Online Learning

— Stochastic Multi-Armed Bandit (Stochastic MAB)

Joseph Chuang-Chieh Lin

Department of Computer Science & Information Engineering,
Tamkang University

Spring 2023

Credits for the resource

The slides are based on the lectures of Prof. Luca Trevisan:
https://lucatrevisan.github.io/40391/index.html

the lectures of Prof. Shipra Agrawal:
https://ieor8100.github.io/mab/

the lectures of Prof. Francesco Orabona:
https://parameterfree.com/lecture-notes-on-online-learning/
the monograph: https://arxiv.org/abs/1912.13213

and also Elad Hazan's textbook:
Introduction to Online Convex Optimization, 2nd Edition.

Outline

Outline

Online Convex Optimization

Goal: Design an algorithm such that

- At discrete time steps $t = 1, 2, \ldots$, output $\boldsymbol{x}_t \in \mathcal{K}$, for each t.
  - $\mathcal{K}$: a convex set of feasible solutions.
- After $\boldsymbol{x}_t$ is generated, a convex cost function $f_t : \mathcal{K} \mapsto \mathbb{R}$ is revealed.
- Then the algorithm suffers the loss $f_t(\boldsymbol{x}_t)$.

And we want to minimize the cost.

The difficulty

- The cost functions f_t is unknown before t.
- $f_1, f_2, \ldots, f_t, \ldots$ are not necessarily fixed.
    - Can be generated dynamically by an adversary.

What's the regret?

- The offline optimum: After T steps,

$$\min_{\boldsymbol{x} \in \mathcal{K}} \sum_{t=1}^{T} f_t(\boldsymbol{x}).$$

- The regret after T steps:

$$\mathrm{regret}_T = \sum_{t=1}^{T} f_t(\boldsymbol{x}_t) - \min_{\boldsymbol{x} \in \mathcal{K}} \sum_{t=1}^{T} f_t(\boldsymbol{x}).$$

What's the regret?

- The offline optimum: After T steps,

$$\min_{\boldsymbol{x} \in \mathcal{K}} \sum_{t=1}^{T} f_t(\boldsymbol{x}).$$

- The regret after T steps:

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

- The rescue: $\text{regret}_T \leq o(T)$.

What's the regret?

- The offline optimum: After T steps,

$$\min_{\boldsymbol{x}\in\mathcal{K}} \sum_{t=1}^{T} f_t(\boldsymbol{x}).$$

- The regret after T steps:

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

- The rescue: $\text{regret}_T \leq o(T)$. $\Rightarrow$ **No-Regret** in average when $T \to \infty$.
  - For example, $\text{regret}_T / T = \frac{\sqrt{T}}{T} \to 0$ when $T \to \infty$.

Multi-Armed Bandit

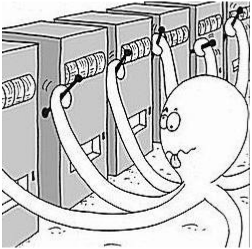


Fig.: Image credit: Microsoft Research

The setting

- We can see N arms as N experts.
- Arms give are independent.
- We can only pull an arm and observe the reward of it.
  - It's NOT possible to observe the reward of pulling the other arms...
- Each arm i has its own reward $r_i \in [0, 1]$.

The setting

- We can see N arms as N experts.
- Arms give are independent.
- We can only pull an arm and observe the reward of it.
  - It's NOT possible to observe the reward of pulling the other arms...
- Each arm i has its own reward $r_i \in [0, 1]$.
  - μ_i: the mean of reward of arm i
    - $\hat{\mu}_i$: the empirical mean of reward of arm i
  - μ^*: the mean of reward of the BEST arm.
  - $\Delta_i : \mu^* - \mu_i$.
  - Index of the best arm: $I^* := \arg\max_{i \in \{1,\dots,N\}} \mu_i$.
  - The associated highest expected reward: $\mu^* = \mu_{I^*}$.

The regret formulation for MAB

Let I_t be the arm played by the algorithm at time t.
The regret of the algorithm in T rounds is

$$\mathrm{regret}_T = \sum_{t=1}^{T}(\mu^* - \mu_{I_t})$$

The regret formulation for MAB

Let I_t be the arm played by the algorithm at time t.
The regret of the algorithm in T rounds is

$$\text{regret}_T = \sum_{t=1}^{T}(\mu^* - \mu_{I_t}) = \sum_{i=1}^{N}\sum_{t:I_t=i}(\mu^* - \mu_i)$$

The regret formulation for MAB

Let I_t be the arm played by the algorithm at time t.
The regret of the algorithm in T rounds is

$$\begin{aligned}\text{regret}_T &= \sum_{t=1}^{T}(\mu^* - \mu_{I_t}) = \sum_{i=1}^{N}\sum_{t:I_t=i}(\mu^* - \mu_i) \\ &= \sum_{i=1}^{N} n_{i,T}\Delta_i\end{aligned}$$

The regret formulation for MAB

Let I_t be the arm played by the algorithm at time t.
The regret of the algorithm in T rounds is

$$\begin{aligned}\text{regret}_T &= \sum_{t=1}^{T}(\mu^* - \mu_{I_t}) = \sum_{i=1}^{N}\sum_{t:I_t=i}(\mu^* - \mu_i)\\ &= \sum_{i=1}^{N} n_{i,T}\Delta_i\\ &= \sum_{i:\mu_i<\mu^*} n_{i,T}\Delta_i.\end{aligned}$$

Outline

A Naïve Greedy Algorithm

Greedy Algorithm

1. For $t \leq cN$, select a random arm with probability $1/N$ and pull it.
2. For $t > cN$, pull the arm $I_t := \arg\max_{i=1,\ldots,N} \hat{\mu}_{i,t}$.

- Here c is a constant.

A Naïve Greedy Algorithm

Greedy Algorithm

1. For $t \leq cN$, select a random arm with probability $1/N$ and pull it.
2. For $t > cN$, pull the arm $I_t := \arg\max_{i=1,\ldots,N} \hat{\mu}_{i,t}$.

- Here c is a constant.
- This algorithm is of linear regret, hence is not a no-regret algorithm.

A Naïve Greedy Algorithm

Greedy Algorithm

1. For $t \le cN$, select a random arm with probability $1/N$ and pull it.
2. For $t > cN$, pull the arm $I_t := \arg\max_{i=1,\ldots,N} \hat{\mu}_{i,t}$.

- Here c is a constant.
- This algorithm is of linear regret, hence is not a no-regret algorithm.
- For example,
  - Arm 1: $0/1$ reward with mean $3/4$.
  - Arm 2: Fixed reward of $1/4$.
  - After $cN = 2c$ steps, with constant probability, we have $\hat{\mu}_{1,cN} < \hat{\mu}_{2,cN}$.

A Naïve Greedy Algorithm

Greedy Algorithm

1. For $t \leq cN$, select a random arm with probability $1/N$ and pull it.
2. For $t > cN$, pull the arm $I_t := \arg\max_{i=1,\ldots,N} \hat{\mu}_{i,t}$.

- Here c is a constant.
- This algorithm is of linear regret, hence is not a no-regret algorithm.
- For example,
  - Arm 1: $0/1$ reward with mean $3/4$.
  - Arm 2: Fixed reward of $1/4$.
  - After $cN = 2c$ steps, with constant probability, we have $\hat{\mu}_{1,cN} < \hat{\mu}_{2,cN}$.
  - If this is the case, the algorithm will keep pulling arm 2 and will never change!

ϵ-Greedy Algorithm

ϵ-Greedy Algorithm

For all $t = 1, 2, \ldots, N$:

- With probability $1 - \epsilon$, pull arm $I_t := \arg\max_{i=1,\ldots,N} \hat{\mu}_{i,t}$.
- With probability ϵ, select an arm uniformly at random (i.e., each with probability $1/N$).

ϵ-Greedy Algorithm

ϵ-Greedy Algorithm

For all $t = 1, 2, \ldots, N$:

- With probability $1 - \epsilon$, pull arm $I_t := \arg\max_{i=1,\ldots,N} \hat{\mu}_{i,t}$.
- With probability ϵ, select an arm uniformly at random (i.e., each with probability $1/N$).

- It looks good.

ϵ-Greedy Algorithm

ϵ-Greedy Algorithm

For all $t = 1, 2, \ldots, N$:

- With probability $1 - \epsilon$, pull arm $I_t := \arg\max_{i=1,\ldots,N} \hat{\mu}_{i,t}$.
- With probability ϵ, select an arm uniformly at random (i.e., each with probability $1/N$).

- It looks good.
- Unfortunately, this algorithm still incurs linear regret.

ϵ-Greedy Algorithm

ϵ-Greedy Algorithm

For all $t = 1, 2, \ldots, N$:

- With probability $1 - \epsilon$, pull arm $I_t := \arg\max_{i=1,\ldots,N} \hat{\mu}_{i,t}$.
- With probability ϵ, select an arm uniformly at random (i.e., each with probability $1/N$).

- It looks good.
- Unfortunately, this algorithm still incurs linear regret.
- Indeed,
  - Each arm is pulled in average $\epsilon T/N$ times.

ϵ-Greedy Algorithm

ϵ-Greedy Algorithm

For all $t = 1, 2, \ldots, N$:

- With probability $1 - \epsilon$, pull arm $I_t := \arg\max_{i=1,\ldots,N} \hat{\mu}_{i,t}$.
- With probability ϵ, select an arm uniformly at random (i.e., each with probability $1/N$).

- It looks good.
- Unfortunately, this algorithm still incurs linear regret.
- Indeed,
  - Each arm is pulled in average $\epsilon T/N$ times.
  - Hence the (expected) regret will be at least $\frac{\epsilon T}{N} \sum_{i:\mu_i<\mu^*} \Delta_i$.

Outline

1. Introduction
   - Online Learning
   - Regret
   - Multi-Armed Bandit
2. Solving the Stochastic Multi-Armed Bandit Problem
   - Greedy Algorithms
   - Upper Confidence Bound (UCB)
   - Time-Decay ϵ-Greedy

The upper confidence bound algorithm (UCB)

- At each time step (round), we simply pull the arm with the highest "empirical reward estimate + high-confidence interval size".
- The empirical reward estimate of arm i at time t:

$$\hat{\mu}_{i,t} = \frac{\sum_{s=1}^{t} I_{s,i} \cdot r_s}{n_{i,t}}.$$

  $n_{i,t}$: the number of times arm i is played.
  $I_{s,i}$: 1 if the choice of arm is i at time s and 0 otherwise.
- Reward estimate + confidence interval:

$$\mathrm{UCB}_{i,t} := \hat{\mu}_{i,t} + \sqrt{\frac{\ln t}{n_{i,t}}}.$$

Algorithm UCB

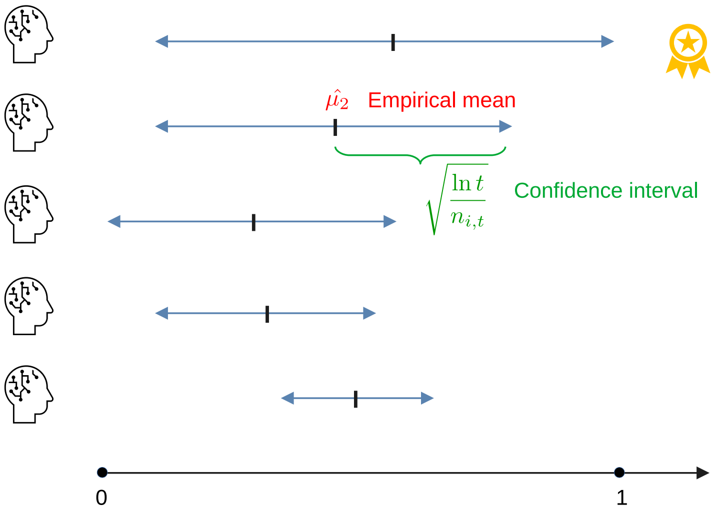
UCB Algorithm

N arms, T rounds such that $T \geq N$.

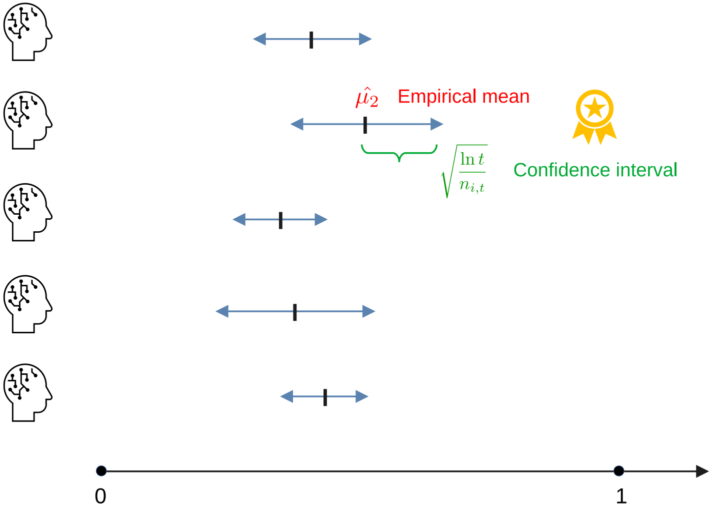
1. For $t = 1, \ldots, N$, play arm t.
2. For $t = N + 1, \ldots, T$, play arm

$$A_t = \arg\max_{i \in \{1,\ldots,N\}} \mathrm{UCB}_{i,t-1}.$$

Algorithm UCB



Algorithm UCB (after more time steps...)



From the Chernoff bound (proof skipped)

For each arm i at time t, we have

$$|\hat{\mu}_{i,t} - \mu_i| < \sqrt{\frac{\ln t}{n_{i,t}}}$$

with probability $\geq 1 - 2/t^2$.

Immediately, we know that

- with prob. $\geq 1 - 2/t^2$, $\mathrm{UCB}_{i,t} := \hat{\mu}_{i,t} + \sqrt{\frac{\ln t}{n_{i,t}}} > \mu_i$.
- with prob. $\geq 1 - 2/t^2$, $\hat{\mu}_{i,t} < \mu_i + \frac{\Delta_i}{2}$ when $n_{i,t} \geq \frac{4 \ln t}{\Delta_i^2}$.

From the Chernoff bound (proof skipped)

For each arm i at time t, we have

$$|\hat{\mu}_{i,t} - \mu_i| < \sqrt{\frac{\ln t}{n_{i,t}}}$$

with probability $\geq 1 - 2/t^2$.

To understand why, please take my Randomized Algorithms course. :)
Immediately, we know that

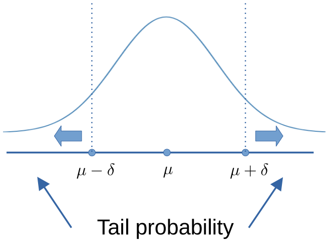
- with prob. $\geq 1 - 2/t^2$, $\mathrm{UCB}_{i,t} := \hat{\mu}_{i,t} + \sqrt{\frac{\ln t}{n_{i,t}}} > \mu_i$.
- with prob. $\geq 1 - 2/t^2$, $\hat{\mu}_{i,t} < \mu_i + \frac{\Delta_i}{2}$ when $n_{i,t} \geq \frac{4 \ln t}{\Delta_i^2}$.

Tail probability by the Chernoff/Hoeffding bound

The Chernoff/Hoeffding bound

For independent and identically distributed (i.i.d.) samples $x_1, \ldots, x_n \in [0, 1]$ with $\mathbb{E}[x_i] = \mu$, we have

$$\Pr\left[\left|\frac{\sum_{i=1}^{n} x_i}{n} - \mu\right| \geq \delta\right] \leq 2e^{-2n\delta^2}.$$



Very unlikely to play a suboptimal arm

Lemma 3

At any time step t, if a suboptimal arm i (i.e., $\mu_i < \mu^*$) has been played for $n_{i,t} \geq \frac{4 \ln t}{\Delta_i^2}$ times, then $\mathrm{UCB}_{i,t} < \mathrm{UCB}_{I^*,t}$ with probability $\geq 1 - 4/t^2$. Therefore, for any t,

$$\Pr\left[I_{t+1,i} = 1 \,\middle|\, n_{i,t} \geq \frac{4 \ln t}{\Delta_i^2} \right] \leq \frac{4}{t^2}.$$

Proof of Lemma 3

With probability $< 2/t^2 + 2/t^2$ (union bound) that

$$\begin{aligned} \mathrm{UCB}_{i,t} = \hat{\mu}_{i,t} + \sqrt{\frac{\ln t}{n_{i,t}}} &\leq \hat{\mu}_{i,t} + \frac{\Delta_i}{2} \\ &< \left(\mu_i + \frac{\Delta_i}{2}\right) + \frac{\Delta_i}{2} \\ &= \mu^* < \mathrm{UCB}_{i^*,t} \end{aligned}$$

does NOT hold.

Playing suboptimal arms for very limited number of times

Lemma 4

For any arm i with $\mu_i < \mu^*$,

$$\mathbb{E}[n_{i,T}] \leq \frac{4 \ln T}{\Delta_i^2} + 8.$$

$$\begin{aligned}
\mathbb{E}[n_{i,T}] &= 1 + \mathbb{E}\left[\sum_{t=N}^{T} \mathbb{1}\left\{I_{t+1,i} = 1\right\}\right] \\
&= 1 + \mathbb{E}\left[\sum_{t=N}^{T} \mathbb{1}\left\{I_{t+1,i} = 1, n_{i,t} < \frac{4 \ln t}{\Delta_i^2}\right\}\right] \\
&\quad + \mathbb{E}\left[\sum_{t=N}^{T} \mathbb{1}\left\{I_{t+1,i} = 1, n_{i,t} \geq \frac{4 \ln t}{\Delta_i^2}\right\}\right]
\end{aligned}$$

Proof of Lemma 4 (contd.)

$$
\begin{aligned}
\mathbb{E}[n_{i,T}] &\leq \frac{4\ln T}{\Delta_i^2} + \mathbb{E}\left[\sum_{t=N}^{T} \mathbb{1}\left\{I_{t+1,i} = 1, n_{i,t} \geq \frac{4\ln t}{\Delta_i^2}\right\}\right] \\
&= \frac{4\ln T}{\Delta_i^2} + \sum_{t=N}^{T} \Pr\left[I_{t+1,i} = 1, n_{i,t} \geq \frac{4\ln t}{\Delta_i^2}\right] \\
&= \frac{4\ln T}{\Delta_i^2} + \sum_{t=N}^{T} \Pr\left[I_{t+1,i} = 1 \,\middle|\, n_{i,t} \geq \frac{4\ln t}{\Delta_i^2}\right] \cdot \Pr\left[n_{i,t} \geq \frac{4\ln t}{\Delta_i^2}\right] \\
&\leq \frac{4\ln T}{\Delta_i^2} + \sum_{t=N}^{T} \frac{4}{t^2} \\
&\leq \frac{4\ln T}{\Delta_i^2} + 8.
\end{aligned}
$$

The regret bound for the UCB algorithm

Theorem 4

For all $T \geq N$, the (expected) regret by the UCB algorithm in round T is

$$\mathbb{E}[\text{regret}_T] \leq 5\sqrt{NT \ln T} + 8N.$$

Proof of Theorem 4

- Divide the arms into two groups:
  1. Group ONE (G_1): "almost optimal arms" with $\Delta_i < \sqrt{\frac{N}{T} \ln T}$.
  2. Group TWO (G_2): "bad" arms with $\Delta_i \geq \sqrt{\frac{N}{T} \ln T}$.

$$\sum_{i \in G_1} n_{i,T} \Delta_i \leq \left(\sqrt{\frac{N}{T} \ln T} \right) \sum_{i \in G_1} n_{i,T} \leq T \cdot \sqrt{\frac{N}{T} \ln T} = \sqrt{NT \ln T}.$$

By Lemma 4,

$$\begin{aligned} \sum_{i \in G_2} \mathbb{E}[n_{i,T}] \Delta_i \leq \sum_{i \in G_2} \frac{4 \ln T}{\Delta_i} + 8\Delta_i \quad &\leq \quad \sum_{i \in G_2} 4 \sqrt{\frac{T \ln T}{N}} + 8 \\ &\leq \quad 4\sqrt{NT \ln T} + 8N. \end{aligned}$$

□

Outline

Time Decaying ϵ-Greedy Algorithm

What if the horizon T is known in advance when we run ϵ-Greedy?

Time-Decaying ϵ-Greedy Algorithm

For all $t = 1, 2, \ldots, N$, set $\epsilon := N^{1/3}/T^{1/3}$:

- With probability $1 - \epsilon$, pull arm $I_t := \arg\max_{i=1,\ldots,N} \hat{\mu}_{i,t}$.
- With probability ϵ, select an arm uniformly at random (i.e., each with probability $1/N$).

Time Decaying ϵ-Greedy Algorithm

What if the horizon T is known in advance when we run ϵ-Greedy?

Time-Decaying ϵ-Greedy Algorithm

For all $t = 1, 2, \ldots, N$, set $\epsilon := N^{1/3}/T^{1/3}$:

- With probability $1 - \epsilon$, pull arm $I_t := \arg\max_{i=1,\ldots,N} \hat{\mu}_{i,t}$.
- With probability ϵ, select an arm uniformly at random (i.e., each with probability $1/N$).

Claim

Time-Decaying ϵ-Greedy Algorithm gets roughly $O(N^{1/3}T^{2/3})$ regret.

Sketch of proving the claim

- The expected regret $\mathbf{E}[R(T)] = \sum_{t=1}^{T} \mathbf{E}[\mu^* - \mu_{T_t}]$.
- Using the greedy choice that $\hat{\mu}_{I_t} \geq \hat{\mu}_{I^*}$, we have

$$
\begin{aligned}
\mathbf{E}[R(T)] &\leq \sum_{t=1}^{T}(1-\epsilon)\mathbf{E}[(\mu_{I^*} - \hat{\mu}_{I^*} + \hat{\mu}_{I_t} - \mu_{I_t}) \mid \text{greedy choice of } I_t] + \epsilon T \\
&\leq \sum_{t=1}^{T}\left(\sqrt{\frac{\ln T}{n_{I^*,t}}} + \sqrt{\frac{\ln T}{n_{I_t,t}}}\right) + \frac{1}{T}\cdot 1 \cdot T + \epsilon T \quad \text{(Chernoff)} \\
\approx\leq & \sum_{t=1}^{T}\left(\sqrt{\frac{\ln T}{\epsilon t/N}} + \sqrt{\frac{\ln T}{\epsilon t/N}}\right) + \epsilon T + 1 \\
&\leq \sqrt{\frac{N}{\epsilon}}\sqrt{T \log T} + \epsilon T + 1 = O(N^{1/3}T^{2/3}\sqrt{logT}).
\end{aligned}
$$

Discussions