EE 290 Theory of Multi-armed Bandits and Reinforcement Learning Lecture 22 - 2/16/2021

Lecture 9: Weighted Majority and Randomized Weighted Majority

Lecturer: Jiantao Jiao *Scribe: Aviral Kumar*

In this lecture, we review the full feedback model and the adversarial setting. We will then review the weighted majority algorithm from last lecture, provide a proof for it, and then discuss a randomized version of the weighted majority algorithm. Finally, we will discuss the Hedge algorithm.

1 Review of Full-Feedback Model from Previous Lecture

In this model, the cost for all arms is observed alongside the arm which is pulled. We can view this as a game with an adversary who selects the cost of each arm. To summarize, in each round $t \in [T]$, first the adversary sets the cost $c_t(a)$ for each arm $a \in [K]$, then the learner picks an arm a, and observes the cost $c_t(a)$ for each arm and the game continues.

The goal in this setting is to do as well as the best expert in hindsight, given a family of experts. In each round $t \in [T]$, we observe a new observation, and each expert predicts an arm (out of two arms A/B) for that round. The algorithm we design then observes the expert predictions and chooses one of the arms to respond. The theorem we want to show here is:

Theorem 1. *Given a pool of N experts, assume the best expert makes L mistakes. Then,*

1. *There is an efficient deterministic algorithm that can guarantee at most $2(1+\varepsilon)L + \frac{2\log N}{\varepsilon}$ mistakes.*
2. *There is a randomized algorithm s.t. the expected number of mistakes is $\leq (1+\varepsilon)L + \frac{\log N}{\varepsilon}$,*

where $\varepsilon \in (0, \frac{1}{2})$ is a parameter of the algorithm.

To prove the first part of this theorem, we construct the weighted majority algorithm that intuitively follows the majority of the experts weighted by their accuracy over the previous rounds. To do so, a weight is maintained for each expert, and it is decreased whenever the expert makes a mistake. We present the algorithm next and provide a proof.

2 Weighted Majority Algorithm

We present the weighted majority algorithm below and then analyze the number of mistakes made.

Algorithm 1: Weighted Majority Algorithm

Parameter $\varepsilon \in (0, 0.5)$. N experts, T rounds, Arms $j \in \{A, B\}$. Initialize $\forall i \in [N], W_1(i) \leftarrow 1$.
for *round* $t \in [1, 2, \cdots]$ **do**
  Let $S_t(A)$ and $S_t(B)$ be the set of experts that choose A and B respectively.
  Define $W_t(A) = \sum_{i \in S_t(A)} W_t(i)$, $W_t(B) = \sum_{i \in S_t(B)} W_t(i)$.
  Choose $a_t \leftarrow A$ if $W_t(A) \geq W_t(B)$ and $a_t \leftarrow B$ otherwise
  for each expert $i \in [N]$, $W_{t+1}(i) \leftarrow W_t(i)$ if i is correct, and $W_{t+1}(i) \leftarrow W_t(i)(1-\varepsilon)$ otherwise.
end

Proof We now prove the weighted majority algorithm. Define $\Phi_t = \sum_{i=1}^{T} W_t(i)$. By definition, note that $\Phi_1 = N$ and $\Phi_{t+1} \leq \Phi_t$, since weights $W_t(i)$ are non-increasing for all t. Each time the algorithm makes a mistake, the weighted majority makes a mistake, and thus at least half the input weights would have made a mistake, else the algorithm will not choose the wrong arm due to the weighted majority condition. Assuming the correct arm was A (without loss of generality), we obtain that,

$$\Phi_{t+1} = \sum_{i=1}^{N} W_t(i) = \sum_{i \in S_t(A)} W_t(i) + \sum_{i \in S_t(B)} (1-\varepsilon) W_t(i) = \Phi_t - \varepsilon \sum_{i \in S_t(B)} W_t(i) \leq \Phi_t \left(1 - \frac{\varepsilon}{2}\right),$$

where the last step follows from the fact that $\sum_{i \in S_t(B)} W_t(i) \geq \Phi_t/2$, since the weighted majority makes a mistake.

Thus, if M_T is the total number of mistakes till round T, we get $\Phi_T \leq \Phi_1 \left(1 - \frac{\varepsilon}{2}\right)^{M_T}$. Also note that if the i-th expert makes $M_T(i)$ mistakes till time T, then $W_T(i) = (1-\varepsilon)^{M_T(i)}$. Next, note that since $W_T(i) \leq \sum_{i \in [N]} W_T(i) := \Phi_T$, we get:

$$\begin{aligned}
(1-\varepsilon)^{M_T(i)} &\leq \Phi_1 \left(1 - \frac{\varepsilon}{2}\right)^{M_T} = N \left(1 - \frac{\varepsilon}{2}\right)^{M_T} \\
\Rightarrow M_T(i) \log(1-\varepsilon) &\leq \log N + M_T \log \left(1 - \frac{\varepsilon}{2}\right).
\end{aligned}$$

Using the inequality that $-x - x^2 \leq \log(1-x) \leq -x \ \forall x \in (0, \frac{1}{2})$, we can simplify the above expression:

$$\begin{aligned}
-M_T(i)(\varepsilon + \varepsilon^2) &\leq \log N - M_T \frac{\varepsilon}{2} \\
&\Rightarrow M_T \leq 2(1+\varepsilon) M_T(i) + \frac{2 \log N}{\varepsilon} \ \forall i \in [N] \\
&\Rightarrow M_T \leq 2(1+\varepsilon) \min_{i \in [N]} M_T(i) + \frac{2 \log N}{\varepsilon} \leq 2(1+\varepsilon) L + \frac{2 \log N}{\varepsilon},
\end{aligned}$$

which completes the proof of part (1) of the theorem. □

Remark Note that by optimizing the bound with respect to ε, we obtain that $L(1+\varepsilon) + \frac{\log N}{\varepsilon} \geq L + 2\sqrt{\varepsilon \cdot \frac{\log N}{\varepsilon}} = L + 2\sqrt{\log N}$, where the second step holds by an application of AM-GM inequality. Hence, the number of mistakes made are only $O(\sqrt{\log N})$ more than the L mistakes made by the best expert.

3 Randomized Weighted Majority Algorithm

Next, we prove the second part of the theorem, where we devise a randomized efficient algorithm for the full-feedback adversarial setting. Instead of computing a weighted majority vote of all the experts, randomized weighted majority chooses and plays expert i at round t with probability $p_t(i) = \frac{W_t(i)}{\sum_{j=1}^{N} W_t(j)}$, and updates the weights $W_t(i)$ similarly to the weighted majority algorithm.

Formally we wish to show that the expected number of mistakes, $\mathbb{E}[M_T] \leq (1+\varepsilon) \min_i M_T(i) + \frac{\log N}{\varepsilon}$.

Proof Define $\Phi_t = \sum_{j=1}^{N} W_t(i)$. Define two indicator variables $\tilde{m}_t = \mathbb{I}(\text{we make mistake at time t})$ and $m_t(i) = \mathbb{I}(\text{expert } i \text{ makes a mistake at time } t)$. Then, we can express Φ_{t+1} as:

$$\Phi_{t+1} = \sum_{i \in [N]} W_t(i)(1 - \varepsilon m_t(i)) = \sum_{i \in [N]} W_t(i) - \sum_i \varepsilon W_t(i) m_t(i).$$

Now note that $p_t(i) = \frac{W_t(i)}{\Phi_t}$ and $\mathbb{E}[\tilde{m}_t] = \sum_i p_t(i) m_t(i)$, which means that we can express the second term in the expression for Φ_{t+1} as: $\sum_i W_t(i) m_t(i) = \Phi_t \sum_i p_t(i) m_t(i) = \Phi_t \mathbb{E}[\tilde{m}_t]$. Thus,

$$\Phi_{t+1} = \Phi_t - \varepsilon \Phi_t \mathbb{E}[\tilde{m}_t] = \Phi_t (1 - \varepsilon \mathbb{E}[\tilde{m}_t]) \leq \Phi_t \exp(-\varepsilon \mathbb{E}[\tilde{m}_t]),$$

where the last step follows from the inequality that $1 + x \leq \exp(x)$.

Next, following a similar strategy as the weighted majority algorithm, we note that $W_T(i) = (1-\varepsilon)^{M_T(i)}$, and that $W_T(i) \leq \Phi_T$, we note that:

$$
\begin{aligned}
(1-\varepsilon)^{M_T(i)} &\leq N \exp(-\varepsilon \mathbb{E}[\sum_{t=1}^{T} \tilde{m}_t]) = N \exp(-\varepsilon \mathbb{E}[M_T]) \\
\Rightarrow M_T(i) \log(1-\varepsilon) &\leq \log N - \varepsilon \mathbb{E}[M_T] \\
\Rightarrow -M_T(i)(\varepsilon + \varepsilon^2) &\leq \log N - \varepsilon \mathbb{E}[M_T],
\end{aligned}
$$

where the last inequality follows from the fact that for $x \in (0, 0.5), -x - x^2 \leq \log(1-x) \leq -x$. Thus, we get the required final statement:

$$\mathbb{E}[M_T] \leq (1+\varepsilon) \min_i M_T(i) + \frac{\log N}{\varepsilon}.$$

□

Next, we briefly discuss the Hedge algorithm, which will be formally proved in the next lecture.

4 Hedge Algorithm

The Hedge algorithm generalizes the randomized weighted majority algorithm for the case when we do not have binary mistake variables, but a (continuous) loss value that is obtained when the algorithm makes a mistake. We assume that each expert i incurs a loss value, $l_t(i) \geq 0$ at time t, and out goal is to minimize the cumulative loss incurred by our algorithm $\sum_{t=1}^{T} l_t(i_t)$, where i_t denotes the expert selected by our algorithm in round t against the best expert in hindsight, which attains a loss of $\min_i \sum_{t=1}^{T} l_t(i)$. The algorithm is shown below:

Algorithm 2: Hedge Algorithm

Parameter $\varepsilon > 0$. N experts, T rounds. Initialize $\forall i \in [N], W_1(i) \leftarrow 1$.
for *round* $t \in [1, 2, \cdots]$ **do**
  $l_t(i) \leftarrow$ loss of expert $i \in [N]$.
  Pick expert i with probability $p_t(i) = \frac{W_t(i)}{\sum_{j=1}^{N} W_t(j)}$.
  Incur loss $l_t(i_t)$, $i_t \sim p_t$
  Update weights for each expert $i \in [N]$, $W_{t+1}(i) \leftarrow W_t(i) \exp(-\varepsilon l_t(i))$.
end

In the next lecture, we will show that the Hedge algorithm achieves a performance bound shown below.

Theorem 2 (Hedge algorithm cumulative loss)**.** *The cumulative loss obtained by the Hedge algorithm is given bounded as:*

$$\mathbb{E}\left[\sum_{t=1}^{T} l_t(i_t)\right] \leq \underbrace{\min_i \sum_{i=1}^{T} l_t(i)}_{\text{best expert}} + \varepsilon \cdot \underbrace{\varepsilon \cdot \sum_{t=1}^{T} \mathbb{E}\left[l_t^2(i_t)\right]}_{\text{sum of second moment of loss}} + \frac{\log N}{\varepsilon}.$$