Adversarial Bandits

CS245: Online Optimization and Learning

Xin Liu
SIST, ShanghaiTech University

Review of Online Learning with Full Information

Online Learning with Full Information

Initialization: $x_1 \in \mathcal{K}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(\cdot)$.
- **Update:** $x_{t+1} = \mathrm{Alg}(f_1, f_2, \cdots, f_t)$.

Online learning with full information:

- We know the complete information of loss functions $f_t(\cdot)$.
- We studied OMD and FTRL and obtain $O(\sqrt{T})$ regret.
- We studied some variants such as online learning with the prediction and delayed feedback, which can be addressed with "Optimistic FTRL".

Online Learning with Bandit Feedback

Online Learning with Bandit Feedback

Initialization: $x_1 \in \mathcal{K}$.

For $t = 1, \cdots, T$:

- **Learner:** Submit x_t.
- **Environment:** Observe the convex loss $f_t(x_t)$.
- **Update:** $x_{t+1} = \mathrm{Alg}(f_1(x_1), \nabla \hat{f}_1(x_1), \cdots, f_t(x_t), \nabla \hat{f}_t(x_t))$.

Online learning with bandit feedback:

- We know the bandit information of loss functions at the decision point $f_t(x_t)$.
- We need to use these bandit feedback to estimate and the loss function or the gradient.

From Expert Problem to (Adversarial) Bandits problem

Expert problem:

Initialization: N experts/models.

For each day $t = 1, \cdots, T$:

- **Learner:** Obtain predictions from N experts/models and sample an expert i from a probability simplex x_t.
- **Environment:** Observe the loss of each model ℓ_t.

Bandit problem:

Initialization: K arms.

For each round $t = 1, \cdots, T$:

- **Learner:** Pull an arm $i \in [K]$.
- **Environment:** Observe the reward of the arm $r_t(i)$.

(Adversarial) Bandits problem

Stochastic Bandit problem:

Initialization: K arms.

For each round $t = 1, \cdots, T$:

- **Learner:** Pull an arm $a_t \in [K]$.
- **Environment:** Observe the reward of the arm $r_t(a_t)$, which is stochastic from some unknown distribution.

Adversarial Bandit problem:

Initialization: K arms.

For each round $t = 1, \cdots, T$:

- **Learner:** Pull an arm $a_t \in [K]$.
- **Environment:** Observe the reward of the arm $r_t(a_t)$, which could be arbitrary and adversarial.

(Adversarial) Bandits problem

We define the regret of adversarial bandit given a sequence of actions $\{a_t\}$ by an algorithm

$$\text{Regret}(\{a_t\}) = \max_i \sum_{t=1}^{T} r_t(i) - \sum_{t=1}^{T} r_t(a_t).$$

The expected reward of an algorithm is

$$\text{Regret}(T) = \mathbb{E}\left[\max_i \sum_{t=1}^{T} r_t(i) - \sum_{t=1}^{T} r_t(a_t)\right].$$

Online Mirrored Descent for Expert Problem

Hedge as Online Mirrored Descent:

Initialization: $x_1 = [1/K, \cdots, 1/K]$ and η.

For each day $t = 1, \cdots, T$:

- **Learner:** Sample an expert i from x_t.
- **Environment:** Observe the full error ℓ_t.
- **Update:** $x_{t+1} = \arg\min_{\mathcal{K}} \ \langle x, \ell_t \rangle + \frac{1}{\eta} B_\psi(x; x_t)$.

Hedge $\longrightarrow$ Exponentiated Gradient $\longrightarrow$ OMD!

OMD is a strong and general framework to design online algorithms with full information. Can it be used to solve adversarial bandit problems?

Online Mirrored Descent for Adversarial Bandit Problems

Online Mirrored Descent for Adversarial Bandits:

Initialization: $x_1 = [1/K, \cdots, 1/K]$ and η.

For each day $t = 1, \cdots, T$:

- **Learner:** Sample an arm a_t from x_t.
- **Environment:** Observe the reward of arm a_t : $r_t(a_t)$.
- **Update:** $x_{t+1} = \arg\min_{\mathcal{K}} \ \langle x, -\hat{r}_t \rangle + \frac{1}{\eta} B_\psi(x; x_t)$.

As discussed, we only observed the reward of the selected arm i, which is arbitrary and adversarial.

In adversarial bandits, the reward is linear!

In OMD, we use the reward estimator of $\hat{r}_t$ to replace true reward or loss (r_t or ℓ_t). The estimator is super important!

Importance Estimator for Reward

The estimator $\hat{r}_t$ is super important! A naive way is to just consider what we have observed as the estimator

$$\hat{r}_t(i) = r_t(i), \quad \text{if } a_t = i.$$

Does it work?

Importance Estimator for Reward

The estimator $\hat{r}_t$ is super important! A naive way is to just consider what we have observed as the estimator

$$\hat{r}_t(i) = r_t(i), \quad \text{if } a_t = i.$$

Does it work?

Another possible way is to do the importance estimator:

$$\hat{r}_t(i) = \frac{r_t(i)}{x_t(i)}, \text{ if action } a_t = i.$$

or

$$\hat{r}_t(i) = \mathbb{I}(a_t = i)\frac{r_t(i)}{x_t(i)}.$$

Importance Estimator for Reward

Are the Importance Estimators unbiased?

What are the variances of the Importance Estimator?

Importance Estimator for Reward

We have two estimators:

$$\begin{aligned}\hat{r}_t(i) =& 1 - \frac{1 - r_t(i)}{x_t(i)}, \text{ if action } a_t = i,\\ \hat{r}_t(i) =& 1 - \mathbb{I}(a_t = i)\frac{1 - r_t(i)}{x_t(i)}.\end{aligned}$$

which one is unbiased? and why?

Online Mirrored Descent for Adversarial Bandit Problems

Online Mirrored Descent for Adversarial Bandits:

Initialization: $x_1 = [1/K, \cdots, 1/K]$ and η.

For each day $t = 1, \cdots, T$:

- **Learner:** Sample an arm i from x_t.
- **Environment:** Observe the reward of arm i : $r_t(i)$.
- **Reward Estimator:** $\hat{r}_t(i) = r_t(i)/x_t(i)$ and 0 otherwise.
- **Update:** $x_{t+1} = \arg\min_{\mathcal{K}} \ \langle x, -\hat{r}_t \rangle + \frac{1}{\eta} B_\psi(x; x_t)$.

OMD for adversarial bandit is quite straightforward: replace r_t with its unbiased estimator $\hat{r}_t$.

In adversarial bandits, it seems we only update x with each individual coordinate (arm).

B_ψ is KL divergence with ψ being the negative entropy.

Exp3 Algorithm

Exp3 Algorithm:

Initialization: $x_1 = [1/K, \cdots, 1/K]$ and η.

For each day $t = 1, \cdots, T$:

- **Learner:** Sample an arm i from x_t.
- **Environment:** Observe the reward $r_t(i)$.
- **Reward Estimator:** $\hat{r}_t(i) = r_t(i)/x_t(i)$ and 0 otherwise.
- **Update:** $x_{t+1,i} = e^{\eta \sum_{s=1}^{t} \hat{r}_s(i)} / \sum_i e^{\eta \sum_{s=1}^{t} \hat{r}_s(i)}$.

Exp3 represents "exponential-weight algorithm for exploration and exploitation".

Exp3 is very similar with exponential gradient except using the total estimated rewards $\sum_{s=1}^{t} \hat{r}_s(i), \forall i$.

Exp3 Algorithm – Regret and Possible Issue

Since Exp3 is viewed as OMD with bandit feedback, we could do the "reduction" from bandit to full feedback. Recall the regret of OMD with full information to be

Theorem 1 (OMD with Full Info)

Let ψ be the negative entropy function in B_ψ. Let fixed learning rate $\eta_t = \eta$. Online mirrored descent algorithm achieves

$$Regret(T) \leq \frac{\log K}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|r_t\|^2.$$

The results can be refined to be

$$\text{Regret}(T) \leq \frac{\log K}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|r_t\|_\infty^2.$$

which implies the regret is $O(\sqrt{T \log K})$.

Exp3 Algorithm – Regret and Possible Issue

Exp3 Algorithm – Regret and Refined Analysis

Exp3 is motivated by EG with full information and it is supposed to work! Indeed, we need a refined analysis.

Theorem 2

Suppose $\eta = \sqrt{K \log K / T}$. *Exp3 algorithm achieves the regret*

$$\begin{aligned} Regret(T) \leq & \frac{\log K}{\eta} + \frac{\eta}{2}\mathbb{E}\left[\sum_{t=1}^{T} \|r_t\|^2\right]. \\ = & O(\sqrt{TK \log K}). \end{aligned}$$

Exp3 returns the regret $O(\sqrt{T})$! Moreover, Exp3 with bandit feedback only has $O(\sqrt{K})$ loss because EG with full info $O(\sqrt{T \log K})$.

Exp3 Algorithm – Regret and Refined Analysis

For OMD, we have a local and strong version of regret analysis as follows.

Lemma 3

Let ψ be twice-differentiable convex function in B_ψ. Let fixed learning rate $\eta_t = \eta$. Online mirrored descent algorithm achieves

$$\begin{aligned}\langle x_t - x, \ell_t \rangle \leq & \frac{1}{\eta}(B(x, x_t) - B(x, x_{t+1})) \\ & + \frac{\eta}{2} \min\{\|\ell_t\|^2_{(\nabla\psi^2(z_t))^{-1}}, \|\ell_t\|^2_{(\nabla\psi^2(z'_t))^{-1}}\}.\end{aligned}$$

where z_t is between x_t and x_{t+1}; z'_t is between x_t and x'_{t+1} with $x'_{t+1} = \arg\min \ \langle x, \ell_t \rangle + \frac{1}{\eta} B_\psi(x; x_t)$.

The lemma can be proved by using Pushback Lemma.

Exp3 Algorithm – Regret and Refined Analysis