Online Convex Optimization in the Bandit Setting

Suyun Liu

OptML Group Meeting, Lehigh ISE

October 30, 2019

Presentation Outline

Recap: Online Convex Optimization

- At each iteration t, the player chooses x_t in convex set $\mathcal{K}$.
- A convex loss function $f_t \in \mathcal{F} : \mathcal{K} \to \mathbb{R}$ is revealed.
- A cost $f_t(x_t)$ is incurred.
  - $\mathcal{F}$ is a set of bounded functions.
  - f_t is revealed after choosing x_t.
  - f_t can be adversarially chosen.

Recap: Online Convex Optimization

Goal: minimize the regret bound

$$\text{regret}_T = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{K}} \sum_{t=1}^{T} f_t(x)$$

Online Gradient Descent (OGD) (Zinkevich 2003):

$$x_{k+1} = \Pi_{\mathcal{K}}(x_k - \eta_t \nabla f_t(x_t))$$

Regret bound

- if f_t is convex: $O(GD\sqrt{T})$
- if f_t is α-strongly convex: $O(\frac{G^2}{2\alpha}(1 + \log(T)))$

Motivation

- In Ad-placement, the search engine can inspect which ads were clicked through, but cannot know whether different ads would have been click through or not.
- Given a fixed budget, how to allocate resources among the research projects whose outcome is only partially known at the time of allocation and may change through time.

Bandit Convex Optimization

Motivation

- In Ad-placement, the search engine can inspect which ads were clicked through, but cannot know whether different ads would have been click through or not.
- Given a fixed budget, how to allocate resources among the research projects whose outcome is only partially known at the time of allocation and may change through time.

Bandit Setting

- In OCO, player has access to $\nabla f_t(x_t)$
- In BCO, player only has black-box access to the function value $f_t(x_t)$. We only can evaluate each function once.

Exploration vs Exploitation

Balance between exploiting the gathered information and exploring the new data.

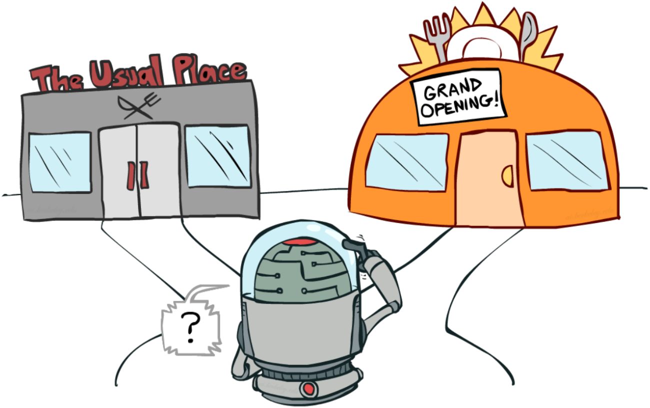

Figure: Where to eat?(Image source: UC Berkeley AI course slide, lecture 11.)

OGD without a gradient

Question: Can we perform OGD without gradients?

- One dim

$$\tilde{\nabla} f(x) \; = \; (f(x+\delta) - f(x-\delta))/2\delta$$

OGD without a gradient

Question: Can we perform OGD without gradients?

- One dim

$$\tilde{\nabla} f(x) = (f(x+\delta) - f(x-\delta))/2\delta$$

- d dim

$$\begin{aligned}\tilde{\nabla} f(x) &\approx \mathbb{E}_{u\in\partial\mathbb{B}}[(f(x+\delta u) - f(x))u]d/\delta \\ &= \mathbb{E}_{u\in\partial\mathbb{B}}[f(x+\delta u)u]d/\delta\end{aligned}$$

Note: $\tilde{g}(x,u) = f(x+\delta u)ud/\delta$

$$\mathbb{E}_{u\in\partial\mathbb{B}}[\tilde{g}(x,u)] = \nabla \hat{f}(x), \quad \text{with } \hat{f}(x) = \mathbb{E}_{v\in\mathbb{B}}[f(x+\delta v)]$$

Bandit gradient descent algorithm

Assumption:

- only access to f_t at one single point x_t.
- function value is bounded, $\{f_t\} : \mathcal{K} \to [-C, C]$.
- f_t can be non-smooth, no bounded gradient assumption.
- $\exists r, R > 0, r\mathbb{B} \subset \mathcal{K} \subset R\mathbb{B}$.

Bandit gradient descent algorithm

Assumption:

- only access to f_t at one single point x_t.
- function value is bounded, $\{f_t\} : \mathcal{K} \to [-C, C]$.
- f_t can be non-smooth, no bounded gradient assumption.
- $\exists r, R > 0, r\mathbb{B} \subset \mathcal{K} \subset R\mathbb{B}$.

Algorithm (Flaxman et al. 2005)

- Let $y_1 = 0$, learning rate η, $\xi \in (0,1), \delta > 0$
- for $t = 1, \ldots, T$:
  - select $u_t \in \partial\mathbb{B}$ uniformly at random
  - $x_t = y_t + \delta u_t$ and receive $f_t(x_t)$
  - $y_{t+1} = \Pi_{(1-\xi)\mathcal{K}}(y_t - \eta f_t(x_t) u_t d/\delta)$
    ($y_{t+1} \in (1-\xi)\mathcal{K}$ ensures $x_t \in \mathcal{K}$ for any $\delta \in [0, \xi r]$)

Bandit gradient descent algorithm

Theorem

For sufficient large T with $\eta = \frac{R}{C\sqrt{T}}$, the expected regret bound is

$$\mathbb{E}[\sum_{t=1}^{T} f_t(x_t)] - \min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x) \leq 6T^{5/6}dC$$

With additional assumption L-Lipschitz function

$$\mathbb{E}[\sum_{t=1}^{T} f_t(x_t)] - \min_{x\in\mathcal{K}} \sum_{t=1}^{T} f_t(x) \leq 6T^{3/4}d(\sqrt{CLR} + C)$$

Parameters: $T > (\frac{3Rd}{2r})^2$, $\delta = (\frac{rR^2d^2}{12T})^{1/3} \leq \xi r$, and $\xi = (\frac{3Rd}{2r\sqrt{T}})^{1/3}$

Multi-Point Bandit Feedback

Recall

$$\tilde{g}_t = \frac{d}{\delta} f_t(u_t) u_t \text{ with } \|\tilde{g}_t\| \leq \frac{dC}{\color{red}{\delta}}$$

Multi-Point Bandit Feedback

Recall

$$\tilde{g}_t = \frac{d}{\delta} f_t(u_t) u_t \text{ with } \|\tilde{g}_t\| \leq \frac{dC}{\delta}$$

Multi-point scheme (Agarwal et al. 2010): use two function values to construct bounded norm gradient estimators for L-Lipschitz continuous functions.

$$\tilde{g}_t = \frac{d}{2\delta}(f_t(x_t + \delta u_t) - f_t(x_t - \delta u_t)) u_t \text{ with } \|\tilde{g}_t\| \leq Ld$$

Multi-Point Bandit Feedback

Recall

$$\tilde{g}_t = \frac{d}{\delta} f_t(u_t) u_t \text{ with } \|\tilde{g}_t\| \leq \frac{dC}{\delta}$$

Multi-point scheme (Agarwal et al. 2010): use two function values to construct bounded norm gradient estimators for L-Lipschitz continuous functions.

$$\tilde{g}_t = \frac{d}{2\delta}(f_t(x_t + \delta u_t) - f_t(x_t - \delta u_t)) u_t \text{ with } \|\tilde{g}_t\| \leq Ld$$

Expected regret bound:

- $\eta = \frac{1}{\sqrt{T}}, \delta = \frac{\log(T)}{T}$ and $\xi = \frac{\delta}{r}$: $(d^2 L^2 + R^2)\sqrt{T} + L\log(T)(3 + \frac{R}{r})$
- α-strong convex, $\eta_t = \frac{1}{\alpha t}, \delta = \frac{\log(T)}{T}$ and $\xi = \frac{\delta}{r}$: $L\log(T)(\frac{d^2 L}{\alpha} + \frac{R}{r} + 3)$.

Summary on regret bounds

Setting	Convex	Linear	Smooth	Str.-Convex	Str.-Convex & Smooth
Full-Info.	$\Theta(\sqrt{T})$			$\Theta(\log T)$	
BCO	$\tilde{O}(T^{3/4})$	$\tilde{O}(\sqrt{T})$	$\tilde{O}(T^{2/3})$		$\tilde{O}(\sqrt{T})$ [Thm. 10]
	$\Omega(\sqrt{T})$				

Figure: Known regret bounds in the Full-Info./BCO setting (Hazan and Levy 2014)

Multi-Armed Bandit

Setting

- At iteration t, player chooses action i_t from a set of discrete actions $\{1, \ldots, n\}$.
- A loss in $[0, 1]$ is independently chosen for each action.
- The loss associated with i_t is revealed.
- Various assumptions and constraints.

Multi-Armed Bandit

Setting

- At iteration t, player chooses action i_t from a set of discrete actions $\{1, \ldots, n\}$.
- A loss in $[0, 1]$ is independently chosen for each action.
- The loss associated with i_t is revealed.
- Various assumptions and constraints.

Example

A gambler pulls one of n slot machines to receive a reward or payoff. Each arm is configured with fixed unknown reward/payoff probability.

What is the best strategy to achieve highest long-term rewards/lowest cumulative loss?

Multi-Armed Bandit

Exploration vs Exploitation: explore more actions or make the best decision using the current estimates of the loss distribution.

Algorithms

- Simple MAB algorithm
- EXP3

Multi-Armed Bandit

Exploration vs Exploitation: explore more actions or make the best decision using the current estimates of the loss distribution.

Algorithms

- Simple MAB algorithm
- EXP3

Let $\mathcal{K} = \Delta_n$ be an n-dimensional simplex. The linear loss function

$$f_t(x_t) = \ell_t^\top x_t = \sum_{i=1}^{n} \ell_t(i) x_t(i) \qquad \forall x_t \in \mathcal{K}$$

Key: to estimate gradient ℓ_t.

Simple MAB algorithm

Separating exploration and exploitation steps (Hazan 2016)

Algorithm 1 Simple MAB algorithm

$\epsilon \in [0,1]$, learning rate $\eta > 0$.
for $t = 1, \ldots, T$ **do**
    $b_t \sim$ Bernoulli(ϵ).
    if $b_t = 1$ **then**
        Choose i_t uniformly at random and receive $\ell_t(i_t)$
        Let
$$\hat{\ell}_t(i) = \begin{cases} n/\epsilon \ell_t(i_t), & \text{for } i = i_t \\ 0, & \text{OW} \end{cases}$$
        $x_{t+1} = \Pi_{\mathcal{K}}(x_t - \eta \hat{\ell}_t)$
    else
        Play $i_t \sim x_t$
        $\hat{\ell}_t = 0, x_{t+1} = x_t$.

- $\mathbb{E}[\hat{\ell}_t] = \ell_t$ and $\mathbb{E}[\hat{f}_t(x_t)] = \mathbb{E}[\hat{\ell}_t^\top x_t] = f_t(x_t)$
- Expected regret bound when $\epsilon = n^{2/3}T^{-1/3}$

$$\mathbb{E}[\sum_{t=1}^{T} \ell_t(i_t)] - \min_i \sum_{t=1}^{T} \ell_t(i) \leq O(T^{2/3} n^{2/3})$$

EXP3

Combining exploration and exploitation steps (Auer et al. 2002b).

Algorithm 2 EXP3 - simple version

1: Choose $\epsilon > 0, x_1 = [1/n, \ldots, 1/n]$.
2: **for** $t = 1, \ldots, T$ **do**
3: Choose $i_t \sim x_t$ and receive $\ell_t(i_t)$.
4: Let

$$\hat{\ell}_t(i) = \begin{cases} \frac{\ell_t(i_t)}{x_t(i_t)}, & \text{for } i = i_t \\ 0, & \text{OW} \end{cases}$$

5: Update $y_{t+1}(i) = x_t(i)e^{-\epsilon\hat{\ell}_t(i)}$, $x_{t+1} = \frac{y_{t+1}}{\|y_{t+1}\|_1}$

- $\mathbb{E}[\hat{\ell}_t] = \ell_t$
- Choose $\epsilon = \sqrt{\frac{\log n}{Tn}}$, expected regret bound $O(\sqrt{Tn\log n})$

Stochastic Multi-armed Bandit

Setting

- Player chooses $i_t \in \{1, \ldots, n\}$.
- Each action i_t has a reward r_{i_t} from a (fixed) probability distribution $\mathbb{P}_{i_t}$ with mean μ_{i_t}.
- The reward revealed to the player is a sample taken from $\mathbb{P}_{i_t}$.

A sub case: Bernoulli Multi-armed Bandit with $\mathbb{P}_i = \text{Bernoulli}(p_i)$, $r_i \in \{0, 1\}$.

General Bernoulli Multi-armed Bandit Algorithm

Algorithm 3 Bernoulli Multi-armed Bandit

Set $N = Q = S = F = 0 \in \mathbb{R}^n$.
for $t = 1, \ldots, T$ **do**
    $i_t =$ PickArm(Q, N, S, F)
    $r_t =$ BernoulliReward(i_t)
    $N[i_t] = N[i_t] + 1$ (number of times arm i is pulled)
    $Q[i_t] = Q[i_t] + \frac{(r_t - Q[i_t])}{N[i_t]}$ (empirical average reward of pulling i)
    $S[i_t] = S[i_t] + r_t$ (number of times a reward of 1 was received)
    $F[i_t] = F[i_t] + (1 - r_t)$(number of times a reward of 0 was received)

Arm Seclection Algorithms for Stochastic MAB

- Random selection
- ϵ-Greedy algorithm
- Boltzmann Exploration
- Upper Confidence Bounds
- Bayesian UCB
- Thompson Sampling
- ...

Upper Confidence Bound Arm selection

Using one sided Hoeffding's inequality

$$\mathbb{P}(\mu_i \geq Q[i] + \epsilon) \leq e^{-2N[i]\epsilon^2}$$

UCB strategy

$$i = \mathrm{argmax}_i (Q[i] + \epsilon), \text{ where } \epsilon = \sqrt{\frac{2\log(t)}{N[i]}}$$

Expected regret bound: $O(\log(T))$ (Auer et al. 2002a)

Thompson Sampling Strategy

Beta distribution $\text{Beta}(\alpha, \beta)$

$$f(x; \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

Thompson Sampling algorithm:

- Initialize $p_i \sim \text{Beta}(1, 1), \forall i$
- **for** $t = 1, \ldots, T$

$$Q[i] \sim \text{Beta}(S[i]+1, F[i]+1), \forall i$$

$$i_t = \text{argmax}_i \{Q[i]\}$$

Thompson Sampling Strategy

Beta distribution Beta(α, β)

$$f(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

Thompson Sampling algorithm:

- Initialize $p_i \sim$ Beta$(1, 1), \forall i$
- **for** $t = 1, \ldots, T$

$$Q[i] \sim \text{Beta}(S[i] + 1, F[i] + 1), \forall i$$

$$i_t = \text{argmax}_i \{Q[i]\}$$

Expected regret bound: $O(\log(T))$ (Agrawal and Goyal 2012)

Generalize to $\tilde{r} \in [0, 1]$: after observing reward $\tilde{r}_t$, perform $r_t \sim$ BernoulliReward$(\tilde{r}_t)$

References

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

Agrawal, S. and Goyal, N. (2012). Analysis of thompson sampling for the multi-armed bandit problem. In Conference on Learning Theory, volume 23, pages 1–26.

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. Machine learning, 47:235–256.

Auer, P., Cesa-Bianchi, N., Freund, Y., and Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. SIAM journal on computing, 32:48–77.

Flaxman, A. D., Kalai, A. T., Kalai, A. T., and McMahan, H. B. (2005). Online convex optimization in the bandit setting: gradient descent without a gradient. In Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms, pages 385–394. Society for Industrial and Applied Mathematics.

Hazan, E. (2016). Introduction to online convex optimization. Foundations and Trends® in Optimization, 2:157–325.

Hazan, E. and Levy, K. (2014). Bandit convex optimization: Towards tight bounds. In Advances in Neural Information Processing Systems, pages 784–792.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In Proceedings of the 20th International Conference on Machine Learning (ICML-03), pages 928–936.