Lecture 20: The Multiplicative Weights Algorithm

David Woodruff
The Experts Problem

- n “experts” try to predict an outcome on each day

- Expert = someone with an opinion, not necessarily someone who knows anything

- For example, the experts could try to predict the stock market

<table>
<thead>
<tr>
<th>Expt 1</th>
<th>Expt 2</th>
<th>Expt 3</th>
<th>neighbor’s dog</th>
<th>truth</th>
</tr>
</thead>
<tbody>
<tr>
<td>down</td>
<td>up</td>
<td>up</td>
<td>up</td>
<td>up</td>
</tr>
<tr>
<td>down</td>
<td>up</td>
<td>up</td>
<td>down</td>
<td>down</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

...
The Experts Problem

• n “experts” predict an outcome on each of T days, \( t = 1, ..., T \)

• On day \( t \), the i-th expert predicts outcome \( \text{out}_i^t \)

• On day \( t \), you see \( \text{out}_1^t, ..., \text{out}_n^t \) and make your prediction \( \text{guess}^t \)

• Then you see the actual outcome \( \text{out}^t \) on day \( t \)

• You are correct if \( \text{guess}^t = \text{out}^t \) and wrong otherwise
The Experts Problem

- **Goal**: if the best expert is wrong on $M$ days, you want to be wrong on at most $M$ days, plus a little bit

- Don’t make assumptions on the input

- Don’t assume future looks like the past

- Application: experts predict stock market, you want to do as well as the best single expert in hindsight

*How should you choose your guess on each day?*
Simpler Question

• Suppose at least one expert is perfect, i.e., never makes a mistake
  • Don’t know which one

• Suppose each expert predicts one of two values: 0 or 1
  • E.g., stock market will go up or down

• Can we find a strategy that makes no more than \( \lfloor \log_2 n \rfloor \) mistakes?

• **Majority-and-halving**: On each day, take the majority vote of all experts
  • Each time you’re wrong, you can remove at least half the experts
  • After \( \lfloor \log_2 n \rfloor \) mistakes you’re left with the perfect expert

• Same guarantee if experts predict more than 2 values
  • You choose most frequent prediction. If wrong, at least half the experts are wrong
Can You Do Better?

- **Claim:** in the worst case, any strategy makes at least $\log_2 n$ mistakes.
- **Proof:** adversary method

  - **Day 1:** make the first $n/2$ experts say 0, and the second $n/2$ experts say 1
    - If predictor outputs 0, then say the best expert outputs 1
    - If predictor outputs 1, then say the best expert outputs 0
    - Perfect expert is either in $[1, n/2]$ or in $[n/2+1, n]$

  - **Day 2:** in each interval $[1, n/2]$ and $[n/2+1, n]$, make first half of the experts say 0 and second half of the experts say 1
    - If predictor outputs 0, then say the best expert outputs 1
    - If predictor outputs 1, then say the best expert outputs 0
    - Perfect expert is either in $[1, n/4]$, $[n/4+1, n/2]$, $[n/2+1, 3n/4]$, or $[3n/4+1, n]$

  - ...

  - Any strategy is incorrect on at least $\log_2 n$ days
No Perfect Expert

• Suppose best expert makes M mistakes

• How can we guarantee we make at most \((M+1)(\log_2 n + 1)\) mistakes?

• Run Majority-and-Halving, but after throwing away all experts, bring them all back in and start over

• In each “phase”, each expert makes at least 1 mistake, and you make at most \(\log_2 n + 1\) mistakes

• At most M finished phases, plus the last unfinished one
Doing Better

• If best expert makes $M$ mistakes, we make at most $(M+1)(\log_2 n + 1) = O(M \log_2 n)$ mistakes

• Can’t do better than best expert, who makes $M$ mistakes
  • Suppose only one expert who always says 1 and is wrong $M$ times

• Can’t do better than $\log_2 n$ mistakes

• But can we make at most $\approx M + \log_2 n$ mistakes instead of $\approx M \cdot \log_2 n$?
Weighted Majority Algorithm

• Throwing away an expert when it makes a mistake is too drastic

• Assign weight $w_i$ to i-th expert. Initialize all weights to 1

• On t-th day, compute sum of weights of experts who say 0, and sum of weights of experts who say 1

• Choose outcome with larger weight

• If an expert is wrong on day t, cut its weight in half
Weighted Majority Algorithm

• **Theorem:** If the best expert makes $M$ mistakes, then the weighted majority algorithm makes at most $2.41(M + \log_2 n)$ mistakes!

• **Proof:** Let $\Phi = \sum_i w_i$. Initially $\Phi = n$

• When we make a mistake, $\Phi_{\text{new}} \leq \frac{3}{4} \cdot \Phi_{\text{old}}$
  • At least half of the weight (which made the majority prediction) gets halved (because it made a mistake)

• If we don’t make a mistake, $\Phi_{\text{new}} \leq \Phi_{\text{old}}$
Weighted Majority Algorithm

• If we’ve made $m$ mistakes so far, $\Phi_{\text{final}} \leq \left(\frac{3}{4}\right)^m \cdot \Phi_{\text{init}} = \left(\frac{3}{4}\right)^m \cdot n$

• But best expert $i^*$ makes at most $M$ mistakes, so $\Phi_{\text{final}} \geq w_{i^*} \geq \left(\frac{1}{2}\right)^M$

• So $\left(\frac{1}{2}\right)^M \leq \Phi_{\text{final}} \leq \left(\frac{3}{4}\right)^m \cdot n$, or $\left(\frac{4}{3}\right)^m \leq 2^M \cdot n$

• Taking logs, $m \leq \frac{M + \log_2 n}{\log_2\left(\frac{4}{3}\right)} = 2.41(M + \log_2 n)$

• If best expert makes a mistake 10% of the time, we make a mistake 24% of the time (plus $\log_2 n$ which is negligible with enough days)
Improved Weighted Majority Algorithm

• **Only change:** if an expert is wrong on day $t$, multiply its weight by $1 - \epsilon$

• Still choose outcome given by the majority weight of experts in each day

• **Theorem:** If the best expert makes $M$ mistakes, then the weighted majority algorithm makes at most $2(1 + \epsilon)M + O(\frac{\log_2 n}{\epsilon})$ mistakes
Improved Weighted Majority Algorithm

- Each time we make a mistake, $\Phi_{\text{new}} \leq \left(1 - \frac{\varepsilon}{2}\right) \cdot \Phi_{\text{old}}$
  - At least half of the weight gets scaled by $1 - \varepsilon$

- If we’ve made $m$ mistakes so far, $\Phi_{\text{final}} \leq \left(1 - \frac{\varepsilon}{2}\right)^m \cdot \Phi_{\text{init}} = \left(1 - \frac{\varepsilon}{2}\right)^m n$

- $\Phi_{\text{final}} \geq w_i \geq (1 - \varepsilon)^M$

- $(1 - \varepsilon)^M \leq \Phi_{\text{final}} \leq \left(1 - \frac{\varepsilon}{2}\right)^m n$ or $\frac{1}{\left(1 - \frac{\varepsilon}{2}\right)^m} \leq \frac{n}{(1 - \varepsilon)^M}$

- So $m \ln \frac{1}{1 - \varepsilon^2} \leq M \cdot \ln \frac{1}{1 - \varepsilon} + \ln n$
  - Use $\ln \frac{1}{1 - \varepsilon^2} \geq \frac{\varepsilon^2}{2}$ and $\ln \frac{1}{1 - \varepsilon} \leq \varepsilon + \varepsilon^2$ for $\varepsilon \in [0, \frac{1}{2}]$, and divide both sides by $\varepsilon$
  - $m \leq 2M(1 + \varepsilon) + \frac{2 \ln n}{\varepsilon}$
Lower Bound for Deterministic Algorithms

• **Theorem:** If the best expert makes $M$ mistakes, then the weighted majority algorithm makes at most $2(1 + \epsilon)M + O\left(\frac{\log_2 n}{\epsilon}\right)$ mistakes!

• If best expert is wrong 10% of the time, we’re wrong 20% of the time

• 2-approximation is best possible for deterministic algorithms:
  • Suppose we have two experts - one always says 0 and one always says 1
  • If algorithm is deterministic, the adversary knows what prediction it will make on each day, so it can choose the opposite outcome
  • So algorithm incorrect on all days, but one expert is correct on at least half of the days
Randomized Weighted Majority Algorithm

• Assign weight $w_i$ to i-th expert. Initialize all weights to 1

• On each day, predict 1 with probability $\frac{\sum_{i \text{ says 1}} w_i}{\sum_i w_i}$, and predict 0 otherwise

• Equivalently, pick a random expert $i$ with probability $\frac{w_i}{\sum_j w_j}$ and choose that expert’s outcome

• When an expert makes a mistake, multiply its weight by $1 - \epsilon$
Randomized Weighted Majority Algorithm

• **Theorem**: If the best expert makes $M$ mistakes, then the expected number of mistakes of the randomized weighted majority algorithm makes at most $(1 + \varepsilon)M + \frac{\ln n}{\varepsilon}$

• Let $\Phi = \sum_i w_i$. Initially $\Phi = n$

• Having fixed the outcome on all days, the potential varies deterministically
Randomized Weighted Majority Algorithm

• Let $F_t$ be the fraction of total weight on the $t$-th day on experts that make a mistake on that day

• The expected number of mistakes we make is $\sum_t F_t$

• On day $t$: $\Phi_{\text{new}} = \Phi_{\text{old}} \cdot (1 - F_t) + \Phi_{\text{old}} \cdot F_t (1 - \epsilon) = \Phi_{\text{old}} (1 - \epsilon \cdot F_t)$

• $\Phi_{\text{final}} \leq n \cdot \prod_t (1 - \epsilon \cdot F_t) \leq n \cdot e^{-\epsilon \sum_t F_t}$ using that $1 + x \leq e^x$ for all $x$

• Also, $\Phi_{\text{final}} \geq (1 - \epsilon)^M$
Randomized Weighted Majority Algorithm

• Have shown: $$(1 - \varepsilon)^M \leq \Phi_{\text{final}} \leq n \cdot e^{-\varepsilon \sum_t F_t}$$

• Taking natural logs, $$\varepsilon \sum_t F_t \leq M \ln \frac{1}{1-\varepsilon} + \ln n$$

• Using $$\ln \frac{1}{1-\varepsilon} \leq \varepsilon + \varepsilon^2$$ for $$\varepsilon \in [0, \frac{1}{2}]$$, and dividing both sides by $$\varepsilon$$ we get:

  Expected number of mistakes = $$\sum_t F_t \leq M(1 + \varepsilon) + \frac{\ln n}{\varepsilon}$$
Understanding the Error Rate

• Expected number of mistakes = $\sum_t F_t \leq M(1 + \epsilon) + \frac{\ln n}{\epsilon}$

• Best expert makes at most $T$ mistakes, so $\sum_t F_t \leq M + \epsilon T + \frac{\ln n}{\epsilon}$

• Let $M/T$ be optimal “error rate”

• Our expected error rate is at most optimal error rate + $\epsilon + \frac{\ln n}{\epsilon T}$

• Setting $\epsilon = \left(\frac{\ln n}{T}\right)^{1/2}$, our error rate $\leq$ optimal rate + $2 \left(\frac{\ln n}{T}\right)^{1/2}$

• The last term is called the “regret”. As $T$ gets larger, the regret goes to 0