Adversarial Bandits

Henry Vu

Feb 9, 2024

Adversarial Bandits: Problem Settings

Given $A = \{1, 2, \ldots, K\}$ the set of action and (possibly) number of rounds $n \geq K$
for $t = 1, 2, \ldots, n$ **do**:

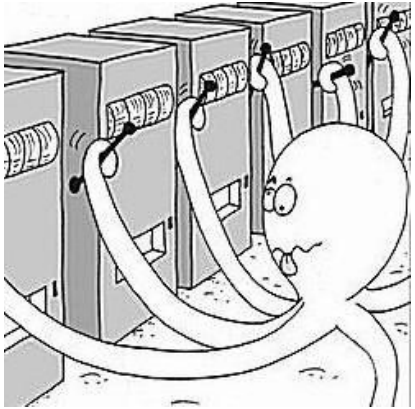
Algorithm pulls arm $A_t \in A$

A reward vector $(x_{t, 1}, x_{t, 2}, \ldots, x_{t, n})$ is given by the adversary (no underlying distribution)

Algorithm gets reward x_{t, A_t}

end for

Goal: Minimizing the static regret, i.e. *gap* between the fixed optimal action and the algorithm's choices



Need for Randomization

Example:

- If algorithm chooses action A, $reward_A = 0$, $reward_B = 1$
- If algorithm chooses action B, $reward_A = 1$, $reward_B = 0$

⇒ R_n is linear in n if the algorithm is *deterministic* since the adversary can always “trick” it.

⇒ The algorithm needs to randomize its actions to achieve sublinear regret

Adversarial vs Stochastic

- In stochastic bandits, total expected reward to compared to the maximum *expected* reward

- In adversarial bandits, total expected reward to compared to the maximum reward. If randomization is present, compared to the expected maximum reward.

$$
\begin{aligned}
R_n(\pi, \nu) &= \max_{i \in [K]} \mathbb{E}\left[\sum_{t=1}^{n} (X_{t_i} - X_{t,A_t})\right] \\
&\leq \mathbb{E}\left[\max_{i \in [K]} \sum_{t=1}^{n} (X_{t_i} - X_{t,A_t})\right] \\
&= \mathbb{E}\left[R_n(\pi, X)\right] \leq R_n^*(\pi)
\end{aligned}
$$

Adversarial Bandits: Full Information

- Algorithm gets reward $x_{A_t, t}$ but also observes the **whole** loss vector $x_t \in [0, 1]^K$ at the end of round t. Also called *prediction with expert advice*

- **Intuition**: Adjust importance of experts based on the loss they incur at the end of round t

Hedge Algorithm

Algorithm 1 Hedge Algorithm

$W_1(i) \leftarrow 1$

for $t = 1, \ldots, n$ **do**

  $\ell_t(i) \leftarrow$ loss of expert i, for each $i = 1, \ldots, K$

  $i_t \sim$ Expert index selected by drawing from $p_t(i) = \frac{W_t(i)}{\sum_{j=1}^{k} W_t(j)}$

  $\ell_t(i_t) \leftarrow$ Loss incurred at time t

  $W_{t+1}(i) \leftarrow W_t(i) \cdot e^{-\epsilon \ell_t(i)}$ (Weight update)

end for

Exponential weight update: more loss = less trust

Hedge Algorithm: Regret

By using exponential weight update, the **hedge algorithm** achieves performance described by:

$$E\left[\sum_{t=1}^{n} \ell_t(i_t)\right] \leq \min_i \sum_{t=1}^{n} \ell_t(i) + \epsilon \cdot E\left[\sum_{t=1}^{n} \ell_t^2(i_t)\right] + \frac{\log K}{\epsilon}$$

- 1nd term: minimum loss of repeatedly pulling a fixed arm
- 2nd term: can be bounded by time horizon n
- 3rd term: sublinear in K

⇒ Sublinear regret by choosing $\epsilon = \sqrt{\frac{8 \log K}{n}}$, $R_n \in \Theta(\sqrt{n \log K})$

Hedge Algorithm: Proof

We define: $\Phi_t := \sum_{i=1}^{N} w_t(i)$

$$\Phi_{t+1} = \sum_{i=1}^{K} w_t(i) e^{-\ell_t(i)} = \Phi_t \sum_{i=1}^{K} p_t(i) e^{-\ell_t(i)} \tag{1}$$

$$\leq \Phi_t \sum_{i=1}^{K} p_t(i)(1 - \ell_t(i) + \ell_t^2(i)) \tag{2}$$

$$= \Phi_t(1 - \epsilon p_t \ell_t + \epsilon^2 p_t \ell_t^2) \tag{3}$$

$$\leq \Phi_t \cdot \exp(-\epsilon p_t \ell_t + \epsilon^2 p_t \ell_t^2). \tag{4}$$

- (2) comes from inequality $e^{-x} \leq 1-x+x^2$ for $x \geq 0$
- (3) comes from writing as inner product and defining $\ell_t^2 := (\ell_t(1)^2, \ell_t(2)^2, \ldots, \ell_t(K)^2)$
- (4) comes from inequality $e^x \geq 1+x$ for $x \in \mathbb{R}$

Hedge Algorithm: Proof (cont.)

- By concatenating the above chain of inequalities for t = 1, …, n, we have for each expert i

$$w_1(i) \exp\left(\epsilon \sum_{t=1}^{n} \ell_t(i))\right) = w_n(i) \leq \Phi_n \leq \Phi_1 \cdot \exp\left(\sum_{t=1}^{n} \left[-\epsilon p_t \ell_t + \epsilon^2 p_t \ell_t^2\right]\right)$$

- Taking the log of both sides gives

$$-\epsilon \sum_{t=1}^{n} \ell_t(i) \leq \log K - \epsilon \cdot \sum_{t=1}^{n} p_t \ell_t + \epsilon^2 \cdot \sum_{t=1}^{n} p_t \ell_t^2.$$

- Finally, dividing both sides by ϵ we get the desired regret statement.

Adversarial Bandits: Partial Information

- Algorithm only observes reward $X_t = x_{A_t, t}$ of the chosen arm A_t and none of other arms.

Algorithm 2 Adversarial Bandits, Setup

$\{x_t\}_{t=1}^n := \{(x_{t,1}, \ldots, x_{t,K}) \in [0,1]^K\}_{t=1}^n$ ▷ Reward vectors selected by the adversary

for time $t = 1, \ldots, n$ **do**

  $P_t(A_t|H_{t-1}) \leftarrow$ Distribution of action at time t conditioned on H_{t-1}, selected by the learner.

  $A_t \sim P_t(A_t|H_{t-1}) \leftarrow$Learner's action at time t, sampled from P_t.

  $X_t := x_{t,A_t} \leftarrow$Reward observed by learner at time t.

end for

Exp3 Algorithm

- **Main idea**: Construct an estimator for the loss functions seen in the Hedge algorithm (in this setting, reward). 2 candidates:

- Candidate 1: $\hat{X}_{t,i} := \frac{\mathbf{1}\{A_t = i\} \cdot x_{t,i}}{P_{t,i}} = \frac{\mathbf{1}\{A_t = i\} \cdot x_{t,A_t}}{P_{t,A_t}} = \frac{\mathbf{1}\{A_t = i\} \cdot X_t}{P_{t,A_t}}$

  Unbiased estimator, since

$$\mathrm{E}[\hat{X}_{t,i}|\mathcal{H}_{t-1}] = \mathrm{E}\left[\frac{\mathbf{1}\{A_t = i\} \cdot X_t}{P_{t,i}}\right] = \frac{P_{t,i} \cdot x_{t,i}}{P_{t,i}} = x_{t,i}$$

  However, its variance can be very large

$$\mathrm{Var}\left[\hat{X}_{t,i}|\mathcal{H}_{t-1}\right] = \mathbb{E}\left[\frac{1_{\{A_t=i\}}}{P_{t,i}^2} \cdot X_t^2\right] - x_{t,i}^2 = x_{t,i}^2 \cdot \frac{1 - P_{t,i}}{P_{t,i}}$$

Exp Algorithm (cont.)

- Another alternative is to use

$$\hat{X}_{t,i} := 1 - \frac{\mathbf{1}\{A_t = i\}}{P_{t,i}} \cdot (1 - X_t) = 1 - \frac{\mathbf{1}\{A_t = i\}}{P_{t,i}} \cdot (1 - x_{t,i})$$

- This estimator is an interpretation of “loss”, and is also unbiased and has similar variance.
  However, the first estimator takes values in $[0,\infty)$, while the second estimator takes on values in $(-\infty,1]$. This observation affects the use of these estimators in Exp3

The Exp3 Algorithm

Algorithm 3 EXP3 Algorithm

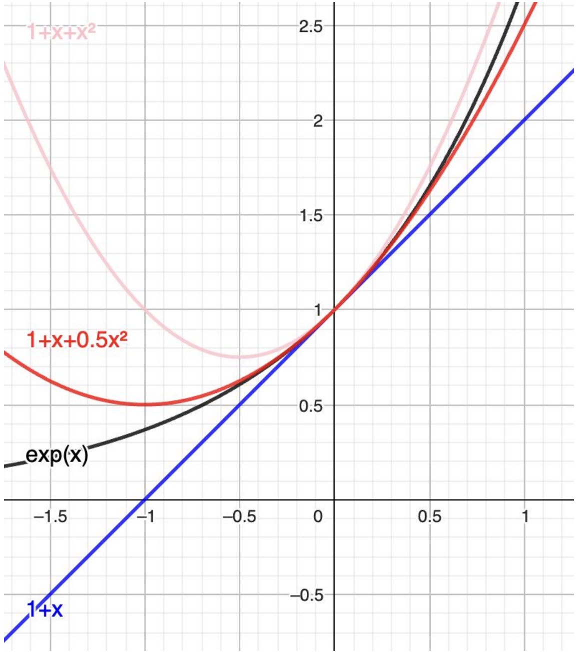
1: **Input:** horizon n, number of arms K, learning rate η;
2: Set $\hat{S}_{0,i} = 0$ for all $1 \leq i \leq K$;
3: **for** $t = 1, 2, \ldots, n$ **do**
4: $\quad P_{t,i} = \frac{\exp(\eta \hat{S}_{t-1,i})}{\sum_{j \in [k]} \exp(\eta \hat{S}_{t-1,j})}$;
5: $\quad$ Sample $A_t \sim P_t$, receive X_t;
6: $\quad$ Update $\hat{S}_{t,i} = \hat{S}_{t-1,i} + 1 - \frac{\mathbb{I}(A_t = i)(1 - X_t)}{P_{t,i}} = \hat{S}_{t-1,i} + \hat{X}_{t,i}$;
7: **end for**

- Similar to Hedge, but reward instead of loss and learning rate η instead of ϵ. Rewards of unrevealed arms are estimated based on observed $x_{t,i}$

Exp3 Algorithm Regret

- Main ideas (similar to Hedge):
  - The weight for a single arm is less than or equal to the sum of weights
  - The total weight doesn't grow too fast

- For proof, use inequalities:
  - $\exp(x) \geq 1+x$ for $x \in \mathbb{R}$
  - $\exp(x) \leq 1+x+x^2$ for $x \leq 1$ (Hedge)
  - $\exp(x) \leq 1+x+x^2/2$ for $x \leq 0$ (Exp3, tighter regret bound!)



Exp3 Regret Proof

- Idea 1:

$$\begin{aligned}
\exp(\eta \hat{S}_{t_i}) &\leq \sum_{j=1}^{K} \exp(\eta \hat{S}_{t_j}) = W_n \\
&= \frac{W_1}{W_0} \cdot \frac{W_2}{W_1} \cdots \frac{W_n}{W_{n-1}} \\
&= K \prod_{t=1}^{n} \frac{W_t}{W_{t-1}}
\end{aligned}$$

Exp3 Regret Proof (cont.)

- Idea 2:

$$\frac{W_t}{W_{t-1}} = \sum_j \frac{\exp(\eta \hat{S}_{t-1,j})}{W_{t-1}} \exp(\eta \hat{X}_{tj}) = \sum_j P_{tj} \exp(\eta \hat{X}_{tj}) \tag{6}$$

$$= \sum_j P_{tj} \exp(\eta) \exp(\eta(\hat{X}_{tj} - 1)) \tag{7}$$

$$\leq \exp(\eta)(1 + \eta \sum_j P_{tj} \hat{X}_{tj} + \eta^2 \sum_j P_{tj} \hat{X}_{tj}^2) \tag{8}$$

$$\leq \exp\left(\eta \sum_j P_{tj} \hat{X}_{tj} + \frac{\eta^2}{2} \sum_j P_{tj} (\hat{X}_{tj} - 1)^2\right) \tag{9}$$

- (8) comes from $\exp(x) \leq 1+x+x^2/2$ for $x \leq 0$
- (9) comes from $\exp(x) \geq 1+x$ for $x \in \mathbb{R}$

Exp3 Regret

- Taking the log both both sides and rearranging some terms, we obtain the following regret bound

Theorem 1 (Lattimore. Theorem 11.2). For rewards $x_{t,i} \in [0,1]$, and the learning rate tuned to $\eta = \sqrt{2\log(K)/(nK)}$, we have for any arm i

$$R_{n,i} \leq \sqrt{2nK\log(K)}$$

Achieves the minimax lower bound upto a factor of log(K)

$$R_n^* \geq c\sqrt{nK}$$

- However, Exp3 works well only in expectation, as we saw the variance can be large!

High-probability Bound on Regret: Exp3-IX

- Large variance when $P_{t,i}$ gets small
  ⇒ “smooth” out $P_{t,i}$ by adding constant $\gamma \geq 0$

$$\hat{Y}_{ti} = \frac{\mathbb{I}\{A_t = i\} Y_t}{P_{ti} + \gamma} \qquad \text{where} \quad \hat{Y}_{ti} = 1 - \hat{X}_{ti}$$

- IX = **I**mplicit E**X**ploration. Actions with large losses for which Exp3 would assign negligible probability are still **explored** occasionally.

- γ must be chosen carefully to not increase bias

Exp3-IX Regret

- Let $\eta_1 = \sqrt{\frac{2\log(K+1)}{nK}}$ and $\eta_2 = \sqrt{\frac{\log(K) + \log(\frac{K+1}{\delta})}{nK}}$
- Then the following holds

1 If Exp3-IX is run with parameters $\eta = \eta_1$ and $\gamma = \eta/2$, then

$$\mathbb{P}\left(\hat{R}_n \geq \sqrt{8.5nK\log(K+1)} + \left(\sqrt{\frac{nK}{2\log(K+1)}} + 1\right)\log\left(\frac{1}{\delta}\right)\right) \leq \delta.$$

2 If Exp3-IX is run with parameters $\eta = \eta_2$ and $\gamma = \eta/2$, then

$$\mathbb{P}\left(\hat{R}_n \geq 2\sqrt{(2\log(K+1) + \log(1/\delta))nK} + \log\left(\frac{K+1}{\delta}\right)\right) \leq \delta.$$

Thank you!