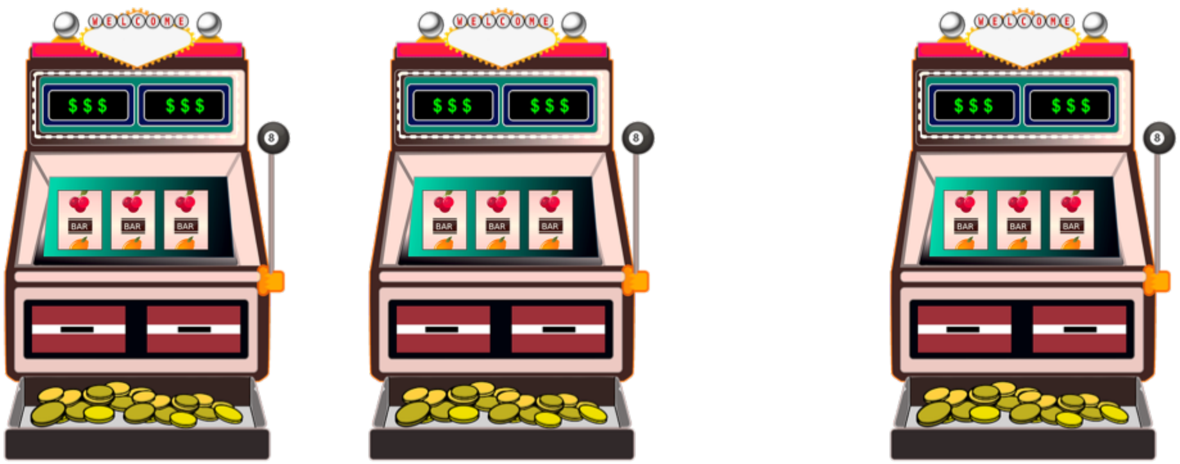
Lecture 11: Adversarial Games

- Full-information setting
- Adversarial bandits
- Hedge/Exp3/Exp3.P/Exp3-IX
- Lower bounds (if we have time)

Recall: Stochastic bandit setting

- Set $\mathcal{K} = \{1, 2, \ldots, K\}$ of K actions (arms, machines)
- You are facing a tuple of distributions $\nu = (P_1, P_2, \ldots, P_K)$



P_1 $\quad$ P_2 $\quad \ldots \quad$ P_K

- Identify the best action by interacting with the environment

What if we are wrong?

All models are wrong, but some are useful. - George E. P. Box

- Stochastic bandit model assumes that rewards are generated at random from a distribution that depends only on the chosen action, i.e. $r_t \sim P_{k_t}$
- What is *truly* stochastic?
- Are distributions always stationary?

Adversarial game

- Remove assumptions about how rewards are generated

$\rightarrow$ The 'environment' becomes an 'adversary'

$\rightarrow$ The adversary has access to the code of your algorithm!

Example: Simple game with two actions on horizon $t = T = 1$

1. You tell your friend your strategy for choosing an action
2. Your friend secretly chooses outcomes $x_1 \in \{0, 1\}$ and $x_2 \in \{0, 1\}$
3. You implement your strategy to select $k_t \in \{1, 2\}$ receive reward x_{k_t}
4. The regret is $R = \max\{x_1, x_2\} - x_{k_t}$

Analyzing the game

Example: Simple game with two actions on horizon $T = 1$

1. You tell your friend your strategy for choosing an action
2. Your friend secretly chooses outcomes $x_1 \in \{0, 1\}$ and $x_2 \in \{0, 1\}$
3. You implement your strategy to select $k_t \in \{1, 2\}$ receive reward x_{k_t}
4. The regret is $R = \max\{x_1, x_2\} - x_{k_t}$

- What happens if your friend chooses $x_1 = x_2$?
- What happens if you have a deterministic strategy?
- What if you have a randomized strategy s.t. $\Pr[k_t = x_1] = \Pr[k_t = x_2] = 1/2$?

Adversarial setting

- Set $\mathcal{K} = \{1, 2, \ldots, K\}$ of $K > 1$ actions
- You are facing a tuple of vectors $\nu = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ where $\mathbf{x}_t = (x_{1,t}, x_{2,t}, \ldots, x_{K,t}) = [0, 1]^K$ for each $t = 1 \ldots T$

For each round t:

- You select action $k_t \in \mathcal{K}$ using policy $\pi_t(\cdot | k_1, r_1, k_2, r_2, \ldots, k_{t-1}, r_{t-1})$
- You observe reward $r_t = x_{k_t,t}$

Measuring the performance

- Recall that in the stochastic bandit setting we tried to minimize the cumulative (expected) regret:

$$\sum_{t=1}^{T} (\mu_\star - \mu_{k_t})$$

$\rightarrow$ $\mu_k = \mathbb{E}[r_t | k_t = k]$, i.e. expectation of P_k
$\rightarrow$ $\mu_\star = \max_{k \in \mathcal{K}} \mu_k$
$\rightarrow$ Benchmark against '*always play the arm with highest expected reward*'

Does that make any sense in an adversarial setting?

A different notion of regret

- Future rewards have no relationship with previous ones
- Goal: Pull the arm with the highest reward at every step
- Minimize cumulative random regret:

$$\hat{R}_T = \max_{k \in \mathcal{K}} \sum_{t=1}^{T} x_{k,t} - \sum_{t=1}^{T} x_{k_t,t}$$

$\rightarrow$ Actual deficit of the learner relative to the best arm in hindsight

$\rightarrow$ Compete with the strategy that always picks a single arm

$\rightarrow$ Learn over time which is the best single arm

How to achieve this based on the information that we have?

Full-information setting

- Set $\mathcal{K} = \{1, 2, \ldots, K\}$ of $K > 1$ actions
- You are facing a tuple of vectors $\nu = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ where $\mathbf{x}_t = (x_{1,t}, x_{2,t}, \ldots, x_{K,t}) = [0, 1]^K$ for each $t = 1 \ldots T$

At each round t:

- You select action $k_t \in \mathcal{K}$
- You receive a reward $r_t = x_{k_t,t}$
- You observe $\mathbf{x}_t \leftarrow$ rewards associated with all arms at this round!

This is also known has *experts problem*!

Experts problem

- Set of available *experts*
- On each round:
  1. You receive a question to answer/problems to solve
  2. Each expert gives you a recommendation
  3. You select an expert to follow
  4. You receive the answer to the question/problem
- Which expert's advice should you follow?

Hedge algorithm

- Based on multiplicative weights update
- Weights accumulate the rewards of actions:

$$w_{k,t} = \exp\left(\eta \sum_{s=1}^{t} x_{k,s}\right) \qquad \text{for } \eta > 0$$

- Initially $w_{k,0} = 1$ for all $k \in \mathcal{K}$
- Compute selection probability for each action k: $p_{k,t} = \dfrac{w_{k,t-1}}{\sum_{k' \in \mathcal{K}} w_{k',t-1}}$
- Select action k_t at random (given action selection probabilities)
- Receive reward $r_t = x_{k_t,t}$
- Observe $\mathbf{x}_t$

$\rightarrow$ This is called a *soft max*

What happens if $\eta \rightarrow \infty$? What if $\eta \rightarrow 0$?

Hedge algorithm guarantees

Theorem 1. *Assume $x_{k,t} \in [0,1]$ for all $k \in \mathcal{K}$ and for all $t \geq 1$. Then*

$$\mathbb{E}[\hat{R}_T] = \max_{k \in \mathcal{K}} \sum_{t=1}^{T} x_{k,t} - \mathbb{E}\left[\sum_{t=1}^{T} x_{k_t,t}\right] \leq 2\eta \max_{k \in \mathcal{K}} \sum_{t=1}^{T} x_{k,t} + \frac{\ln K}{\eta}$$

for any choice of $\eta \in [0,1]$.

- We know that $\max_{k \in \mathcal{K}} \sum_{t=1}^{T} x_{k,t} \leq T$
- If we let $\eta = \sqrt{\frac{\ln K}{T}} \leq 1$ we get $\mathbb{E}[R_T] \leq 3\sqrt{T \ln K}$

$\rightarrow$ This requires observing $\mathbf{x}_t$ (i.e. rewards of non-chosen actions)

Adversarial bandit setting

- Set $\mathcal{K} = \{1, 2, \ldots, K\}$ of $K > 1$ actions
- You are facing a tuple of vectors $\nu = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$ where $\mathbf{x}_t = (x_{1,t}, x_{2,t}, \ldots, x_{K,t}) = [0, 1]^K$ for each $t = 1 \ldots T$

For each round t:

- You select action $k_t \in \mathcal{K}$ using policy $\pi_t(\cdot | k_1, r_1, k_2, r_2, \ldots, k_{t-1}, r_{t-1})$
- You observe reward $r_t = x_{k_t,t}$

$\rightarrow$ You **do not** observe rewards for actions $k \neq k_t$

How should we estimate the weights in the Hedge algorithm then?

Importance sampling

- When we can sample from one distribution $p(x)$
- ...but we are interested in the expectation when we sample with respect to another distribution $q(x)$
- Simple idea: take samples x from $p(x)$ and modify them:

$$\hat{x} = \frac{xq(x)}{p(x)}$$

$\rightarrow \mathbb{E}_{x\sim p}[\hat{x}] = \mathbb{E}_{\hat{x}\sim q}[x]$

- How we can use this:

$$\hat{x}_{k,t} = \frac{x_{k,t}\mathbb{I}\{k_t = k\}}{p_{k,t}} = \frac{r_t\mathbb{I}\{k_t = k\}}{p_{k,t}}$$

Importance sampling: Expectation

$$\text{Recall:} \quad \hat{x}_{k,t} = \frac{\mathbb{I}\{k_t = k\}}{p_{k,t}} x_{k,t} = \frac{\mathbb{I}\{k_t = k\}}{p_{k,t}} r_t$$

- Let $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot | k_1, r_1, k_2, r_2, \ldots, k_{t-1}, r_{t-1}]$
- Expectation of estimator:

$$\begin{aligned}
\mathbb{E}[\hat{x}_{k,t} | k_1, r_1, k_2, r_2, \ldots, k_{t-1}, r_{t-1}] &= \mathbb{E}_t \left[\frac{\mathbb{I}\{k_t = k\}}{p_{k,t}} x_{k,t} \right] \\
(p_{k,t} \text{ is a function of } k_1, r_1, \ldots, k_{t-1}, r_{t-1}) &= \frac{x_{k,t}}{p_{k,t}} \mathbb{E}_t[\mathbb{I}\{k_t = k\}] \\
&= \frac{x_{k,t}}{p_{k,t}} p_{k,t} \\
&= x_{k,t}
\end{aligned}$$

Importance sampling: Variance

$$\text{Recall:} \quad \hat{x}_{k,t} = \frac{\mathbb{I}\{k_t = k\}}{p_{k,t}} x_{k,t} = \frac{\mathbb{I}\{k_t = k\}}{p_{k,t-1}} r_t$$

$$\mathbb{E}_t[\hat{x}_{k,t}] = x_{k,t}$$

- Let $\mathbb{V}_t[\cdot] = \mathbb{V}[\cdot | k_1, r_1, k_2, r_2, \ldots, k_{t-1}, r_{t-1}]$
- Variance of estimator:

$$\begin{aligned}
\mathbb{V}[\hat{x}_{k,t} | k_1, r_1, k_2, r_2, \ldots, k_t, r_t] &= \mathbb{E}_t[\hat{x}_{k,t}^2] - \mathbb{E}_t[\hat{x}_{k,t}]^2 \\
&= \mathbb{E}_t\left[\frac{\mathbb{I}\{k_t = k\}}{p_{k,t}^2} x_{k,t}^2\right] - x_{k,t}^2 \\
&= x_{k,t}^2 \frac{1 - p_{k,t}}{p_{k,t}}
\end{aligned}$$

Exp3 algorithm

- '**Exp**onential-weight algorithm for **Exp**loration and **Exp**loitation'
- Weights accumulate the rewards of actions:

$$w_{k,t} = \exp\left(\eta \sum_{s=1}^{t} \hat{x}_{k,s}\right) \qquad \text{with} \quad \hat{x}_{k,s} = \frac{\mathbb{I}\{k_s = k\}}{p_{k,s}} r_s$$

- Initially $w_{k,0} = 1$ for all $k \in \mathcal{K}$
- Compute selection probability for each action k:

$$p_{k,t} = \frac{w_{k,t-1}}{\sum_{k' \in \mathcal{K}} w_{k',t-1}}$$

- Select action k_t at random (given action selection probabilities)
- Receive reward $r_t = x_{k_t,t}$
- η is called *learning rate* $\rightarrow$ Link to the exploration/exploitation tradeoff?

Exp3 guarantees

Theorem 2. *With learning rate* $\eta = \sqrt{\frac{\ln K}{TK}}$*, then*

$$\hat{R}_T \leq 2\sqrt{TK \ln K}.$$

- We lose a factor $\sqrt{K}$ compared with the full-information case

$\rightarrow$ Price to pay for missing information

Reducing variance: High rewards $\rightarrow$ Low losses

Recall: $w_{k,t} = \exp\left(\eta \sum_{s=1}^{t} \hat{x}_{k,s}\right)$ with $\hat{x}_{k,s} = \frac{\mathbb{I}\{k_s = k\}}{p_{k,s}} x_{k,s}$

What happens if $x_{k,s}$ is bounded away from 0?

- Recall: rewards $\mathbf{x}_{k,t} \in [0,1]$ for all $k \in \mathcal{K}$, for all $t \geq 1$
- Loss: $y_{k,t} = 1 - x_{k,t}$, $\ell_t = 1 - r_t$

$$\hat{y}_{k,t} = \frac{\mathbb{I}\{k_t = k\}}{p_{k,t}} y_{k,t}$$

$$\rightarrow \mathbb{V}[\hat{y}_{k,t} | k_1, r_1, k_2, r_2, \ldots, k_t, r_t] = y_{k,t}^2 \frac{1 - p_{k,t}}{p_{k,t}}$$

How would you rewrite the random regret in terms of losses?

Random regret with losses

$$
\begin{aligned}
\hat{R}_T &= \max_{k \in \mathcal{K}} \sum_{t=1}^{T} x_{k,t} - \sum_{t=1}^{T} x_{k_t,t} \\
&= -\min_{k \in \mathcal{K}} \sum_{t=1}^{T} -x_{k,t} - \sum_{t=1}^{T} x_{k_t,t} \\
&= -\min_{k \in \mathcal{K}} \sum_{t=1}^{T} -x_{k,t} - T + T - \sum_{t=1}^{T} x_{k_t,t} \\
&= -(T + \min_{k \in \mathcal{K}} \sum_{t=1}^{T} -x_{k,t}) + T - \sum_{t=1}^{T} x_{k_t,t} \\
&= -\min_{k \in \mathcal{K}} \sum_{t=1}^{T} (1 - x_{k,t}) + \sum_{t=1}^{T} (1 - x_{k_t,t}) = \sum_{t=1}^{T} y_{k_t,t} - \min_{k \in \mathcal{K}} \sum_{t=1}^{T} y_{k,t}
\end{aligned}
$$

How would you rewrite Exp3 with losses?

Exp3 with losses

- Weights accumulate the losses of actions:

$$w_{k,t} = \exp\left(-\eta \sum_{s=1}^{t} \hat{y}_{k,s}\right) \qquad \text{with} \quad \hat{y}_{k,s} = \frac{\mathbb{I}\{k_s = k\}}{p_{k,s}} \ell_s$$

- Initially $w_{k,0} = 1$ for all $k \in \mathcal{K}$
- Compute selection probability for each action k:

$$p_{k,t} = \frac{w_{k,t-1}}{\sum_{k' \in \mathcal{K}} w_{k',t-1}}$$

- Select action k_t at random (given action selection probabilities)
- Receive loss $\ell_t = y_{k_t,t}$

Increasing stability

Recall: $w_{k,t} = \exp\left(-\eta \sum_{s=1}^{t} \hat{y}_{k,s}\right)$ with $\hat{y}_{k,s} = \frac{\mathbb{I}\{k_s = k\}}{p_{k,s}} \ell_s$

What happens if $p_{k,t}$ becomes very small?

- Trick: Ensure that $p_{k,t} \geq$ something
- Two approaches:
  1. Mix the sampling probabilities with a uniform distribution
  2. Be optimistic

Increasing stability: Blending with uniform distribution

Recall: $w_{k,t} = \exp\left(-\eta \sum_{s=1}^{t} \hat{y}_{k,s}\right)$ with $\hat{y}_{k,s} = \frac{\mathbb{I}\{k_s = k\}}{p_{k,s}} \ell_s$

- Let $\gamma \in (0,1)$, redefine

$$p_{k,t} = (1-\gamma)\frac{w_{k,t-1}}{\sum_{k' \in \mathcal{K}} w_{k',t-1}} + \frac{\gamma}{K}$$

$\rightarrow$ This ensures that $p_{k,t} \geq \frac{\gamma}{K}$

Exp3.P algorithm

- Weights accumulate the losses of actions:

$$w_{k,t} = \exp\left(-\eta \sum_{s=1}^{t} \hat{y}_{k,s}\right) \qquad \text{with} \quad \hat{y}_{k,s} = \frac{\mathbb{I}\{k_s = k\}}{p_{k,s}} \ell_s$$

- Initially $w_{k,0} = 1$ for all $k \in \mathcal{K}$
- Compute selection probability for each action k:

$$p_{k,t} = (1-\gamma)\frac{w_{k,t-1}}{\sum_{k' \in \mathcal{K}} w_{k',t-1}} + \frac{\gamma}{K} \quad \text{for} \quad \gamma \in (0,1)$$

- Select action k_t at random (given action selection probabilities)
- Receive loss $\ell_t = y_{k_t,t}$

Exp3.P guarantees

Theorem 3. *There exists a universal constant* $C > 0$ *such that for any* $\delta \in (0, 1)$ *and an appropriate choice of* η *and* γ*, it holds that*

$$\hat{R}_T \leq C\sqrt{TK \ln(K/\delta)}$$

with probability at least $1 - \delta$*.*

Increasing stability: Being optimistic

- Let $\gamma > 0$, redefine

$$\hat{y}_{k,t} = \frac{\mathbb{I}\{k_t = k\}}{p_{k,t} + \gamma} y_{k,t}$$

$\rightarrow$ Bias/variance tradeoff:

$$\begin{aligned} \mathbb{E}[\hat{y}_{k,t} | k_1, r_1, k_2, r_2, \ldots, k_{t-1}, r_{t-1}] &= \mathbb{E}_t \left[\frac{\mathbb{I}\{k_t = k\}}{p_{k,t} + \gamma} y_{k,t} \right] = \frac{y_{k,t}}{p_{k,t} + \gamma} p_{k,t} \\ &\leq y_{k,t} \end{aligned}$$

How about variance?

Exp3-IX algorithm

- Exp3 with **I**mplicit e**X**ploration
- Weights accumulate the rewards of actions:

$$w_{k,t} = \exp\left(-\eta \sum_{s=1}^{t} \hat{y}_{k,s}\right) \qquad \text{with} \quad \hat{y}_{k,s} = \frac{\mathbb{I}\{k_s = k\}}{p_{k,s} + \gamma}(1 - r_s) \quad \text{for} \quad \gamma > 0$$

- Initially $w_{k,0} = 1$ for all $k \in \mathcal{K}$
- Compute selection probability for each action k:

$$p_{k,t} = \frac{w_{k,t-1}}{\sum_{k' \in \mathcal{K}} w_{k',t-1}}$$

- Select action k_t at random (given action selection probabilities)
- Receive reward $r_t = x_{k_t,t}$

Exp3-IX guarantees

Theorem 4. *Let $\delta \in (0,1)$ and define*

$$\eta_1 = \sqrt{\frac{2\ln(K+1)}{TK}} \qquad \textit{and} \qquad \eta_2 = \sqrt{\frac{\ln K + \ln\frac{K+1}{\delta}}{TK}}.$$

1. *If Exp-IX is run with parameters $\eta = \eta_1$ and $\gamma = \eta/2$, then*

$$\Pr\left[\hat{R}_T \geq \sqrt{8.5TK\ln(K+1)} + \left(\sqrt{\frac{TK}{2\ln(K+1)}} + 1\right)\ln 1/\delta\right] \leq \delta.$$

2. *If Exp-IX is run with parameters $\eta = \eta_2$ and $\gamma = \eta/2$, then*

$$\Pr\left[\hat{R}_T \geq 2\sqrt{(2\ln(K+1) + \ln(1/\delta)TK} + \ln\frac{TK}{\delta}\right] \leq \delta.$$

Summary

- Stochasticity is necessary in the adversarial setting
- Importance sampling allows to extend algorithms from the full-information setting to the bandit setting
- We can formulate algorithms in terms of rewards or losses
- We can gain stability/reduce variance by enforcing exploration
- We can gain stability/reduce variance by being optimistic
- Tightness vs generality of the bounds

Lower bounds

- Regret upper bound $\rightarrow$ control how bad we can perform
- How about the best performance that we can expect?

$$R_T \geq \text{something}$$

- Prove that no algorithm can do better
- Forces people to understand what is hard about the problem

$\rightarrow$ Derive the right algorithm for the right problem

Worst case regret

"For any policy that you give me, I will give you an instance of a bandit problem ν on which the regret is at least L"

- Worst case regret of policy π on environment class $\mathcal{E}$:

$$R_T(\pi, \mathcal{E}) = \sup_{\nu \in \mathcal{E}} R_T(\pi, \nu)$$

- Example:
  - Consider Bernoulli rewards
  - $R_T(\pi, \mathcal{E})$ looks for the K-arms Bernoulli bandit that is the most difficult to solve using policy π

Minimax optimality

- Let Π be the set of all policies
- Minimax regret:

$$R_T^*(\mathcal{E}) = \inf_{\pi \in \Pi} R_T(\pi, \mathcal{E}) = \inf_{\pi \in \Pi} \sup_{\nu \in \mathcal{E}} R_T(\pi, \nu)$$

- Compare all possible policies on their most challenging environment
- Small $R_T^*(\mathcal{E}) \to$ underlying bandit problem is less challenging
- Understand what makes $R_T^*(\mathcal{E})$ large/small
- A policy π is called minimax optimal for $\mathcal{E}$ if

$$R_T(\pi, \mathcal{E}) = R_T^*(\mathcal{E})$$

- Minimax optimality is a property of π, $\mathcal{E}$ and T

Game-theoretic interpretation

- Imagine a game between two-players: the protagonist and the antagonist
- For $K > 1$ and $T \geq K$:
  - The protagonist proposes a policy π
  - The antagonist looks at π and chooses a bandit instance $\nu \in \mathcal{E}$
- Utility for the antagonist: expected regret
- Utility for the protagonist: negation of the expected regret

$\rightarrow$ Zero-sum game!

Pareto optimality interpretation

- The regret of policies Π on environments in $\mathcal{E}$ is multi-objective

$\rightarrow$ Some policies are good on some instances, bad on others

- Policy π is *Pareto optimal* if there does not exist another policy π' that is a strict improvement:

$$R_T(\pi', \nu) \leq R_T(\pi, \nu) \quad \forall \nu \in \mathcal{E}$$

and

$$R_T(\pi', \nu) < R_T(\pi, \nu) \quad \text{for at least one } \nu \in \mathcal{E}$$

Deriving lower bounds: Key ideas

Select two bandit problem instances, ν_1 and ν_2, in such a way that the following conditions hold simultaenously:

Competition: A sequence of actions that is good for ν_1 is not good for ν_2

Similarity: ν_1 and ν_2 are *close* enough that the policy interacting with either of the two instances cannot statistically identify the true bandit with reasonable statistical accuracy

Conflict!

$\rightarrow$ Lower bound: optimize the tradeoff

Example of results: Adversarial bandits

Theorem 5. *Let $c, C > 0$ be sufficiently small/large universal constants and $K \geq 2$, $n \geq 1$ and $\delta \in (0,1)$ be such that $n \geq CK \ln(1/(2\delta))$. Then there exists a reward sequence $x \in [0,1]^{TK}$ such that*

$$\Pr\left[\hat{R}_T(x) \geq c\sqrt{TK \ln \frac{1}{2\delta}}\right] \geq \delta$$

$\rightarrow$ Pay attention to the inequalities