Adaptive Algorithms and Data-Dependent Guarantees for Bandit Convex Optimization

Mehryar Mohri
Courant Institute and Google
251 Mercer Street
New York, NY 10012
mohri@cims.nyu.edu

Scott Yang
Courant Institute
251 Mercer Street
New York, NY 10012
yangs@cims.nyu.edu

Abstract

We present adaptive algorithms with strong data-dependent regret guarantees for the problem of bandit convex optimization. In the process, we develop a general framework from which the main previous results in this setting can be recovered. The key method is the introduction of adaptive regularization. By appropriately adapting the exploration scheme, we show that one can derive regret guarantees that can be significantly more favorable than those previously known. Moreover, our analysis also modularizes the problematic quantities in achieving the conjectured minimax optimal rates in the most general setting of the problem.

1 INTRODUCTION

Bandit convex optimization (BCO) is a general scenario for sequential decision making under uncertainty. In contrast to the standard full information setting of online convex optimization, in the BCO scenario, at each round, the learner only receives only the value of the loss function and no other feedback, in particular no information about the function derivatives.

BCO extends the well-known multi-armed bandit scenario and is an instance of the exploration-exploitation dilemma inherent in many online machine learning problems: at each round, the learner must decide between exploring new actions and exploiting the best actions determined thus far.

The partial information assumption also captures many real-world problems where the exact form of the loss function is not readily available at each step. These include settings such as ad prediction and medical diagnosis. In these problems, the learner is typically able to see only the value returned from the current action, as the exact loss function and its gradients are often quite abstract and complex.

From a theoretical vantage point, bandit convex optimization also remains an area of online learning in which existing regret guarantees in several regimes are known to be sub-optimal and where optimal methods have still yet to be discovered.

We now formalize the setting that we consider. Let $\mathcal{K} \subset \mathbb{R}^n$ be a compact convex set, and let $\{f_t\}_{t=1}^{\infty}$ be a sequence of convex functions. At each round $t = 1, 2, \ldots, T$, the learner selects a point $x_t \in \mathcal{K}$ and incurs loss $f_t(x_t)$. The learner's objective is to minimize his regret, defined by:

$$\mathrm{Reg}_T := \max_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x_t) - f_t(x),$$

that is the difference between his cumulative loss and that of the best fixed point $x^* \in \mathcal{K}$ in hindsight. In contrast to the standard online learning or online convex optimization scenarios, in bandit convex optimization, the learner has access only to the value $f_t(x_t)$ and not any higher-order information. This scenario was first studied by Flaxman et al. (2005), where they proved that for sequences of Lipschitz functions, one can achieve a regret that is in $\mathcal{O}(T^{3/4})$. The seminal work of Abernethy et al. (2008) showed that, for linear functions, one can attain a regret in $\mathcal{O}(\sqrt{T})$. Agarwal et al. (2010) showed that one can improve upon Flaxman's bound and attain $\mathcal{O}(T^{2/3})$ in the strongly convex setting, and Saha and Tewari (2011) showed that one can achieve $\mathcal{O}(T^{2/3})$ in the strongly smooth setting. Hazan and Levy (2014) showed that when the functions are guaranteed to be both strongly smooth and strongly convex, one can attain $\mathcal{O}(\sqrt{T})$ regret. Most recently, Bubeck and Eldan (2015) presented a non-constructive proof demonstrating that a $\mathcal{O}(\sqrt{T})$ bound is also theoretically attainable in the general setting, albeit with a much heavier dependence on the dimension of the domain $\mathcal{O}(n^{11})$ than in the other references mentioned above.

It still remains an open question whether one can efficiently obtain the desired $\mathcal{O}(\sqrt{T})$ regret in the purely strongly convex, purely strongly smooth, or purely Lipschitz settings. To make progress in this direction, we will build upon recent advances in other areas of the online convex optimization literature. Specifically, we will draw from the techniques in adaptive regularization presented in (Bartlett et al., 2007; Duchi et al., 2010; McMahan and Streeter, 2010) as well as ideas from the "learning faster from easy data" paradigm studied in (Even-Dar et al., 2007; Bubeck and Slivkins, 2012; Sani et al., 2014; de Rooij et al., 2014) to derive two efficient adaptive algorithms with minimal assumptions on the function's loss sequence.

Our algorithms will provide strong data-dependent guarantees, so that while their regret will never be worse than that of previous algorithms in the same setting, they can also be much better depending on how favorable and "easy" the actual data is. Moreover, the algorithms we present are anytime and automatically adjust to the data, so that they can run without any a priori tuning or unreasonable parameter specification. Perhaps most importantly, analyzing the resulting bounds provides insight into both whether the conjectured optimal bounds are truly achievable as well as how they might viably be attained.

We will start off by introducing some mathematical notation for the rest of our paper. Then, in Section 3, we will describe the general methodology of BCO, and in the process, introduce several key concepts and tools as well as our intuition and contribution to this framework. This will be formalized in Sections 4, 5, and 6, where we introduce concrete algorithms and guarantees. Finally, we will highlight the main implications of our results in Section 7, both in terms of new regret guarantees as well as added insight for the general bandit convex optimization setting with only the Lipschitz loss assumption.

2 NOTATION

In what follows, we will denote by $\mathcal{B}^n$ the n-dimensional unit ball under the Euclidean norm, and $\mathcal{S}^n = \partial\mathcal{B}^n$ the $(n-1)$-dimensional unit sphere. For any sequence of functions $\{c_t\}_{t=1}^{\infty}$, we will write $c_{1:t} = \sum_{s=1}^{t} c_t$. Given a function f_t and a point x_t, we will denote by $g_t \in \partial f_t(x_t)$ an element of the subgradient of f_t at x_t, such that for any y, $f_t(y) \geq f_t(x_t) + g_t^\top(y - x_t)$. Given any norm $\|\cdot\|$, we will denote its dual by $\|\cdot\|_*$, so that $\|x\|_* = \sup_{\|y\|\leq 1} x^\top y$. Moreover, given any symmetric positive semi-definite (SPSD) matrix A, we define the semi-norm $\|x\|_A = \sqrt{x^\top Ax}$, and we denote the j-th eigenvalue of A by $\lambda_j(A)$ (in decreasing order).

Definition 1 (Strongly Smooth and Strongly Convex)**.** *Let A be an SPSD matrix. A function f is said to be*

- *A-strongly smooth if $f(x) \leq f(y) + \nabla f(y)^\top(x-y) + \frac{1}{2}\|x-y\|_A^2$;*
- *A-strongly convex if $f(x) \geq f(y) + \nabla f(y)^\top(x-y) + \frac{1}{2}\|x-y\|_A^2$.*

For a scalar $\beta \in \mathbb{R}_+$, f is said to be β-strongly smooth (β-strongly convex) if it is βI-strongly smooth (respectively strongly convex). For functions that are not C^1, we replace the gradients by subgradients in these definitions.

3 OVERVIEW OF BANDIT CONVEX OPTIMIZATION

Bandit convex optimization, and bandit problems in general, can be viewed as online learning problems with partial information. In this context, the natural approach is to estimate the missing data from the full information setting, and to then apply online learning methods to the problem.

One online learning method that is commonly used in bandit convex optimization is the Follow-the-Regularized-Leader (FTRL) (Kalai and Vempala, 2005) algorithm, which is based on the update:

$$x_{t+1} = \operatorname*{argmin}_{x\in\mathcal{K}} \eta g_{1:t}^\top x + \mathcal{R}(x),$$

where $\mathcal{R}$ is some regularization function.

However, in bandit convex optimization, the missing data at each round is the gradient. Since the learner only knows the value of the loss function at each round, the FTRL algorithm cannot be readily applied (nor can most other online learning algorithms, which also typically use gradient information). This is what makes bandit convex optimization significantly more difficult than standard online onvex optimization.

A key step toward addressing this issue has been the insight that by playing an action randomly near the intended one, it is possible to estimate the gradient of a smoothed version of the loss function. More formally, given any $f\colon \mathbb{R}^n \to \mathbb{R}$ and $A \in \mathbb{R}^{n\times n}$ an SPSD matrix, we define $\widehat{f}(x) = \mathbb{E}_{v\in\mathcal{B}^n}[f(x+Av)]$, the average of f at x over the ellipsoid generated by A, and $\widehat{g}_t = nf(x+Au)A^{-1}u$, its one-point gradient estimate. Then the following result holds:

Lemma 1 (Saha and Tewari (2011))**.** $\mathbb{E}_{u\sim\mathcal{S}^n}[\widehat{g}_t] = \nabla\widehat{f}(x)$.

For completeness, we provide a proof of this result in Appendix A.

This implies that by sampling a point $x + Av$ in an ellipsoid around the intended action, we can estimate the gradient of a smoothed version of our loss function even if we are only able to play a single action. Moreover, by playing these gradient estimates, our regret will be the regret of this smoothed loss up to the approximation error of smoothing.

In practice, the smoothing ellipsoid is defined by scaling the inverse Hessian of the regularization function $\mathcal{R}$, i.e. $A = \delta\nabla^2\mathcal{R}(x)^{-1/2}v$. Thus, the choice of regularization becomes crucial towards determining how much to explore and how much approximation error to incur. The key work along this direction has been (Abernethy et al., 2008), which showed that one can use the notion of self-concordant barrier to find a good tradeoff.

For completeness, we briefly introduce this concept and summarize the key results that we will use in our analysis.

3.1 BACKGROUND ON SELF-CONCORDANT FUNCTIONS

The use of self-concordant functions can be traced back to Nesterov's work on Newton's method (see (Nesterov, 2004) for a comprehensive treatment).

Definition 2 (Self-concordant barrier)**.** *A C^3 function $\mathcal{R}\colon \operatorname{int}(K) \to \mathbb{R}$ is a ν-self concordant barrier if for any $h \in \mathbb{R}^n$:*

1. *$\mathcal{R}$ approaches infinity for any sequence of points approaching the boundary of $\mathcal{K}$.*
2. $\left|\nabla^3\mathcal{R}(x)[h,h,h]\right| \leq 2(\nabla^2\mathcal{R}(x)[h,h])^{3/2}$.
3. $|\nabla\mathcal{R}(x)h| \leq (\nu\nabla^2\mathcal{R}(x)[h,h])^{1/2}$.

Definition 3 (Dikin Ellipsoid)**.** *Let $\mathcal{R}$ be a self-concordant function and $x \in \operatorname{int}(K)$. Then, the Dikin Ellipsoid $W_1(x)$ is the ellipsoid induced by the Hessian of $\mathcal{R}$ at x:*

$$W_1(x) = \{z \in \mathbb{R}^n | \|z - x\|_{\nabla^2\mathcal{R}(x)} \leq 1\} \subset \mathcal{K}.$$

Definition 4 (Newton Decrement)**.** *Given any C^2 function $\mathcal{R}$ whose Hessian is invertible at a point x, the Newton decrement of $\mathcal{R}$ at x is defined to be $\lambda(x,\mathcal{R}) = \|\nabla\mathcal{R}(x)\|_{\nabla^2\mathcal{R}(x)^{-1}}$.*

The following two results can be found in (Nemirovski and Todd, 2008) and will be the most important properties for our analysis.

Lemma 2. *Let $\mathcal{R}$ be a self concordant function and $x \in \operatorname{int}(\mathcal{K})$ a point such that $\lambda(x,\mathcal{R}) \leq \frac{1}{2}$. Then, $\|x - \mathit{argmin}_u\, \mathcal{R}(u)\|_{\nabla^2\mathcal{R}(x)} \leq 2\lambda(x,\mathcal{R})$.*

Given $x, y \in \operatorname{int}(\mathcal{K})$, the Minkowsky function is defined as $\pi_x(y) = \inf\left\{t \geq 0 \mid x + \frac{1}{t}(y - x)\right\}$.

Lemma 3. *Let $\mathcal{R}$ be a ν-self concordant barrier. Then for any $x, y \in \operatorname{int}(\mathcal{K})$: $\mathcal{R}(y) - \mathcal{R}(x) \leq \nu \log\left(\frac{1}{1-\pi_x(y)}\right)$.*

Thus, the current state-of-the-art approach to bandit convex optimization problem has been to play a FTRL-type algorithm with the update:

$$x_{t+1} = \operatorname*{argmin}_{x\in\mathcal{K}} \eta\widehat{g}_{1:t}^{\top}x + \mathcal{R}(x),$$

where $\widehat{g} = \widehat{g}(\delta, \nabla^2\mathcal{R})$, and $\mathcal{R}$ is a self-concordant barrier.

For global σ-strongly convex loss functions, one can also add an associated quadratic term to the optimization problem and a σ-ball to the sampling ellipsoid.

In this paper, we extend the above framework with the concept of adapting to the data. Specifically, we will tune the learning rate and sampling ellipsoid at each step of the algorithm according to the local data that we see. The goal of this approach is two-fold. On the one hand, we want to design any-time algorithms with general regret bounds that recover all existing approaches in a unified manner. Previous algorithms assumed various levels of global regularity information, had different sampling schemes for each, and had to be tuned with a posteriori knowledge. On the other hand, and perhaps more importantly, we also want to derive data-dependent guarantees that can reveal new insight into the difficulties of the problem.

4 ADAPTIVE BANDIT CONVEX OPTIMIZATION

Using the motivation above, we now present AdaBCO, an adaptive procedure for bandit convex optimization. AdaBCO is a skeleton algorithm that we will use as a *launching point* for our two data-dependent algorithms. As such, it is not meant to be implemented on its own, and some of its parameters, such as η_t and δ_t, are not specified precisely. These will be selected carefully in Algorithms 2 and 3.

Unlike previous algorithms in the literature, AdaBCO does not need the learner to specify a priori a fixed level of global convexity for the entire sequence of loss functions encountered during learning. This is often an unreasonable requirement, particularly in a truly online setting, and so instead, AdaBCO allows the learner to specify the convexity of functions as it sees them. The algorithm is designed such that the regret bound will automatically adapt to this data. This is achieved via dynamic tuning of the sampling ellipsoids and learning rates, which will be prescribed more explicitly in Algorithms 2 and 3, when we also take into account the level of function smoothness.

Moreover, it is important to realize that computing parameters in real-time is never more difficult than computing bounds that hold uniformly over all rounds at the start in a truly online scenario. Thus, AdaBCO is never more difficult to implement than previous algorithms.

AdaBCO also differs from previous work in that it treats strong convexity as a matrix parameter instead of a scalar parameter. This is based on the insight that, for minimizing regret, convexity of the loss function is closely tied to convexity of the self-concordant barrier's Hessian, and that one can bound regret in terms of the average eigenvalue of the sum of these matrices as opposed to the minimal eigenvalue. Essentially, the algorithm can "borrow" convexity from the self-concordant barrier if the convexity of the loss function is not strong enough to achieve the desired regret. This becomes particularly useful when the learner is querying points near the decision set's boundary, and the Hessian of $\mathcal{R}$ has large eigenvalues in the direction of the boundary (often the case because $\mathcal{R} \nearrow \infty$ at $\partial\mathcal{K}$). This will become more clear with the data-dependent guarantees and discussion in Sections 5, 6, and 7.

We will first show that AdaBCO yields a strong data-dependent regret bound on the sequence of smoothed loss functions. The proof technique is based on a few key steps. We first use convexity of the loss function to change the problem into bounding the regret of quadratic functions. Then we use the fact that our original algorithm can be seen as a Follow-the-Regularized-Leader algorithm played on this sequence of surrogate loss functions to bound the regret. From here, we leverage the fact that part of our loss function is proximal, along with the properties of self-concordant barriers that we stated, to show that in the local norm, the incremental update can be bounded by the gradient of the (smoothed) loss function. Then we can estimate the gradient in the local norm in terms of the quantities that we have prescribed. Finally, we use more properties about self-concordant barriers to bound their growth near the center of the domain.

In the process, we will require a few technical lemmas about the properties of smoothed functions as well as some results from the general online learning literature. For completeness, all proofs are provided in Appendix A.

4.1 TECHNICAL LEMMAS

The following result is a mild generalization of Lemma 7 in (Hazan and Levy, 2014) and states that a smoothed loss function retains the same strong convexity properties as the original.

Lemma 4. *Let A be an SPSD matrix, and let $f: \mathbb{R}^n \to \mathbb{R}$ be A-strongly convex. Then $\widehat{f}$ is also A-strongly convex.*

Next we state a lemma by Zinkevich (2003), which shows that we can bound the regret of any sequence of loss functions by a lower barrier. This will be useful for switching between our loss functions and the quadratic lower bounds induced by their strong convexity.

Lemma 5. *Let $\{f_t\}_{t=1}^{\infty}$ be a sequence of functions and $\{x_t\}_{t=1}^{\infty} \subset \mathcal{K}$. Suppose there exists a sequence of lower barrier functions $\{h_t\}_{t=1}^{\infty}$ such that $h_t(x_t) = f_t(x_t)$ and $h_t \le f_t$. Then, the following inequality holds:*

$$\max_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x_t) - f_t(x) \le \max_{x\in\mathcal{K}} \sum_{t=1}^{T} h_t(x_t) - h_t(x).$$

The final technical lemma in this section extends the well-known "be-the-leader"-based result of follow-the-regularized-leader type algorithms (originally from (Kalai and Vempala, 2005), to algorithms with adaptive regularization.

Algorithm 1 AdaBCO

```
Input: $\eta_0 = \frac{1}{2nC}$, $\nu$-self concordant barrier $\mathcal{R}$.
Initialize: $x_1 = \text{argmin}_{x\in\mathcal{K}} \mathcal{R}(x)$.
for $t = 1, \dots, T$: do
    Choose matrix $Q_t \succcurlyeq 0$ such that $f_t(x) \ge f_t(x_t) + g_t^\top(x - x_t) + \frac{1}{2}\|x - x_t\|_{Q_t}^2$.
    Define $\eta_t \le \eta_{t-1}$.
    Set $B_t = \left[\nabla^2\mathcal{R}(x_t) + \eta_t Q_{1:t}\right]^{-1/2}$.
    Sample $u \sim \mathcal{S}^n$ uniformly.
    Define $\delta_t$ and set $y_t = x_t + \delta_t B_t u \in W_1(x_t) \subset \mathcal{K}$.
    Play $y_t$ and incur loss $f_t(y_t)$.
    Compute the estimate $\widehat{g}_t = n f_t(y_t)(\delta_t B_t)^{-1} u$.
    Update $x_{t+1} = \text{argmin}_{x\in\mathcal{K}} \widehat{g}_{1:t}^\top x + \frac{1}{2}\sum_{s=1}^{t} \|(x - x_s)\|_{Q_s}^2 + \frac{1}{\eta_t}\mathcal{R}(x)$.
end for
```

Lemma 6. *Let $\{f_t\}_{t=1}^{\infty}$ be a sequence of convex functions defined on a closed convex set $\mathcal{K}$, and let $\{x_t\}_{t=1}^{\infty}$ be a sequence of points in $\mathcal{K}$ such that the subgradient of f_t at x_t is denoted as g_t. Let $\{r_t\}_{t=1}^{\infty}$ be a sequence of non-negative convex functions. Then the update $x_{t+1} =$ argmin$_x$ $g_{1:t}^\top x + r_{0:t}(x)$ incurs regret at most*

$$\sum_{t=1}^{T} f_t(x_t) - f_t(x) \le r_{0:T}(x) + \sum_{t=1}^{T} g_t^\top(x_t - x_{t+1}).$$

We are now able to present the regret guarantee for Algorithm 1:

Theorem 1 (AdaBCO)**.** *Let $\mathcal{K}$ be a convex set with diameter $\mathcal{D}_{\mathcal{K}}$, and $\mathcal{R}$ a ν-self-concordant barrier over $\mathcal{K}$. Assume that $|f| \le C$. Then, for $0 < \eta_t \le \frac{1}{2nC}$, the following regret bound holds for Algorithm 1:*

$$\max_{w\in\mathcal{K}} \sum_{t=1}^{T} \mathbb{E}[\widehat{f}_t(y_t) - \widehat{f}_t(w)] \\ \le \sum_{t=1}^{T} \frac{\eta_t}{\delta_t^2}\mathbb{E}\left[(n f_t(x_t + \delta_t B_t u))^2 + \frac{1}{\eta_T}\nu\log(T)\right].$$

Proof. We will refer back to the series of lemmas presented in the prior sections in our analysis.

Let r_0 and r_t be defined as follows:

$$r_0(x) = \frac{1}{\eta}\mathcal{R}(x), \qquad r_t(x) = \frac{1}{2}\|x - x_t\|_{Q_t}^2.$$

Let $h_0 = r_0$, and for $t \ge 1$, let $h_t(x) = \widehat{g}_t^\top x + r_t(x)$. Then, $h_{0:t}(x) = \widehat{g}_{1:t}^\top x + r_{0:t}(x)$, and the update in Algorithm 1 can be written as $x_{t+1} = \text{argmin}\, h_{0:t}(x)$.

Now, define $b_t(x) = \widehat{g}_t^\top(x - x_t) + r_t(x)$. By our choice of Q_t and the fact that smoothed functions preserve strong convexity (Lemma 4), it follows that $\widehat{f}_t(x) \ge b_t(x) + \widehat{f}_t(x_t)$ for all $x \in \mathcal{K}$, with equality at x_t. Thus, if we define

$\tilde{f}_t(x) = \widehat{g}_t^\top x + r_t(x)$, we can apply Lemma 5 to obtain $\max_x \sum_{t=1}^T \widehat{f}_t(x_t) - \widehat{f}_t(x) \leq \max_x \sum_{t=1}^T \tilde{f}_t(x_t) - \tilde{f}_t(x)$. This helps us reduce upper bounding the regret of our theorem by the regret of quadratic functions plus a self-concordant barrier.

Note that $x_{t+1} = \text{argmin}\ \tilde{f}_{1:t}(x) + \frac{1}{\eta_t}\mathcal{R}(x)$ is a FTRL-style update. Thus, by Lemma 6, the following holds:

$$\sum_{t=1}^T \mathbb{E}\left[\tilde{f}_t(x_t) - \tilde{f}_t(x)\right] \\ \leq \sum_{t=1}^T \mathbb{E}\left[\nabla \tilde{f}_t(x_t)^\top (x_t - x_{t+1}) + \frac{1}{\eta_T}\mathcal{R}(x)\right].$$

The first term can be bounded by

$$\sum_{t=1}^T \mathbb{E}\left[\nabla \tilde{f}_t(x_t)^\top (x_t - x)\right] \\ = \sum_{t=1}^T \mathbb{E}\left[\widehat{g}_t^\top (x_t - x)\right] \\ \leq \sum_{t=1}^T \|\widehat{g}_t\|_{\nabla^2 \eta_t h_{0:t}(x_t),*}\ \|x_t - x_{t+1}\|_{\nabla^2 \eta_t h_{0:t}(x_t)}.$$

Since $x_{t+1} = \text{argmin}_x\, \eta_t h_{0:t}(x)$, Lemma 2 tells us that if the Newton decrement $\lambda(x_t, \eta_t h_{0:t}) = \|\nabla \eta_t h_{0:t}(x_t)\|_{(\nabla^2 \eta_t h_{0:t}(x_t))^{-1}} \leq \frac{1}{2}$, then $\|x_t - x_{t+1}\|_{\nabla^2 \eta_t h_{0:t}(x_t)} \leq 2\lambda(x_t, \eta_t h_{0:t})$. This implies that $\sum_{t=1}^T \nabla \tilde{f}_t(x_t)^\top (x_t - x) \leq \sum_{t=1}^T 2\eta_t \|\widehat{g}_t\|_{\nabla^2 \eta h_{0:t}(x_t),*} \|\nabla h_{0:t}(x_t)\|_{\nabla^2 \eta_t h_{0:t}(x_t),*}$.

At the same time, since

$$x_t = \underset{x}{\text{argmin}}\, h_{0:t-1}(x), \\ h_{0:t-1}(x) + r_t(x) = h_{0:t}(x) - \widehat{g}_t^\top x,$$

and x_t also minimizes r_t, it follows that $0 = \nabla(h_{0:t-1} + r_t)(x_t) = \nabla h_{0:t}(x_t) - \widehat{g}_t$. Thus, the following holds: $\sum_{t=1}^T \nabla \tilde{f}_t(x_t)^\top (x_t - x) \leq \sum_{t=1}^T 2\eta_t \|\widehat{g}_t\|^2_{\nabla^2 \eta_t h_{0:t}(x_t),*}$.

Putting everything together yields:

$$\sum_{t=1}^T \mathbb{E}\left[\widehat{f}_t(x_t) - \widehat{f}_t(x)\right] \\ \leq \left(\sum_{t=1}^T \mathbb{E}\left[2\eta_t \|\widehat{g}_t\|^2_{\nabla^2 \eta_t h_{0:t}(x_t),*}\right]\right) + \mathbb{E}\left[\frac{1}{\eta_T}\mathcal{R}(x)\right].$$

In addition, we know that

$$\mathbb{E}\left[\|\widehat{g}_t\|^2_{\nabla^2 \eta_t h_{0:t}(x_t),*}\right] \\ = \mathbb{E}\left[\|n f_t(y_t)(\delta_t B_t)^{-1} u\|^2_{\nabla^2 \eta_t h_{0:t}(x_t),*}\right] \\ = \mathbb{E}[(n f_t(y_t)(\delta_t B_t)^{-1} u)^\top \nabla^2 \eta_t h_{0:t}(x_t)^{-1} \\ (n f_t(y_t)(\delta_t B_t)^{-1} u)] \\ \leq \mathbb{E}\left[\frac{1}{\delta_t^2}(n f_t(x_t + \delta_t B_t u))^2\right].$$

To bound the self-concordant function, Lemma 3 shows that if $\mathcal{R}$ is any ν-self concordant function over $\mathcal{K}$, then

$$\mathcal{R}(y) - \mathcal{R}(x) \leq \nu \log\left(\frac{1}{1 - \pi_x(y)}\right), \quad \forall x, y \in \text{int}(\mathcal{K})$$

where $\pi_{\mathcal{K},x}(y) = \inf\{t \geq 0 \colon x + \frac{1}{t}(y - x) \in \mathcal{K}\}$ is the Minkowsky functional over $\mathcal{K}$ at x.

By definition, $x_1 = \text{argmin}_{x \in \mathcal{K}}\, \mathcal{R}(x)$. Now, any point $y \in \mathcal{K}$ satisfying $\pi_{\mathcal{K},x_1}(w) \leq 1 - \frac{1}{T}$ must satisfy $\mathcal{R}(w) - \mathcal{R}(x_1) \leq \nu \log(T)$. Since $\mathcal{R} \geq 0$ on $\mathcal{K}$ by assumption, this also means that $\mathcal{R}(w) \leq \nu \log(T)$.

On the other hand, if $\pi_{\mathcal{K},x_1}(w) > 1 - \frac{1}{T}$, then the fact that $\pi_{\mathcal{K},x_1}(w) \leq 1$ implies that there exists $0 < \epsilon \leq \frac{1}{T}$ such that $\pi_{\mathcal{K},x_1}(w) \leq 1 - \frac{1}{T} + \epsilon$. Defining $z = x_1 + (w - x_1)\frac{1 - \frac{1}{T}}{1 - \frac{1}{T} + \epsilon} \in \mathcal{K}$ yields that $\|w - z\| = \|\frac{\epsilon}{1 - \frac{1}{T}} x_1 - w\| \leq \mathcal{O}\left(\frac{1}{T}\right)$ and $\pi_{\mathcal{K},x_1}(z) \leq 1 - \frac{1}{T}$. But $\mathcal{R}$ is Lipschitz on any compact subset of $\text{int}(\mathcal{K})$, so we get that $\mathcal{R}(w) \leq \nu \log(T) + \mathcal{O}\left(\frac{1}{T}\right)$. Since the last term does not grow as a function of T, we will ignore it in the regret bound.

Combining the above estimates shows that if $\eta_t \|\widehat{g}_t\|_{\nabla^2 \mathcal{R}(x_t) + \eta_t Q_t,*} \leq \frac{1}{2}$ (i.e. if $\eta_t \leq \frac{1}{2nC}$), then

$$\sum_{t=1}^T \mathbb{E}[\widehat{f}_t(y_t) - \widehat{f}_t(w)] \\ \leq \sum_{t=1}^T \mathbb{E}\left[\frac{\eta_t}{\delta_t^2}(n f_t(x_t + \delta_t B_t u))^2 + \frac{1}{\eta_T}\nu \log(T)\right].$$

□

The extrapolation of the regret bound in Theorem 1 for smoothed loss functions to a regret bound on the original loss requires taking into account some local regularity assumptions. Factoring in local second-order regularity yields Algorithm 2.

5 AdaBCO FOR SMOOTH FUNCTIONS

The major differences between Algorithm 2 with the previous algorithm are that we now account for the local smoothness parameter β_t and that we also specify precisely the

Algorithm 2 AdaBCO-Smooth

```
Input: $\eta_0 = \frac{1}{2nC}$, $\nu$-self concordant barrier $\mathcal{R}$, $C > 0$ constant.
Initialize: $x_1 = \text{argmin}_{x \in \mathcal{K}} \mathcal{R}(x)$.
for $t = 1, \dots, T$: do
    Choose a constant $\beta_t > 0$ such that $f_t(x) \leq f_t(y) + \nabla f_t(y)^\top (x - y) + \frac{\beta_t}{2} \|x - y\|_2^2$.
    Choose matrix $Q_t \succcurlyeq 0$ such that $f_t(x) \geq f_t(x_t) + g_t^\top (x - x_t) + \frac{1}{2} \|x - x_t\|_{Q_t}^2$.
    Define $\tilde{B}_{t,s} = \left(\nabla^2 R(x_s) + \eta_s 1_{\{s<t\}} Q_{1:s}\right)^{-1/2}$ and
        $$\eta_t = \left( \sum_{s=1}^{t} \sqrt{4\beta_s \frac{1}{n} \sum_{j=1}^{n} \lambda_j(\tilde{B}_{t,s}^2) n^2 C^2} \right)^{-2/3} (\nu \log(T))^{2/3}.$$
    Let $B_t = \left[\nabla^2 \mathcal{R}(x_t) + \eta_t Q_{1:t}\right]^{-1/2}$.
    Define $\delta_t = \left( \frac{n^3 C^2 \eta_t}{\beta_t \sum_{j=1}^n \lambda_j(B_t^2)} \right)^{1/4}$.
    Sample $u \sim \mathcal{S}^n$ uniformly.
    Set $y_t = x_t + \delta_t B_t u \in W_1(x_t) \subset \mathcal{K}$.
    Play $y_t$ and incur loss $f_t(y_t)$.
    Compute the estimate $\widehat{g}_t = n f_t(y_t) (\delta_t B_t)^{-1} u$.
    Update $x_{t+1} = \text{argmin}_{x \in \mathcal{K}} g_{1:t}^\top x + \frac{1}{2} \sum_{s=1}^t \|(x - x_s)\|_{Q_s}^2 + \frac{1}{\eta_t} \mathcal{R}(x)$.
end for
```

dynamic learning rate η_t and sampling radius δ_t. Thus, Algorithm 2 is also an upgrade from previous algorithms in the literature in the sense that it does not require a priori assumptions on the global convexity or smoothness of the loss functions. One can adjust these parameters online, and the algorithm's regret will adapt.

The proof of the regret bound relies first on comparing the true loss function with the smoothed one by using the regularity parameters at each step. In contrast to previous algorithms which used global regularity parameters in a coarse manner (e.g. (Saha and Tewari, 2011)), we analyze the random sampling of the ellipsoid in greater depth in order to produce data-dependent estimates that we can leverage. This requires some general results about random variables and sampling that we present in Lemmas 8 and 9. After analyzing the approximation error, we use the result of Theorem 1 to derive a tight data-dependent bound in terms of all the relevant controllable quantities. From here, the sampling ellipsoid and learning rate at each iteration are adjusted dynamically to achieve a tight bound on the regret.

The choice of these ellipsoids and learning rates is fairly subtle and cannot be done directly due to their interdependence. The optimal a posteriori learning rate depends on the sampling ellipsoid, and the radius of the sampling ellipsoid depends on the learning rate. To get around this chicken-and-egg type of phenomenon, we force the learner to first hallucinate a different set of sampling ellipsoids based on history from which the learner can determine good learning rates. From here, the learner is then able to define an efficient true sampling ellipsoid. Deriving a tight on-line approximation to the a posteriori optimal parameters also involves an abstract calculation on normalized sums, which we present in Lemma 7.

We first formerly state the technical lemmas that we will need in this section. Their proofs are provided in Appendix A.

5.1 TECHNICAL LEMMAS

Lemma 7. *Let $\alpha_t \geq 0$, $\gamma > 0$, $\beta > 1$, and $\eta_t = \beta^{\frac{1}{1+\gamma}} (\alpha_{1:t})^{\frac{-1}{1+\gamma}}$. Then*

$$\left(\sum_{t=1}^{T} \eta_t^\gamma \alpha_t \right) + \frac{\beta}{\eta_T} \leq (2 + \gamma) \beta^{\frac{\gamma}{1+\gamma}} (\alpha_{1:T})^{\frac{1}{1+\gamma}}.$$

To derive finer estimates on the approximation error, we will use the following facts about quadratic forms of random variables and the statistical properties of sampling from the unit sphere.

Lemma 8. *Let $x \sim \mathcal{D}$ be a random vector and A be a symmetric matrix. Then, the following identity holds:*

$$\mathbb{E}_{x \sim \mathcal{D}}[x^\top A x] = \textit{trace}(A \textit{cov}(x)) + \mathbb{E}[x]^\top A \mathbb{E}[x],$$

where $\textit{cov}(x) = \mathbb{E}[x x^\top] - \mathbb{E}[x]\mathbb{E}[x]^\top$ is the covariance matrix associated to x.

Lemma 9. *Let $u \sim \mathcal{S}^n$. Then $\textit{cov}(u) = \frac{1}{n} I$ and $\mathbb{E}[u] = 0$.*

We are now ready to present the regret guarantee of Algorithm 2:

Theorem 2 (AdaBCO using dynamic smoothness bounds)**.** *Let $\mathcal{K}$ be a convex set and $\mathcal{R}$ a ν-self-concordant barrier over $\mathcal{K}$. Assume that $|f| \leq C$. Then the following regret bound holds for Algorithm 2:*

$$\max_{x \in \mathcal{K}} \sum_{t=1}^{T} \mathbb{E}[f_t(y_t) - f_t(x)]$$
$$\leq \mathbb{E}\left[\frac{5}{2} (\nu \log(T))^{\frac{1}{3}} \left(\sum_{t=1}^{T} \sqrt{4\beta_t n C^2 \sum_{j=1}^{n} \lambda_j(B_t^2)} \right)^{\frac{2}{3}} \right]$$

Proof. We will show first that Algorithm 2 yields regret of at most:

$$\begin{aligned}\sum_{t=1}^{T}&\mathbb{E}[f_t(y_t)-f_t(x)]\\ &\le \sum_{t=1}^{T}\mathbb{E}\left[\delta_t^2\beta_t\frac{1}{n}\sum_{j=1}^{n}\lambda_j(B_t^2)\right]\\ &+\mathbb{E}\left[\left(\sum_{t=1}^{T}\frac{\eta_t}{\delta_t^2}(nf_t(x_t+B_tu))^2\right)+\frac{1}{\eta_T}\nu\log(T)\right]\end{aligned}$$

for any schedule of $\{\delta_t\}_{t=1}^T$ and $\{\eta_t\}_{t=1}^T$.

The expected regret can be decomposed as follows:

$$\begin{aligned}\mathbb{E}&[\mathrm{Reg}_T(w)]\\ &=\sum_{t=1}^{T}\mathbb{E}[f_t(y_t)-f_t(w)]\\ &=\sum_{t=1}^{T}\mathbb{E}[f_t(y_t)-f_t(x_t)]+\mathbb{E}[f_t(x_t)-\widehat{f}_t(x_t)]\\ &\quad+\mathbb{E}[\widehat{f}_t(w)-f_t(w)]+\mathbb{E}[\widehat{f}_t(x_t)-\widehat{f}_t(w)].\end{aligned}$$

The first three terms reflect the approximation error from running our algorithm against the true loss functions versus the smoothed out versions, and the last term can be bounded via Theorem 1. To bound the first three, we use the β_t-strongly smooth property.

For the first term, we can use the smooothness constant of the particular loss function along with the results on random sampling to derive the following bound:

$$\begin{aligned}\mathbb{E}&[f_t(y_t)-f_t(x_t)]\\ &=\mathbb{E}\Big[\mathbb{E}_{u\sim\mathcal{S}^n}[f_t(x_t+\delta_tB_tu)-f_t(x_t)|x_t]\Big]\\ &\le\mathbb{E}\Big[\mathbb{E}_{u\sim\mathcal{S}^n}[\nabla f_t(x_t)\delta_tB_tu+\tfrac{\beta_t}{2}\|\delta_tB_tu\|_2^2|x_t]\Big]\\ &=\mathbb{E}\Big[\mathbb{E}\Big[\tfrac{\beta_t}{2}\|\delta_tB_tu\|_2^2|x_t\Big]\Big]\\ &=\mathbb{E}\left[\frac{\beta_t}{2}\,\mathrm{trace}\left(\delta_t^2B_t^2\frac{1}{n}I\right)\right]\ \text{(by Lemmas 8 and 9)}\\ &=\mathbb{E}\left[\frac{\beta_t}{2}\delta_t^2\frac{1}{n}\sum_{j=1}^{n}\lambda_j(B_t^2)\right].\end{aligned}$$

The second term can be bounded using Jensen's inequality:

$$\begin{aligned}\mathbb{E}&[f_t(x_t)-\widehat{f}_t(x_t)]\\ &=\mathbb{E}\left[f_t(x_t)-\mathbb{E}_{v\sim\mathcal{B}^n}\left[f_t(x_t+\delta_tB_tv)\right]\right]\\ &\le\mathbb{E}\left[f_t(x_t)-f_t\left(\mathbb{E}_{v\sim\mathcal{B}^n}\left[x_t+\delta_tB_tv\right]\right)\right]\\ &=0.\end{aligned}$$

The third term can be analyzed in a way similar to the first term, using the smoothness constant of the particular loss function as well as the results on random sampling:

$$\begin{aligned}\mathbb{E}&[\widehat{f}_t(w)-f_t(w)]\\ &=\mathbb{E}[\mathbb{E}_{v\sim\mathcal{B}^n}[f_t(w+\delta_tB_tv)]-f_t(w)]\\ &\le\mathbb{E}\left[\mathbb{E}_{v\sim\mathcal{B}^n}\left[\nabla f_t(w)\delta_tB_tv+\frac{\beta_t}{2}\|\delta_tB_tv\|_2^2\right]\right]\\ &=\mathbb{E}\left[\frac{\beta_t}{2}\delta_t^2\mathbb{E}_{v\sim\mathcal{B}^n}\left[\|B_tv\|_2^2\right]\right]\\ &\le\mathbb{E}\left[\frac{\beta_t}{2}\delta_t^2\frac{1}{n}\sum_{j=1}^{n}\lambda_j(B_t^2)\right]\ \text{(by Lemmas 8 and 9).}\end{aligned}$$

By putting together all of the estimates above, we can arrive at the following intermediate inequality:

$$\begin{aligned}\sum_{t=1}^{T}&\mathbb{E}[f_t(y_t)-f_t(x)]\\ &\le\sum_{t=1}^{T}\mathbb{E}\left[\delta_t^2\beta_t\frac{1}{n}\sum_{j=1}^{n}\lambda_j(B_t^2)\right]\\ &\quad+\mathbb{E}\left[\left(\sum_{t=1}^{T}\frac{\eta_t}{\delta_t^2}(nf_t(x_t+B_tu))^2\right)+\frac{1}{\eta_T}\nu\log(T)\right]\\ &\le\sum_{t=1}^{T}\mathbb{E}\left[\delta_t^2\beta_t\frac{1}{n}\sum_{j=1}^{n}\lambda_j(B_t^2)\right]\\ &\quad+\mathbb{E}\left[\left(\sum_{t=1}^{T}\frac{\eta_t}{\delta_t^2}n^2C^2\right)+\frac{1}{\eta_T}\nu\log(T)\right],\end{aligned}$$

and by our choice of δ_t, it follows that

$$\begin{aligned}\sum_{t=1}^{T}&\mathbb{E}[f_t(y_t)-f_t(x)]\\ &\le\mathbb{E}\left[\sum_{t=1}^{T}2\sqrt{\beta_t\eta_tnC^2\sum_{j=1}^{n}\lambda_j(B_t^2)}+\frac{1}{\eta_T}\nu\log(T)\right].\end{aligned}$$

Finally, our choice of η_t, the fact that $\eta_t\le\eta_{t-1}$, and Lemma 7 with $\gamma=\frac{1}{2}$, $\alpha_t=2\sqrt{\beta_tnC^2\sum_{j=1}^{n}\lambda_j(B_t^2)}$, $\beta=\nu\log(T)$ yield:

$$\begin{aligned}\sum_{t=1}^{T}&\mathbb{E}[f_t(y_t)-f_t(x)]\\ &\le\mathbb{E}\left[\frac{5}{2}(\nu\log(T))^{\frac{1}{3}}\left(\sum_{t=1}^{T}\sqrt{4\beta_tnC^2\sum_{j=1}^{n}\lambda_j(B_t^2)}\right)^{\frac{2}{3}}\right].\end{aligned}$$

□

Algorithm 3 AdaBCO-Lipschitz

```
**Input:** $\eta_0 = \frac{1}{2nC}$, $\nu$-self concordant barrier $\mathcal{R}$, $C > 0$ constant.
**Initialize:** $x_1 = \operatorname{argmin}_{x \in \mathcal{K}} \mathcal{R}(x)$.
**for** $t = 1, \ldots, T$: **do**
    Choose a constant $L_t \geq 0$ such that $|f_t(x) - f_t(y)| \leq L_t|x - y|$.
    Choose matrix $Q_t \succcurlyeq 0$ such that $f_t(x) \geq f_t(x_t) + g_t^\top(x - x_t) + \frac{1}{2}\|x - x_t\|^2_{Q_t}$.
    Define $\tilde{B}_{t,s} = \left(\nabla^2 R(x_s) + (\eta_s 1_{\{s<t\}})Q_{1:s}\right)^{-1/2}$ and
        $$\eta_t = \left(\sum_{s=1}^{t} 2\left(2L_s \frac{1}{n}\sum_{j=1}^{n} \lambda_j(\tilde{B}_{t,s})n^2C^2\right)^{1/3}\right)^{-3/4} \left(\frac{\nu \log(T)}{2}\right)^{3/4}.$$
    Let $B_t = \left[\nabla^2 \mathcal{R}(x_t) + \eta_t Q_{1:t}\right]^{-1/2}$.
    Define $\delta_t = \left(2\frac{n^3C^2\eta_t}{L_t \sum_{j=1}^n \lambda_j(B_t)}\right)^{1/3}$.
    Sample $u \sim \mathcal{S}^n$ uniformly.
    Set $y_t = x_t + \delta_t B_t u \in W_1(x_t) \subset \mathcal{K}$.
    Play $y_t$ and incur loss $f_t(y_t)$.
    Compute the estimate $\widehat{g}_t = nf_t(y_t)(\delta_t B_t)^{-1}u$.
    Update $x_{t+1} = \operatorname{argmin}_{x \in \mathcal{K}} g_{1:t}^\top x + \frac{1}{2}\sum_{s=1}^t \|(x - x_s)\|^2_{Q_s} + \frac{1}{\eta_t}\mathcal{R}(x)$.
**end for**
```

6 AdaBCO FOR LIPSCHITZ FUNCTIONS

Using first-order regularity instead of second motivates the design of Algorithm 3. Like Algorithm 2, the major difference here is that we factor in the local Lipschitz constant L_t and that we specify precisely η_t and δ_t. In the process, we also need to hallucinate a separate set of ellipsoids to circumvent the chicken-and-egg phenomenon.

Using similar techniques as in Theorem 2, one can derive the following regret bound:

Theorem 3 (AdaBCO using dynamic Lipschitz bounds)**.** *Let $\mathcal{K}$ be a convex set and $\mathcal{R}$ a ν-self-concordant barrier over $\mathcal{K}$. Assume that $|f| \leq C$. Then Algorithm 2 provides the regret bound:*

$$\max_{x \in \mathcal{K}} \sum_{t=1}^{T} \mathbb{E}[f_t(y_t) - f_t(x)]$$

$$\leq \mathbb{E}\left[5(\nu \log(T))^{\frac{1}{4}} \left(\sum_{t=1}^{T}\left(L_t nC^2 \sum_{j=1}^{n} \lambda_j(\tilde{B}_t)\right)^{\frac{1}{3}}\right)^{\frac{3}{4}}\right]$$

The proof of this result is similar to that of Theorem 2, and is provided in Appendix B.

7 APPLICATIONS AND COMPARISON WITH PREVIOUS RESULTS

The data-dependent nature of Algorithms 2 and 3 provide two important implications.

The first is that they allow us to easily produce regret bounds in a variety of new situations, where the learner experiences loss functions with various levels of local smoothness and convexity. In particular, we can identify new scenarios where the optimal $\tilde{\mathcal{O}}(\sqrt{T})$ regret is achievable by navigating the relationship between smoothnesss and convexity.

The second is that these algorithms also automatically adapt to the smoothness and convexity of these scenarios. These new cases do not require any a priori insight or tuning. The algorithms presented in this paper adaptively determine optimal sampling ellipsoids and learning rates, which lead to strong guarantees.

In particular, they allow the learner to recover existing regret bounds without modifying the algorithms. Properties such as strong convexity or smoothness are processed adaptively and online, so that if, e.g., a sequence of loss functions is found to be approximately strongly convex (which will become clear in the following results), then the strongly convex guarantee will apply. If the sequence of loss functions is better than strongly convex, then the algorithm will give an even better guarantee. Thus, these algorithms are prime examples of algorithms that "learn faster from easy data".

We present first the results for Algorithm 2.

Corollary 1 (Power law asymptotics for the dynamically smooth and strongly convex scenario)**.** *Assume that there exists $\alpha \in \mathbb{R}$ such that*

$$\beta_t nC^2 \sum_{j=1}^{n} \lambda_j((\nabla^2 \mathcal{R}(x_t) + \eta_t Q_{1:t})^{-1}) = \mathcal{O}(t^\alpha).$$

Then the inequality $\sum_{t=1}^{T} \mathbb{E}[f_t(y_t) - f_t(x)] \leq \tilde{\mathcal{O}}(T^{\frac{2+\alpha}{3}})$ holds.

In particular, $\tilde{\mathcal{O}}(\sqrt{T})$ regret is attainable for $\alpha \leq \frac{-1}{2}$.

Moreover, $\tilde{\mathcal{O}}(T^{1/2})$ regret is adaptively attained for smooth and strongly convex functions, while $\tilde{\mathcal{O}}(T^{2/3})$ regret is adaptively attained for smooth functions.

For purely strongly smooth and strongly convex functions, $\beta_t \equiv \beta > 0$, and $Q_t \equiv Q \succ 0$, such that $Q_{1:t} = tQ$. Algorithm 2 then implies that $\eta_t = \tilde{\mathcal{O}}(t^{-1/2})$ (provable via induction), so that the corollary above applies with $\alpha \leq \frac{-1}{2}$. Thus, we adaptively attain the bound of $\tilde{\mathcal{O}}(\sqrt{T})$ in

(Hazan and Levy, 2014) without a priori knowledge of the function's regularity or any extra tuning.

For purely strongly smooth functions, $\beta_t \equiv \beta$ and in the worst case $Q_t \equiv 0$. This implies that the expression above reduces to $\beta n C^2 \sum_{j=1}^n \lambda_j(\nabla^2 \mathcal{R}(x_t)^{-1})$, so that the regret in t depends entirely on the average eigenvalue of the inverse Hessian, $\sum_{j=1}^n \lambda_j(\mathcal{R}(x_t)^{-1})$. In the worst case, this expression is $\mathcal{O}(1)$, which gives us the bound of $\tilde{\mathcal{O}}(T^{2/3})$ in (Saha and Tewari, 2011).

From another perspective, the corollary can be interpreted as saying that as long as $\beta_t \asymp \frac{1}{t \sum_{j=1}^n \lambda_j(Q_{1:t}^{-1})} \asymp t^\gamma$ for any $\gamma \in \mathbb{R}$, then $\eta_t = \tilde{\mathcal{O}}(\frac{1}{\sqrt{t}})$, and a regret of at most $\tilde{\mathcal{O}}(\sqrt{T})$ regret is guaranteed. In other words, we can extend the result of (Hazan and Levy, 2014) to not just the case where the smoothness and strong convexity are fixed and local, but in fact to any setting where the smoothness and average strong convexity parameters are locally changing at the same rate.

Moreover, we would like to stress that the above reductions are worst-case guarantees. The data-dependent nature of the regret bound above implies that it can do much better on easier data. We also do not need to know about these optimistic settings in advance of running the algorithms, as they will be adaptively and automatically obtained.

In particular, our algorithms factor in and leverage the convexity of the self-concordant barrier, so that the algorithm's bounds are much stronger when the algorithm plays points at which the Hessian of the barrier has large average eigenvalues. For common self-concordant barriers such as the log-barrier function, this corresponds to being closer to the boundary of the action set. This insight is actually somewhat surprising, because being further from the boundary generally implies that the learner will use a wider sampling ellipsoid and be able to explore more. This suggests that the self-concordant barrier regularization introduced by Abernethy et al. (2008) might not ellicit the best tradeoff between exploration and exploitation for general convex functions as it does for linear functions. Previous algorithms in bandit convex optimization did not reveal this phenomenon because they were not adaptive and did not provide data-dependent guarantees.

We now present the accompanying results for Algorithm 3.

Corollary 2 (Power law asymptotics for the dynamically Lipschitz and strongly convex scenario)**.** *Assume that there exists $\alpha \in \mathbb{R}$ such that*

$$L_t n C^2 \sum_{j=1}^n \lambda_j((\nabla^2 \mathcal{R}(x_t) + \eta_t Q_{1:t})^{-1/2}) = \mathcal{O}(t^\alpha).$$

Then the inequality $\sum_{t=1}^T \mathbb{E}[f_t(y_t) - f_t(x)] \leq \tilde{\mathcal{O}}(T^{\frac{3+\alpha}{4}})$ holds.

In particular, $\tilde{\mathcal{O}}(\sqrt{T})$ regret is attainable for $\alpha \leq -1$.

Moreover, $\tilde{\mathcal{O}}(T^{2/3})$ regret is adaptively attained for strongly convex functions, and $\tilde{\mathcal{O}}(T^{3/4})$ regret is adaptively attained for Lipschitz functions.

For purely Lipschitz and strongly convex functions, $L_t \equiv L > 0$, and $Q_t \equiv Q \succ 0$, such that $Q_{1:t} = tQ$. Algorithm 2 then implies that $\eta_t = \tilde{\mathcal{O}}(t^{-1/3})$, so that the corollary above applies with $\alpha = \frac{-1}{3}$. Thus, we adaptively attain the bound of $\tilde{\mathcal{O}}(T^{2/3})$ of (Agarwal et al., 2010) without a priori knowledge of the regularity or any extra tuning.

For purely Lipschitz functions, $L_t \equiv L$ and in the worst case $Q_t \equiv 0$. This implies that the expression above reduces to $LnC^2 \sum_{j=1}^n \lambda_j(\nabla^2 \mathcal{R}(x_t)^{-1/2})$, so that the regret now depends entirely on the average eigenvalue of the square root of the inverse Hessian, $\sum_{j=1}^n \lambda_j(\mathcal{R}(x_t)^{-1/2})$. In the worst case, this expression is $\mathcal{O}(1)$, which gives us the bound of $\tilde{\mathcal{O}}(T^{3/4})$ in (Flaxman et al., 2005).

Moreover, as long as $L_t \asymp \frac{1}{t \sum_{j=1}^n \lambda_j(Q_{1:t}^{-1/2})} \asymp t^\gamma$ for $\gamma \in \mathbb{R}$, it follows that $\eta_t = \tilde{\mathcal{O}}(t^{-1/3})$ and will attain a regret of at least $\tilde{\mathcal{O}}(T^{2/3})$.

However, as we mentioned before, these bounds can be more favorable in optimistic settings with easier data, and our algorithm will automatically adapt to these scenarios.

8 CONCLUSION

We presented two efficient and adaptive algorithms for bandit convex optimization. Unlike previous algorithms, ours do not require a priori assumptions of global strong convexity or smoothness. Instead, they can process these parameters locally and online, which is much more suitable for the setting of online convex optimization.

They also provide data-dependent guarantees, so that on "easier data", the algorithms learn faster and the bounds become tighter. In particular, we present and characterize many new data-dependent scenarios under which one can obtain the desired $\tilde{\mathcal{O}}(\sqrt{T})$ regret, including in the purely Lipschitz and purely smooth settings.

Moreover, our algorithms characterize easy data to be situations where the local smoothness and convexity of our loss functions grow at the same rate as well as when the loss function guides the learner to play points closer to the boundary. This bias in optimal exploration suggests that the self-concordant barrier may be a sub-optimal regularizer in the case of general Lipschitz convex functions.

9 ACKNOWLEDGEMENTS

This work was partly funded by the NSF awards IIS-1117591 and CCF-1535987 and was also supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 1342536.

References

Abernethy, J., E. Hazan, and A. Rakhlin (2008). Competing in the dark: An efficient algorithm for bandit linear optimization. In *COLT*, pp. 263–274.

Agarwal, A., O. Dekel, and L. Xiao (2010). Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pp. 28–40.

Bartlett, P. L., E. Hazan, and A. Rakhlin (2007). Adaptive online gradient descent. In *NIPS*, pp. 65–72.

Bubeck, S. and R. Eldan (2015). Multi-scale exploration of convex functions and bandit convex optimization. *CoRR abs/1507.06580*.

Bubeck, S. and A. Slivkins (2012). The best of both worlds: Stochastic and adversarial bandits. In *COLT*, Volume 23, pp. 42.1–42.23.

de Rooij, S., T. van Erven, P. D. Grünwald, and W. M. Koolen (2014). Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research 15*, 1281–1316.

Duchi, J. C., E. Hazan, and Y. Singer (2010). Adaptive subgradient methods for online learning and stochastic optimization. In *COLT*, pp. 257–269.

Even-Dar, E., M. Kearns, Y. Mansour, and J. Wortman (2007). Regret to the best vs. regret to the average. In *COLT*, Volume 4539, pp. 233–247.

Flaxman, A. D., A. T. Kalai, and H. B. McMahan (2005). Online convex optimization in the bandit setting: Gradient descent without a gradient. In *SODA*, pp. 385–394.

Hazan, E. and K. Y. Levy (2014). Bandit convex optimization: Towards tight bounds. In *NIPS*, pp. 784–792.

Kalai, A. T. and S. Vempala (2005). Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 291–307.

McMahan, H. B. and M. J. Streeter (2010). Adaptive bound optimization for online convex optimization. In *COLT*, pp. 244–256.

Nemirovski, A. S. and M. J. Todd (2008). Interior-point methods for optimization. *Acta Numerica*, 191–234.

Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization: A Basic Course*. New York, NY, USA: Springer.

Saha, A. and A. Tewari (2011). Improved regret guarantees for online smooth convex optimization with bandit feedback. In *AISTATS*, pp. 636–642.

Sani, A., G. Neu, and A. Lazaric (2014). Exploiting easy data in online optimization. In *NIPS*, pp. 810–818.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *ICML*, pp. 928–936.