15-451/651 Algorithm Design & Analysis
Fall 2022, Recitation #9

Objectives

- Understand the analysis of the multiplicative weights algorithm by trying the analysis with different weight update rules
- See how the experts framework and the multiplicative weights algorithm can be used algorithmically to solve zero-sum games!

Recitation Problems

1. **(Expert analysis)** In lecture we saw that the simple procedure that multiplied the weight of each expert by $\frac{1}{2}$ whenever the expert made a mistake, resulted in

$$m = \text{ \#mistakes of algorithm } \leq 2.41(M + \log_2 n),$$

where M = #mistakes made by the best expert and n = # of experts. If we multiply the weight by 2/3 at each time, how does this analysis change? Let's see

- The total weight of the experts starts at _____
- Each time we make a mistake, the new total weight is at most _____ times the old weight
- If we make m mistakes and the best expert makes M mistakes, then

$$\_\_\_\_\_\_\_\_\_\_ \leq \text{final total weight of all experts} \leq \_\_\_\_\_\_\_\_\_\_$$

- Therefore, $m \leq$ __________________________

2. **(Playing games like an expert)** A cool application of the randomized weighted majority algorithm is that it can be used to play two-player zero-sum games! Suppose we have a two-player zero-sum games where all of the payoffs are zero (column player wins) or one (row player wins). We will play as the column player, and use the experts framework to help us pick our strategy. Specifically, we will treat each column as an expert and use the randomized weighted majority algorithm. Each column (expert) begins with a weight of 1.

    - According to the randomized weighted majority algorithm, each round, we should play column i with probability

    - We should reduce the weights of ____________________

      ____________________ by ____________________

    - How well does this strategy perform in the long run? Lets appeal to the theorem that we proved in lecture, which says that

      $$\text{our error rate} \leq \text{optimal error rate} + 2\sqrt{\frac{\ln n}{T}}$$

      How should we interpret "our error rate" and "optimal error rate" in game-theoretic terms in relation to this problem? (Hint: first think about what a "mistake" corresponds to)

      Our error rate: ____________________

      Optimal error rate: ____________________

    - Lastly, let's see how close this strategy is to optimal. Suppose that over the T rounds of the game, the row player played row i some portion p_i of the time (in other words, they played row i with probability p_i). Write the "optimal error rate" quantity from the previous part in terms of p. Using facts that we have learned about zero-sum games, what is an upper bound on this quantity that relates it to the game?

- Use the error rate bound for multiplicative weights that we wrote above and combine it with the upper bound from the previous part to show how good our strategy was

$$\underbrace{\rule{6cm}{0.4pt}}_{\text{"Our error rate"}} \leq \underbrace{\rule{6cm}{0.4pt}}_{\text{"Optimal error rate"}} + 2\sqrt{\frac{\ln n}{T}}$$

- What happens as $T \to \infty$?

Further Review

1. **(More expert analysis)** In lecture we saw that the simple procedure that multiplied the weight of each expert by $\frac{1}{2}$ whenever the expert made a mistake, resulted in

$$m = \text{ \#mistakes of algorithm } \leq 2.41(M + \log_2 n),$$

where M = #mistakes made by the best expert and n = # of experts. In Problem 1, we saw what happens if we replace 1/2 with 2/3. In general, it turns out that the closer this factor gets to 1, the better the bound. Show that if we reduce the weights by a factor of $(1-\epsilon)$ for $\epsilon \leq 1/2$, then the number of mistakes is

$$m \leq 2(1+\epsilon)M + O\left(\frac{\log n}{\epsilon}\right).$$

2. **(Fun with experts)** In class we saw the randomized weighted majority (RWM) algorithm, in which we were given n experts. Then over any sequence of T rounds, and any expert i, we had

$$\mathbb{E}[\text{number of mistakes by RWM}] \leq (1+\epsilon)m_i + \frac{\ln n}{\epsilon}.$$

Here m_i is the number of mistakes made by expert i until time T. In Section 4 of the notes, we observed that $m_i \leq T$, so dividing the above by T and choosing $\epsilon := \sqrt{\frac{\ln n}{T}}$ we get that for any i

$$\mathbb{E}[\text{rate of mistakes by RWM}] \leq \frac{m_i}{T} + 2\sqrt{\frac{\ln n}{T}}.$$

I.e., for any expert i (which includes the best expert at time T) the *average regret (versus that expert)*, which is our mistake rate minus that of the expert's mistake rate, goes to zero as $T \to \infty$. Above, we assumed we knew the time horizon T and hence could set $\epsilon = \sqrt{\frac{\ln n}{T}}$. What if we don't know T? Here's one algorithm: for $s = 1, 2, \ldots$, play 2^s rounds of RWM (starting from scratch) with $\epsilon = \sqrt{\frac{\ln n}{2^s}}$. Show that the average regret of this algorithm after time T is $O\left(\sqrt{\frac{\ln n}{T}}\right)$.

3. **(Experts with fractional loss – hard/optional)** In lecture we saw the randomized weighted majority algorithm, which scales the weight by $(1-\epsilon)$ when an expert makes a mistake. We bound the number of mistakes we make with the number of mistakes the best expert makes. Here, we are interested in generalizing this framework.

First, we will allow more than binary outcomes, so the experts are predicting from a set of possible outcomes. Then, instead of just being right or wrong, an expert's prediction can be valued from 0 to 1 (where 0 could mean a perfect prediction and 1 the worst prediction, with other values in between). We call this value the "loss", which generalizes the "mistakes" from our original framework. Once again, this is the quantity that we want to minimize.

Let $\mathcal{P}$ be all the possible outcomes. We define a matrix $M_{i,j}$, with $i \in [n], j \in \mathcal{P}$ to be the loss that expert i experiences when the outcome is j. For all i, j, we have $M_{i,j} \in [0,1]$. Similar to the algorithm in class, we initialize the weight of each expert to 1. To make a prediction, we randomly sample an expert with the weight. Our expected loss could be measured by summing over the expected loss of each round.

To use the loss matrix to update the weights, if the outcome of round t is j_t, for each expert, $w^{(t+1)} = w^{(t)}(1-\epsilon)^{M_{i,j_t}}$. Intuitively, expert i tends to make a decision that incurs M_{i,j_t} loss when the outcome is j_t.

Our expected loss each round, given that the outcome is j_t, is

$$\left(\sum_{i=1}^{n} w_i^{(t)} M_{i,j_t}\right) \Big/ \sum_{i=1}^{n} w_i^{(t)}$$

Let this be denoted by $M(E^t, j_t)$. We are interested in upper bounding $\sum_{t=1}^{T} M(E^t, j_t)$.

Let $\epsilon < \frac{1}{2}$. After T rounds, for any expert i, we want to show that

$$\sum_{t=1}^{T} M(E^t, j_t) \leq \frac{\ln n}{\epsilon} + (1+\epsilon)\sum_{t} M_{i,j_t}$$

Use these inequalities:

$$\begin{aligned}
(1-\epsilon)^x &\leq (1-\epsilon x) && \forall x \in [0,1] \\
(1+\epsilon)^{-x} &\leq (1-\epsilon x) && \forall x \in [-1,0] \\
\ln\left(\frac{1}{1-\epsilon}\right) &\leq \epsilon + \epsilon^2 && \forall \epsilon : 0 < \epsilon < \frac{1}{2} \\
\ln(1+\epsilon) &\geq \epsilon - \epsilon^2 && \forall \epsilon : 0 < \epsilon < \frac{1}{2}
\end{aligned}$$

and a similar potential function approach as in class to prove the bound above.

4. **(Fickle experts)** In class you saw the randomized weighted majority theorem, in which we were given n experts. Now suppose you don't just want to compare yourself to the best you could have done by choosing a single expert and sticking with them. Call an deterministic algorithm K-*fickle* if over the time horizon T, it follows the advice of some expert i_1 for the first t_1 steps, then i_2 for the next t_2 steps, etc, and then i_K for the last t_K steps, where each $i_j \in [n]$, $t_j \geq 0$ and $\sum_{j=1}^{K} t_j = T$. Give an algorithm such that for any K-fickle (deterministic) algorithm A,

$$\mathbb{E}[\text{\# mistakes by your algo}] \leq (\text{\# mistakes by } A)(1+\varepsilon) + \frac{O(K \log(nT))}{\varepsilon}.$$

Your algorithm is allowed to run in time $(nT)^{O(K)}$.