Learning from experts & Multiplicative Weights

In this lecture we start a new algorithm design technique called Multiplicative Weights Algorithms or online learning or learning from experts.

The expert Problem (or prediction with expert advice)

Suppose we have n experts that make predictions for something. e.g. It is going to rain or not, market goes up or down, etc.

- let $\mathcal{U}$ be the universe of possible out comes and at the start of day t each expert $i \in [n]$ makes a prediction $p_i^t \in \mathcal{U}$
- The algorithm receives all these predictions & must come up with its own prediction $o^t \in \mathcal{U}$.
- We then are given the actual outcome a^t.

The goal is to minimize the mistakes we make i.e. # of days our prediction was false.

- We obviously cannot do better than the best expert. Our goal is to be as close to the best expert in hindsight:

After, say, T number of days, we want our mistakes not (much) larger than that of the expert with min # of mistakes in those T days.

- Say m : # of mistakes by our alg.

  m_i : # of mistakes of expert $i \in [n]$

Goal: $m \approx m^* = \min_i m_i$

Lemma: There is an algorithm that makes $\lceil \log n \rceil$ mistakes if there is a perfect expert.

Proof: Suppose $m^* = 0$, i.e. there is an expert that makes no mistake.

- Each day let o^t be the majority of predictions
- At the end of each day remove experts that were wrong.

Note that for each day we make a mistake the # of remaining experts goes down by a factor of ≥ 2 (because the correct answer was not majority). $\longrightarrow$ # of mistakes $\leq \lceil \log n \rceil$ ∎

Now consider the general case (no perfect expert). We can generalize:

Lemma: There is an algorithm that makes at most $(m^* + 1)\lceil \log n \rceil$ mistakes.

proof: - think of time divided into "epochs"

- We run the algorithm of "perfect expert" until experts are removed.
- Bring back all experts & start a new epoch.
- Suppose $m^* = m_{i^*}$. Note that in each epoch we make at most $\lceil \log n \rceil$ mistakes.
- Also # of epochs $\leq m^*$: each time all experts are eliminated $\equiv$ i^* made a mistake
- Finally in the last epoch (after m^*) there are at most $\lceil \log n \rceil$ mistakes. ∎

The Weighted Majority Algorithm (WM)

The Weighted Majority Algorithm (WM)

We can improve by softening our penalty:
instead of eliminating an expert who erred,
discount their weight!

[Littlestone & Warmuth '89]

- For each expert i maintain weight w_i^t for each day t and let $w_i^t = 1$
- Each day $t \geq 1$, return weighted majority

$$a^t \leftarrow \underset{u \in \mathcal{U}}{\arg\max} \sum_{i:\, i \text{ predicts } u} w_i^t$$

- After seeing the outcome for each wrong expert i set: $w_i^{t+1} = w_i^t / 2$

Theorem: For any sequence of predictions, the number of mistakes made by WM is $\leq 2.41\,(m^* + \log n)$

proof: We the # of mistakes $\leq 2.41\,(m_i + \log n)$ $\forall i$

- Define potential function

$$\Phi^t = \sum_{1 \leq i \leq n} w_i^t$$

- Note:

$1 \le i \le n$

- Note:

1) $\Phi_1 = n$,

2) $\Phi^{t+1} \le \Phi^t \quad \forall t$

3) if WM algorithm makes a mistake in round t then total weight of wrong experts is more than correct ones:

$$\Phi^{t+1} = \sum_{i \text{ wrong}} w_i^{t+1} + \sum_{i \text{ correct}} w_i^{t+1}$$

$$= \frac{1}{2}\sum_{i \text{ wrong}} w_i^t + \sum_{i \text{ correct}} w_i^t$$

$$= \Phi^t - \frac{1}{2}\sum_{i \text{ wrong}} w_i^t$$

$$\le \frac{3}{4}\Phi^t$$

if after T rounds, i has made m_i mistakes and WM has made m mistakes then

$$\left(\frac{1}{2}\right)^{m_i} = w_i^{T+1} \le \Phi^{T+1} \le \Phi^1\left(\frac{3}{4}\right)^m = n\left(\frac{3}{4}\right)^m$$

$$\longrightarrow \quad m \le \frac{m_i + \log n}{\log \frac{4}{3}} \le 2.41(m_i + \log n)$$

Using a similar analysis and choosing $\varepsilon \in (0, \frac{1}{2})$

Theorem: For $\varepsilon \in (0, \frac{1}{2})$, Penalizing each wrong expert by a factor of $(1-\varepsilon)$ guarantees the # of mistakes by WM to be $m \leq 2(1+\varepsilon)m^* + O\left(\frac{\log n}{\varepsilon}\right)$

Proof: Same analysis implies

$\Phi^{t+1} \leq (1-\frac{\varepsilon}{2})\Phi^t$ and

$(1-\varepsilon)^{m_i} \leq \Phi^{T+1} \leq \Phi^1(1-\frac{\varepsilon}{2})^m = n(1-\frac{\varepsilon}{2})^m \longrightarrow$

$$m \leq \frac{-m_i \log(1-\varepsilon) + \ln n}{\varepsilon/2} \leq \frac{2m_i(\varepsilon+\varepsilon^2)}{\varepsilon} + O\left(\frac{\log n}{\varepsilon}\right)$$

(using the fact that $-\ln(1-\varepsilon) = \varepsilon + \frac{\varepsilon^2}{2} + \frac{\varepsilon^3}{3} + \cdots \leq \varepsilon + \varepsilon^2$)
$\varepsilon \in [0,1]$

This means the ratio gets close to 2. $\blacksquare$

Proposition: Any deterministic algorithm makes $\geq 2m^*$ mistakes on some input.

proof: If the algorithm is deterministic adversary knows the prediction based on history & prediction of experts.

suppose $n=2$: A always predicts 1
B always predicts 0

So the adversary can choose the outcome s.t. the algorithm is always wrong but at least one of A or B has $\leq \frac{1}{2}$ error rate!

The Multiplicative Weight update (randomized WM)

- Now we see the randomized version of weighted majority that is known as the MWU algorithm.
- We use randomization to bypass the issue of scenario in the prev. proposition.

Intuition: Use the weights exactly as before but now the prediction is drawn randomly proportional to current weights.

$$\Pr[o^t = u] = \frac{\sum_{i:\, i \text{ predicts } u} w_i^t}{\sum_i w_i^t}$$

- the update of w_i^t's is the same as before.

Theorem: For any $\varepsilon \leq \frac{1}{2}$, the expected number of mistakes made by randomized WM is:

$$\mathbb{E}[m] \leq (1+\varepsilon)m_i + O\left(\frac{\log n}{\varepsilon}\right)$$

Proof: Again define potential function

$$\Phi^t = \sum_i w_i^t \quad , \quad F^t = \frac{\sum_{i \text{ wrong}} w_i^t}{\sum_i w_i^t}$$

$\mathbb{E}[m] = \sum F^t$

$\sum_i w_i$

Note: $\mathbb{E}[m] = \sum_{t \in [T]} F^t$ as we make a mistake at each step t with probability F^t.

$$\Phi^{t+1} = \sum_{i \text{ correct}} w_i^{t+1} + \sum_{i \text{ wrong}} w_i^{t+1}$$

$$= \sum_{i \text{ correct}} w_i^t + \sum_{i \text{ wrong}} (1-\varepsilon) w_i^t$$

$$= \Phi^t \left((1-F_t) + (1-\varepsilon) F_t \right) = \Phi^t (1 - \varepsilon F_t)$$

$$\to (1-\varepsilon)^{m_i} \le \Phi^{T+1} = \Phi^1 \prod_{t=1}^{t} (1-\varepsilon F_t) \le n e^{-\varepsilon \sum_t F_t} = n e^{-\varepsilon \mathbb{E}[m]}$$

$$\to m_i \log(1-\varepsilon) \le \ln n - \varepsilon \mathbb{E}[m] \to$$

$$\mathbb{E}[m] \le (1+\varepsilon) m_i + O\left(\frac{\ln n}{\varepsilon}\right) \qquad \blacksquare$$