CSE 632: Analysis of Algorithms II: Randomized Algorithms **Spring 2024**

Lecture 19: Prediction with Expert Advice

Lecturer: Zongchen Chen

1 Binary Prediction

Consider the following prediction process involving three parties: the algorithm (player), n experts, and the nature (adversary).

for $t = 1$ to T **do**
  For each $i \in [n]$, expert i makes a prediction $x_i^t \in \{0, 1\}$
  Algorithm makes a prediction $\hat{x}^t \in \{0, 1\}$ ▷ Based on experts' predictions and all history
  Nature reveals the outcome $y^t \in \{0, 1\}$
end for

For each $i \in [n]$, let m_i be the number of mistakes made by expert i; i.e.,

$$m_i = \left|\left\{t \in [T] : x_i^t \neq y^t\right\}\right|.$$

Let m^* be the number of mistakes made by the best expert; i.e.,

$$m^* = \min_{i \in [n]} m_i.$$

Finally, let $\hat{m}$ be the number of mistakes made by the algorithm; i.e.,

$$\hat{m} = \left|\left\{t \in [T] : \hat{x}^t \neq y^t\right\}\right|.$$

The goal of the algorithm is to minimize $\hat{m}$ compared to m^*. We want to establish that $\hat{m} \leq \alpha m^* + \beta$ where α, β may depend on n, T. Ideally, we want to have $\alpha = 1$ and $\beta = o_n(T)$, which means that the fraction of mistakes in T rounds made by the algorithm and the best expert are asymptotically the same as T goes to infinity.

2 Halving Algorithm

As a warm-up, consider the special case where one of the experts is perfect and makes no mistake in all T rounds, and so $m^* = 0$. Our goal is to minimize $\hat{m}$. Let S^t be the set of experts that make no mistake up to time t. The algorithm maintains the set S^t for all t.

In each round of Algorithm 1:

- Either $\hat{x}^t = y^t$ (the algorithm is correct), in which case we have trivially $|S^t| \leq |S^{t-1}|$;
- Or $\hat{x}^t \neq y^t$ (the algorithm makes a mistake), in which case we have $|S^t| \leq \frac{1}{2}|S^{t-1}|$ since the majority of the experts in S^{t-1} are wrong and will be excluded from S^t.

Therefore, we have

$$\left|S^T\right| \leq \left(\frac{1}{2}\right)^{\hat{m}} \left|S^0\right| = \frac{n}{2^{\hat{m}}}.$$

Algorithm 1 Halving algorithm

$S^0 = \{1, \dots, n\}$
for $t = 1$ to T **do**
  For each $i \in [n]$, expert i predicts $x_i^t \in \{0, 1\}$
  Algorithm predicts $\hat{x}^t \in \{0, 1\}$ as the majority of x_i^t's where $i \in S^{t-1}$; i.e.,

$$\hat{x}^t = \begin{cases} 1, & \text{if } |\{i \in S^{t-1} : x_i^t = 1\}| > |\{i \in S^{t-1} : x_i^t = 0\}| \\ 0, & \text{otherwise} \end{cases}$$

  Nature reveals $y^t \in \{0, 1\}$
  Algorithm updates S^t from S^{t-1} by removing all $i \in S^{t-1}$ such that $x_i^t \neq y^t$
end for

Meanwhile, we have

$$\left|S^T\right| \geq 1,$$

since at least one of the experts is perfect. It follows that

$$1 \leq \left|S^T\right| \leq \frac{n}{2^{\hat{m}}} \quad \Longrightarrow \quad \hat{m} \leq \log_2 n.$$

Namely, Algorithm 1 makes at most $\log_2 n$ mistakes, which in particular is independent of T.

3 Weighted Majority Algorithm

In the general case where a perfect expert does not necessarily exist, we maintain a vector $w^t = (w_1^t, \dots, w_n^t)$ of weights of all experts which represents our confidence/evaluation for each expert. When making a prediction, we use a weighted majority vote among all experts.

Algorithm 2 Weighted majority algorithm

$w^0 = (1, \dots, 1)$
for $t = 1$ to T **do**
  For each $i \in [n]$, expert i predicts $x_i^t \in \{0, 1\}$
  Algorithm predicts $\hat{x}^t \in \{0, 1\}$ as the (w^{t-1})-weighted majority of all x_i^t's; i.e.,

$$\hat{x}^t = \begin{cases} 1, & \text{if } \sum_{i \in [n]:\, x_i^t = 1} w_i^{t-1} > \sum_{i \in [n]:\, x_i^t = 0} w_i^{t-1} \\ 0, & \text{otherwise} \end{cases}$$

  Nature reveals $y^t \in \{0, 1\}$
  for $i = 1$ to n **do**
    $w_i^t = \begin{cases} (1 - \varepsilon) w_i^{t-1}, & \text{if } x_i^t \neq y^t \\ w_i^{t-1}, & \text{otherwise} \end{cases}$
  end for
end for

Let $Z^t = \sum_{i=1}^n w_i^t$. In each round of Algorithm 2:

- Either $\hat{x}^t = y^t$ (the algorithm is correct), in which case we have trivially $Z^t \leq Z^{t-1}$;

- Or $\hat{x}^t \neq y^t$ (the algorithm makes a mistake), in which case we have

$$Z^t \leq \left(\frac{1}{2} + \frac{1}{2}(1-\varepsilon)\right) Z^{t-1} = \left(1 - \frac{\varepsilon}{2}\right) Z^{t-1},$$

since the weights of experts who are wrong, which have sum at least $\frac{1}{2}Z^{t-1}$, will decrease by a factor of $1-\varepsilon$.

Therefore, we have

$$Z^T \leq \left(1 - \frac{\varepsilon}{2}\right)^{\hat{m}} Z^0 = \left(1 - \frac{\varepsilon}{2}\right)^{\hat{m}} n.$$

Suppose i^* is a best expert with the fewest mistakes (i.e., $i^* \in \arg\min_{i\in[n]} m_i$), and so $m^* = m_{i^*}$. Then we have

$$Z^T = \sum_{i=1}^{n} w_i^T \geq w_{i^*}^T = (1-\varepsilon)^{m_{i^*}} w_{i^*}^0 = (1-\varepsilon)^{m^*}.$$

It follows that

$$(1-\varepsilon)^{m^*} \leq Z^T \leq \left(1 - \frac{\varepsilon}{2}\right)^{\hat{m}} n.$$

Taking logarithms on both sides, we deduce that

$$m^* \log\left(\frac{1}{1-\varepsilon}\right) \geq \hat{m} \log\left(\frac{1}{1-\varepsilon/2}\right) - \log n.$$

Since we have

$$\log\left(\frac{1}{1-\varepsilon}\right) = \varepsilon \pm O(\varepsilon^2) \quad \text{and} \quad \log\left(\frac{1}{1-\varepsilon/2}\right) = \frac{\varepsilon}{2} \pm O(\varepsilon^2),$$

we conclude that

$$\hat{m} \leq (2 + O(\varepsilon))m^* + O\left(\frac{\log n}{\varepsilon}\right). \tag{1}$$

4 Randomized Weighted Majority Algorithm

We hope to improve the constant $2 + O(\varepsilon)$ in Eq. (1) to $1 + O(\varepsilon)$, so that the performance of the algorithm matches the best expert in the long run, even though we may have no clue which expert is the best. Instead of taking a weighted majority vote as in Algorithm 2, we sample a random expert with probability proportional to the weight and then follow the prediction of this selected expert.

For each $i \in [n]$ and all $t \in [T]$, let ℓ_i^t be the indicator of whether expert i makes a mistake in round t of Algorithm 3; i.e.,

$$\ell_i^t = \begin{cases} 1, & \text{if } x_i^t \neq y^t; \\ 0, & \text{otherwise.} \end{cases}$$

Then we have

$$m_i = \sum_{t=1}^{T} \ell_i^t.$$

Algorithm 3 Randomized weighted majority algorithm

$w^0 = (1, \ldots, 1)$
for $t = 1$ to T **do**
  For each $i \in [n]$, expert i predicts $x_i^t \in \{0, 1\}$
  Algorithm chooses a random $i \in [n]$ with probability $\propto w_i^{t-1}$, i.e.,

$$\Pr(\text{pick } i) = \frac{w_i^{t-1}}{Z^{t-1}} \quad \text{where } Z^{t-1} = \sum_{i=1}^{n} w_i^{t-1};$$

  and then follows the prediction of expert i, i.e., set $\hat{x}^t = x_i^t$
  Nature reveals $y^t \in \{0, 1\}$
  for $i = 1$ to n **do**

$$w_i^t = \begin{cases} (1-\varepsilon) w_i^{t-1}, & \text{if } x_i^t \neq y^t \\ w_i^{t-1}, & \text{otherwise} \end{cases}$$

  end for
end for

Since the prediction of the algorithm is random, we define $\hat{\ell}^t$ to be the probability that the algorithm makes a mistake in round t, and $\hat{m}$ to be the expected number of mistakes made by the algorithm. Therefore, we have

$$\hat{\ell}^t = \sum_{i=1}^{n} \Pr(\text{pick } i \text{ in round } t) \cdot \ell_i^t = \sum_{i=1}^{n} \frac{w_i^{t-1}}{Z^{t-1}} \cdot \ell_i^t = \frac{1}{Z^{t-1}} \sum_{i=1}^{n} \ell_i^t w_i^{t-1}, \tag{2}$$

and

$$\hat{m} = \sum_{t=1}^{T} \hat{\ell}^t.$$

We have that

$$\begin{aligned} Z^t &= \sum_{i=1}^{n} w_i^t \\ &= \sum_{i=1}^{n} \left(1 - \varepsilon \ell_i^t\right) w_i^{t-1} \\ &= \sum_{i=1}^{n} w_i^{t-1} - \varepsilon \sum_{i=1}^{n} \ell_i^t w_i^{t-1} \\ &= Z^{t-1} - \varepsilon \hat{\ell}^t Z^{t-1} && \text{(by Eq. (2))} \\ &= \left(1 - \varepsilon \hat{\ell}^t\right) Z^{t-1} \\ &\leq e^{-\varepsilon \hat{\ell}^t} Z^{t-1}. \end{aligned}$$

Thus, it follows that

$$Z^T \leq \exp\left(-\varepsilon \sum_{t=1}^{T} \hat{\ell}^t\right) Z^0 = e^{-\varepsilon \hat{m}} n.$$

Meanwhile, if i^* is a best expert with the fewest mistakes, then we have

$$Z^T = \sum_{i=1}^{n} w_i^T \geq w_{i^*}^T = (1-\varepsilon)^{m_{i^*}} w_{i^*}^0 = (1-\varepsilon)^{m^*}.$$

Therefore, we deduce that

$$(1-\varepsilon)^{m^*} \leq Z^T \leq e^{-\varepsilon \hat{m}} n.$$

Taking logarithms on both sides, we obtain

$$\hat{m} \leq \left(\frac{1}{\varepsilon} \log\left(\frac{1}{1-\varepsilon}\right)\right) m^* + \frac{\log n}{\varepsilon} \leq (1+\varepsilon) m^* + \frac{\log n}{\varepsilon}, \tag{3}$$

where the last inequality is due to $\frac{1}{x}\log(\frac{1}{1-x}) \leq 1 + x$ for all $x \in [0, \frac{1}{2}]$.

We can reinterpret Eq. (3) as in the following theorem.

Theorem 1. *Suppose $T \geq 4 \log n$. If we set $\varepsilon = \sqrt{\frac{\log n}{T}}$, then it holds*

$$\hat{m} \leq m^* + 2\sqrt{T \log n}.$$

Proof. Since $m^* \leq T$, Eq. (3) implies

$$\hat{m} \leq m^* + \varepsilon T + \frac{\log n}{\varepsilon}.$$

The theorem then follows by plugging in $\varepsilon = \sqrt{\frac{\log n}{T}}$, which minimizes the right-hand side. □

5 More General Setting and Multiplicative Weight Update

We can generalize Algorithm 3 to a much more general setting. The predictions of experts can be arbitrary rather than 0/1. The goal of the algorithm is to choose an expert (randomly) and follow the prediction of the chosen expert. The outcome does not need to be directly related to the predictions and can be arbitrary. It suffices to have a penalty/loss function which maps a pair of a prediction and an outcome to a value measuring the performance/accuracy of the prediction. In this sense, the algorithm does not need to know either the predictions of experts or the outcome, but only needs to know the penalty for each expert.

1: **for** $t = 1$ to T **do**
2: Algorithm chooses an expert (randomly)
3: Nature reveals the penalty $\ell_i^t \in [-1, 1]$ for each expert i
4: **end for**

For each $i \in [n]$, the total penalty of expert i is

$$m_i = \sum_{t=1}^{T} \ell_i^t.$$

The total penalty of the best expert is

$$m^* = \min_{i \in [n]} m_i.$$

Algorithm 4 Multiplicative weight update algorithm (hedge algorithm)

$w^0 = (1, \dots, 1)$
for $t = 1$ to T **do**
    Algorithm chooses a random $i \in [n]$ with probability $\propto w_i^{t-1}$, i.e.,

$$\Pr(\text{pick } i) = \frac{w_i^{t-1}}{Z^{t-1}} \quad \text{where } Z^{t-1} = \sum_{i=1}^{n} w_i^{t-1}$$

    Nature reveals $\ell_i^t \in [-1, 1]$ for each $i \in [n]$
    for $i = 1$ to n **do**
        $w_i^t = e^{-\varepsilon \ell_i^t} w_i^{t-1}$
    end for
end for

Let $\hat{\ell}^t$ be the expected penalty of the algorithm in round t, and $\hat{m}$ be the expected total penalty of the algorithm, i.e.,

$$\hat{m} = \sum_{t=1}^{T} \hat{\ell}^t.$$

The following theorem can be proved similarly as Theorem 1.

Theorem 2. *Suppose $T \geq \log n$. If we set $\varepsilon = \sqrt{\frac{\log n}{T}}$, then it holds*

$$\hat{m} \leq m^* + 2\sqrt{T \log n}.$$