Bandit Learning

Yingkai Li

EC4501/EC4501HM

Multi-arm Bandits

Consider an online decision process with T periods and n arms.

- each arm i has stochastic return $F_i \in \Delta([0,1])$ with mean μ_i for each time period;
- the designer cannot observe F_i for any i.

Multi-arm Bandits

Consider an online decision process with T periods and n arms.

- each arm i has stochastic return $F_i \in \Delta([0,1])$ with mean μ_i for each time period;
- the designer cannot observe F_i for any i.

At any time $t \leq T$:

- designer selects an arm i_t^* based on past rewards;
- the payoff $v_{i_t^*}$ is realized according to $F_{i_t^*}$.

Multi-arm Bandits

Consider an online decision process with T periods and n arms.

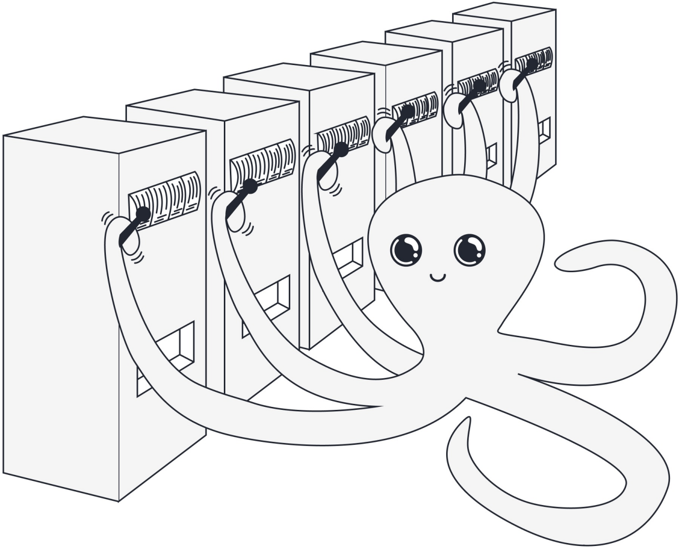
- each arm i has stochastic return $F_i \in \Delta([0,1])$ with mean μ_i for each time period;
- the designer cannot observe F_i for any i.

At any time $t \leq T$:

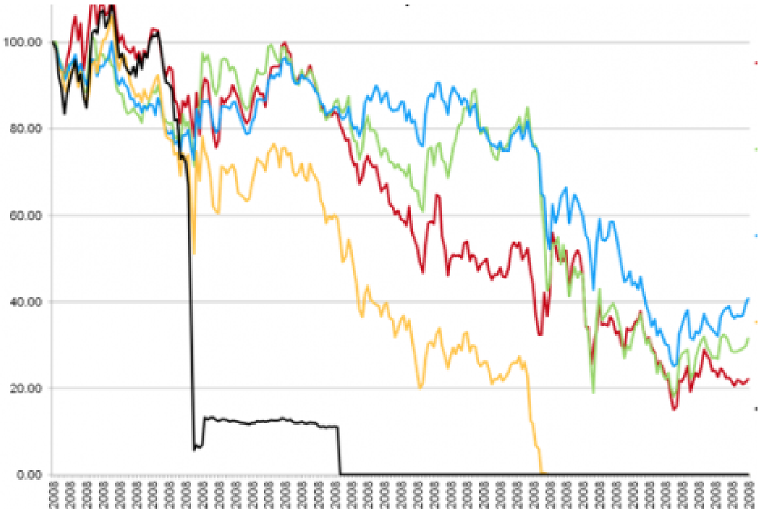
- designer selects an arm i_t^* based on past rewards;
- the payoff $v_{i_t^*}$ is realized according to $F_{i_t^*}$.

Question: how to design online algorithms with good online performance even without knowing $\{F_i\}_{i\in[n]}$?

Applications



Applications



Regret Minimization

Expected reward of the best arm:

$$B_T = T \cdot \max_{i \in [n]} \mathbf{E}_{v_i \sim F_i}[v_i] \,.$$

Regret Minimization

Expected reward of the best arm:

$$B_T = T \cdot \max_{i \in [n]} \mathbf{E}_{v_i \sim F_i}[v_i] \,.$$

(External) Regret:

$$\mathrm{R}_T = B_T - \sum_{t \in T} v_{i_t^*, t}.$$

Regret Minimization

Expected reward of the best arm:

$$B_T = T \cdot \max_{i \in [n]} \mathbf{E}_{v_i \sim F_i}[v_i] \,.$$

(External) Regret:

$$\mathrm{R}_T = B_T - \sum_{t \in T} v_{i_t^*, t}.$$

An algorithm has no-regret if $\mathrm{R}_T = o(T)$.

- Is it possible to design no-regret algorithms without any knowledge about the reward distributions?

Simple Question

PollEv.com/quietsalute502

Myopic Exploitation

Myopic Exploitation Algorithm:

- at any time $t \leq T$, select the arm with highest average reward

$$i_t^* = \underset{i \in [n]}{\operatorname{argmax}} \, \hat{\mu}_{i,t} \quad \text{where} \quad \hat{\mu}_{i,t} \triangleq \frac{\sum_{s<t} v_{i,s} \cdot \mathbf{1}\left(i = i_s^* \right)}{\sum_{s<t} \mathbf{1}\left(i = i_s^* \right)}.$$

Myopic Exploitation

Myopic Exploitation Algorithm:

- at any time $t \leq T$, select the arm with highest average reward

$$i_t^* = \operatorname*{argmax}_{i \in [n]} \hat{\mu}_{i,t} \quad \text{where} \quad \hat{\mu}_{i,t} \triangleq \frac{\sum_{s<t} v_{i,s} \cdot \mathbf{1}\left(i = i_s^* \right)}{\sum_{s<t} \mathbf{1}\left(i = i_s^* \right)}.$$

Myopic exploitation has regret $\mathrm{R}_T = \Theta(T)$.

- two arms, arm 1 has fixed reward $\frac{1}{3}$, arm 2 has reward uniform in $\{0, 1\}$;
- myopic exploitation will always choose the inferior arm 1 if in the first time arm 2 only provides a reward of 0; with expected regret at least $\frac{T}{12}$.

Explore then Exploit

Learn the distributions first and choose the best one later.

Explore then Exploit

Learn the distributions first and choose the best one later.

Given parameter $K \leq \frac{T}{n}$:

- choose each arm one by one for each period $t \leq nK$;
- for any period $t \in [nK+1, T]$, choose arm

$$i_t^* = \operatorname*{argmax}_{i \in [n]} \hat{\mu}_{i,nK}.$$

Concentration Inequalities

Estimation error with large samples.

Lemma (Hoeffding's Inequality)

Let $X_1, X_2, \ldots, X_n$ be independent random variables such that $X_i \in [a_i, b_i]$ almost surely. Then, for the sum of these variables, we have the following concentration bound:

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right]\right| \geq \epsilon\right) \leq 2\exp\left(-\frac{2n^2\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

In the special case where $X_i \in [0,1]$ for all i:

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} X_i\right]\right| \geq \epsilon\right) \leq 2\exp(-2n\epsilon^2).$$

Explore then Exploit

Bound on sample size for an ϵ-estimation with error probability at most δ.

Lemma

Fixing any arm i, for any $\epsilon, \delta > 0$, if $K \geq \frac{1}{2\epsilon^2} \cdot \log \frac{2}{\delta}$, we have $|\hat{\mu}_{i,nK} - \mu_i| \leq \epsilon$ with probability at least $1 - \delta$.

Explore then Exploit

Bound on sample size for an ϵ-estimation with error probability at most δ.

Lemma

Fixing any arm i, for any $\epsilon, \delta > 0$, if $K \geq \frac{1}{2\epsilon^2} \cdot \log \frac{2}{\delta}$, we have $|\hat{\mu}_{i,nK} - \mu_i| \leq \epsilon$ with probability at least $1 - \delta$.

Let X_j be the random variable for pulling arm i for the jth time.

$$\begin{aligned}\mathbb{P}\left(|\hat{\mu}_{i,nK} - \mu_i| \geq \epsilon\right) &= \mathbb{P}\left(\left|\frac{1}{K}\sum_{j=1}^{K} X_j - \mathbb{E}\left[\frac{1}{K}\sum_{j=1}^{K} X_j\right]\right| \geq \epsilon\right) \\ &\leq 2\exp(-2K\epsilon^2) \leq \delta.\end{aligned}$$

Explore then Exploit

Lemma (Union Bound)

For any probability events X, Y, we have

$$\Pr[X \cup Y] \leq \Pr[X] + \Pr[Y].$$

Explore then Exploit

Lemma (Union Bound)

For any probability events X, Y, we have

$$\Pr[X \cup Y] \leq \Pr[X] + \Pr[Y].$$

By the union bound, we have that for $K \geq \frac{1}{2\epsilon^2} \cdot \log \frac{2n}{\delta}$, with probability $1 - \delta$,

$$|\hat{\mu}_{i,nK} - \mu_i| \leq \epsilon \text{ for all } i \in [n].$$

Explore then Exploit

Lemma (Union Bound)

For any probability events X, Y, we have

$$\Pr[X \cup Y] \leq \Pr[X] + \Pr[Y].$$

By the union bound, we have that for $K \geq \frac{1}{2\epsilon^2} \cdot \log \frac{2n}{\delta}$, with probability $1 - \delta$,

$$|\hat{\mu}_{i,nK} - \mu_i| \leq \epsilon \text{ for all } i \in [n].$$

Therefore, letting $\delta = \frac{1}{T}$ and $\epsilon = (\frac{n}{T})^{\frac{1}{3}}$, we have $K = \frac{1}{2} \cdot (\frac{T}{n})^{\frac{2}{3}} \cdot \log 2nT$ and the regret of Explore-then-Exploit is

$$\mathrm{R}_T \leq \underbrace{nK}_{\text{Exploration Regret}} + \underbrace{T((1-\delta) \cdot 2\epsilon + \delta)}_{\text{Exploitation Regret}} \leq nK + T \cdot 2\epsilon + 1 = O(n^{\frac{1}{3}} \cdot T^{\frac{2}{3}} \cdot \log 2nT).$$

Better Algorithms

The Explore-then-Exploit does not exploit the better arms until after $\tilde{O}(n^{\frac{1}{3}} \cdot T^{\frac{2}{3}})$ periods.

Better Algorithms

The Explore-then-Exploit does not exploit the better arms until after $\tilde{O}(n^{\frac{1}{3}} \cdot T^{\frac{2}{3}})$ periods.

Intuition for better algorithms: exploits the better arms more promptly.

- Active arm elimination;
- Upper confidence bound;
- Thompson sampling.

Better Algorithms

The Explore-then-Exploit does not exploit the better arms until after $\tilde{O}(n^{\frac{1}{3}} \cdot T^{\frac{2}{3}})$ periods.

Intuition for better algorithms: exploits the better arms more promptly.

- Active arm elimination;
- Upper confidence bound;
- Thompson sampling.

The worst-case regrets for these three algorithms are $O(\sqrt{nT \cdot \log nT})$

Active Arm Elimination

- Maintain an active set S, which is initialized as $[n]$;
- Choose an arm in S in a sequential order;
- Update the active set S: eliminate arm $i \in S$ if there exists $j \in S$ such that

$$\hat{\mu}_{j,t} \geq \hat{\mu}_{i,t} + 2C_t$$

where $C_t = \sqrt{\frac{\log nT}{K_t}}$ and K_t is the number of times arms in S has been chosen.

Active Arm Elimination

- Maintain an active set S, which is initialized as $[n]$;
- Choose an arm in S in a sequential order;
- Update the active set S: eliminate arm $i \in S$ if there exists $j \in S$ such that

$$\hat{\mu}_{j,t} \geq \hat{\mu}_{i,t} + 2C_t$$

where $C_t = \sqrt{\frac{\log nT}{K_t}}$ and K_t is the number of times arms in S has been chosen.

Intuition: if the history of rewards indicates that an arm is not the best arm with high probability, the algorithm never chooses that arm again in the future.

- in contrast, Explore-then-Exploit keeps exploring bad arms until after $\tilde{O}(n^{\frac{1}{3}} \cdot T^{\frac{2}{3}})$.

Active Arm Elimination

In active-arm-elimination, a worse arm is eliminated earlier.

Active Arm Elimination

In active-arm-elimination, a worse arm is eliminated earlier.

Let i^* be the optimal arm, and let $\Delta_i = \mu_{i^*} - \mu_i$.

Lemma

With probability $1 - \frac{2}{nT}$, arm i^* is never eliminated, and arm $i \neq i^*$ is removed before time

$$T_i \triangleq \frac{16 \log nT}{\Delta_i^2}.$$

Active Arm Elimination

In active-arm-elimination, a worse arm is eliminated earlier.

Let i^* be the optimal arm, and let $\Delta_i = \mu_{i^*} - \mu_i$.

Lemma

With probability $1 - \frac{2}{nT}$, arm i^* is never eliminated, and arm $i \neq i^*$ is removed before time

$$T_i \triangleq \frac{16 \log nT}{\Delta_i^2}.$$

Again by applying Hoeffding's inequality, at any time $t \in [T]$, for any arm $i \in [n]$,

$$\Pr\left[\, |\hat{\mu}_{i,t} - \mu_i| \geq C_t \,\right] \leq \frac{2}{(nT)^2}.$$

We apply the union bound such that they hold simultaneously with probability at most $\frac{2}{nT}$.

Active Arm Elimination

In active-arm-elimination, a worse arm is eliminated earlier.

Let i^* be the optimal arm, and let $\Delta_i = \mu_{i^*} - \mu_i$.

Lemma

With probability $1 - \frac{2}{nT}$, arm i^* is never eliminated, and arm $i \neq i^*$ is removed before time

$$T_i \triangleq \frac{16 \log nT}{\Delta_i^2}.$$

Again by applying Hoeffding's inequality, at any time $t \in [T]$, for any arm $i \in [n]$,

$$\Pr\left[\, |\hat{\mu}_{i,t} - \mu_i| \geq C_t \,\right] \leq \frac{2}{(nT)^2}.$$

We apply the union bound such that they hold simultaneously with probability at most $\frac{2}{nT}$.

$$\hat{\mu}_{i^*,t} - \hat{\mu}_{i,t} > (\mu_{i^*} - C_t) - (\mu_i + C_t) = \Delta_i - 2C_t.$$

To guarantee elimination of i, we require $\Delta_i - 2C_t \geq 2C_t$, or $\Delta_i \geq 4C_t = 4\sqrt{\frac{\log(nT)}{K_t}}$.

Solving for K_t: $K_t \geq \frac{16\log(nT)}{\Delta_i^2}$.

Active Arm Elimination

Instance-dependent Bound:

$$
\begin{aligned}
R_T(\mathcal{E}) &\leq \sum_{i \neq i^*} \Delta_i \cdot T_i \\
&= \sum_{i \neq i^*} \Delta_i \cdot \frac{16 \log(nT)}{\Delta_i^2} \\
&= 16 \log(nT) \cdot \sum_{i \neq i^*} \frac{1}{\Delta_i}.
\end{aligned}
$$

Active Arm Elimination

Lemma (Cauchy-Schwarz inequality)

For two vectors $\mathbf{u} = (u_1, ..., u_k)$ *and* $\mathbf{v} = (v_1, ..., v_k)$,

$$\left(\sum_{i=1}^{k} u_i v_i\right)^2 \leq \left(\sum_{i=1}^{k} u_i^2\right) \cdot \left(\sum_{i=1}^{k} v_i^2\right).$$

Active Arm Elimination

Lemma (Cauchy-Schwarz inequality)

For two vectors $\mathbf{u} = (u_1, ..., u_k)$ *and* $\mathbf{v} = (v_1, ..., v_k)$*,*

$$\left(\sum_{i=1}^{k} u_i v_i\right)^2 \le \left(\sum_{i=1}^{k} u_i^2\right) \cdot \left(\sum_{i=1}^{k} v_i^2\right).$$

Worst-case Bound: Let $L = 16\log(nT)$. Worst case occurs when $\sum_{i \neq i^*} T_i = T$, i.e., $\sum_{i \neq i^*} \frac{L}{\Delta_i^2} = T$.

Active Arm Elimination

Lemma (Cauchy-Schwarz inequality)

For two vectors $\mathbf{u} = (u_1, ..., u_k)$ *and* $\mathbf{v} = (v_1, ..., v_k)$,

$$\left(\sum_{i=1}^{k} u_i v_i\right)^2 \leq \left(\sum_{i=1}^{k} u_i^2\right) \cdot \left(\sum_{i=1}^{k} v_i^2\right).$$

Worst-case Bound: Let $L = 16\log(nT)$. Worst case occurs when $\sum_{i \neq i^*} T_i = T$, i.e., $\sum_{i \neq i^*} \frac{L}{\Delta_i^2} = T$.
The regret of active-arm-elimination is

$$\begin{aligned} R_T &\leq L \cdot \sum_{i \neq i^*} \frac{1}{\Delta_i} \\ &\leq L\sqrt{n \sum_{i \neq i^*} \frac{1}{\Delta_i^2}} && \text{(Cauchy-Schwarz)} \\ &= O(\sqrt{nT \cdot \log nT}). \end{aligned}$$

Upper Confidence Bound (UCB)

Optimism in the face of uncertainty.

Upper Confidence Bound (UCB)

Optimism in the face of uncertainty.

Upper confidence bound: $\bar{u}_{i,t} = \hat{\mu}_{i,t} + \sqrt{\frac{\log nT}{K_{i,t}}}$.

- $\bar{u}_{i,t}$ is the optimistic estimate of μ_i at time t given the historical rewards.
- $K_{i,t}$: the number of times action i is chosen before t.

Upper Confidence Bound (UCB)

Optimism in the face of uncertainty.

Upper confidence bound: $\bar{u}_{i,t} = \hat{\mu}_{i,t} + \sqrt{\frac{\log nT}{K_{i,t}}}$.

- $\bar{u}_{i,t}$ is the optimistic estimate of μ_i at time t given the historical rewards.
- $K_{i,t}$: the number of times action i is chosen before t.

At any time t, UCB chooses the arm that maximizes $\bar{u}_{i,t}$.

Upper Confidence Bound (UCB)

Optimism in the face of uncertainty.

Upper confidence bound: $\bar{u}_{i,t} = \hat{\mu}_{i,t} + \sqrt{\frac{\log nT}{K_{i,t}}}$.

- $\bar{u}_{i,t}$ is the optimistic estimate of μ_i at time t given the historical rewards.
- $K_{i,t}$: the number of times action i is chosen before t.

At any time t, UCB chooses the arm that maximizes $\bar{u}_{i,t}$.

Intuition: somewhat similar to active-arm-elimination (AAE), UCB never chooses suboptimal arms for too many periods.

- AAE rules out all overly pessimistic arms;
- UCB chooses the most optimistic arm.

Upper Confidence Bound (UCB)

Similar to the analysis of AAE: with probability at least $1 - \frac{2}{nT}$, each arm $i \neq i^*$ is pulled by at most $\frac{8 \log nT}{\Delta_i^2}$ times.

Upper Confidence Bound (UCB)

Similar to the analysis of AAE: with probability at least $1 - \frac{2}{nT}$, each arm $i \neq i^*$ is pulled by at most $\frac{8 \log nT}{\Delta_i^2}$ times.

Instance-dependent Bound:

$$R_T(\mathcal{E}) = 8 \log(nT) \cdot \sum_{i \neq i^*} \frac{1}{\Delta_i}.$$

Upper Confidence Bound (UCB)

Similar to the analysis of AAE: with probability at least $1 - \frac{2}{nT}$, each arm $i \neq i^*$ is pulled by at most $\frac{8 \log nT}{\Delta_i^2}$ times.

Instance-dependent Bound:

$$R_T(\mathcal{E}) = 8 \log(nT) \cdot \sum_{i \neq i^*} \frac{1}{\Delta_i}.$$

Worst-case Bound:

$$R_T = O(\sqrt{nT \cdot \log nT}).$$

Thompson Sampling

- At any time t, estimate a reward distribution $\hat{F}_{i,t}$ for each arm i;
- Sample reward r_i from $\hat{F}_{i,t}$;
- Thompson sampling chooses an arm that maximizes r_i.

Thompson Sampling

- At any time t, estimate a reward distribution $\hat{F}_{i,t}$ for each arm i;
- Sample reward r_i from $\hat{F}_{i,t}$;
- Thompson sampling chooses an arm that maximizes r_i.

Intuition: better arms are exploited with higher probability, and bad arms are still explored with a small probability.

- empirically, Thompson sampling usually have better performance than UCB or active-arm-elimination, despite the fact that they have the same worst-case regret.

Unknown Time Horizon

Previous algorithms such as UCB set confidence intervals based on the time horizon.

- for practical applications, the time horizon may be unknown in ex ante.

Unknown Time Horizon

Previous algorithms such as UCB set confidence intervals based on the time horizon.

- for practical applications, the time horizon may be unknown in ex ante.

Doubling trick: consider time horizons $T = 2, 4, 8, 16 \cdots$

- for time $t \in [2^i, 2^{i+1})$, apply bandit algorithms as if the time horizon is 2^{i+1};

Unknown Time Horizon

Previous algorithms such as UCB set confidence intervals based on the time horizon.

- for practical applications, the time horizon may be unknown in ex ante.

Doubling trick: consider time horizons $T = 2, 4, 8, 16 \cdots$

- for time $t \in [2^i, 2^{i+1})$, apply bandit algorithms as if the time horizon is 2^{i+1};

General Reduction: let $m = \log T$ and let R_T^N be the regret without knowing T

$$R_T^N \le \sum_{k=1}^{m} R_{2^k} \le \sum_{k=1}^{m} \sum_{i \neq i^*} \frac{\log(n \cdot 2^k)}{\Delta_i} = \sum_{i \neq i^*} \frac{1}{\Delta_i} \cdot \sum_{k=1}^{m} \log(n \cdot 2^k) = \sum_{i \neq i^*} \frac{1}{\Delta_i} \cdot O((\log nT)^2).$$

Unknown Time Horizon

Previous algorithms such as UCB set confidence intervals based on the time horizon.

- for practical applications, the time horizon may be unknown in ex ante.

Doubling trick: consider time horizons $T = 2, 4, 8, 16 \cdots$

- for time $t \in [2^i, 2^{i+1})$, apply bandit algorithms as if the time horizon is 2^{i+1};

General Reduction: let $m = \log T$ and let R_T^N be the regret without knowing T

$$R_T^N \leq \sum_{k=1}^{m} R_{2^k} \leq \sum_{k=1}^{m} \sum_{i \neq i^*} \frac{\log(n \cdot 2^k)}{\Delta_i} = \sum_{i \neq i^*} \frac{1}{\Delta_i} \cdot \sum_{k=1}^{m} \log(n \cdot 2^k) = \sum_{i \neq i^*} \frac{1}{\Delta_i} \cdot O((\log nT)^2).$$

Similarly, the worst case regret is $R_T^N = O(\sqrt{nT} \cdot \log nT)$