CS 396: Online Markets

Lecture 7: Multi-armed Bandit Learning

Last Time:

- online learning (cont)
- warmup: geometric random variables
- perturbed follow the leader (analysis)
- multi-armed bandit learning

Today:

- multi-armed bandit learning
- reduction to online learning

Exercise: Expected Payoff

Setup:

- online learning, $k = 2$ actions
- probabilities algorithm selects each action in round i are:

  $\boldsymbol{\pi}^i = (\pi_1^i, \pi_2^i) = (2/3, 1/3)$

- payoffs of each action in round i are:

  $\mathbf{v}^i = (v_1^i, v_2^i) = (3, 9)$

Question: What is the expected payoff of the algorithm in round i?

(Online) Multi-armed Bandit Learning

"online learning with partial information"

Model:

- k actions
- n rounds
- action j's payoff in round i: $v_j^i \in [0, h]$
- in round i:
  (a) choose an action j^i
  (b) learn payoffs $v_{j^i}^i$.
  (c) obtain payoff $v_{j^i}^i$.
- payoff ALG $= \sum_{i=1}^n v_{j^i}^i$

Goal: profit close to best action in hindsight

Note: identical to online learning except only learn $v_{j^i}^i$ and not $(v_1^i, \ldots, v_k^i)$.

Note: if don't play an action j, can't learn if j is good.

Challenge: tradeoff **explore** versus **exploit**.

Reducing MAB to Online Learning

Approach: reduce partial information to full information.

"solve multi-armed bandit problem with online learning algorithm"

```
 ---            ---              ---
|   |          |   | (1-eps)pi |   |
|   |    pi    |   |  + eps/k  |   |
|   | ------>  |   | --------> | W |
| O |          | M |           | O |
| L | (0,...   | A |           | R |
| A | vj/pij,  | B |           | L |
|   |   ...,0) |   |    vj     | D |
|   | <------  |   | <-------- |   |
|   |          |   |           |   |
 ---            ---              ---
```

Notation: Online Algorithm (OLA)

- in round i:
- probabilities of actions $\boldsymbol{\pi}^i = (\pi_1^i, \ldots, \pi_k^i)$
- choose action $j^i \sim \boldsymbol{\pi}^i$.
- payoffs $\mathbf{v}^i = (v_1^i, \ldots, v_k^i)$
- expected payoff:

$$\begin{aligned} \mathbf{E}\left[v_{j^i}^i\right] &= \sum_j \mathbf{E}\left[v_{j^i}^i \mid j^i = j\right] \mathbf{Pr}\left[j^i = j\right] \\ &= \sum_j v_j^i \pi_j^i \\ &= \mathbf{v}^i \cdot \boldsymbol{\pi}^i \qquad \text{(vector dot product)} \end{aligned}$$

Challenge 1: what report to the algorithm?

Idea 1: give algorithm unbiased estimator of payoffs.

- if alg uses probabilities $\boldsymbol{\pi}^t = (\pi_1^i, \ldots, \pi_k^i)$
- and samples $j^i \sim \boldsymbol{\pi}^t$
- real payoffs are $\mathbf{v}^i = (v_1^i, \ldots, v_k^i)$
- learn only $v_{j^i}^i$
- report payoff $\tilde{\mathbf{v}}^i = (0, \ldots, \tilde{v}_{j^i}^i / \pi_{j^i}^i, \ldots, 0)$

Lemma 1: reported payoffs are unbiased estimators of true payoffs

Proof:

$$\begin{aligned}
\mathbf{E}\big[\tilde{v}_j^i\big] &= \mathbf{E}\big[\tilde{v}_j^i \,|\, j^i = j\big] \cdot \mathbf{Pr}\big[j^i = j\big] \\
&\qquad + \mathbf{E}\big[\tilde{v}_j^i \,|\, j^i \neq j\big] \cdot \mathbf{Pr}\big[j^i \neq j\big] \\
&= v_j^i/\pi_j^i \cdot \pi_j^i + 0 \cdot (1 - \pi_j^i) \\
&= v_j^i \\
\mathbf{E}\big[\tilde{\mathbf{v}}^i\big] &= \mathbf{v}^i
\end{aligned}$$

Note:

- reported payoffs in $[0, \tilde{h}]$ for $\tilde{h} = \max_{i,j} v_j^i/\pi_j^i$.
- if π_j^i is small, then $\tilde{v}_j^i = v_j^i/\pi_j^i$ can be big!

Challenge 2: keep $\tilde{h}$ small

Idea 2: pick random action with some minimal probability ϵ/k

Lemma 2: if $\pi_j^i \geq \epsilon/k$ then $\tilde{v}_j^i \leq \tilde{h} = kh/\epsilon$

Proof: $\tilde{v}_j^i = v_j^i/\pi_j^i \leq h/\epsilon/k = kh/\epsilon$

Note: explore-vs-exploit tradeoff with ϵ

Alg: MAB Reduction to OLA

In round i:

1. $\boldsymbol{\pi} \leftarrow$ OLA
2. draw $j^i \sim \tilde{\boldsymbol{\pi}}$ with
$$\tilde{\pi}_j^i = (1 - \epsilon)\, \pi_j^i + \epsilon/k$$
3. take action j^i
4. report $\tilde{\mathbf{v}}$ to OLA with
$$\tilde{v}_j^i = \begin{cases} v_j^i/\pi_j^i & \text{if } j = j^i \\ 0 & \text{otherwise.} \end{cases}$$

Thm: for payoffs in $[0, \tilde{h}]$, if OLA satisfies

$$\mathbf{E}[\mathrm{OLA}] \geq (1 - \epsilon)\,\mathrm{OPT} - \tilde{h}/\epsilon \ln k$$

then for payoffs in $[0, h]$, MAB satisfies

$$\mathbf{E}[\mathrm{MAB}] \geq (1 - 2\epsilon)\,\mathrm{OPT} - h\,k/\epsilon^2 \ln k$$

Recall: Exponential Weights (EW) satisfies assumption of Thm.

Cor: for payoffs in $[0, h]$, MAB-EW satisfies vanishing per round regret.

Proof: similar to before.

Exercise: MAB-EW

Recall: the per-round regret of exponential weights is $2h\sqrt{\ln k/n}$

- dependence on h is $O(h)$
- dependence on n is $O(\sqrt{1/n})$
- dependence on k is $O(\sqrt{\log k})$

Setup:

- payoffs in $[0, h]$
- apply the multi-armed-bandit reduction to the exponential weights algorithm
- Theorem: $\mathbf{E}[\mathrm{MAB}] \geq (1 - 2\epsilon)\,\mathrm{OPT} - h\,k/\epsilon^2 \ln k$
- optimally tune the learning rate ϵ for n rounds

Question: analyze the per-round regret, what is dependence on

- maximum payoff h?
- number of rounds n?
- number of actions k?

Analysis

"online learning works with unbiased estimators of payoffs"

Proof of Thm:

"$\mathbf{E}[\mathrm{MAB}] \geq (1-2\epsilon)\,\mathrm{OPT} - {}^{h\,k}\!/\!_{\epsilon^2} \ln k$"

0. let
   - $R = {}^{\tilde{h}}\!/\!_{\epsilon} \ln k$
   - $j^* = \mathrm{argmax}_j \sum_i v_j^i$

1. what does OLA guarantee?

   for any $\tilde{\mathbf{v}}^1, \ldots, \tilde{\mathbf{v}}^n$:

$$
\begin{array}{rll}
\mathrm{OLA} = \displaystyle\sum_i \boldsymbol{\pi}^i \cdot \tilde{\mathbf{v}}^i & \geq (1-\epsilon) \displaystyle\sum_i \tilde{v}^i_{j^*} - R \\
\mathbf{E}_{\boldsymbol{\pi},\tilde{\mathbf{v}}}[\mathrm{OLA}] = \displaystyle\sum_i \mathbf{E}_{\boldsymbol{\pi},\tilde{\mathbf{v}}}\left[\boldsymbol{\pi}^i \cdot \tilde{\mathbf{v}}^i\right] & \geq (1-\epsilon) \displaystyle\sum_i \mathbf{E}_{\tilde{\mathbf{v}}}\left[\tilde{v}^i_{j^*}\right] - R \\
\shortparallel & \qquad\qquad \shortparallel \\
\displaystyle\sum_i \mathbf{E}_{\boldsymbol{\pi}}\left[\boldsymbol{\pi}^i \cdot \mathbf{v}^i\right] & \geq (1-\epsilon) \displaystyle\sum_i v^i_{j^*} - R
\end{array}
$$

For left-hand side:

$$
\begin{aligned}
\mathbf{E}_{\boldsymbol{\pi}^i,\tilde{\mathbf{v}}^i}\left[\boldsymbol{\pi}^i \cdot \tilde{\mathbf{v}}^i\right] &= \sum_{\boldsymbol{\pi}^i} \mathbf{E}_{\boldsymbol{\pi}^i,\tilde{\mathbf{v}}^i}\left[\boldsymbol{\pi}^i \cdot \tilde{\mathbf{v}}^i \,|\, \boldsymbol{\pi}^i\right] \,\mathbf{Pr}\left[\boldsymbol{\pi}^i\right] \\
&= \sum_{\boldsymbol{\pi}^i} \left[\boldsymbol{\pi}^i \cdot \mathbf{v}^i\right] \,\mathbf{Pr}\left[\boldsymbol{\pi}^i\right] \\
&= \mathbf{E}_{\boldsymbol{\pi}^i}\left[\boldsymbol{\pi}^i \cdot \mathbf{v}^i\right]
\end{aligned}
$$

2. What is MAB performance?

$$
\begin{aligned}
\mathrm{MAB} &= \sum_i \tilde{\boldsymbol{\pi}}^i \cdot \mathbf{v}^i \\
&= (1-\epsilon) \sum_i \boldsymbol{\pi}^i \cdot \mathbf{v}^i + \frac{\epsilon}{k} \sum_j v_j^i \\
&\geq (1-\epsilon) \sum_i \boldsymbol{\pi}^i \cdot \mathbf{v}^i
\end{aligned}
$$

3. Combine (1) and (2), plug in R, Lemma~2:

$$
\begin{aligned}
\mathbf{E}[\mathrm{MAB}] &\geq (1-2\epsilon)\,\mathrm{OPT} - R \\
&= (1-2\epsilon)\,\mathrm{OPT} - {}^{\tilde{h}}\!/\!_{\epsilon} \ln k \\
&= (1-2\epsilon)\,\mathrm{OPT} - {}^{h\,k}\!/\!_{\epsilon^2} \ln k
\end{aligned}
$$