EE 290 Theory of Multi-armed Bandits and Reinforcement Learning Lecture 22 - 2/19/2021

Lecture 10: Hedge and Exp3 Algorithms

Lecturer: Jiantao Jiao *Scribe: Qiyang Li, Chih-Yuan Chiu*

1 Agenda

- Review the Hedge algorithm, presented in the previous lecture, and present its proof.
- Present the Exp3 algorithm in the restrictive bandit feedback setting.

2 Hedge Algorithm

Consider the Hedge algorithm (Algorithm 1), as presented in the previous lecture (Slivkins, Chapter 5.3 [1], Lattimore and Szepesvari, Chapter 11 [2]). Choose some fixed $\epsilon > 0$.

Algorithm 1: Hedge Algorithm.

```
$W_1(1) \leftarrow 1$
for time $t = 1, \cdots, T$ do
    $\ell_t(i) \leftarrow$ loss of expert $i$, for each $i = 1, \cdots, k$
    $i_t \sim$ 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).
```

The following theorem provides a guarantee of the performance of the Hedge algorithm (Lattimore and Szepesvari, Chapter 11 [2]). Below, we use the notation $[K] := \{1, 2, \cdots, K\}$

Theorem 1 (**Hedge Algorithm Performance**, [1], Section 5.3)**.** *The hedge algorithm achieves performance described by:*

$$\mathrm{E}\left[\sum_{t=1}^{T} \ell_t(i_t)\right] \leq \underbrace{\min_i \sum_{t=1}^{T} \ell_t(i)}_{(a)} + \underbrace{\epsilon \cdot \sum_{t=1}^{T} \mathrm{E}[l_t^2(i_t)]}_{(b)} + \underbrace{\frac{\log N}{\epsilon}}_{(c)}. \tag{1}$$

Remark The terms above can be interpreted as follows:

- (a) denotes the minimum loss incurred if a fixed arm were repeatedly pulled throughout the time horizon $[T]$.
- (b) describes the sum of the second moment of the losses. It can be ("trivially") upper bounded by T, if the losses $\ell_t(i)$ are (uniformly) bounded. When more structure regarding $\ell_t(i)$ is known, the second moment can be more explictly computed, resulting in a tighter upper bound for this term. This idea will appear in the upcoming discussion of the Exp3 algorithm.
- (c) This term describes how, as the number of experts, N, increases, it becomes more difficult to achieve performance close to that of the best expert. However, this term scales only sub-linearly in N, and thus will not deteriorate the algorithm performance too much. This term, together with the "trivial" bound for term in (b), yields a performance of $\Theta(\sqrt{T \log N})$ for a appropriately chosen $\epsilon > 0$.

Proof As with the proof of the Weighted Majority algorithm's performance, we define:

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

In particular, $\Phi_1 = N$, since $w_t(1) = 1$ at initialization. Thus, for each $t \in \{1, \cdots, T-1\}$,

$$\begin{aligned}
\Phi_{t+1} &= \sum_{i=1}^{N} w_t(i) e^{-\epsilon \ell_t(i)} \\
&= \Phi_t \sum_{i=1}^{N} \left[p_t(i) e^{-\epsilon \ell_t(i)} \right] && (2) \\
&\leq \Phi_t \sum_{i=1}^{N} \left[p_t(i)(1 - \epsilon \ell_t(i) + \epsilon^2 \ell_{t,sq}(i)) \right] && (3) \\
&= \Phi_t \left(1 - \epsilon p_t^\top \ell_t + \epsilon^2 p_t^\top \ell_{t,sq} \right) && (4) \\
&\leq \Phi_t \cdot \exp\left(-\epsilon p_t^\top \ell_t + \epsilon^2 p_t^\top \ell_{t,sq} \right). && (5)
\end{aligned}$$

Explanations for lines (2)-(5) are as given below:

- (2) follows from $p_t(i) = \frac{w_t(i)}{\sum_{j=1}^{N} w_t(j)}$,
- (3) follows by observing that $e^{-x} \leq 1 - x + x^2$ for each $x \geq 0$,
- (4) follows by interpreting the sum as an inner product, and defining $\ell_{t,sq} := \left(\ell_t(1)^2, \cdots, \ell_t(N)^2 \right)$,
- (5) follows by observing that $1 + x \leq e^x$ for each $x \in \mathbb{R}$.

By concatenating the above chain of inequalities across $t = 1, \cdots, T$, we have, for each expert i,

$$w_T(i) \leq \Phi_T \leq \Phi_1 \cdot \exp\left(\sum_{t=1}^{T} \left[-\epsilon p_t^\top \ell_t + \epsilon^2 p_t^\top \ell_{t,sq} \right] \right).$$

Taking the logarithm on both sides gives

$$-\epsilon \cdot \sum_{t=1}^{T} \ell_t(i) \leq \log N - \epsilon \cdot \sum_{t=1}^{T} p_t^\top l_t + \epsilon^2 \cdot \sum_{t=1}^{T} p_t^\top \ell_{t,sq}.$$

Finally, by dividing both sides by ϵ and rearranging terms, we obtain the desired theorem statement. □

Remark The Hedge algorithm has the above guarantee in expectation, but not (directly) with high probability. It took researchers a lot of time to make this algorithm perform reasonably well with high probability, and thus be useful in practice.

3 Exp3 Algorithm for Adversarial Bandits

3.1 Adversarial Bandits: Problem Setting

Here, we present the setting of the adversarial bandits problem that we will consider in this section. First, let K denote the (finite) number of actions, or "arms", let T denote the (finite) time horizon, and denote $[K] := \{1, \cdots, K\}, [T] := \{1, \cdots, T\}$. As before, let A_t and X_t denote the action taken and reward received at time t, for each $t \in [T]$.

Before the learner takes any action, an adversary secretly chooses an arbitrary sequence of rewards $(x_t)_{t=1}^T$, where $x_t := (x_{t,1}, \cdots, x_{t,K}) \in [0,1]^k$ denotes the k-dimensional reward vectors associated with time t, for each $t \in [T]$. In particular, $x_{t,k}$ corresponds to the reward that the adversary would give if the k^{th} arm is choosen at time t. At each $t \in [T]$, the learner produces a distribution P_t over the action space on $[K]$, conditioned on the history $\mathcal{H}_{t-1} := \{A_1, X_1, \cdots, A_{t-1}, X_{t-1}\}$. The learner's action at time t, A_t, follows the distribution P_t, and observes reward x_{t,A_t}. The sequence of above events is summarized in Algorithm 2 below.

Algorithm 2: Adversarial Bandits, Setup.

```
$\{x_t\}_{t=1}^T := \{(x_{t,1}, \cdots, x_{t,K}) \in [0,1]^k\}_{t=1}^T \leftarrow$ Reward vectors, secretly selected by the adversary.
for time $t = 1, \cdots, T$ do
    $P_t(A_t|\mathcal{H}_{t-1}) \leftarrow$ Distribution of action at time $t$ conditioned on $\mathcal{H}_{t-1}$, selected by the learner.
    $A_t \sim P_t(A_t|\mathcal{H}_{t-1}) \leftarrow$ Learner's action at time $t$, sampled from $P_t$
    $X_t := x_{t,A_t} \leftarrow$ Reward observed by learner at time $t$.
```

We use the *expected regret* $R_n(\pi, x)$, defined below, as a metric of performance for adversarial bandit algorithms:

$$R_n(\pi, x) := \max_{i \in [k]} \sum_{t=1}^T x_{t,i} - \mathrm{E}\left[\sum_{t=1}^T x_{t,A_t}\right]. \tag{6}$$

Note that $\max_{i \in [k]} \sum_{t=1}^T x_{t,i}$ denotes the maximum possible reward received, in hindsight, by pulling any fixed arm repeatedly throughout the time horizon $[T]$, i.e. the reward by repeatedly pulling the best fixed arm $i \in [K]$. Meanwhile, $\mathrm{E}\left[\sum_{t=1}^T x_{t,A_t}\right]$ denotes the expectation of the learner's total reward. The expected regret $R_n(\pi, x)$ captures the gap between these two quantities.

Another metric of performance is furnished by the *worst-case regret* $R_n^\star(\pi)$, defined by finding the most unfortunate expected regret over the set of all possible arbitrary reward sequences chosen by the adversary:

$$R_n^\star(\pi) := \sup_{x \in [0,1]^{T \times K}} R_n(\pi, x).$$

Here, $x := (x_1, \cdots, x_T) = (x_{1,1}, \cdots, x_{1,K}, \cdots, x_{T,1}, \cdots, x_{T,K}) \in [0,1]^{T \times K}$.

As shown below, the worst-case regret can be used to demonstrate that deterministic policies perform poorly on the adversarial bandit problem.

Proposition 2. *For any deterministic policy π:*

$$R_n^\star(\pi) \geq T\left(1 - \frac{1}{K}\right).$$

Remark

- Recall that algorithms for the full feedback setup had a performance of $\Theta(\sqrt{T \log K})$. Here, we can show that algorithms operating in this bandit setting can yield a performance of $\Theta(\sqrt{TK})$.
- Another motivation for why adversarial bandits are interesting to study: Algorithms that work for adversarial bandits would also work for stochastic bandit. However, the UCB1 and successive elimination algorithms that yield reasonable regret guarantees for stochastic bandits would not be reasonable for adversarial bandits, because of the proposition above.

3.2 Main Idea of The Exp3 Algorithm

The main idea of the Exp3 algorithm for adversarial bandits is to construct an unbiased estimator for the loss functions seen in the problem formulation of the Hedge algorithm. Below, we introduce two candidate unbiased estimators for $\{\hat{X}_{t,i} | t \in [T], i \in [K]\}$. As described above, let A_t be the learner policy, following the distribution defined by $P_t = (P_{t,1}, \cdots, P_{t,K})$ as follows:

$$P_{t,i} := \mathbb{P}(A_t = i | \mathcal{H}_{t-1}). \tag{7}$$

One choice of an unbiased estimator is given by:

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

Note that $\hat{X}_{t,i}$ is a well-defined estimator (i.e., it depends only on information in $\mathcal{H}_{t-1}$), since $\hat{X}_{t,i} = x_{t,A_t}/P_{t,A_t}$ if $i = A_t$, and equals 0 otherwise. In other words, for each $t \in [T]$, $i \in [K]$, the estimator $\hat{X}_{t,i}$ does not depend on any unobserved award chosen by the adversary. Moreover, $\hat{X}_{t,i}$ is unbiased since

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

Although $\hat{X}_{t,i}$ is unbiased, its (conditional) variance may be very large:

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

In particular, if $p_{t,i}$ is small for some arm $i \in [K]$ at some time $t \in [T]$, the estimator $\hat{X}_{t,i}$ would incur large variance. Surprisingly, this does not severely deteriorate the performance of the Exp3 algorithm, as will be demonstrated in the next lecture.

Another unbiased estimator—the one we will actually use in the Exp3 proof—is given by:

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

Similarly, this estimator can be verified to be well-defined and unbiased, with a variance that becomes extremely large when $p_{t,i}$ is small:

$$\begin{aligned}\mathrm{E}[\hat{X}_{t,i}] &= 1 - \mathrm{E}\left[\frac{\mathbf{1}(A_t = i)}{P_{t_i}} \middle| \mathcal{H}_{t-1}\right] + \mathrm{E}\left[\frac{\mathbf{1}(A_t = i) X_{t_i}}{P_{t_i}} \middle| \mathcal{H}_{t-1}\right] = x_{t_i} \\ &= 1 - 1 + x_{t,i} = x_{t,i}, \\ \mathrm{Var}[\hat{X}_{t,i}] &= (1 - x_{t,i})^2 \cdot \frac{1 - P_{t,i}}{P_{t,i}}.\end{aligned}$$

Remark The most difficult part of the proof for the Exp3 algorithm's performance lies in the construction of the unbiased loss estimator. The proof for the Exp3 algorithm's performance greatly resembles that of the Hedge algorithm. As such, we will need to control its variance term, analogous to term (b) in (1). This is where the unbiased loss estimator will play an important role.

References

[1] A. Slivkins, "Introduction to multi-armed bandits," *ArXiV*, 2019.

[2] T. Lattimore and C. Szepesvári, *Bandit Algorithms.* Cambridge University Press, 2020.