
Optimal Algorithms for Online Convex Optimization with Multi-Point Bandit Feedback

Alekh Agarwal*
University of California
Berkeley, CA
alekh@cs.berkeley.edu

Ofer Dekel
Microsoft Research
Redmond, WA
oferd@microsoft.com

Lin Xiao
Microsoft Research
Redmond, WA
lin.xiao@microsoft.com

Abstract

Bandit convex optimization is a special case of online convex optimization with partial information. In this setting, a player attempts to minimize a sequence of adversarially generated convex loss functions, while only observing the value of each function at a single point. In some cases, the minimax regret of these problems is known to be strictly worse than the minimax regret in the corresponding full information setting. We introduce the multi-point bandit setting, in which the player can query each loss function at multiple points. When the player is allowed to query each function at two points, we prove regret bounds that closely resemble bounds for the full information case. This suggests that knowing the value of each loss function at two points is almost as useful as knowing the value of each function everywhere. When the player is allowed to query each function at $d+1$ points (d being the dimension of the space), we prove regret bounds that are exactly equivalent to full information bounds for smooth functions.

1 Introduction

Online convex optimization is best understood as a repeated game between a player and an adversary. On round t of the game, the player begins by choosing a point x_t from a fixed and known convex set $\mathcal{K} \subseteq \mathbb{R}^d$. We adopt the game-theoretic terminology and say that x_t is the player's *move* on round t . The adversary observes x_t and chooses a convex loss function $\ell_t : \mathcal{K} \rightarrow \mathbb{R}$. Then, the loss function ℓ_t is revealed to the player, who incurs the loss $\ell_t(x_t)$. The goal of the player is to minimize his *regret*, defined as

$$\sum_{t=1}^T \ell_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^T \ell_t(x) .$$

Regret measures the difference between the cumulative loss of the player's strategy and the loss of the best constant point chosen in hindsight. Although the adversary has the advantage of playing second on each round, there exist strategies for which we can prove non-trivial upper-bounds on regret. Zinkevich (2003) presents a strategy based on gradient descent that guarantees a regret of $O(\sqrt{T})$ when the set \mathcal{K} is compact and the loss functions are Lipschitz continuous. If the loss functions are strongly convex (defined below), Hazan et al. (2007) analyze a similar gradient descent strategy with a regret bound of $O(\log(T))$. They also prove a similar regret bound for the larger class of exp-concave loss functions using an online Newton-step algorithm. The $O(\sqrt{T})$ and $O(\log(T))$ regret bounds, for convex and strongly convex loss functions respectively, are known to be minimax optimal (see eg. Abernethy et al. (2009); Cesa-Bianchi and Lugosi (2006) and references therein).

The online convex optimization problem becomes more challenging when the player only receives partial feedback on the choices of the adversary. One particularly interesting type of feedback is *bandit* feedback, where the adversary only reveals the value of the loss function at x_t , instead of revealing the entire function ℓ_t . Specifically, the player does not know the gradient of ℓ_t at x_t and a simple gradient descent algorithm is inapplicable.

In the bandit setting, the player does not stand a chance against the *completely adaptive* adversary described above, who chooses ℓ_t after observing the player's move. Specifically, a regret bound for

*This research was conducted while AA was a research intern in Microsoft Research at Redmond. AA gratefully acknowledges the support of the NSF through grants 0707060 and 0830410 for travel expenses

Algorithm 1 Template for k -point bandit algorithm

```

for  $t = 1, \dots, T$  do
    Adversary (secretly) chooses convex loss function  $\ell_t$ .
    Player chooses and reveals  $y_{t,1}, \dots, y_{t,k} \in \mathcal{K}$ .
    Adversary reveals  $\ell_t(y_{t,1}), \dots, \ell_t(y_{t,k})$ .
    Player incurs the loss  $(1/k) \sum_{i=1}^k \ell_t(y_{t,i})$ .
end for

```

any strategy in this setting is necessarily $\Omega(T)$. To see this, let \mathcal{K} be the interval $[0, 1]$ and define the following two completely adaptive adversaries: the first chooses $\ell_t(z) = z - x_t$ while the second chooses $\ell'_t(z) = x_t - z$. In both cases, the player observes the loss value $\ell_t(x_t) = \ell'_t(x_t) = 0$ on every round, and has no way of knowing which of the two adversaries he is facing. Therefore, the player will play the same sequence of moves against both adversaries. If at least half of the player's moves lie in the interval $[\frac{1}{2}, 0]$ then the player's regret against the first adversary is at least $T/4$. Otherwise, the player's regret against the second adversary is at least $T/4$. Overall, any strategy will suffer at least a linear regret against one of these adversaries.

The example above indicates that we need to level the playing field by slightly limiting the power of the adversary. Therefore, an *adaptive* adversary in the bandit setting is allowed to choose ℓ_t based only on the player's past moves x_1, \dots, x_{t-1} , and not on his current move x_t . Put another way, the adversary chooses ℓ_t at the beginning of round t , before the player makes his move. Then, the player chooses x_t without knowing ℓ_t and reveals his move to the adversary. Finally, the adversary notifies the player of his loss $\ell_t(x_t)$. We note in passing that an even weaker adversary is the *oblivious* adversary, who chooses ℓ_t without knowing any of the player's moves. However, in practice, an analysis for the oblivious case is often an intermediate step towards an analysis for the adaptive case.

In the special case where the loss functions are linear, Abernethy and Rakhlin (2009) present a randomized algorithm with an $\tilde{O}(\sqrt{T})$ regret-bound that holds with high-probability against adaptive adversaries. This algorithm relies on a non-trivial application of self-concordant barrier regularization, and differs significantly from the typical algorithms that are used in the full information case. This $\tilde{O}(\sqrt{T})$ regret bound is optimal due to an information-theoretic argument from Auer et al. (2003). For general convex loss functions, Flaxman et al. (2005) present a simple modification of the full information gradient descent algorithm, where the gradient is replaced by a randomized estimate. The expected regret of this algorithm is shown to be $O(T^{3/4})$ against oblivious adversaries. Flaxman et al. also sketch a high probability extension of their bound to adaptive adversaries. For smooth and strongly convex loss functions, the regret bound of Flaxman et al. (2005) can be strengthened to $O(T^{2/3})$, and furthermore, if \mathcal{K} is a linear vector space (namely, the optimization is unconstrained) then the bound can be improved to $O(\sqrt{T})$ ¹. Finding an optimal algorithm for the general bandit convex optimization setting remains an open problem. However, a lower bound due to Dani et al. (2008) implies that the regret of this optimal algorithm will be $\Omega(\sqrt{T})$, even when the functions are strongly convex. This is to be contrasted with the full-information case where $O(\log(T))$ regret is achieved by online gradient descent when the loss functions are strongly convex.

Overall, we observe significant gaps between the optimal regret bounds for the full information and bandit settings, as well as gaps in the complexity of the algorithms that attain these bounds. This leads to the natural question of whether we can study a continuum of problems ranging between these two extremes. Through such an inquiry, we can hope to find the point along this continuum where the regret bounds for the full information case deteriorate to become the inferior bounds in the bandit setting. We can also hope to find the point on this continuum where simple gradient descent approaches stop giving optimal bounds, and the need for specialized algorithms arises. Answering these questions is the focus of our paper.

To this end, we extend the bandit setting and introduce the multi-point bandit setting, where the player queries each loss function at k randomized points, rather than at a single point. The template for a k -point bandit algorithm is given in Algorithm 1. In this setting, we define the expected regret,

$$\mathbb{E} \frac{1}{k} \sum_{t=1}^T \sum_{i=1}^k \ell_t(y_{t,i}) - \min_{x \in \mathcal{K}} \mathbb{E} \sum_{t=1}^T \ell_t(x),$$

where expectation is taken over the randomness of the player. When $k = 1$, the multi-point bandit

¹Both of these results are novel but their proof is omitted due to space constraints.

setting reduces to the bandit case discussed above.

For $k = 2$, we show that a variant of the randomized gradient descent algorithm proposed by Flaxman et al. (2005) performs optimally. We prove a *high-probability* regret bound of $\tilde{O}(\sqrt{T})$ for convex Lipschitz-continuous loss functions chosen by an adaptive adversary. We also prove an *expected* regret bound of $O(\log(T))$ for strongly convex loss functions chosen by an adaptive adversary. Thus, by allowing the player to query only two points on each round, we can already obtain bounds that closely resemble the optimal bounds for the full-information setting. In the strongly convex case, we see a sharp phase transition of the minimax regret from $\Omega(\sqrt{T})$ to $O(\log(T))$ when moving from $k = 1$ to $k = 2$. For general convex loss functions, optimal bounds for the $k = 1$ setting achievable in a computationally efficient manner are not yet known, but it seems evident that optimal algorithms and bounds will rely on techniques that are significantly different from those used in the full information-case. In contrast, for $k = 2$ we can rely on algorithms and analysis techniques that are quite similar to those used in the full information case. Overall, we conclude that having the ability to evaluate the loss function at two points on each round is *almost* as powerful as observing the entire loss function.

When $k = d + 1$, where d is the dimension of the space, we show that a *deterministic* algorithm can obtain a regret of $O(\sqrt{T})$ against convex Lipschitz and smooth loss functions, and a regret of $O(\log(T))$ against strongly convex and smooth loss functions. Moreover, both of these bounds hold for the case of *completely adaptive* adversaries, precisely as in the full information case. This algorithm uses gradient descent based on a deterministic approximation of the gradient. Applying a similar approximation to the online Newton-step algorithm (Hazan et al., 2007) also gives an $O(\log(T))$ regret for exp-concave and smooth loss functions. We conclude that having the ability to evaluate the loss function at $d + 1$ points on each round is as powerful as observing the entire function when the functions are smooth.

The algorithms we propose are efficient, with the computational complexities being no greater than their full information counterparts.

Similar schemes for constructing gradient estimators using multiple function evaluations have also been analyzed in the framework of stochastic optimization (Polyak & Tsyplkin, 1973; Spall, 2003) and derivative-free optimization (Conn et al., 2009). However, there seem to be relatively few relevant results concerning the rate of convergence.

Notation and Assumptions: Before proceeding, we define our notation and state assumptions that will appear in various parts of the analysis. Let $\|\cdot\|$ denote the Euclidean norm in \mathbb{R}^d , and let $\mathcal{B} = \{x \in \mathbb{R}^d : \|x\| \leq 1\}$ be the unit ball centered at the origin. We assume that the convex set \mathcal{K} is compact and has a nonempty interior. If not, we can always map \mathcal{K} to a lower dimensional space. More specifically, we assume that there exist $r, D > 0$ such that

$$r\mathcal{B} \subseteq \mathcal{K} \subseteq D\mathcal{B}. \quad (1)$$

We assume w.l.o.g. that the set contains 0, as we can always translate \mathcal{K} . We assume that each loss function ℓ_t is Lipschitz continuous, i.e., there exists a constant G such that

$$|\ell_t(x) - \ell_t(y)| \leq G\|x - y\|, \quad \forall x, y \in \mathcal{K}, \forall t. \quad (2)$$

For $\sigma \geq 0$, the function ℓ is called σ -strongly convex on the set \mathcal{K} if

$$\ell(x) \geq \ell(y) + \nabla \ell(y)^\top (x - y) + \frac{\sigma}{2}\|x - y\|^2, \quad \forall x, y \in \mathcal{K}. \quad (3)$$

The case $\sigma = 0$ merely implies convexity of ℓ . We call a function L -smooth on \mathcal{K} if it is differentiable on an open set containing \mathcal{K} and its gradient is Lipschitz continuous with a constant L , i.e.,

$$\|\nabla \ell(x) - \nabla \ell(y)\| \leq L\|x - y\|, \quad \forall x, y \in \mathcal{K}. \quad (4)$$

We use the notation \mathbb{E}_t to denote the conditional expectation conditioned on all randomness in the first $t - 1$ rounds.

2 Expected gradient descent using two queries per round

In this section, we present optimal algorithms (up to logarithmic factors) that query the loss function at two points on each round. We assume that the adversary's choice of ℓ_t can depend on all of the information available up to round $t - 1$, but does not depend on the player's random moves on round t . In the full information case, the player can follow the online gradient descent strategy (Zinkevich, 2003) and make the move

$$x_{t+1} = \Pi_{\mathcal{K}}(x_t - \eta_t \nabla \ell_t(x_t)), \quad (5)$$

Algorithm 2 Expected Gradient Descent with two queries per round

Input: Learning rates η_t , exploration parameter δ and shrinkage coefficient ξ .

Set $x_1 = 0$

for $t = 1, \dots, T$ **do**

 Pick a unit vector u_t uniformly at random.

 Observe $\ell_t(x_t + \delta u_t)$ and $\ell_t(x_t - \delta u_t)$.

 Set $\tilde{g}_t = \frac{d}{2\delta}(\ell_t(x_t + \delta u_t) - \ell_t(x_t - \delta u_t))u_t$.

 Update $x_{t+1} = \Pi_{(1-\xi)\mathcal{K}}(x_t - \eta_t \tilde{g}_t)$.

end for

where $\Pi_{\mathcal{K}}(x)$ denotes the Euclidean projection of x onto the set \mathcal{K} , and η_t is the step size or *learning rate*. In our partial information setting, we follow Flaxman et al. (2005) and replace $\nabla \ell_t(x_t)$ with an estimate \tilde{g}_t , obtained by evaluating the loss function at two random points around x_t . The player makes the move

$$x_{t+1} = \Pi_{(1-\xi)\mathcal{K}}(x_t - \eta_t \tilde{g}_t), \quad (6)$$

where $\xi \in (0, 1)$ and $(1 - \xi)\mathcal{K}$ is shorthand for $\{(1 - \xi)x : x \in \mathcal{K}\}$. The projection is made onto the shrunk set $(1 - \xi)\mathcal{K}$ to ensure that the random query points around x_{t+1} belong to \mathcal{K} . In particular, for any $x \in (1 - \xi)\mathcal{K}$ and any unit vector u it holds that $(x + \delta u) \in \mathcal{K}$ for any δ in $[0, \xi r]$ (Flaxman et al., 2005, Observation 2). Algorithm 2 gives a complete description of the Expected Gradient Descent method with two queries per round. The intuition behind this method is that any random realization of \tilde{g}_t is a good proxy for the directional derivative $\nabla \ell_t(x_t)^\top u_t$. Hence taking expectation over a random choice of u_t gives us a good estimator for $\nabla \ell_t(x_t)$.

The difference between Algorithm 2 and the algorithm proposed by Flaxman et al. (2005) is that the latter relies on a single function evaluation on each round. The single value $\ell_t(x_t + \delta u_t)$ is used to construct the gradient estimator

$$g_t = \frac{d}{\delta} \ell_t(x_t + \delta u_t) u_t. \quad (7)$$

Let v be a uniform random vector in the unit ball \mathcal{B} , and define the smoothed loss function

$$\hat{\ell}_t(x) = \mathbb{E}_v \ell_t(x + \delta v).$$

Note that $\hat{\ell}_t$ is Lipschitz continuous with the same constant G , and $\hat{\ell}_t$ is always differentiable even if ℓ_t is not. Through a clever use of Stokes' theorem, Flaxman et al. (2005) showed that g_t in (7) is a conditionally unbiased estimator of $\nabla \hat{\ell}_t(x_t)$, i.e.,

$$\mathbb{E}_t[g_t] = \nabla \hat{\ell}_t(x_t). \quad (8)$$

Their analysis uses the facts that the update of Equation (6) performs expected gradient descent on the functions $\hat{\ell}_t$ and that $\hat{\ell}_t(x_t)$ and $\ell_t(x_t)$ are close when δ is small. However, the resulting bounds on expected regret are much worse than bounds for gradient descent in the full information case. This is partly due to the large norm of the gradient estimator in (7) when δ is small.

The key insight in this section is that, for the entire class of Lipschitz continuous functions, one can use two function evaluations to construct gradient estimators that have a bounded norm, which leads to much improved regret bounds. First, we note that the gradient estimator in Algorithm 2,

$$\tilde{g}_t = \frac{d}{2\delta}(\ell_t(x_t + \delta u_t) - \ell_t(x_t - \delta u_t))u_t, \quad (9)$$

also satisfies the unbiasedness condition (8) as the distribution of u_t is symmetric. To show that it has bounded norm, we have

$$\begin{aligned} \|\tilde{g}_t\| &= \frac{d}{2\delta} \|(\ell_t(x_t + \delta u_t) - \ell_t(x_t - \delta u_t))u_t\| \\ &= \frac{d}{2\delta} |\ell_t(x_t + \delta u_t) - \ell_t(x_t - \delta u_t)| \\ &\leq \frac{dG}{2\delta} \|2\delta u_t\| = Gd, \end{aligned}$$

where in the inequality above we use the Lipschitz property (2).

In order to analyze the regret of Algorithm 2, we also define the functions

$$h_t(x) = \hat{\ell}_t(x) + (\tilde{g}_t - \nabla \hat{\ell}_t(x_t))^\top x, \quad \forall t.$$

It is easily seen that $\nabla h_t(x_t) = \tilde{g}_t$, and therefore $\|\nabla h_t(x_t)\| \leq Gd$ for all t . Also $\mathbb{E}_t h_t(x) = \hat{\ell}_t(x)$ for any x that is independent of u_t . Moreover, for any fixed $x, y \in \mathcal{K}$,

$$\begin{aligned} |h_t(x) - h_t(y)| &\leq |\hat{\ell}_t(x) - \hat{\ell}_t(y)| + |(\tilde{g}_t - \nabla \hat{\ell}_t(x_t))^\top (x - y)| \\ &\leq G\|x - y\| + (\|\tilde{g}_t\| + \|\nabla \hat{\ell}_t(x_t)\|)\|x - y\| \\ &\leq G(d + 2)\|x - y\| \leq 3Gd\|x - y\| \quad \text{since } \hat{\ell}_t \text{ is } G \text{ Lipschitz.} \end{aligned}$$

Therefore the Lipschitz constant of h_t is bounded by $3Gd$. In addition, the convexity of h_t follows from that of ℓ_t . With this definition, Algorithm 2 amounts to performing *deterministic* gradient descent on the convex functions h_t , with projections on the shrunk convex set $(1 - \xi)\mathcal{K}$. In the analysis below, we first state a regret bound for deterministic gradient descent on the functions h_t due to Bartlett et al. (2008b), then use it to prove the desired regret bound on the functions ℓ_t .

We assume that ℓ_t is σ_t -strongly convex (3) for some $\sigma_t \geq 0$. It follows that the functions $\hat{\ell}_t$ and h_t are also σ_t -strongly convex. Assume, without loss of generality, that $\sigma_1 > 0$. As we see later, this can always be done by adding a strongly convex regularization term, and it does not affect the overall regret bound. We use the shorthand $\sigma_{s:t}$ to denote $\sum_{\tau=s}^t \sigma_\tau$. Then $\sigma_{1:t}$ is always positive if $\sigma_1 > 0$.

We begin our analysis by stating a general upper bound on the regret of Online Gradient Descent, due to Bartlett et al. (2008b).

Lemma 1 (Bartlett et al. (2008b)) *If the Online Gradient Descent algorithm (5) is performed over a convex set \mathcal{S} on σ_t -strongly convex functions h_t with $\eta_t = 1/\sigma_{1:t}$, then for any $x \in \mathcal{S}$,*

$$\sum_{t=1}^T h_t(x_t) - \sum_{t=1}^T h_t(x) \leq \frac{d^2 G^2}{2} \sum_{t=1}^T \frac{1}{\sigma_{1:t}}.$$

We will also require the following lemma, which relates the desired regret on the losses ℓ_t with the moves $y_{t,1} = x_t + \delta u_t$ and $y_{t,2} = x_t - \delta u_t$, to the regret on the losses $\hat{\ell}_t$ with the move x_t .

Lemma 2 *For any point $x \in \mathcal{K}$,*

$$\sum_{t=1}^T \frac{1}{2}(\ell_t(y_{t,1}) + \ell_t(y_{t,2})) - \sum_{t=1}^T \ell_t(x) \leq \sum_{t=1}^T \hat{\ell}_t(x_t) - \sum_{t=1}^T \hat{\ell}_t((1 - \xi)x) + 3TG\delta + TGD\xi.$$

Proof: By the Lipschitz property (2),

$$\ell_t(y_{t,1}) = \ell_t(x_t + \delta u_t) \leq \ell_t(x_t) + G\delta\|u_t\| \quad \text{and} \quad \ell_t(y_{t,2}) = \ell_t(x_t - \delta u_t) \leq \ell_t(x_t) + G\delta\|u_t\|.$$

Since $\|u_t\| = 1$ for all t , we get

$$\frac{1}{2}(\ell_t(y_{t,1}) + \ell_t(y_{t,2})) \leq \ell_t(x_t) + G\delta. \quad (10)$$

The Lipschitz property and the assumption $\|x\| \leq D$ also imply that, for all $x \in \mathcal{K}$,

$$\ell_t((1 - \xi)x) \leq \ell_t(x) + GD\xi. \quad (11)$$

We can also relate $\ell_t(x_t)$ and $\hat{\ell}_t(x_t)$ using the Lipschitz property:

$$|\ell_t(x_t) - \hat{\ell}_t(x_t)| = |\ell_t(x_t) - \mathbb{E}_t \ell_t(x_t + \delta v)| \leq \mathbb{E}_t |\ell_t(x_t) - \ell_t(x_t + \delta v)| \leq \mathbb{E}_t G\delta\|v\|,$$

where we use the fact that the random vector v in the definition of $\hat{\ell}_t(x_t)$ is independent of x_t and ℓ_t . Using the fact $\|v\| \leq 1$ gives

$$\ell_t(x_t) \leq \hat{\ell}_t(x_t) + G\delta \quad \text{and} \quad \hat{\ell}_t((1 - \xi)x) \leq \ell_t((1 - \xi)x) + G\delta \quad \forall x \in \mathcal{K}. \quad (12)$$

Combining the inequalities (10), (11) and (12), we have

$$\frac{1}{2}(\ell_t(y_{t,1}) + \ell_t(y_{t,2})) + \hat{\ell}_t((1 - \xi)x) \leq \hat{\ell}_t(x_t) + \ell_t(x) + 3G\delta + GD\xi.$$

Finally, summing both sides of the above inequality for all t from 1 to T and rearranging terms gives the desired statement. \blacksquare

With the above lemma, it suffices to bound regret on the losses $\hat{\ell}_t$ for the sequence x_t against the set $\mathcal{K}(1 - \xi)$. We can now state a bound on the expected regret of Algorithm 2.

Theorem 3 Let assumptions (1) and (2) hold, and ℓ_t be σ_t -strongly convex where $\sigma_1 > 0$. If Algorithm 2 is run with $\eta_t = \frac{1}{\sigma_{1:t}}$, $\delta = \frac{\log T}{T}$ and $\xi = \frac{\delta}{r}$, then for any $x \in \mathcal{K}$,

$$\mathbb{E} \sum_{t=1}^T \frac{1}{2} (\ell_t(y_{t,1}) + \ell_t(y_{t,2})) - \mathbb{E} \sum_{t=1}^T \ell_t(x) \leq \frac{d^2 G^2}{2} \sum_{t=1}^T \frac{1}{\sigma_{1:t}} + G \log(T) \left(3 + \frac{D}{r} \right).$$

Proof: As pointed out earlier, since $\nabla h_t(x_t) = \tilde{g}_t$, Algorithm 2 is actually performing gradient descent (as if with full information) on the functions h_t restricted to the convex set $(1 - \xi)\mathcal{K}$. Using Lemma 1, we have that

$$\sum_{t=1}^T h_t(x_t) - \sum_{t=1}^T h_t((1 - \xi)x) \leq \frac{d^2 G^2}{2} \sum_{t=1}^T \frac{1}{\sigma_{1:t}}.$$

Taking expectations, and using Equation (8), we conclude that

$$\mathbb{E} \sum_{t=1}^T \hat{\ell}_t(x_t) - \mathbb{E} \sum_{t=1}^T \hat{\ell}_t((1 - \xi)x) \leq \frac{d^2 G^2}{2} \sum_{t=1}^T \frac{1}{\sigma_{1:t}}.$$

Using Lemma 2, we get that

$$\mathbb{E} \sum_{t=1}^T \frac{1}{2} (\ell_t(y_{t,1}) + \ell_t(y_{t,2})) - \mathbb{E} \sum_{t=1}^T \ell_t(x) \leq \frac{d^2 G^2}{2} \sum_{t=1}^T \frac{1}{\sigma_{1:t}} + 3TG\delta + TGD\xi.$$

Plugging in the stated values of δ and ξ completes the proof. \blacksquare

We can take δ arbitrarily close to 0, but that doesn't improve the bound beyond constant factors. With Theorem 3 handy, we are all set to prove the following corollaries.

Corollary 4 If assumptions (1) and (2) hold and Algorithm 2 is run with $\eta_t = \frac{1}{\sqrt{T}}$, $\delta = \frac{\log T}{T}$ and $\xi = \frac{\delta}{r}$, then

$$\mathbb{E} \sum_{t=1}^T \frac{1}{2} (\ell_t(y_{t,1}) + \ell_t(y_{t,2})) - \min_{x \in \mathcal{K}} \mathbb{E} \sum_{t=1}^T \ell_t(x) \leq (d^2 G^2 + D^2) \sqrt{T} + G \log(T) \left(3 + \frac{D}{r} \right).$$

Proof: Since we do not assume that the loss functions ℓ_t , for $t \geq 1$, are strongly convex, we add a fictitious round in the beginning with $\ell_0(x) = \frac{\sqrt{T}}{2} \|x\|^2$, which has $\sigma_0 = \sqrt{T}$. Setting $\sigma_t = 0$ for $t \geq 1$ leads to $\sigma_{0:t} = \sqrt{T}$ for all $t \geq 0$. Plugging this in the result of Theorem 3, we have for all $x \in \mathcal{K}$,

$$\mathbb{E} \sum_{t=0}^T \frac{1}{2} (\ell_t(y_{t,1}) + \ell_t(y_{t,2})) - \mathbb{E} \sum_{t=0}^T \ell_t(x) \leq \frac{d^2 G^2}{2} \sum_{t=0}^T \frac{1}{\sqrt{T}} + G \log(T) \left(3 + \frac{D}{r} \right). \quad (13)$$

For the regret in the initial fictitious round, we have

$$\frac{1}{2} (\ell_0(y_{t,1}) + \ell_0(y_{t,2})) - \ell_0(x) = \frac{1}{2} \left(\frac{\sqrt{T}}{2} \|y_{0,1}\|^2 + \frac{\sqrt{T}}{2} \|y_{0,2}\|^2 \right) - \frac{\sqrt{T}}{2} \|x\|^2 \geq -\frac{\sqrt{T}}{2} D^2.$$

Rearranging the inequality (13) gives

$$\begin{aligned} \mathbb{E} \sum_{t=1}^T \frac{1}{2} (\ell(y_{t,1}) + \ell(y_{t,2})) - \mathbb{E} \sum_{t=1}^T \ell_t(x) &\leq \frac{d^2 G^2}{2} \frac{T+1}{\sqrt{T}} + \frac{\sqrt{T}}{2} D^2 + G \log(T) \left(3 + \frac{D}{r} \right) \\ &\leq (d^2 G^2 + D^2) \sqrt{T} + G \log(T) \left(3 + \frac{D}{r} \right). \end{aligned}$$

Since this holds for all $x \in \mathcal{K}$, it is certainly true for $\arg \min_{x \in \mathcal{K}} \mathbb{E} \sum_{t=1}^T \ell_t(x)$. \blacksquare

Corollary 5 Suppose that assumptions (1) and (2) hold, and each loss function ℓ_t is σ -strongly convex for some $\sigma > 0$. If Algorithm 2 is run with $\eta_t = \frac{1}{\sigma t}$, $\delta = \frac{\log T}{T}$ and $\xi = \frac{\delta}{r}$, then

$$\mathbb{E} \sum_{t=1}^T \frac{1}{2} (\ell_t(y_{t,1}) + \ell_t(y_{t,2})) - \min_{x \in \mathcal{K}} \mathbb{E} \sum_{t=1}^T \ell_t(x) \leq G \log(T) \left(\frac{d^2 G}{\sigma} + 3 + \frac{D}{r} \right).$$

Proof: In this case, $\sigma_t = \sigma > 0$ and therefore $\sigma_{1:t} = \sigma t$. Hence we can directly plug these values into Theorem 3 to conclude

$$\begin{aligned} \mathbb{E} \sum_{t=1}^T \frac{1}{2} (\ell_t(y_{t,1}) + \ell_t(y_{t,2})) - \min_{x \in \mathcal{K}} \mathbb{E} \sum_{t=1}^T \ell_t(x) &\leq \frac{d^2 G^2}{2} \sum_{t=1}^T \frac{1}{\sigma t} + G \log(T) \left(3 + \frac{D}{r} \right) \\ &\leq \frac{d^2 G^2}{2\sigma} (1 + \log(T)) + G \log(T) \left(3 + \frac{D}{r} \right). \\ &\leq \frac{d^2 G^2}{\sigma} \log(T) + G \log(T) \left(3 + \frac{D}{r} \right) \quad \text{for } T \geq 3. \end{aligned}$$

■

While expected regret bounds are interesting, the regret may still have a high variance. In order to argue that Algorithm 2 often enjoys a small regret, we need to prove a bound that holds with high probability. For that, we turn to concentration inequalities for martingales. For any point $x \in (1 - \xi)\mathcal{K}$, define

$$Z_t = \hat{\ell}_t(x_t) - \hat{\ell}_t(x) - h_t(x_t) + h_t(x).$$

Since $\mathbb{E}_t h_t(x) = \hat{\ell}_t(x)$ for any x that is independent of u_t , we have $\mathbb{E}_t Z_t = 0$. In addition,

$$|Z_t| \leq |\hat{\ell}_t(x_t) - \hat{\ell}_t(x)| + |h_t(x_t) - h_t(x)| \leq G\|x_t - x\| + 3Gd\|x_t - x\| \leq 8GdD.$$

Thus the sequence Z_t is a bounded martingale difference sequence. Hence, we can use the Hoeffding-Azuma inequality to derive a high probability guarantee for the case of convex, Lipschitz functions.

Theorem 6 Suppose assumptions (1) and (2) hold. If Algorithm 2 is run with $\eta_t = \frac{1}{\sqrt{T}}$, $\delta = \frac{\log T}{T}$ and $\xi = \frac{\delta}{r}$. Then for any fixed $x \in \mathcal{K}$ and any $\delta_1 > 0$, with probability at least $1 - \delta_1$,

$$\sum_{t=1}^T \frac{1}{2} (\ell_t(y_{t,1}) + \ell_t(y_{t,2})) - \sum_{t=1}^T \ell_t(x) \leq (d^2 G^2 + D^2)\sqrt{T} + G \log(T) \left(3 + \frac{D}{r} \right) + 8dGD\sqrt{2T \log(1/\delta_1)}.$$

Proof: Using the Hoeffding-Azuma inequality, we conclude that

$$\mathbb{P} \left(\sum_{t=1}^T Z_t > \epsilon \right) \leq \exp \left(-\frac{\epsilon^2}{2TB^2} \right),$$

where $B = 8GdD$ as shown before. Let $\delta_1 = \exp(-\frac{\epsilon^2}{2TB^2})$. Then $\epsilon = B\sqrt{2T \log(1/\delta_1)}$. Hence we know that with probability at least $1 - \delta_1$

$$\sum_{t=1}^T \hat{\ell}_t(x_t) - \hat{\ell}_t(x(1 - \xi)) \leq \sum_{t=1}^T h_t(x_t) - h_t(x(1 - \xi)) + B\sqrt{2T \log(1/\delta_1)}.$$

Now using Lemma 2,

$$\sum_{t=1}^T \frac{1}{2} (\ell_t(y_{t,1}) + \ell_t(y_{t,2})) - \sum_{t=1}^T \ell_t(x) \leq \sum_{t=1}^T h_t(x_t) - h_t(x(1 - \xi)) + B\sqrt{2T \log(1/\delta_1)} + 3TG\delta + TGD\xi.$$

To bound the regret $\sum_{t=1}^T h_t(x_t) - h_t(x(1 - \xi))$, we use Lemma 1 on the regularized loss sequence exactly like in the proof of Corollary 4. Plugging in the stated values of η_t and δ proves the theorem. ■

When the adversary is adaptive, however, the loss sequence depends on the player's moves. As a result, the best comparator, which minimizes $\sum_{t=1}^T \ell_t(x)$, depends on the player's moves too. The high probability result stated above doesn't allow us to compete with such a comparator, as the comparator x in the theorem statement is fixed ahead of time and is not allowed to depend on the random x_t 's. To obtain a regret bound against $\min_{x \in \mathcal{K}} \sum_{t=1}^T \ell_t(x)$, we need to take the high probability bound for a fixed $x \in \mathcal{K}$ described above, and combine it with an appropriate union bound over the entire set to account for all possible minimizers of the cumulative loss sequence. This union bound can be taken either over a barycentric spanner, or over a cover of size at most

$(4dT)^{d/2}$. This step is the same as the one described in Bartlett et al. (2008a) or Dani and Hayes (2006), and is not repeated here.

While this result provides nearly tight (up to log factors and constants) guarantees on high probability regret for general convex functions, it falls short for the case of strongly convex functions. We note that in applying Hoeffding–Azuma’s concentration inequality, we incur an additional term of $\tilde{O}(\sqrt{T})$. Showing an $O(\log T)$ regret bound for strongly convex functions that holds with high probability remains an open question. We would like to mention that the techniques of Kakade and Tewari (2009), which provide such a result in the online to batch conversion setting, are not applicable to our case. This is because in stochastic optimization of strongly convex functions, the sequence x_t converges rapidly to the optimal point. This rapid concentration doesn’t occur in an adversarial bandit optimization setting; the sequence x_t might actually track the oscillations of the losses ℓ_t and give a much lower regret than any constant point. Such cases present a novel difficulty and resolving this difficulty is an interesting question for future research.

3 A general class of gradient estimators for smooth functions

In the previous section, we focused on a gradient estimator based on two function evaluations along a random direction uniformly distributed on the unit sphere centered at x_t . It is natural to ask if one can obtain similar regret bounds for other estimators based on random sampling with different probability distributions. It turns out that with an additional smoothness assumption on the loss functions, we can indeed show similar regret bounds with more general gradient estimators.

The additional smoothness assumption is that each loss function ℓ_t is L -smooth, i.e., they satisfy the condition (4). We recall that a direct consequence of (4) is the following second-order property

$$\ell_t(x) \leq \ell_t(y) + \langle \nabla \ell_t(y), x - y \rangle + \frac{L}{2} \|x - y\|^2, \quad \forall x, y \in \mathcal{K}. \quad (14)$$

If the loss functions are L -smooth, then we can show that the expectation of the gradient estimator in Equation (9) is close to the true gradient. We start with a useful fact from linear algebra.

Lemma 7 *For any fixed $v \in \mathbb{R}^d$ and for $u \in \mathbb{R}^d$ chosen randomly with $\|u\| = 1$, it holds that $v = d\mathbb{E} \langle v, u \rangle u$.*

Proof: The lemma is based on the observation that $\mathbb{E} uu^\top = \frac{1}{d}I$, where u is a random unit vector and I is the $d \times d$ identity matrix. To see this, recall that symmetry implies $\mathbb{E} u_i u_j = 0$ for any two coordinates i, j . Also u is a unit vector so $\mathbb{E} \sum_i u_i^2 = 1$. Again by symmetry, this implies that $\mathbb{E} u_i^2 = \frac{1}{d}$ for all i . Therefore, $Iv = (d\mathbb{E} uu^\top)v$, which equals $d\mathbb{E} \langle v, u \rangle u$. \blacksquare

The above lemma allows us to conclude

$$\begin{aligned} \|\mathbb{E}_t \tilde{g}_t - \nabla \ell_t(x_t)\| &= \|\mathbb{E}_t \tilde{g}_t - d\mathbb{E}_t \langle \nabla \ell_t(x_t), u_t \rangle u_t\| \leq \mathbb{E}_t \|\tilde{g}_t - d \langle \nabla \ell_t(x_t), u_t \rangle u_t\| \\ &\leq d\mathbb{E}_t \left| \frac{1}{2\delta} (\ell_t(x_t + \delta u_t) - \ell_t(x_t - \delta u_t)) - \langle \nabla \ell_t(x_t), u_t \rangle \right| \quad \text{since } \|u_t\| = 1. \end{aligned}$$

By the smoothness assumption, more specifically (14), we have

$$\ell_t(x_t + \delta u_t) \leq \ell_t(x_t) + \delta \langle \nabla \ell_t(x_t), u_t \rangle + \frac{L\delta^2}{2}.$$

Also by convexity of ℓ_t , $\ell_t(x_t - \delta u_t) \geq \ell_t(x_t) - \delta \langle \nabla \ell_t(x_t), u_t \rangle$. Combining the two inequalities above, we get

$$\frac{1}{2\delta} (\ell_t(x_t + \delta u_t) - \ell_t(x_t - \delta u_t)) - \langle \nabla \ell_t(x_t), u_t \rangle \leq \frac{L\delta}{4}.$$

Similarly by using (14) on $\ell_t(x_t - \delta u_t)$ and the convexity inequality on $\ell_t(x_t + \delta u_t)$, we can lower bound the left-hand side of the above inequality by $-\frac{L\delta}{4}$. This allows us to obtain

$$\|\mathbb{E}_t \tilde{g}_t - \nabla \ell_t(x_t)\| \leq \frac{dL\delta}{4}.$$

It turns out that the boundedness of the gradient estimator \tilde{g}_t along with the closeness of its expectation to the true gradient is sufficient to reproduce the regret bounds of the previous section. In particular, we do not need to rely on random vectors uniformly distributed on the unit sphere centered at x_t and we do not need to use the function $\hat{\ell}_t(x)$. To illustrate this point, we only show how a corresponding version of Theorem 3 is proved in this general case. The corresponding versions of other results follow in a similar fashion.

Theorem 8 Assume that (1) and (2) hold, each function ℓ_t is σ_t -strongly convex, and $\sigma_1 > 0$. Suppose that on round t the player issues k random queries $y_{t,1}, \dots, y_{t,k}$, constructs a gradient estimator \tilde{g}_t , and uses the algorithm $x_{t+1} = \Pi_{\mathcal{K}(1-\xi)}(x_t - \eta_t \tilde{g}_t)$ with $\eta_t = \frac{1}{\sigma_{1:t}}$, $\delta = \frac{\log T}{T}$ and $\xi = \frac{\delta}{r}$. If the gradient estimator satisfies the following conditions for all $t \geq 1$:

- (i) $\|x_t - y_{t,i}\| \leq \delta$ for $i = 1, \dots, k$.
- (ii) $\|\tilde{g}_t\| \leq G_1$ for some constant G_1 .
- (iii) $\|\mathbb{E}_t \tilde{g}_t - \nabla \ell_t(x_t)\| \leq c\delta$ for some constant c .

Then for any fixed $x \in \mathcal{K}$ we have

$$\mathbb{E} \sum_{t=1}^T \frac{1}{k} \sum_{i=1}^k \ell_t(y_{t,i}) - \mathbb{E} \sum_{t=1}^T \ell_t(x) \leq \frac{G_1^2}{2} \sum_{t=1}^T \frac{1}{\sigma_{1:t}} + G \log(T) \left(1 + 2c + \frac{D}{r} \right).$$

Proof: We start by defining $h_t(x) = \ell_t(x) + (\tilde{g}_t - \nabla \ell_t(x))^\top x$. Then it is clear that h_t has the same convexity properties as ℓ_t and $\nabla h_t(x_t) = \tilde{g}_t$. So Lemma 1 still holds, with a gradient bound of G_1 instead of dG , and we get

$$\sum_{t=1}^T h_t(x_t) - h_t(x) \leq \frac{G_1^2}{2} \sum_{t=1}^T \frac{1}{\sigma_{1:t}}.$$

Then we can take expectations to conclude

$$\begin{aligned} \mathbb{E} \sum_{t=1}^T [\ell_t(x_t) - \ell_t(x)] &= \mathbb{E} \sum_{t=1}^T [h_t(x_t) - h_t(x)] + \mathbb{E} \sum_{t=1}^T [\ell_t(x_t) - h_t(x_t) - \ell_t(x) + h_t(x)] \\ &\leq \frac{G_1^2}{2} \sum_{t=1}^T \frac{1}{\sigma_{1:t}} + \mathbb{E} \sum_{t=1}^T (\mathbb{E}_t \tilde{g}_t - \nabla \ell_t(x_t))^\top (x_t - x) \\ &\leq \frac{G_1^2}{2} \sum_{t=1}^T \frac{1}{\sigma_{1:t}} + 2c\delta DT. \end{aligned}$$

In the first inequality above, we used convexity of ℓ_t and h_t . We can now use arguments similar to the proof of Lemma 2 to conclude

$$\mathbb{E} \sum_{t=1}^T \frac{1}{k} \sum_{i=1}^k \ell_t(y_{t,i}) - \mathbb{E} \sum_{t=1}^T \ell_t(x) \leq \frac{G_1^2}{2} \sum_{t=1}^T \frac{1}{\sigma_{1:t}} + 2cD\delta T + GT\delta + GDT\xi.$$

Plugging in the values of δ and ξ proves the theorem. ■

As shown in the beginning of this section, the assumptions of Theorem 8 are satisfied with the gradient estimator (9) if the loss functions ℓ_t are L -smooth. In that case, we have $G_1 = dG$ and $c = dL/4$, and $k = 2$ queries per round suffice.

Theorem 8 also generalizes the special case of optimizing with noisy gradients, which has been previously studied in the optimization community. Suppose that for each query point x_t , the player only has access to a noisy version of the gradient \tilde{g}_t . In this case, we have $k = 1$ and $y_{t,1} = x_t$, and it is not even necessary to assume the smoothness condition. If the approximate gradient satisfies the conditions (ii) and (iii) in Theorem 8, then we have the stated regret bound in expectation. If the approximate gradient has a deterministic bound, i.e., $\|\tilde{g}_t - \nabla \ell_t(x_t)\| \leq c\delta$ for all t , then the regret bound becomes deterministic.

As a more interesting application of this general result, we demonstrate another estimator \tilde{g}_t that meets the conditions of Theorem 8. On each round t , the player picks an integer i_t from the set $\{1, \dots, d\}$ uniformly at random, queries the function values at two points along the corresponding coordinate axes e_{i_t} ; $y_{t,1} = x_t + \delta e_{i_t}$, $y_{t,2} = x_t - \delta e_{i_t}$. Then it constructs the estimator

$$\tilde{g}_t = \frac{d}{2\delta} (\ell_t(y_{t,1}) - \ell_t(y_{t,2})) e_{i_t}.$$

It is clear that $\|x_t - y_{t,i}\| \leq \delta$. As in the previous section, this gradient estimator has a bounded norm with $G_1 = dG$. Moreover,

$$\mathbb{E}_t \tilde{g}_t = \frac{1}{2\delta} \sum_{i=1}^d (\ell_t(x_t + \delta e_i) - \ell_t(x_t - \delta e_i)) e_i.$$

Algorithm 3 Gradient descent with deterministic estimator based on $d + 1$ points

Input: Learning rates η_t , exploration parameter δ and shrinkage coefficient ξ .
Set $x_1 = 0$
for $t = 1, \dots, T$ **do**
 Observe $\ell_t(x_t), \ell_t(x_t + \delta e_i)$ for $i = 1, \dots, d$.
 Set $\tilde{g}_t = \frac{1}{\delta} \sum_{i=1}^d (\ell_t(x_t + \delta e_i) - \ell_t(x_t)) e_i$.
 Update $x_{t+1} = \Pi_{(1-\xi)\mathcal{K}}(x_t - \eta_t \tilde{g}_t)$.
end for

Then

$$\begin{aligned} \|\mathbb{E}_t \tilde{g}_t - \nabla \ell_t(x_t)\| &= \sqrt{\sum_{i=1}^d \left| \frac{1}{2\delta} (\ell_t(x_t + \delta e_i) - \ell_t(x_t - \delta e_i)) - \langle \nabla \ell_t(x_t), e_i \rangle \right|^2} \\ &\leq \frac{\sqrt{d}L\delta}{4}. \end{aligned}$$

Thus Theorem 8 applies to this estimator with $G_1 = dG$ and $c = \sqrt{d}L/4$. This means that a coordinate descent style algorithm that uses only 1-dimensional gradient estimators will also exhibit a low regret for optimizing smooth convex Lipschitz functions.

Since the gradient estimator is 1-dimensional along a coordinate axis, gradient updates become computationally more efficient. We also note that this improvement in per-iteration cost comes at no worsening of the regret bound when compared to using a random direction on the unit sphere. In the full information case, stochastic coordinate descent approaches usually have a convergence rate slower by a factor of d compared to gradient descent, which offsets the computational gains (Shalev-Shwartz & Tewari, 2009; Tseng & Yun, 2009). In this case, however, both the gradient descent method and coordinate descent have to rely on an estimator with the same amount of variance, which leads to similar regret guarantees in the two cases and gives the coordinate descent approach a clear computational edge. This example is just a hint of the strength of Theorem 8.

4 Deterministic algorithms for completely adaptive adversaries

In the previous sections, we considered adversaries that know the player's past moves, but do not know the player's current random move. This seems reasonable for a randomized algorithm, because if the adversary can see the player's random bits, then randomization is futile. However, in some cases, the online interaction mandates an adversarial response dependent not only on the player's previous moves, but also on the current move. That is, after the player plays a point x_t , the loss function ℓ_t is chosen with the knowledge of x_t . While such a scenario is readily tackled in the full information setting with standard gradient based algorithms, it is impossible to adapt to such feedback in the bandit setting, as explained in Section 1. This prompts the question if there is any partial feedback setting where we might be able to effectively compete against such an adversary. In this section, we show one special partial feedback scenario where it is possible to compete against such a *completely adaptive* adversary.

We assume that the player is allowed to query each loss function at up to $d + 1$ points. In this case, the player can play the points $x_t, x_t + \delta e_i$ for $i = 1, \dots, d$, where e_i are the standard unit basis vectors, and then construct a *deterministic* gradient approximation

$$\tilde{g}_t = \frac{1}{\delta} \sum_{i=1}^d (\ell_t(x_t + \delta e_i) - \ell_t(x_t)) e_i. \quad (15)$$

The details of the player's moves are described in Algorithm 3.

We assume that each loss function ℓ_t is L -smooth and σ_t -strongly convex on \mathcal{K} . Such functions are always Lipschitz continuous, and we can derive a bound on the Lipschitz constant based on L and D . However, for convenience, we simply assume (2) holds with the constant G . We can derive just as in previous sections that

$$\|\tilde{g}_t\| \leq dG, \quad \|\tilde{g}_t - \nabla \ell_t(x_t)\| \leq \frac{\sqrt{d}L\delta}{2}. \quad (16)$$

These properties of \tilde{g}_t are the deterministic version of the properties discussed in the previous section for randomized algorithms. They immediately give us partial feedback algorithms based on using approximate gradients in the corresponding full information analogues. For instance, we can show the same regret guarantee against a stronger class of adversaries.

Theorem 9 Let $\{\ell_t\}_{t=1}^T$ be convex functions chosen by a completely adaptive adversary. Suppose the conditions (1) and (2) hold. In addition, let each ℓ_t be σ_t strongly convex and L -smooth. If Algorithm 3 is run with $\eta_t = \frac{1}{\sigma_{1:t}}$, $\delta = \frac{\log T}{T}$ and $\xi = \frac{\delta}{r}$, then

$$\sum_{t=1}^T \frac{1}{d+1} \left(\ell_t(x_t) + \sum_{i=1}^d \ell_t(x_t + \delta e_i) \right) - \min_{x \in \mathcal{K}} \sum_{t=1}^T \ell_t(x) \leq \frac{d^2 G^2}{2} \sum_{t=1}^T \frac{1}{\sigma_{1:t}} + G \log(T) \left(1 + \frac{\sqrt{d} L \delta}{2} + \frac{D}{r} \right).$$

Proof: The proof essentially mimics that of Theorem 8. We define $h_t(x) = \ell_t(x) + (\tilde{g}_t - \nabla \ell_t(x_t))^\top x$. Then $\|\nabla h_t(x_t)\| = \|\tilde{g}_t\| \leq dG$. So we obtain from Lemma 1 for any $x \in (1 - \xi)\mathcal{K}$,

$$\sum_{t=1}^T h_t(x_t) - h_t(x) \leq \frac{d^2 G^2}{2} \sum_{t=1}^T \frac{1}{\sigma_{1:t}}.$$

Proceeding as in the proof of Theorem 8 and using $\|\tilde{g}_t - \nabla \ell_t(x_t)\| \leq \frac{\sqrt{d} L \delta}{2}$ gives the result. \blacksquare

It turns out that we can say more about this deterministic gradient estimator. More specifically, we can use it to develop quasi-Newton type methods that can achieve a sharper regret bound for the class of exp-concave functions. Recall that a loss function ℓ_t is α -exp-concave if $\exp(-\alpha \ell_t(x))$ is a concave function. For example, if a function has a Lipschitz constant G and is σ -strongly convex, then the exp-concave property holds with $\alpha = G^2/\sigma$. However, the exp-concave condition is weaker than assuming the bounds G and σ . These functions arise in applications like portfolio optimization, squared-error regression etc.

All the algorithms discussed until now are gradient-descent based. Gradient-descent methods are often insufficient to obtain the optimal regret for the general class of exp-concave functions. Hazan et al. (2007) developed an online Newton-step algorithm that achieves an optimal $O(\log T)$ regret on these functions. In order to implement approximate second-order methods using gradient estimators, we also need to control error in second order gradient terms. To demonstrate good second-order properties of the gradient estimator (15), we use the following fact from linear algebra.

Lemma 10 Let u and v be two vectors in \mathbb{R}^d such that $\|u - v\| \leq \theta$. Then for any x

$$x^\top (uu^\top - vv^\top)x \leq \|x\|^2 \theta (2\|v\| + \theta).$$

Proof: We have

$$\begin{aligned} x^\top (uu^\top - vv^\top)x &= (x^\top u)^2 - (x^\top v)^2 = x^\top (u + v)x^\top (u - v) \\ &\leq \|x\|^2 \|u + v\| \|u - v\| \leq \|x\|^2 \theta (2\|v\| + \theta), \end{aligned}$$

where in the last inequality we used $\|u + v\| = \|2v + (u - v)\| \leq 2\|v\| + \|u - v\| \leq 2\|v\| + \theta$. \blacksquare

We apply this lemma with $u = \tilde{g}_t$ and $v = \nabla \ell_t(x_t)$ to conclude that for any vector x

$$x^\top (\tilde{g}_t^\top \tilde{g}_t - \nabla \ell_t(x_t)^\top \nabla \ell_t(x_t))x \leq \|x\|^2 \frac{\sqrt{d} L \delta}{2} \left(2G + \frac{\sqrt{d} L \delta}{2} \right). \quad (17)$$

With this second-order approximation bound in mind, we present the Bandit Online Newton-Step algorithm (BONES). This algorithm is a bandit version of the Online Newton Step algorithm, presented in Hazan et al. (2007).

We begin the analysis of the BONES algorithm by first proving the following lemma.

Lemma 11 Let \tilde{g}_t and β be as defined in Algorithm 4. Then for any $x \in (1 - \xi)\mathcal{K}$,

$$\sum_{t=1}^T \tilde{g}_t^\top (x_t - x) - \frac{\beta}{2} (x - x_t)^\top \tilde{g}_t \tilde{g}_t^\top (x - x_t) \leq \frac{d}{2\beta} \log (TG^2 d^4 \beta^2 + 1) + \frac{1}{2\beta}.$$

Proof: This result is based on the observation that the updates of BONES are equivalent to performing online Newton-step algorithm of Hazan et al. (2007) on the functions

$$\tilde{g}_t^\top (x - x_t) - \frac{\beta}{2} (x - x_t)^\top A_t (x - x_t).$$

Algorithm 4 Bandit Online Newton Step (BONES) algorithm

Set $x_1 = 0, A_0 = \epsilon I_d, \beta = \frac{1}{2} \min \left\{ \frac{1}{4GD}, \alpha \right\}$.
for $t = 1, \dots, T$ **do**
 Observe $\ell_t(x_t), \ell_t(x_t + \delta e_i)$ for $i = 1, \dots, d$.
 $\tilde{g}_t = \frac{1}{\delta} \sum_{i=1}^d (\ell_t(x_t + \delta e_i) - \ell_t(x_t)) e_i$.
 $A_t = A_{t-1} + \tilde{g}_t \tilde{g}_t^\top$.
 $x_{t+1} = \Pi_{(1-\xi)\mathcal{K}}^{A_t} \left(x_t - \frac{1}{\beta} A_t^{-1} \tilde{g}_t \right)$, where $\Pi_{\mathcal{S}}^{A_t}(y) = \arg \min_{x \in \mathcal{S}} (y - x)^\top A_t (y - x)$.
end for

Following the proof of Theorem 2 in (Hazan et al., 2007), we conclude that for any $x \in (1 - \xi)\mathcal{K}$,

$$\sum_{t=1}^T \tilde{g}_t^\top (x - x_t) - \frac{\beta}{2} (x - x_t)^\top A_t (x - x_t) \leq \frac{1}{2\beta} \sum_{t=1}^T \tilde{g}_t^\top A_t^{-1} \tilde{g}_t + \frac{1}{2\beta}.$$

Further following their analysis, we note that the first summand on the right-hand side can be bounded by controlling the eigenvalues of the matrices A_t . Together with $\|\tilde{g}_t\| \leq Gd$, this gives

$$\sum_{t=1}^T \tilde{g}_t^\top (x - x_t) - \frac{\beta}{2} (x - x_t)^\top A_t (x - x_t) \leq \frac{d}{2\beta} \log \left(\frac{(Gd)^2 T}{\epsilon} + 1 \right) + \frac{1}{2\beta}.$$

Finally setting $\epsilon = \frac{1}{\beta^2 d^2}$ gives the result. ■

This lemma now allows us to prove a bound on the regret incurred by the BONES algorithm against completely adaptive adversaries.

Theorem 12 *Let $\{\ell_t\}_{t=1}^T$ be chosen by a completely adaptive adversary. Suppose the conditions (1) and (2) hold. In addition, let each ℓ_t be α -exp-concave and L -smooth. If the BONES Algorithm is run with $\delta = \frac{\log T}{T}$ and $\xi = \frac{\delta}{r}$, then*

$$\begin{aligned} \sum_{t=1}^T \frac{1}{d+1} \left(\ell_t(x_t) + \sum_{i=1}^d \ell_t(x_t + \delta e_i) \right) - \min_{x \in \mathcal{K}} \sum_{t=1}^T \ell_t(x) &\leq 4d \left(GD + \frac{1}{\alpha} \right) \log \left(1 + \frac{Td^4 D^2}{64} \right) \\ &\quad + \log(T) \left(\frac{DL\sqrt{d}}{2} + \frac{2GD}{r} \right) + o(\log(T)). \end{aligned}$$

Proof: We will only outline the main steps in the proof. The omitted details can be found in (Hazan et al., 2007). We start by recalling a second-order property of exp-concave functions (Hazan et al., 2007, Lemma 3):

$$\ell_t(x) \geq \ell_t(x_t) + \nabla \ell_t(x_t)^\top (x - x_t) + \frac{\beta}{2} (x - x_t)^\top \nabla \ell_t(x_t) \nabla \ell_t(x_t)^\top (x - x_t).$$

In particular, this observation allows us to relate regret on the function ℓ_t with that of a local quadratic approximation.

$$\begin{aligned} \sum_{t=1}^T \ell_t(x_t) - \ell_t(x) &\leq \sum_{t=1}^T \nabla \ell_t(x_t)^\top (x_t - x) - \frac{\beta}{2} (x - x_t)^\top \nabla \ell_t(x_t) \nabla \ell_t(x_t)^\top (x - x_t) \\ &\leq \sum_{t=1}^T \tilde{g}_t(x_t)^\top (x_t - x) - \frac{\beta}{2} (x - x_t)^\top \tilde{g}_t \tilde{g}_t^\top (x - x_t) \\ &\quad + \sum_{t=1}^T \|\tilde{g}_t - \nabla \ell_t(x_t)\| \|x_t - x\| + \sum_{t=1}^T \frac{\beta}{2} (x_t - x) (\tilde{g}_t \tilde{g}_t^\top - \nabla \ell_t(x_t) \nabla \ell_t(x_t)^\top) (x - x_t) \end{aligned}$$

Now the first two terms on the right-hand side are bounded in Lemma 11. For the remaining terms, we use the equations (16) and (17). As before, we also need to relate this regret to the average regret on the $y_{t,i}$'s against a point in \mathcal{K} rather than $\mathcal{K}(1 - \xi)$. These can be done as before by appealing to Lemma 2. Plugging in all the constants gives the result. ■

5 Discussion

This paper introduces the multi-point bandit feedback setting for online convex optimization under partial information. While this setting is just a slight generalization of the bandit setting, we are able to show certain sharp phase transitions based on the number of queries per round, both in the nature of the bounds and in the nature of the adversaries against which these bounds hold. We provide optimal algorithms at several points along this continuum.

For future research, it will be interesting to investigate more general partial feedback models, for instance allowing an adaptive choice of the number of queries at every round. Proving an $O(\log(T))$ regret bound with high probability for the case of strongly convex functions also remains an open question for $k = 2$ queries.

References

Abernethy, J., Agarwal, A., Rakhlin, A., & Bartlett, P. L. (2009). A stochastic view of optimal regret through minimax duality. *Proceedings of the 22nd Annual Conference on Learning Theory*.

Abernethy, J., & Rakhlin, A. (2009). Beating the adaptive bandit with high probability. *Proceedings of the 22nd Annual Conference on Learning Theory*.

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (2003). The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32, 48–77.

Bartlett, P. L., Dani, V., Hayes, T., Kakade, S., Rakhlin, A., & Tewari, A. (2008a). High-probability regret bounds for bandit online linear optimization. *Proc. of the 21st Annual COLT*.

Bartlett, P. L., Hazan, E., & Rakhlin, A. (2008b). Adaptive online gradient descent. *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press.

Cesa-Bianchi, N., & Lugosi, G. (2006). *Prediction, learning and games*. Cambridge University Press.

Conn, A. R., Scheinberg, K., & Vicente, L. N. (2009). *Introduction to derivative-free optimization*. Philadelphia, PA: SIAM.

Dani, V., & Hayes, T. P. (2006). Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary. *ACM-SIAM Symposium on Discrete Algorithms*.

Dani, V., Hayes, T. P., & Kakade, S. M. (2008). Stochastic linear optimization under bandit feedback. *Proceedings of the 21st Annual Conference on Learning Theory*.

Flaxman, A. D., Kalai, A. T., & McMahan, B. H. (2005). Online convex optimization in the bandit setting: gradient descent without a gradient. *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 385–394).

Hazan, E., Agarwal, A., & Kale, S. (2007). Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69, 169–192.

Kakade, S. M., & Tewari, A. (2009). On the generalization ability of online strongly convex programming algorithms. *Advances in Neural Information Processing Systems 21* (pp. 801–808).

Polyak, B. T., & Tsyplkin, Y. Z. (1973). Pseudogradient adaption and training algorithms. *Automation and Remote Control*, 34, 377–397.

Shalev-Shwartz, S., & Tewari, A. (2009). Stochastic methods for ℓ_1 regularized loss minimization. *Proceedings of the 26th International Conference on Machine Learning*.

Spall, J. C. (2003). *Introduction to stochastic search and optimization: Estimation, simulation, and control*. Hoboken, NJ: John Wiley and Sons.

Tseng, P., & Yun, S. (2009). A block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization. *J. of Optimization Theory and Applications*, 140, 513–535.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. *Proceedings of the 20th International Conference on Machine Learning* (pp. 928–936).