Initialization: for all  $i \in [n], j \in [k]$ , set  $S_{i,j} = 0$ . Set  $\mathbf{x}_1 = \arg \min_{\mathbf{x} \in \mathcal{K}} [\mathcal{R}(\mathbf{x})]$  and  $\tilde{\mu}_0 = 0$ .
  - 3: **for**  $t = 1$  to  $T$  **do**
  - 4: Let  $r \leftarrow 1$  with probability  $\frac{nk}{nk+t}$  (and 0 otherwise)
  - 5: **if**  $r = 1$  **then**
  - 6:  $\tilde{\mu}_t \leftarrow \text{SIMPLEXSAMPLE}$ .
  - 7:  $\tilde{\mathbf{f}}_t \leftarrow 0$ .
  - 8: **else**
  - 9:  $\tilde{\mu}_t \leftarrow \tilde{\mu}_{t-1}$ .
  - 10:  $\tilde{\mathbf{f}}_t \leftarrow \text{ELLIPSOIDSAMPLE}(\mathbf{x}_t)$ .
  - 11: **end if**
  - 12: Update  $\mathbf{x}_{t+1} = \arg \min_{\mathbf{x} \in \mathcal{K}} \underbrace{\left[ \eta \sum_{\tau=1}^t \tilde{\mathbf{f}}_\tau^\top \mathbf{x} + \mathcal{R}(\mathbf{x}) \right]}_{\Phi_t(\mathbf{x})}$
  - 13: **end for**
- 

Now, the `SIMPLEXSAMPLE` procedure is invoked with probability  $\frac{nk}{nk+t}$  for any time period  $t > 0$ . Once invoked, it samples a coordinate  $i_t \in [n]$  with the uniform distribution. The point  $\mathbf{y}_t$  chosen by the algorithm is the corresponding vertex  $\mathbf{e}_{i_t}$  of the  $n$ -dimensional simplex (which is assumed to be contained inside of  $\mathcal{K}$ ) to obtain the coordinate  $\mathbf{f}_t(i_t)$  as the cost.

It then chooses one of the samples  $S_{i_t,1}, S_{i_t,2}, \dots, S_{i_t,k}$  uniformly at random and replaces it with the value  $\mathbf{f}_t(i_t)$ , and updates  $\tilde{\mu}_t$ . This exactly implements the reservoir sampling for each coordinate. The algorithm is given in the following figure.

---

**Algorithm 2** `SIMPLEXSAMPLE`


---

- 1: Choose  $i_t$  uniformly at random from  $\{1, \dots, n\}$  and  $j$  uniformly at random from  $\{1, \dots, k\}$ .
  - 2: Predict  $\mathbf{y}_t = \mathbf{e}_{i_t}$ , i.e. the  $i_t$ ’th standard basis vector.
  - 3: Observe the cost  $\mathbf{f}_t^\top \mathbf{y}_t = \mathbf{f}_t(i_t)$ .
  - 4: Update the sample  $S_{i_t,j} = \mathbf{f}_t(i_t)$ .
  - 5: Update  $\tilde{\mu}_t(i) = \frac{1}{k} \sum_{j=1}^k S_{i,j}$ , for all  $i \in [n]$ .
- 

As for the `ELLIPSOIDSAMPLE` procedure, it is a modification of the sampling procedure of [1]. The point  $\mathbf{y}_t$  chosen by the algorithm is uniformly at random chosen from the endpoints of the principal axes of the Dikin ellipsoid  $W_1(\mathbf{x}_t)$  centered at  $\mathbf{x}_t$ . The analysis of [1] already does the “hard work” of making certain that the ellipsoidal sampling is unbiased and has low variation with respect to the regularization. However, to take advantage of the low variation in the data, we incorporate the previous information in the form of  $\tilde{\mu}$ .

This modification seems to be applicable more generally, not only to the algorithm of [1]. However, plugged into this recent algorithm we obtain the best possible regret bounds and also an efficient algorithm.

---

**Algorithm 3** ELLIPSOIDSAMPLE( $\mathbf{x}_t$ )

---

- 1: Let  $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$  and  $\{\lambda_1, \dots, \lambda_n\}$  be the set of eigenvectors and eigenvalues of  $\nabla^2 \mathcal{R}(\mathbf{x}_t)$ .
- 2: Choose  $i_t$  uniformly at random from  $\{1, \dots, n\}$  and  $\varepsilon_t = \pm 1$  with probability  $1/2$ .
- 3: Predict  $\mathbf{y}_t = \mathbf{x}_t + \varepsilon_t \lambda_{i_t}^{-1/2} \mathbf{v}_{i_t}$ .
- 4: Observe the cost  $\mathbf{f}_t^\top \mathbf{y}_t$ .
- 5: Define  $\tilde{\mathbf{f}}_t$  to be:

$$\tilde{\mathbf{f}}_t = \tilde{\mu}_t + \tilde{\mathbf{g}}_t$$

---

Where  $\tilde{\mathbf{g}}_t := n(\mathbf{f}_t^\top \mathbf{y}_t - \tilde{\mu}_t^\top \mathbf{y}_t) \varepsilon_t \lambda_{i_t}^{1/2} \mathbf{v}_{i_t}$ .

---

#### 4 Analysis

In this section, we prove a slightly weaker regret bound, in a setting where we know an upper bound  $Q$  on the total variation  $Q_T$ . The main theorem proved here is the following:

**THEOREM 6.** *Let  $Q$  be an estimated upper bound on  $Q_T$ . Suppose that Algorithm 1 is run with  $\eta = \min\left\{\sqrt{\frac{\log(T)}{n^2(Q+nk)}}, \frac{1}{2(n+1)}\right\}$  and  $k = \log(T)$ . Then, if  $Q_T \leq Q$ , the expected regret is bounded as follows:*

$$\mathbf{E}[\text{Regret}_T] = O\left(n\sqrt{\vartheta Q \log(T)} + \sqrt{\vartheta} n^{1.5} \log^2(T)\right).$$

Although this bound requires an estimate of the total variation, we show in the Appendix A how to remove this dependence, thereby proving Theorem 4. In this section we sketch the simpler proof of Theorem 6 and give precise proofs of the main lemmas involved.

*Proof.* [Proof sketch] We first relate the expected regret of Algorithm 1 which plays the points  $\mathbf{y}_t$ , for  $t = 1, 2, \dots$  with the  $\mathbf{f}_t$  cost vectors to the expected regret of another algorithm that plays the points  $\mathbf{x}_t$  with the  $\tilde{\mathbf{f}}_t$  cost vectors. The proof of the following lemma appears in Appendix B.

**LEMMA 7.** *For any  $\mathbf{u} \in \mathcal{K}$ ,*

$$\mathbf{E}\left[\sum_{t=1}^T \mathbf{f}_t^\top (\mathbf{y}_t - \mathbf{u})\right] \leq \mathbf{E}\left[\sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u})\right] + O(n \log^2(T)).$$

Intuitively, this bound holds since in every ELLIPSOIDSAMPLE step, the expectation of  $\tilde{\mathbf{f}}_t$  and  $\mathbf{x}_t$  (conditioned

on all previous randomization) are  $\mathbf{f}_t$  and  $\mathbf{x}_t$  respectively, the expected costs for both algorithms is the same in such rounds. In the SIMPLEXSAMPLE steps, we pessimistically bound the difference in costs of the two algorithms by  $|\mathbf{f}_t^\top (\mathbf{x}_t - \mathbf{u})|$ , which is at most 2. The expected number of such steps is  $O(nk \log(T)) = O(n \log^2(T))$ , which yields the extra additive term.

We therefore turn to bounding  $\sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u})$ . Using standard techniques and previous observations, the regret can be re-written as follows (for proof see Appendix B):

**LEMMA 8.** *For any sequence of cost functions  $\tilde{\mathbf{f}}_1, \dots, \tilde{\mathbf{f}}_T \in \mathbb{R}^n$ , the FTRL algorithm with a  $\vartheta$ -self concordant barrier  $\mathcal{R}$  has the following regret guarantee: for any  $\mathbf{u} \in \mathcal{K}$ , we have*

$$\sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u}) \leq \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) + \frac{2}{\eta} \vartheta \log T.$$

We now turn to bounding the term  $\tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1})$ . The following main lemma give such bounds, and forms the main part of the theorem. We go into detail of its proof in the next section, as it contains the main new ideas.

**LEMMA 9.** *Let  $t$  be an ELLIPSOIDSAMPLE step. Then we have*

$$\tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \leq 8\eta n^2 \|\mathbf{f}_t - \mu_t\|^2 + 10\eta n^2 \|\mu_t - \tilde{\mu}_t\|^2 + 2\mu_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}).$$

A similar but much easier statement can be made for SIMPLEXSAMPLE steps. Trivially, since we set  $\tilde{\mathbf{f}}_t = 0$  in such steps, we have  $\mathbf{x}_t = \mathbf{x}_{t+1}$ . Thus, we have

$$\tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) = 0 = 2\mu_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}).$$

Summing up over all time periods  $t$  we get

$$(4.5) \quad \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \leq 8\eta n^2 \sum_{t=1}^T \|\mathbf{f}_t - \mu_t\|^2 + 10\eta n^2 \sum_{t=1}^T \|\mu_t - \tilde{\mu}_t\|^2 + 2 \sum_{t=1}^T \mu_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1})$$

We bound each term of the inequality (4.5) above separately. The first term can be easily bounded by the total variation, even though it is the sum of squared deviations from changing means. Essentially, the means don't change very much as time goes on. We omit the proof from this extended abstract, but it is based on Lemma 7 from [11].

**LEMMA 10.**  $\sum_{t=1}^T \|\mathbf{f}_t - \mu_t\|^2 \leq Q_T + nk$ .

The second term, in expectation, is just the variance of the estimators  $\tilde{\mu}_t$  of  $\mu_t$ , which can be bounded in terms of the size of the reservoir and the total variation (see Lemma 3). The proof appears in Appendix B.

LEMMA 11.  $\mathbf{E} \left[ \sum_{t=1}^T \|\mu_t - \tilde{\mu}_t\|^2 \right] \leq \frac{\log(T)}{k} (Q_T + nk)$ .

The third term can be bounded by the sum of successive differences of the means, which, in turn, can be bounded the logarithm of the total variation. The proof is omitted from this extended abstract, but it is based on Lemma 8 from [12].

LEMMA 12.  $\sum_{t=1}^T \mu_t(\mathbf{x}_t - \mathbf{x}_{t+1}) \leq 2 \log(Q_T + nk)$ .

Plugging the bounds from Lemmas 10, 11, and 12 into (4.5), and using the value  $k = \log(T)$ , we obtain the following bound on the expected regret of the algorithm: for any  $\mathbf{u} \in \mathcal{K}$ , we have

$$\mathbf{E} \left[ \sum_{t=1}^T \mathbf{f}_t^\top(\mathbf{y}_t - \mathbf{u}) \right] \leq O \left( \eta n^2 (Q_T + nk) + (\vartheta/\eta) \log(T) + \log(Q_T + nk) + n \log^2(T) \right).$$

Now, choosing  $\eta = \min \left\{ \sqrt{\frac{\vartheta \log(T)}{n^2(Q_T + nk)}}, \frac{1}{n} \right\}$ , for some upper bound  $Q \geq Q_T$ , and we get the following regret bound (using the value  $k = \log(T)$ ):

$$\mathbf{E} \left[ \sum_{t=1}^T \mathbf{f}_t^\top(\mathbf{y}_t - \mathbf{u}) \right] \leq O \left( n \sqrt{\vartheta Q \log(T)} + \sqrt{\vartheta} n^{1.5} \log^2(T) \right).$$

**4.1 Proof of Lemma 9.** In order to prove Lemma 9, we first develop some machinery to assist us. The following lemma is a generalization of Lemma 6 in [1] to the case in which we have both sampling and ellipsoidal steps, and its proof is omitted from this extended abstract.

LEMMA 13. *For any time period  $t$ , the next minimizer  $\mathbf{x}_{t+1}$  is “close” to  $\mathbf{x}_t$ :*

$$\mathbf{x}_{t+1} \in W_{\frac{1}{2}}(\mathbf{x}_t).$$

LEMMA 14. *For any time period  $t$ , we have*

$$\|\mathbf{x}_t - \mathbf{x}_{t+1}\|_{\mathbf{x}_t}^2 \leq 4\eta \tilde{\mathbf{f}}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}).$$

*Proof.* Applying the Taylor series expansion to the function  $\Phi_t$  around the point  $\mathbf{x}_t$ , we get that for some

point  $\mathbf{z}_t$  on the line segment joining  $\mathbf{x}_t$  to  $\mathbf{x}_{t+1}$ , we have

$$\begin{aligned} \Phi_t(\mathbf{x}_t) &= \Phi_t(\mathbf{x}_{t+1}) + \nabla \Phi_t(\mathbf{x}_{t+1})^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \\ &\quad + (\mathbf{x}_{t+1} - \mathbf{x}_t)^\top \nabla^2 \Phi_t(\mathbf{z}_t) (\mathbf{x}_{t+1} - \mathbf{x}_t) \\ &= \Phi_t(\mathbf{x}_{t+1}) + \|\mathbf{x}_{t+1} - \mathbf{x}_t\|_{\mathbf{z}_t}^2, \end{aligned}$$

because  $\nabla \Phi_t(\mathbf{x}_{t+1}) = 0$  since  $\mathbf{x}_{t+1}$  is the unique minimizer of  $\Phi_t$  in  $\mathcal{K}$ . We also used the fact that  $\nabla^2 \Phi_t(\mathbf{z}_t) = \nabla^2 \mathcal{R}(\mathbf{z}_t)$ . Thus, we have

$$\begin{aligned} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|_{\mathbf{z}_t}^2 &= \Phi_t(\mathbf{x}_t) - \Phi_t(\mathbf{x}_{t+1}) \\ &= \Phi_{t-1}(\mathbf{x}_t) - \Phi_{t-1}(\mathbf{x}_{t+1}) \\ &\quad + \eta \mathbf{f}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}) \\ &\leq \eta \tilde{\mathbf{f}}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}), \end{aligned}$$

since  $\mathbf{x}_t$  is the minimizer of  $\Phi_{t-1}$  in  $\mathcal{K}$ . It remains to show that  $\frac{1}{4} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|_{\mathbf{x}_t}^2 \leq \|\mathbf{x}_{t+1} - \mathbf{x}_t\|_{\mathbf{z}_t}^2$ . Thus follows because  $\mathbf{z}_t \in W_{1/2}(\mathbf{x}_t)$  since  $\mathbf{x}_{t+1} \in W_{1/2}(\mathbf{x}_t)$  by Lemma 13, and hence, we have  $\frac{1}{4} \nabla^2 \mathcal{R}(\mathbf{x}_t) \preceq \nabla^2 \mathcal{R}(\mathbf{z}_t)$  by (2.2). ■

*Proof. [Lemma 9]*

First, we have

$$\begin{aligned} &(\tilde{\mathbf{f}}_t - \mu_t)^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \\ &\leq \|\tilde{\mathbf{f}}_t - \mu_t\|_{\mathbf{x}_t}^* \cdot \|\mathbf{x}_t - \mathbf{x}_{t+1}\|_{\mathbf{x}_t} \quad (\text{by (2.1)}) \\ &\leq \|\tilde{\mathbf{f}}_t - \mu_t\|_{\mathbf{x}_t}^* \cdot \sqrt{4\eta \tilde{\mathbf{f}}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1})} \quad (\text{Lemma 14}) \\ &\leq 2\eta \|\tilde{\mathbf{f}}_t - \mu_t\|_{\mathbf{x}_t}^{*2} + \frac{1}{2} \tilde{\mathbf{f}}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}). \end{aligned}$$

The last inequality follows using the fact that  $ab \leq \frac{1}{2}(a^2 + b^2)$  for real numbers  $a, b$ . Simplifying, we get that

$$\begin{aligned} &\tilde{\mathbf{f}}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}) \\ &\leq 4\eta \|\tilde{\mathbf{f}}_t - \mu_t\|_{\mathbf{x}_t}^{*2} + 2\mu_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}) \\ &\leq 8\eta \left( \|\tilde{\mathbf{f}}_t - \tilde{\mu}_t\|_{\mathbf{x}_t}^{*2} + \|\mu_t - \tilde{\mu}_t\|_{\mathbf{x}_t}^{*2} \right) + 2\mu_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}) \\ &\leq 4\eta \left( \|\tilde{\mathbf{g}}_t\|_{\mathbf{x}_t}^{*2} + \|\mu_t - \tilde{\mu}_t\|^2 \right) + 2\mu_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}). \end{aligned}$$

The last inequality is because  $\|\cdot\|_{\mathbf{x}}^* \leq \|\cdot\|$  from (2.3).

Using the definition of  $\tilde{\mathbf{g}}_t$  from Algorithm 3, we get that

$$\begin{aligned} \|\tilde{\mathbf{g}}_t\|_{\mathbf{x}_t}^{*2} &= n^2 \left( (\mathbf{f}_t - \tilde{\mu}_t)^\top \mathbf{y}_t \right)^2 \lambda_{i_t} \cdot \left( \mathbf{v}_{i_t}^\top [\nabla^2 \mathcal{R}(\mathbf{x}_t)]^{-1} \mathbf{v}_{i_t} \right) \\ &= n^2 \left( (\mathbf{f}_t - \tilde{\mu}_t)^\top \mathbf{y}_t \right)^2 \\ &\leq n^2 \|\mathbf{f}_t - \tilde{\mu}_t\|^2 \\ &\leq 2n^2 \left[ \|\mathbf{f}_t - \mu_t\|^2 + \|\mu_t - \tilde{\mu}_t\|^2 \right]. \end{aligned}$$

The first inequality follows by applying Cauchy-Schwarz and using the fact that  $\|\mathbf{y}_t\| \leq 1$ . Plugging this bound

into the previous bound we conclude that

$$\begin{aligned} \tilde{\mathbf{f}}_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}) &\leq 8\eta n^2 \|\mathbf{f}_t - \mu_t\|^2 + 10\eta n^2 \|\mu_t - \tilde{\mu}_t\|^2 \\ &\quad + 2\mu_t^\top(\mathbf{x}_t - \mathbf{x}_{t+1}). \end{aligned}$$

■

## 5 Conclusions and Open Problems

In this paper, we gave the first bandit online linear optimization algorithm whose regret is bounded by the square-root of the total quadratic variation of the cost vectors. These bounds naturally interpolate between the worst-case and stochastic models of the problem.

This algorithm continues a line of work which aims to prove variation-based regret bounds for any online learning framework. So far, such bounds have been obtained for four major online learning scenarios: expert prediction, online linear optimization, portfolio selection (and exp-concave cost functions), and bandit online linear optimization. Future work includes proving such bounds for other online learning scenarios.

A specific open problem that remains from our work is to remove the dependence on  $\text{poly}(\log(T))$  in the regret bound, and replace it by  $\text{poly}(\log(Q))$  dependence. This seems to be quite a challenging problem.

## References

- [1] J. Abernethy, E. Hazan, and A. Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *COLT*, 2008.
- [2] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2003.
- [3] B. Awerbuch and R. D. Kleinberg. Adaptive routing with end-to-end feedback: distributed learning and geometric approaches. In *STOC*, pages 45–53, 2004.
- [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] N. Cesa-Bianchi, Y. Mansour, and G. Stoltz. Improved second-order bounds for prediction with expert advice. *Mach. Learn.*, 66(2-3):321–352, 2007.
- [6] Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- [7] V. Dani, T. Hayes, and S. Kakade. The price of bandit information for online optimization. In *NIPS*, 2008.
- [8] V. Dani and T. P. Hayes. Robbing the bandit: less regret in online geometric optimization against an adaptive adversary. In *SODA*, pages 937–943, 2006.
- [9] A. D. Flaxman, A. T. Kalai, and H. B. McMahan. Online convex optimization in the bandit setting: gradient

descent without a gradient. In *SODA*, pages 385–394, 2005.

- [10] J. Hannan. Approximation to bayes risk in repeated play. In M. Dresher, A. W. Tucker, and P. Wolfe, editors, *Contributions to the Theory of Games, volume III*, pages 97–139, 1957.
- [11] E. Hazan and S. Kale. Extracting certainty from uncertainty: Regret bounded by variation in costs. In *COLT*, 2008.
- [12] E. Hazan and S. Kale. How to invest in calmer markets (without compromising universality). *Manuscript*, 2008.
- [13] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*.
- [14] H. B. McMahan and A. 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] H. Robbins. Some aspects of the sequential design of experiments. *Bull. Amer. Math. Soc.*, 58(5):527–535, 1952.
- [18] J. S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Softw.*, 11(1):37–57, 1985.

## A Tuning the learning rate: Proof of Theorem 4

Theorem 6 requires *a priori* knowledge of a good bound  $Q$  on the total quadratic variation  $Q_T$ . This may not be possible in many situations. Typically, in online learning scenarios where a regret bound of  $O(\sqrt{A_T})$  for some quantity  $A_T$  which grows with  $T$  is desired, one first gives an online learning algorithm  $L(\eta)$  where  $\eta \leq 1$  is a learning rate parameter which obtains a regret bound of

$$\text{Regret}_T \leq \eta A_T + O(1/\eta).$$

Then, we can obtain a master online learning algorithm whose regret grows like  $O(\sqrt{A_T})$  as follows. We start with  $\eta = 1$ , and run the learning algorithm  $L(\eta)$ . Then, the master algorithm tracks how  $A_T$  grows with  $T$ . As soon as  $A_T$  quadruples, the algorithm resets  $\eta$  to half its current value, and restarts with  $L(\eta)$ . This simple trick can be shown to obtain  $O(\sqrt{A_T})$  regret.

Unfortunately, this trick doesn't work in our case, where  $A_T = Q_T$ , since we cannot even compute  $Q_T$  accurately in the bandit setting. For this reason, obtaining a regret bound of  $\tilde{O}(\sqrt{Q_T})$  becomes quite non-trivial. In this section, we give a method to obtain such a regret bound. At its heart, we still make use of the  $\eta$ -halving trick, but in a subtle way. We assume that we know a good bound on  $\log(T)$  in advance. This

is not a serious restriction, it can be circumvented by standard tricks, but we make this assumption in order to simplify the exposition.

We design our master algorithm in the following way. Let  $L(\eta)$  be Algorithm 1 with the given parameter  $\eta$  and  $k = \log(T)$ . We initialize  $\eta_0 = 1/2(n+1)$ . The master algorithm then runs in phases indexed by  $i = 0, 1, 2, \dots$ . In phase  $i$ , the algorithm runs  $L(\eta_i)$  where  $\eta_i = \eta_0/2^i$ . The decision to end a phase  $i$  and start phase  $i+1$  is taken in the following manner: let  $t_i$  be first period of phase  $i$ , and let  $t$  be the current period. We start phase  $i+1$  as soon as

$$\sum_{\tau=t_i}^t \tilde{\mathbf{f}}_{\tau}^{\top}(\mathbf{x}_{\tau} - \mathbf{x}_{\tau+1}) \geq \frac{2}{\eta_i} \vartheta \log(T)$$

(thus, phase  $i$  ends at time period  $t-1$ , and we don't actually use the  $\mathbf{x}_t$  computed by  $L(\eta_i)$ , since  $L(\eta_{i+1})$  starts at this point). Note that this sum can be computed by the algorithm. Define  $I_i = \{t_i, t_i+1, \dots, t_{i+1}-1\}$ , i.e. the interval of time periods which constitute phase  $i$ .

By Lemma 8, for any  $\mathbf{u} \in \mathcal{K}$ , we have

$$\begin{aligned} \sum_{t \in I_i} \tilde{\mathbf{f}}_t^{\top}(\mathbf{x}_t - \mathbf{u}) &\leq \sum_{t \in I_i} \tilde{\mathbf{f}}_t^{\top}(\mathbf{x}_t - \mathbf{x}_{t+1}) + \frac{2}{\eta_i} \vartheta \log(T) \\ &\leq \frac{4}{\eta_i} \vartheta \log(T). \end{aligned}$$

Note that this inequality uses the fact that  $\sum_{\tau=t_i}^t \tilde{\mathbf{f}}_{\tau}^{\top}(\mathbf{x}_{\tau} - \mathbf{x}_{\tau+1})$  is a monotonically increasing function of  $t$ , since by Lemma 14, we have that  $\tilde{\mathbf{f}}_t^{\top}(\mathbf{x}_t - \mathbf{x}_{t+1}) \geq 0$ .

Let  $i^*$  be the index of the final phase. Summing up this bound over all phases, we have

$$\sum_{t=1}^T \tilde{\mathbf{f}}_t^{\top}(\mathbf{x}_t - \mathbf{u}) \leq \sum_{i=0}^{i^*} \frac{4}{\eta_i} \vartheta \log(T) \leq \frac{8}{\eta_{i^*}} \vartheta \log(T).$$

Then, using Lemma 7 we get that the expected regret of this algorithm is bounded by

$$\begin{aligned} \mathbf{E} \left[ \sum_{t=1}^T \mathbf{f}_t^{\top}(\mathbf{y}_t - \mathbf{u}) \right] &\leq \mathbf{E}[1/\eta_{i^*}] \cdot 8\vartheta \log(T) \\ (1.6) \quad &+ O(n \log^2(T)). \end{aligned}$$

We now need to bound  $\mathbf{E}[1/\eta_{i^*}]$ . For this, consider the phase  $i^*-1$ . For brevity, let  $J = I_{i^*-1} \cup \{t_{i^*}\}$ . For this interval, we have (here, we assume that  $\mathbf{x}_{t_{i^*}}$  is computed by  $L(\eta_{i^*-1})$ ):

$$\sum_{t \in J} \tilde{\mathbf{f}}_t^{\top}(\mathbf{x}_t - \mathbf{x}_{t+1}) \geq \frac{2}{\eta_{i^*-1}} \vartheta \log(T) = \frac{1}{\eta_{i^*}} \vartheta \log(T).$$

Applying the bound (4.5), and using the fact that  $\eta_{i^*-1} = 2\eta_{i^*}$ , we get

$$\begin{aligned} \sum_{t \in J} \tilde{\mathbf{f}}_t^{\top}(\mathbf{x}_t - \mathbf{x}_{t+1}) &\leq 16\eta_{i^*} n^2 \sum_{t \in J} \|\mathbf{f}_t - \mu_t\|^2 \\ &+ 20\eta_{i^*} n^2 \sum_{t \in J} \|\mu_t - \tilde{\mu}_t\|^2 \\ &+ 2 \sum_{t \in J} \mu_t^{\top}(\mathbf{x}_t - \mathbf{x}_{t+1}). \end{aligned}$$

Putting these together, and dividing by  $\eta_{i^*}$ , we get

$$\begin{aligned} \frac{1}{\eta_{i^*}} \vartheta \log(T) &\leq 16n^2 \sum_{t \in J} \|\mathbf{f}_t - \mu_t\|^2 \\ (1.7) \quad &+ 20n^2 \sum_{t \in J} \|\mu_t - \tilde{\mu}_t\|^2 \\ &+ \frac{2}{\eta_{i^*}} \sum_{t \in J} \mu_t^{\top}(\mathbf{x}_t - \mathbf{x}_{t+1}). \end{aligned}$$

Lemmas 10 and 12 enable us to upper bound  $\sum_{t \in J} \|\mathbf{f}_t - \mu_t\|^2 \leq Q_T + nk$  and  $\sum_{t \in J} \mu_t^{\top}(\mathbf{x}_t - \mathbf{x}_{t+1}) \leq 2 \log(Q_T + nk)$ . Now, we take the expectation on both sides of the inequality over the randomness in phase  $i^*-1$ . Denoting this expectation by  $\mathbf{E}_{i^*-1}$ , we have the bound

$$\mathbf{E}_{i^*-1} \left[ \sum_{t \in J} \|\mu_t - \tilde{\mu}_t\|^2 \right] \leq \frac{\log(T)}{k} (Q_T + nk),$$

by Lemma 11. Plugging these bounds into (1.7), and using  $k = \log(T)$ , we get

$$\frac{1}{\eta_{i^*}} \vartheta \log(T) \leq 36n^2(Q_T + nk) + \frac{4}{\eta_{i^*}} \log(Q_T + nk).$$

Now, one of  $36n^2(Q_T + nk)$  or  $\frac{4}{\eta_{i^*}} \log(Q_T + nk)$  must be at least  $\frac{1}{2\eta_{i^*}} \vartheta \log(T)$ . In the first case, we get the bound  $\frac{1}{\eta_{i^*}} \leq 10n \sqrt{\frac{Q_T + nk}{\vartheta \log(T)}}$ . In the second case, we get the bound  $\frac{1}{\eta_{i^*}} \leq \frac{8 \log(Q_T + nk)}{\vartheta \log(T)}$ . In either case, we can bound

$$\mathbf{E}[1/\eta_{i^*}] \cdot 8\vartheta \log(T) \leq O\left(n \sqrt{\vartheta(Q_T + nk) \log(T)}\right).$$

Plugging this into (1.6), and for  $k = \log(T)$ , we get that the expected regret is bounded by

$$\begin{aligned} \mathbf{E} \left[ \sum_{t=1}^T \mathbf{f}_t^{\top}(\mathbf{y}_t - \mathbf{u}) \right] \\ \leq O\left(n \sqrt{\vartheta Q_T \log(T)} + \sqrt{\vartheta} n^{1.5} \log^2(T)\right). \end{aligned}$$

## B Proofs of some lemmas

*Proof.* [Lemma 7]

Let  $t$  be an ELLIPSOIDSAMPLE step. We first show that  $\mathbf{E}[\tilde{\mathbf{f}}_t] = \mathbf{f}_t$ . We condition on all the randomness prior to this step, thus,  $\tilde{\mu}_t$  is fixed. In the following,  $\mathbf{E}_t$  denotes this conditional expectation. Now, condition on the choice  $i_t$  and average over the choice of  $\varepsilon_t$ :

$$\begin{aligned} \mathbf{E}_t[\tilde{\mathbf{g}}_t] &= \frac{1}{2}n \left( (\mathbf{f}_t - \tilde{\mu}_t)^\top (\mathbf{x}_t + \lambda_{i_t}^{-1/2} \mathbf{v}_{i_t}) \right) \lambda_{i_t}^{1/2} \mathbf{v}_{i_t} \\ &\quad - \frac{1}{2}n \left( (\mathbf{f}_t - \tilde{\mu}_t)^\top (\mathbf{x}_t - \lambda_{i_t}^{-1/2} \mathbf{v}_{i_t}) \right) \lambda_{i_t}^{1/2} \mathbf{v}_{i_t} \\ &= n((\mathbf{f}_t - \tilde{\mu}_t)^\top \mathbf{v}_{i_t}) \mathbf{v}_{i_t}. \end{aligned}$$

Hence,

$$\mathbf{E}_t[\tilde{\mathbf{g}}_t] = \sum_{i=1}^n \frac{1}{n} \cdot n((\mathbf{f}_t - \tilde{\mu}_t)^\top \mathbf{v}_i) \mathbf{v}_i = \mathbf{f}_t - \tilde{\mu}_t,$$

since the  $\mathbf{v}_i$  form an orthonormal basis. Thus,  $\mathbf{E}_t[\tilde{\mathbf{f}}_t] = \mathbf{E}_t[\tilde{\mathbf{g}}_t] + \tilde{\mu}_t = \mathbf{f}_t$ .

Furthermore, it is easy to see that  $\mathbf{E}_t[\mathbf{y}_t] = \mathbf{x}_t$ , since  $\mathbf{y}_t$  is drawn from a symmetric distribution centered at  $\mathbf{x}_t$  (namely, the uniform distribution on the endpoints of the principal axes of the Dikin ellipsoid centered at  $\mathbf{x}_t$ ). Thus, we conclude that

$$\mathbf{E}_t(\mathbf{f}_t^\top (\mathbf{y}_t - \mathbf{u})) = \mathbf{f}_t^\top (\mathbf{x}_t - \mathbf{u}) = \mathbf{E}_t(\tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u})),$$

and hence, taking expectation over all the randomness, we have

$$\mathbf{E}(\mathbf{f}_t^\top (\mathbf{y}_t - \mathbf{u})) = \mathbf{E}(\tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u})).$$

Now, let  $t$  be a SIMPLEXSAMPLE step. In this case, we have  $\|\mathbf{f}_t^\top (\mathbf{y}_t - \mathbf{u})\| \leq \|\mathbf{f}_t\| \|\mathbf{y}_t - \mathbf{u}\| \leq 2$ , and  $\tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u}) = 0$ . Thus,

$$\mathbf{E}(\mathbf{f}_t^\top (\mathbf{y}_t - \mathbf{u})) \leq \mathbf{E}(\tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u})) + 2.$$

Overall, if  $X$  is the number of SIMPLEXSAMPLE sampling steps, we have

$$\mathbf{E}(\mathbf{f}_t^\top (\mathbf{y}_t - \mathbf{u})) \leq \mathbf{E}_t(\tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u})) + 2\mathbf{E}[X].$$

Finally, using the fact that  $\mathbf{E}[X] = \sum_{t=1}^T \frac{nk}{t} \leq nk \log(T) \leq n \log^2(T)$ , the proof is complete. ■

*Proof.* [Lemma 8]

The following bound on the regret is fairly standard for the FTRL algorithm (for proof see Lemma 2 in [1]):

$$\begin{aligned} \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{u}) &\leq \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) \\ &\quad + \frac{1}{\eta} [\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1) - \nabla \mathcal{R}(\mathbf{x}_1)^\top (\mathbf{u} - \mathbf{x}_1)] \\ &= \sum_{t=1}^T \tilde{\mathbf{f}}_t^\top (\mathbf{x}_t - \mathbf{x}_{t+1}) + \frac{1}{\eta} [\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1)] \end{aligned}$$

since  $\nabla \mathcal{R}(\mathbf{x}_1) = 0$  because  $\mathbf{x}_1$  minimizes  $\mathcal{R}$  over  $\mathcal{K}$ .

As in [1], we restrict our attention to a particular set of  $\mathbf{u}$ 's: viz. the Minkowski section  $\mathcal{K}_{1/T}$ . By equation (2.4), we have that  $\mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{x}_1) \leq 2\vartheta \log(T)$ .

Thus, to compare the performance of the algorithm with respect to any point  $\mathbf{v} \in \mathcal{K}$ , we may instead compare the performance with respect to a point  $\mathbf{u} \in \mathcal{K}_{1/T}$  instead. We can bound the difference in costs as

$$\sum_{t=1}^T |\mathbf{f}_t^\top (\mathbf{v} - \mathbf{u})| \leq \sum_{t=1}^T \|\mathbf{f}_t\| \|\mathbf{v} - \mathbf{u}\| \leq \sum_{t=1}^T 1/T = 1. \quad \blacksquare$$

*Proof.* [Lemma 11]

Fix any time period  $t$ . Now, for any coordinate  $i$ ,  $\tilde{\mu}_t(i)$  is the average of a  $k$  samples chosen *without* replacement from  $F_t := \{\mathbf{f}_\tau(i) \mid \tau = -(nk-1), \dots, 0, 1, \dots, t\}$ , where we set  $\mathbf{f}_\tau = 0$  for  $\tau = -(nk-1), \dots, 0$ . Thus, we have  $\mathbf{E}[\tilde{\mu}_t(i)] = \mu_t(i)$ , and hence  $\mathbf{E}[(\tilde{\mu}_t(i) - \mu_t(i))^2] = \text{VAR}[\tilde{\mu}_t(i)]$ .

Now consider another estimator  $\nu_t(i)$ , which averages  $k$  samples chosen *with* replacement from  $F_t$ . It is easy to check that  $\text{VAR}[\tilde{\mu}_t(i)] \leq \text{VAR}[\nu_t(i)]$ . Thus, we bound  $\text{VAR}[\nu_t(i)]$  instead.

Let  $\mu = \frac{1}{T} \sum_{t=1}^T \mathbf{f}_t$ . We have

$$\begin{aligned} \text{VAR}[\nu_t(i)] &= \mathbf{E}[(\nu_t(i) - \mu_t(i))^2] \\ &\leq \mathbf{E}[(\nu_t(i) - \mu(i))^2] \\ &= \frac{1}{k} \sum_{\tau=-(nk-1)}^t \frac{1}{t+nk} (\mathbf{f}_t(i) - \mu(i))^2 \\ &\leq \frac{1}{tk} \left[ nk\mu(i)^2 + \sum_{\tau=1}^T (\mathbf{f}_t(i) - \mu(i))^2 \right]. \end{aligned}$$

Summing up over all coordinates  $i$ , we get

$$\mathbf{E}[\|\tilde{\mu}_t - \mu_t\|^2] \leq \sum_i \text{VAR}[\nu_t(i)] \leq \frac{1}{tk} (Q_T + nk).$$

Summing up over all  $t$ , we get

$$\begin{aligned} \mathbf{E} \left[ \sum_{t=1}^T \|\tilde{\mu}_t - \mu_t\|^2 \right] &\leq \sum_{t=1}^T \frac{1}{tk} (Q_T + nk) \\ &\leq \frac{\log(T)}{k} (Q_T + nk). \end{aligned} \quad \blacksquare$$