Online Bandit Linear Optimization

An efficient Algorithm

V. Mullachery
S. Tiwari

Courant Institute of Mathematical Sciences

April 20, 2018

A Recap - Online Convex Optimization

- The player predicts a vector $\mathbf{x}_t \in \mathcal{K}$ at each time t
- The adversary (equivalently the environment) chooses a function $\mathbf{f}_t$ that is then passed to the learner.
- $\mathcal{K}$ is a convex compact set and $\mathbf{f}_t$ is a convex function of $\mathcal{K}$
- Loss of learner is $\mathbf{f}_t(\mathbf{x}_t)$
- Regret of algorithm $\mathcal{A}$ is

$$R_T(\mathcal{A}) = \sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x}_t) - \inf_{\mathbf{x}\in\mathcal{K}} \sum_{t=1}^{T} \mathbf{f}_t(\mathbf{x})$$

- Under mild and intuitively obvious assumptions, $O(\sqrt{T})$ regret guarantees are possible.

A Recap - Bandit Convex Optimization

1. Bandit Convex Optimization differs from Online Convex Optimization due to the lack of feedback available to the learner.
2. Learner receives the loss $\mathbf{f}_t(\mathbf{x}_t)$ but does not receive the function $\mathbf{f}_t$
3. Same formulation of regret, but now it is much harder to attain $O(\sqrt{T})$ guarantees.

Special Case - Bandit Linear Optimization

The convex functions that the adversary chooses are now linear functions. If we assume $\mathcal{K} \in \mathbb{R}^n$, then a linear function is nothing but a vector in $\mathbb{R}^n$ as well. We say that the adversary chooses $\mathbf{f}_t \in \mathbb{R}^n$ and that the learner receives loss $\mathbf{f}_t^T \mathbf{x}_t$. We place the mild and intuitively obvious condition that $|\mathbf{f}_t^T \mathbf{x}_t| \leq 1 \ \forall \mathbf{x}_t \in \mathcal{K}$.
Note: removal of this condition results in a situation where the adversary can chooose ever larger vectors $\mathbf{f}_t$ to inflict linear regret on any arbitrary learning algorithm.

Main Application - Online Shortest Path

1. This is the canonical motivation for Bandit Linear Optimization algorithms.
2. $G = (V, E)$ is a graph, with $s, t \in V$ as the source-sink pair, and with $|E| = n$.
3. Learner is looking for a path from s to t, and the adversary chooses the time required to traverse every edge.
4. Learner is only given the final travel time and no other information.

Online Shortest Path = Multi-Armed Bandit?

1. The set of all possible paths may be exponential in n (the number of edges), and thus, while this is technically a multi-armed bandit problem, there are far too many arms.
2. Problem is better formulated in $\mathbb{R}^{|E|} = \mathbb{R}^n$. A path is thus a vector in $\mathbb{R}^n$ with each coordinate either 0 or 1.
3. Let $\mathcal{K}$ be the convex hull of all possible path vectors. This is well known to be the *set of all flows* in a Graph.
4. Our mild assumption on $\mathbf{f}_t$ corresponds to the requirement that no path takes longer than 1 time unit to traverse.

Intuition

Since the problem setting is adversarial, the learner must be probabilistic. Follow The Leader, for instance, suffers linear regret at the hands of an adaptive adversary.
Usually, an optimum point $\mathbf{x}_t$ is evaluated, and then $\mathbf{y}_t$, a perturbation of $\mathbf{x}_t$, is played. This perturbation is also employed to evaluate $\tilde{\mathbf{f}}_t$, an estimate of $\mathbf{f}_t$. The learner proceeds to evaluate $\mathbf{x}_{t+1}$ using this, and prior estimates.

Intuition

$\tilde{\mathbf{f}}_t$ are single-point gradient estimates, and are random variables such that:

$$\mathbb{E}[\tilde{\mathbf{f}}_t] = \mathbf{f}_t$$

Since the adversary's $\mathbf{f}_t$ are not only convex but in fact linear, these estimates work very well. This idea transforms the bandit setting into a full information setting.

What can go wrong? Earlier Attempts

Projection methods, that are special cases of mirror descent, require one to work with a $\mathcal{K}_\delta \subset \mathcal{K}$. The points $\mathbf{x}_t$ are necessarily in $\mathcal{K}_\delta$ and a perturbation of size δ results in $\mathbf{y}_t$ in $\mathcal{K}$.
The problem arises because the next $\mathbf{x}_{t+1}$ is a projection onto $\mathcal{K}_\delta$.
With δ, η as hyper-parameters, BanditPGD obtains the regret:

$$R_T(\text{BanditPGD}) = \frac{C_1}{\eta} + \frac{C_2 \eta T}{\delta^2} + C_3 T \delta$$

for constants C_1, C_2, C_3.

What can go wrong? Earlier Attempts

Since the optimal point $\mathbf{x}^*$ is guaranteed to be on the boundary of $\mathcal{K}$ (due to the linearity of $\mathbf{f}_t$), and since projection requires $\mathbf{x}_t \in \mathcal{K}_\delta$, we suffer a $O(\delta T)$ cost by staying away from the boundary.

The $O(\frac{T}{\delta^2})$ appears due to the $\mathbb{E}[\tilde{\mathbf{f}_t^T}(\mathbf{x}_t - \mathbf{x}^*)]$ term. Using Cauchy-Schwarz to bound this, one encounters $\mathbb{E}|\tilde{\mathbf{f}_t}|^2 = O(\frac{1}{\delta^2})$

The result is a suboptimal $O(T^{\frac{3}{4}})$ regret guarantee.

What can go wrong? Earlier Attempts

Dani, Hayes and Kakade (Insert citation of some form) modified Auer's EXP3 algorithm for multi-armed adversarial bandits and achieved $O(n^3\sqrt{T})$ regret[1].
While regret is optimal in time, it does not admit an efficient implementation, with a computational complexity that is exponential in n.
Such computational costs are impractical for applications to Online Shortest Path - recall that n represents the number of edges in the graph.

[1] http://papers.nips.cc/paper/3371-the-price-of-bandit-information-for-online-optimization.pdf

Enter: BanditFTRL

1. Uses superior barrier functions to deal with the geometry of $\mathcal{K}$, instead of projections.
2. Since the cost of projection is removed from regret analysis, expected regret is $O(\sqrt{T})$ and not $O(T^{\frac{3}{4}})$.
3. Finding the argmin of a strongly convex function is the most computationally challenging task at each round. If one were to modify the algorithm and replace the argmin step by a single iteration of the Damped Newton method, the expected regret provably enjoys the same asymptotics.
4. The Damped Newton step allows for an implementation of this algorithm (and hence a solution to the Online Shortest Path problem) with $O(n^3 T)$ computational complexity.

Ingredients - I

- The full-information Follow The Regularized Leader (FTRL) algorithm is the underlying algorithm.
- A regularization function $\mathcal{R} : \mathcal{K} \to \mathbb{R}$
- Specific regularization functions yield well-known algorithms.
- If $\mathcal{R} : \mathbf{x} \mapsto \|\mathbf{x}\|^2$ is chosen, FTRL results in Online Gradient Descent algorithm. Similarly, choosing entropy function gives the Exponentiated Gradient algorithm.
- Thus, the first ingredient of the algorithm is an appropriate $\mathcal{R}$. A self-concordant barrier function is used.

Ingredients - II

- To tranform the bandit problem to the full-information case, we use single-point gradient estimates $\tilde{\mathbf{f}}_t$.
- This ingredient of the algorithm is straightforward.
- As observed earlier with BanditPGD, $\mathbb{E}\|\tilde{\mathbf{f}}_t\|^2$ is the troublesome term.
- This variance term is avoided by considering local norms instead of the standard Euclidean norm

Ingredients - III

- The final ingredient is a sampling scheme.
- Like in BanditPGD, the learner never plays $\mathbf{x}_t$, but rather plays a nearby point $\mathbf{y}_t$.
- BanditPGD samples according to a sphere of radius δ
- Instead, we use a sampling scheme based on the Dikin ellipsoid.

So what is a Self-Concordant Barrier?

$\mathcal{R} : \text{int}(\mathcal{K}) \to \mathbb{R}$ is a self-concordant function if for any $\mathbf{h} \in \mathbb{R}^n$, we have:

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

Additionally, $\lim_{\mathbf{x}\to\delta\mathcal{K}} \mathcal{R}(\mathbf{x}) = \infty$ (compare to Legendre-type functions from Mirror Descent lecture)

A self-concordant $\mathcal{R}$ is a ϑ-self-concordant barrier if $\forall \mathbf{h} \in \mathbb{R}^n$:

$$|D\mathcal{R}(\mathbf{x})[\mathbf{h}]| \leq \sqrt{\vartheta D^2\mathcal{R}(\mathbf{x})[\mathbf{h},\mathbf{h}]}$$

So what is the Dikin Ellipsoid?

The Hessian of $\mathcal{R}$ is a positive definite symmetric matrix, so that for any point $\mathbf{x} \in \text{int}(\mathcal{K})$, we can define

$$\langle \mathbf{y}_1, \mathbf{y}_2 \rangle_{\mathbf{x}} = \mathbf{y}_1^T \nabla^2 \mathcal{R}(\mathbf{x}) \mathbf{y}_2$$

This inner product gives a *local norm* at $\mathbf{x}$, denoted by $\|\|_{\mathbf{x}}$.
The Dikin Ellipsoid (of radius 1) at $\mathbf{x}$ is given by

$$W_1(\mathbf{x}) = \{\mathbf{y} \in \mathcal{K} : \|\mathbf{y} - \mathbf{x}\|_{\mathbf{x}} < 1\}$$

The Dikin ellipsoid is therefore nothing but an open ball corresponding to a specific inner product on $\mathbb{R}^n$.

Why do we use Dikin ellipsoids?

Important property: for any interior point x of $\mathcal{K}$, the Dikin ellipsoid at $\mathbf{x}$ is also contained within the interior of $\mathcal{K}$. This is a consequence of the fact that $\mathcal{R}$ is a *barrier* function - the Dikin ellipsoids get flatter as one approaches the boundary of our set.
Thus, by sampling points in the Dikin ellipsoid, one does not need to perform any projections in the algorithm.

Algorithm 1 SCRiBLe (Self-Concordant Regularization in Bandit Learning)

Input : $\eta > 0$ ϑ-self-concordant barrier $\mathcal{R}$
Let $\mathbf{x}_1 = \text{argmin}_{\mathbf{x}\in\mathcal{K}}[\mathcal{R}(\mathbf{x})]$
for $t = 1$ to T **do**
    Compute $\{\mathbf{e_1} \cdots \mathbf{e_n}\}, \{\lambda_1 \cdots \lambda_n\}$ eigenvectors and eigenvalues of $\nabla^2\mathcal{R}(\mathbf{x}_t)$
    Choose i uniformly at random from $\{1 \cdots n\}$ and $\varepsilon = \pm 1$ with probability $1/2$
    Predict $\mathbf{y}_t = \mathbf{x}_t + \varepsilon\lambda_i^{-1/2}\mathbf{e}_i$
    Observe the cost $\mathbf{f}_t^T\mathbf{y}_t \in \mathbb{R}$
    Define $\tilde{\mathbf{f}}_t := \eta(\mathbf{f}_t^T\mathbf{y}_t)\varepsilon\lambda_i^{1/2}\mathbf{e}_i$
    Update $\mathbf{x}_{t+1} = \text{argmin}_{\mathbf{x}\in\mathcal{K}}[\eta\sum_{s=1}^{t}\tilde{\mathbf{f}}_s^T\mathbf{x} + \mathcal{R}(\mathbf{x})]$
end for

The explore-exploit trade-off

How is the explore-exploit trade-off encoded in this algorithm?

1. As $\eta \sum_{i=1}^{t} \tilde{\mathbf{f}}_t$ grows in norm, the effect of the self-concordant barrier diminishes. Since $\mathbf{x}_{t+1}$ is chosen as the argmin of:

$$\eta \sum_{i=1}^{t} \tilde{\mathbf{f}}_t \mathbf{x} + \mathcal{R}(x)$$

2. Via the geometry of the Dikin ellipsoid: As $\mathbf{x}_t$ approaches an edge of $\mathcal{K}$, the Dikin ellipsoid at $\mathbf{x}_t$ becomes flatter (illustration would be nice) and orients itself along the edge. Consequently, the sampled $\mathbf{y}_t$ is less likely to deviate from $\mathbf{x}_t$ in this direction.

Lemma (Bandit Reduction Lemma)

Assume we are given any full information algorithm $\mathcal{A}$ and unbiased sampling and estimating schemes ***sampler, guesser****. If we let the associated Bandit algorithm be*
$\mathcal{A}' = BanditReduction(\mathcal{A}, \textbf{sampler, guesser})$*, then the expected regret of the (randomized) algorithm $\mathcal{A}'$ on the fixed sequence $\{\mathbf{f}_t\}$ is equal to the expected regret of the (deterministic) algorithm $\mathcal{A}$ on the random sequence $\{\tilde{\mathbf{f}}_t\}$.*

$$\mathbb{E}[Regret^{\mathbf{u}}(\mathcal{A}; \mathbf{f}_1, \mathbf{f}_2 \dots \mathbf{f}_T] = \mathbb{E}[Regret^{\mathbf{u}}(\mathcal{A}'; \tilde{\mathbf{f}}_1, \tilde{\mathbf{f}}_2, \dots \tilde{\mathbf{f}}_T)]$$

Lemma (Regret Bound Lemma)

Assume that $\eta\|\mathbf{f}_t\|^*_{\mathbf{x}_t} \leq \frac{1}{4}$ *and that* $\mathcal{R}$ *is a self-concordant barrier with* $\min \mathcal{R}(\mathbf{x}) = 0$. *Then for any* $\mathbf{u} \in \mathcal{K}$,

$$\mathit{Regret}^{\mathbf{u}}(\mathit{FTRL}(\mathcal{R}, \mathbf{f}_{1:t})) \leq 2\eta \sum_{t=1}^{T} \|\mathbf{f}_t\|^{*2}_{\mathbf{x}_t} + \eta^{-1}\mathcal{R}(\mathbf{u})$$

Theorem (Main)

Let $\mathcal{K}$ be a compact convex set $\subset \mathbb{R}^n$, and $\mathcal{R}$ be a ϑ-self-concordant barrier on $\mathcal{K}$. Assume $|\mathbf{f}_t^T \mathbf{x}| \leq L$ for any $\mathbf{x} \in \mathcal{K}$ and any t. Setting $\eta = \sqrt{\frac{\vartheta \log T}{2n^2 L^2 T}}$, the regret of SCRiBLe is bounded as $\mathbb{E}[Regret^u(SCRiBLe; \mathbf{f}_1 \cdots \mathbf{f}_T)] \leq nL\sqrt{8\vartheta T \log T} + 2L$ whenever $\frac{T}{\log T} > 8\vartheta$

Proof.

By Bandit Reduction lemma, we can write the regret as: $\mathbb{E}[Regret^u(\mathcal{A}; \mathbf{f}_1 \cdots \mathbf{f}_T)] = \mathbb{E}[Regret^u(FTRL_{\mathcal{R}}; \widetilde{\mathbf{f}}_1 \cdots \widetilde{\mathbf{f}}_T)]$ Now applying Regret Bound theorem:

$$
\begin{aligned}
\eta \, ||\mathbf{f}_t||^*_{\mathbf{x}_t} &= \eta \sqrt{\widetilde{\mathbf{f}}_t^T \nabla^{-2} \mathcal{R}(\mathbf{x}_t) \widetilde{\mathbf{f}}_t} \\
&= \eta n |\mathbf{f}_t^T \mathbf{y}_t| \sqrt{\lambda_i \mathbf{e}_i \nabla^{-2} \mathcal{R}(\mathbf{x}_t) \mathbf{e}_i} \\
&= \eta n |\widetilde{\mathbf{f}}_t \mathbf{y}_t| \\
&\leq \eta n L \\
&\leq nL \sqrt{\frac{\vartheta \log T}{2n^2 L^2 T}} \\
&\leq \frac{1}{4} && (\tfrac{T}{\log T} > 8\vartheta)
\end{aligned}
$$

Proof (Cont.)

Thus, $||\mathbf{f}_t||_{\mathbf{x}_t}^{*2} \leq n^2 L^2$. So $\forall \mathbf{u} \in \mathcal{K}$,

$$\mathbb{E}[Regret^{\mathbf{u}}(FTRL_{\mathcal{R}}; \tilde{\mathbf{f}}_1 \cdots \tilde{\mathbf{f}}_T)] \leq 2\eta \mathbb{E}[\sum_{t=1}^{T} \left|\left|\tilde{\mathbf{f}}_t\right|\right|_{\mathbf{x}_t}^{*2}] + \eta^{-1}\mathcal{R}(\mathbf{u})$$
$$\leq 2\eta n^2 L^2 T + \eta^{-1}\mathcal{R}(\mathbf{u})$$

If $\pi_{\mathbf{x}_1}(\mathbf{u}) \leq 1 - 1/T$, $\mathcal{R}(\mathbf{u}) \leq \vartheta \log T$. Even otherwise, the regret is bounded by $\leq 2\eta n^2 L^2 T + \vartheta \eta^{-1} \log T + 2L$ □