Dynamic Regret Bounds for Online Gradient Descent with $d + 1$ point Bandit Feedback

Dhamma Kimpara

December 9, 2020

1 Introduction

Online convex optimization (OCO) is an area of great interest with wide ranging applications in machine learning and controls. In general, the goal of OCO is to minimize a stream of convex loss functions without prior information about the function. This can also be viewed as a game between the algorithm (player) and the environment (adversary). The algorithm chooses an action $x_t \in \mathcal{K} \subseteq \mathbb{R}^d$ and the environment gives a loss $\ell_t : \mathcal{K} \to \mathbb{R}$.

The algorithm may have several different levels of access to the loss function, corresponding to different problem settings. The most typical setting is that the algorithm has access to gradient information. However, here we focus on the Bandit setting where the algorithm only has access to the functional evaluation(s) of the action(s) that the algorithm plays.

We measure the performance of various algorithms with regret. However, here in contrast to the majority of the work on Bandit OCO, we focus on dynamic or time-varying regret:

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

In the Bandit setting, the algorithm cannot compete against a completely adaptive adversary who chooses the loss function ℓ_t after observing the action x_t Agarwal et al. [2010]. Hence for this setting, we will show an algorithm that is no-regret against an adaptive adversary that can only observe information of the $t-1$ rounds preceding round t.

Overall in the literature we observe a large gap between between the optimal regret bounds between the full information and bandit settings. In particular, in the full information there are algorithms that are no-regret against a completely adaptive adversary. This raises the question of studying a range of problem between bandit (single function evaluation) and full information extremes. We thus begin with studying a generic k-point bandit feedback setting. Then we show that for $k = d + 1$-point feedback, there is a deterministic algorithm that is no-regret against a completely adaptive adversary, therefore matching the full-information bound.

For this k-point feedback setting, we define the expected regret:

Definition 1. k-point Bandit Regret

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

where the expectation is taken over the randomness of the player.

As is typical in Bandit OCO, the algorithm relies on Online Gradient Descent (OGD) which uses a gradient estimator $\tilde{g}_t$. Thus the regret bounds are based on the regret bounds of OGD. Here we draw extensively from Agarwal et al. [2010] which studies our problem but with static regret and Mokhtari et al. [2016] that gives dynamic regret bounds for OGD.

2 Preliminaries

2.1 Problem Setting

First we formally introduce the problem. Our objective is:

$$\min_{x_t \in \mathcal{K}} \ell_t(x_t)$$

over rounds $t = 1, \ldots, T$. Where $\mathcal{K} \subset \mathbb{R}^d$ is our action set and ℓ_t are adversarially chosen time varying loss functions. The key in our setting is that we do not have first order gradient information $\nabla \ell_t$ but we are able to get zeroth-order (bandit) feedback with $k = d + 1$ points. First we outline our assumptions.

We assume that $\mathcal{K}$ is compact and has a nonempty interior (otherwise project $\mathcal{K}$ to a lower dimensional space). For this work, when we implicitly or explicitly refer to norms, we will be using the euclidean norm. We also have the following assumptions for each loss function ℓ_t:

Assumption 1. Let $\mathcal{B}$ denote the unit ball centered at the origin. There exists $r, D > 0$ such that

$$r\mathcal{B} \subseteq \mathcal{K} \subseteq D\mathcal{B}$$

Assumption 2. Strong Convexity. For $\mu \geq 0$, ℓ_t is μ-strongly convex over the set $\mathcal{K}$:

$$\ell_t(x) \geq \ell_t(y) + \nabla \ell_t(y)^\intercal (x - y) + \frac{\mu}{2}\|x - y\|^2, \quad \forall x, y \in \mathcal{K}$$

Assumption 3. ℓ_t is L-smooth on $\mathcal{K}$ if it is differentiable on an open set containing $\mathcal{K}$ and its gradient is Lipschitz continuous with constant L:

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

.

Assumption 4. ℓ_t is G-Lipschitz:

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

.

Hence the gradient of ℓ_t over $\mathcal{K}$ is bounded:

$$\|\nabla \ell_t(x)\| \leq G \quad \forall t, \forall x \in \mathcal{K}$$

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

3 Projected Gradient Descent with k queries per round

Algorithm 1 Generic k-point Bandit Online Gradient Descent

Input: Step size η, exploration parameter δ and shrinkage coefficient ξ.
Set $x_1 = 0$
for $t = 1, ..., T$ **do**
  Observe randomized queries around x_t: $\ell_t(y_{t,1}), \ldots, \ell_t(y_{t,k})$
  Estimate the gradient $\tilde{g}_t = g(\ell_t(y_{t,1}), \ldots, \ell_t(y_{t,k}))$.
  Update $x_{t+1} = proj_{(1-\xi)\mathcal{K}}(x_t - \eta \tilde{g}_t)$.
end for

In this section we present a no-regret result for any algorithm that is an instantiation of Algorithm 1 for some k and where its randomized gradient estimate satisfies boundedness and closeness of its expectation to the true gradient. We assume the reader is familiar with OGD.

We first explain the algorithm. It plays k actions, constructs a gradient estimate $\tilde{g}_t$ from its k actions in round t and performs the OGD step to produce the next action x_{t+1}:

$$x_{t+1} = proj_{(1-\xi)\mathcal{K}}(x_t - \eta\tilde{g}_t),$$

where $\xi \in (0,1)$ and $(1-\xi)\mathcal{K}$ is shorthand for $\{(1-\xi)x : x \in \mathcal{K}\}$. Note that our gradient estimators will randomly query around the action x_t in order to estimate the gradient. Thus the projection is made onto the shrunk set to ensure that the random queries around the point x_{t+1} belong to $\mathcal{K}$. For any $x \in (1-\mathcal{K})$ and any unit vector u it holds that $(x+\delta u) \in \mathcal{K}$ for any $\delta \in [0, \xi r]$ [Flaxman et al., 2004].

We now present the main result. But first, we need two Lemmas the first is the dynamic regret bound of OGD and the second is the difference in regret between playing x_t on $\mathcal{K}$ and $\{y_{t,i}\}_{i=1}^k$ on $(1-\xi)\mathcal{K}$. [DK: check η]

Lemma 1. [Mokhtari et al., 2016]. Assume that the functions h_t are strongly convex and L-smooth. Assume further that the gradient norms are bounded and the step size is chosen such that $\eta \leq 1/L$. Then, the dynamic regret $Regret_T^d$ for the sequence of actions x_t generated by OGD is bounded by

$$Regret_T^d(OGD, G) := \sum_{t=1}^T h_t(x_t) - \sum_{t=1}^T \min_{x_t^* \in \mathcal{K}} h_t(x_t^*) \leq GK_1 \sum_{t=2}^T \left\|x_t^* - x_{t-1}^*\right\| + GK_2$$

where the constants K_1 and K_2 are explicitly given by

$$K_1 := \frac{\|x_1 - x_1^*\| - \rho\|x_T - x_T^*\|}{(1-\rho)}, \quad K_2 := \frac{1}{1-\rho}.$$

Where $0 \leq \rho := (1-\eta\mu)^{1/2} < 0$. Is our linear convergence constant.

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

$$\frac{1}{k}\sum_{i=1}^k \ell_t(y_{t,i}) - \ell_t(x) \leq \ell_t(x_t) - \ell_t((1-\xi)x) + G\delta + GD\xi.$$

Proof. By assumption of Lipschitz continuity,

$$\ell_t(y_{t,i}) \leq \ell_t(x_t) + G\delta.$$

We also have that by the Lipschitz property and $\|x\| \leq D$, for all $x \in \mathcal{K}$,

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

Combining the above two inequalities we get

$$\frac{1}{k}\sum_{i=1}^k \ell_t(y_{t,i}) + \ell_t((1-\xi)x) \leq \ell_t(x_t) + \ell_t(x) + G\delta + GD\xi.$$

Rearranging terms gives us the Lemma.

□

Theorem 1. Assume that the assumptions hold. Suppose on round t the algorithm plays k random queries $y_{t,1}, ..., y_{t,k}$, constructs a gradient estimator $\tilde{g}_t = g(y_{t,1}, ..., y_{t,k})$ and uses the the algorithmic step $x_{t+1} = proj_{(1-\xi)\mathcal{K}}(x_t - \eta\tilde{g}_t)$ with $\eta \leq \frac{1}{2G_1}$, $\delta = \frac{\log(T)}{T}$, and $\xi = \frac{\delta}{r}$. If the gradient estimator satisfies the following conditions for all $t \geq 1$:

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

Then for any sequence $\{x_t^*\}_{t=1}^T, x_t^* \in \mathcal{K}$ we have

$$\mathbb{E}\frac{1}{k}\sum_{t=1}^{T}\sum_{i=1}^{k}\ell_t(y_{t,i}) - \mathbb{E}\sum_{t=1}^{T}\min_{x_t^*\in\mathcal{K}}\ell_t(x_t^*) \leq G_1 K_1 \sum_{t=2}^{T}\big\|x_t^* - x_{t-1}^*\big\| + G_1 K_2 + G\log(T)\big(1 + 2c + \frac{D}{r}\big).$$

where the constants K_1 and K_2 are explicitly given by

$$K_1 := \frac{\|x_1 - x_1^*\| - \rho\|x_T - x_T^*\|}{(1-\rho)}, \quad K_2 := \frac{1}{1-\rho}.$$

Where $0 \leq \rho := (1-\eta\mu)^{1/2} < 0$ is our linear convergence constant.

Proof. Start by defining $h_t(x) = \ell_t(x) + (\tilde{g}_t - \nabla\ell_t(x))^\intercal x$. Then h_t has the same convexity properties as ℓ_t and is L_1-smooth. Note also that $\nabla h_t(x_t) = \tilde{g}_t$. So we can pretend that the algorithm is actually performing deterministic gradient descent, as if with full information, on the functions h_t restricted to $(1-\xi)\mathcal{K}$. Using the OGD regret bound from Lemma 1 we have that since $\nabla h_t(x_t) = \tilde{g}_t$ and hence $\|\tilde{g}_t\| \leq G_1$ (So h_t is $2G_1$-smooth since ∇h_t is $2G_1$-Lipschitz),

$$\sum_{t=1}^{T} h_t(x_t) - \sum_{t=1}^{T} h_t(x_t^*) \leq G_1 K_1 \sum_{t=2}^{T}\big\|x_t^* - x_{t-1}^*\big\| + 2G_1K_2 := Regret_T^d(OGD, G_1).$$

Then taking expectations,

$$\begin{aligned}
\mathbb{E}\sum_{t=1}^{T}[\ell_t(x_t) - \ell_t(x_t^*)] &= \mathbb{E}\sum_{t=1}^{T}[h_t(x_t) - h_t(x_t^*)] + \mathbb{E}\sum_{t=1}^{T}[\ell_t(x_t) - h_t(x_t) - \ell_t(x_t^*) + h_t(x_t^*)] \\
&\leq Regret_T^d(OGD, G_1) + \mathbb{E}\sum_{t=1}^{T}(\mathbb{E}_t\tilde{g}_t - \nabla\ell_t(x_t))^\intercal(x_t - x_t^*) \\
&\leq Regret_T^d(OGD, G_1) + 2c\delta DT.
\end{aligned}$$

Where the first inequality is by the convexity of ℓ_t and h_t. Now we use Lemma 2 and obtain

$$\mathbb{E}\frac{1}{k}\sum_{t=1}^{T}\sum_{i=1}^{k}\ell_t(y_{t,i}) - \mathbb{E}\sum_{t=1}^{T}\min_{x_t^*\in\mathcal{K}}\ell_t(x_t^*) \leq Regret_T^d(OGD, G_1) + 2c\delta DT + TG\delta + GDT\xi.$$

To finish the proof, plug in the values for δ and ξ.

□

4 $d+1$ point feedback

In this section, we show that we can construct a deterministic gradient estimator using $d+1$ point feedback. Thus we obtain a deterministic version of Theorem 1. Hence, the algorithm is no-regret even against completely adaptive adversaries meaning that the adversary can choose the loss ℓ_t after the algorithm plays x_t. Hence we match the full-information bound.

Algorithm 2 Gradient descent with deterministic estimator based on $d+1$ points

Input: Step size η, exploration parameter δ and shrinkage coefficient ξ.
Set $x_1 = 0$
for $t = 1, ..., T$ **do**
  Observe $\ell_t(x_t)$, $\ell_t(x_t + \delta e_i)$ for $i = 1, ..., d$.
  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 $x_{t+1} = proj_{(1-\xi)\mathcal{K}}(x_t - \eta\tilde{g}_t)$.
end for

The algorithm constructs the deterministic gradient estimator

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

Where e_i's are the standard unit basis vectors. We further need only the assumptions on strong convexity and L-smoothness, since they imply a bound on the gradient which we denote G. We can thus derive a bound on the norm of the estimator

$$\begin{aligned}
\|\tilde{g}_t\| &= \left\|\frac{1}{\delta}\sum_{i=1}^{d}(\ell_t(x_t+\delta e_i) - \ell_t(x_t))e_i\right\| \\
&\leq \frac{d}{\delta}\max_i\|\ell_t(x_t+\delta e_i) - \ell_t(x_t)\| \\
&\leq \frac{d}{\delta}\delta G \\
&= dG.
\end{aligned}$$

Where the second inequality is by the Lipschitz property. We can also derive the divergence of the estimator:

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

By the smoothness assumption, we have for all i

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

And by convexity we have $\ell_t(x_t+\delta e_i) \geq \ell_t(x_t) + \delta\langle\nabla\ell_t(x_t), e_i\rangle$. Hence

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

So we conclude that

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

Hence we have that the properties of $\tilde{g}_t$ are the deterministic version of the properties of the estimator outline in theorem 1. Hence we have an algorithm that can guarantee no-regret against a completely adaptive adversary.

Theorem 2. Suppose a completely adaptive adversary chooses the sequence of loss functions $\{\ell_t\}_{t=1}^T$ subject to the same assumptions as above. If Algorithm 2 is run with the $\eta \leq \frac{1}{2dG}$, $\delta = \frac{\log(T)}{T}$, and $\xi = \frac{\delta}{r}$, then

$$\sum_{t=1}^{T} \frac{1}{d+1}(\ell_t(x_t) + \sum_{i=1}^{d} \ell_t(x_t + \delta e_i)) - \sum_{t=1}^{T} \ell_t(x_t^*) \leq Regret_T^d(OGD, dG) + G\log(T)\big(1 + \frac{\sqrt{d}L\delta}{2} + \frac{D}{r}\big).$$

Proof. This is a modification of the proof of Theorem 1. Define $h_t(x) = \ell_t(x) + (\tilde{g}_t - \nabla\ell_t(x))^\intercal x$. Then since $\|\nabla h_t(x_t)\| \leq dG$. Hence from Lemma 1 for any sequence $\{x_t^*\}_{t=1}^T$,

$$\sum_{t=1}^{T} h_t(x_t) - h_t(x_t^*) \leq Regret_T^d(OGD, dG).$$

Then we proceed as in the proof of Theorem 1, use $\|\tilde{g}_t - \nabla\ell_t(x_t)\| \leq \frac{\sqrt{d}L\delta}{2}$. Apply Lemma 2 and plug in our parameters and this gives us the result.

□

5 Experiments

In this section we present empirical results based on our implementation the algorithm from the previous section (see appendix for the python code). As proved in the previous section, the algorithm should succeed with high probability when $n \sim \Omega(m^2)$ so we tested values for n ranging between $n = m^{1.5}$ to $m^{2.5}$ for each m between 5 and 50. For $m \in [50, 100]$ we ran experiments for $n = m^{2.1}$ to $m^{2.5}$ Note that the experimental values for m were not evenly spaced apart. Each pair of n, m was run for 100 trials, each trial consisting of generating an election uniformly at random, picking a random candidate to test, and running on that instance. A success was when returned "definitely", as in it was self-knowingly correct.

At 100 candidates, the 100 trials took 9 hours, while at 5 candidates, the trials took about 3 minutes.

Figure 1

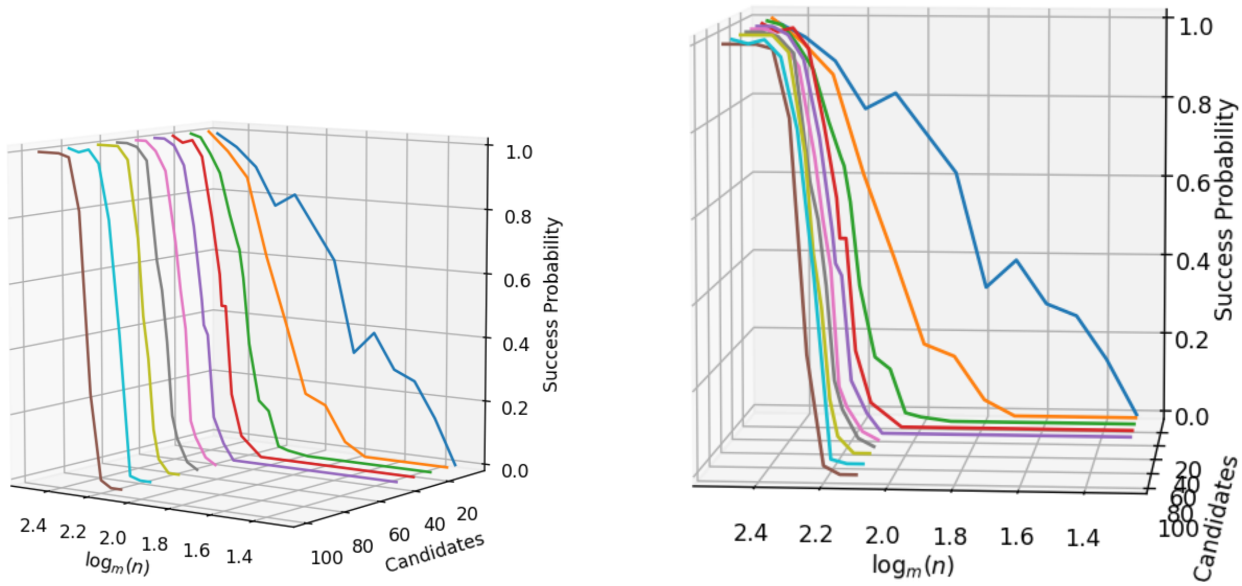

Figure 1: 3D line plot showing the success probability of the algorithm. Each line color corresponds to a particular number of candidates. 100 trials were run for each pair of n, m. Error bars were omitted to not clutter the graph. For reference they are at most 0.08.

References

Alekh Agarwal, Ofer Dekel, and Lin Xiao. Optimal algorithms for online convex optimization with multi-point bandit feedback. In *COLT*, pages 28–40. Citeseer, 2010.

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*, 2004.

Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7195–7201. IEEE, 2016.

A Code

A.1 Algorithm

A.2 Data Collection