Lecture 24: Online Decision Making and Expert Problem

- Mistake Bounded Model
- MW/WM Algorithm.

Fix $\epsilon > 0$

$$\#\text{algo mistakes} \leq (1+\epsilon)\left(\#\text{mistakes of best expert}\right) + \frac{\log N}{\epsilon}$$

on this sequence

N experts 1 2 . . N

	1	2	3	. .	N	
day 1	Y	Y	N		Y ←	Y decision $\in \{Y/N\}$
actual	x	x	☺		x N ☹	mistake
day 2	Y	N	N		Y	

make a small # of mistakes $\leq$ (#mistakes made by the best of these experts) + ( )

- Record!
- HW?
- Thursday lecture
- Final?

① Use primitive

② Algo nice

③ Idea arises in diff contexts

mistakes we make $\leq$ [ ] $\times$ # mistakes of best expert

$+$ [ ]

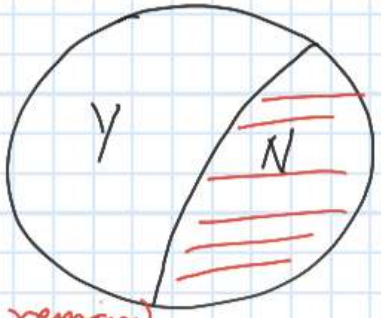
① Suppose $\exists$ expert makes 0 mistakes.

m = # mistakes of best expert $= 0$

Alg : — · — algo.

1	2		N
Y	Y	. . N	Y

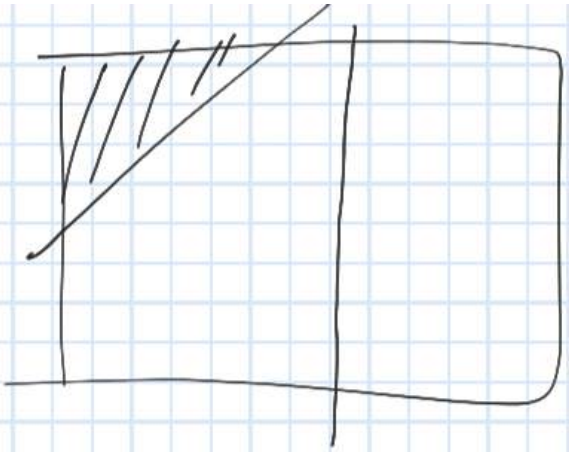
$Alg \leq \log_2 N$ mistakes



throw away

Every time I err, # remaining experts halves

$\Rightarrow$ # errors $\leq \log_2 N$. (finally $\exists$ at least one expert remain)



every time I err ⇒ # remaining experts halves

⇒ # mistakes ≤ $\log_2 N$

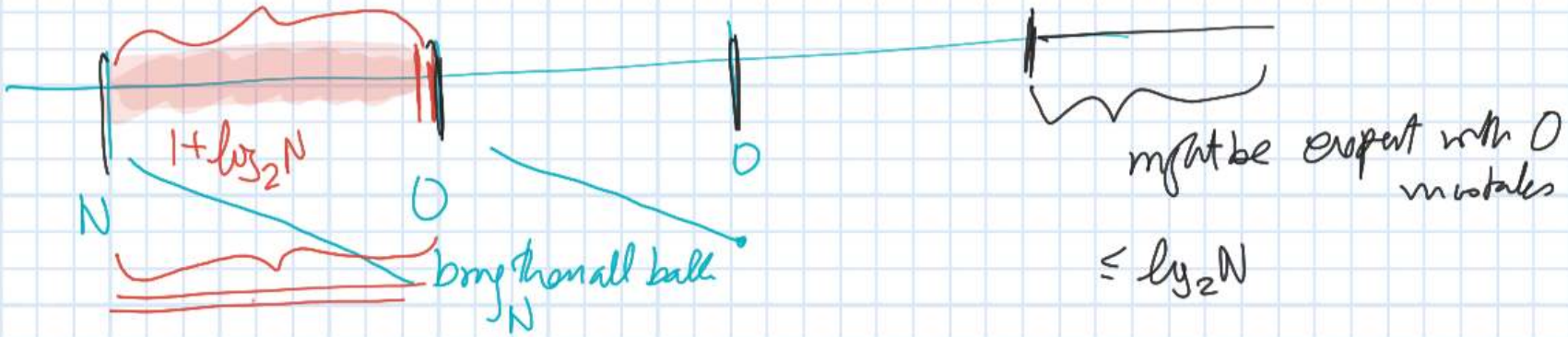
mistakes best expert

② $Alg \le (\log_2 N)(m+1)$

$m=0$

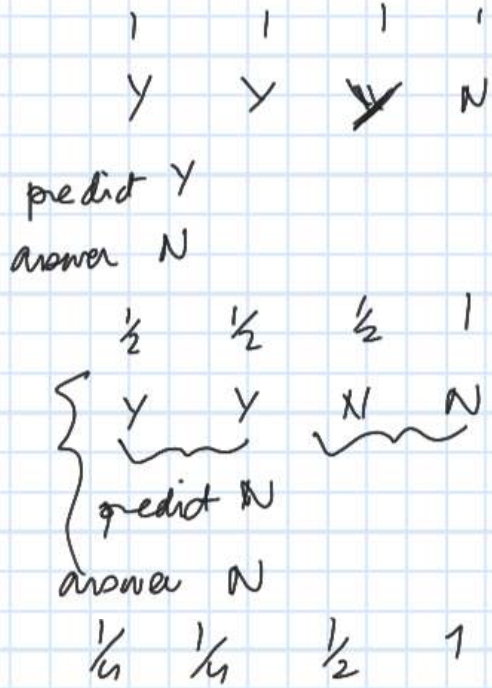
$Alg \le \log_2 N$

Run "perfect expert algo"



Rand. Multiplicative Weights / Weighted Majority (MW/WM)

- start with weight 1 on all N experts
- at each timestep,
  - Prediction = ~~weighted majority of current experts~~ random
    Y with prob $\frac{\text{weight on experts who say Y}}{\text{total weight}}$
  - look at truth,
    At wrong experts, ~~halve weight~~ reduce weight by $(1-\varepsilon)$



Thm: $Alg \leq 2.4\, m + O(\log_2 N)$

~~$(1+\varepsilon)$~~

$\forall$ expert i $\quad Alg \leq 2.4\, m_i + O(\log_2 N)$

(m_i = # mistakes i makes)

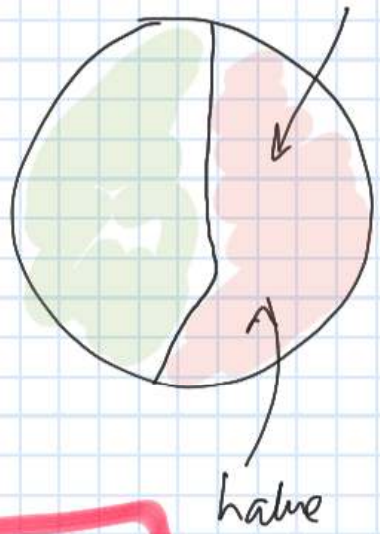
$$\text{Alg} \le 2.4\, m_i + O(\lg_2 N)$$

Pf: total weight W_t $\qquad W_0 = N$

① $W_t \le W_{t-1}$ $\forall t$

② If I make mistake, $W_t \le \left(\frac{3}{4}\right) W_{t-1}$

③ $W_T \ge$ weight of ith expert $= \left(\frac{1}{2}\right)^{m_i}$



after time T,

$$\left(\frac{1}{2}\right)^{m_i} \le W_T \le \left(\frac{3}{4}\right)^{\text{Alg}} \cdot W_0 = \left(\frac{3}{4}\right)^{\text{Alg}} \cdot N$$

now take log on both sides and simplify :)

Fact [Exercise] Reduce weight $\frac{1}{2} \longrightarrow (1-\varepsilon)$

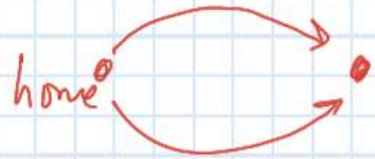
$$\Rightarrow Alg \leq \underbrace{2(1+\varepsilon)\, m_i} + \left(\frac{\log n}{\varepsilon}\right)$$

Do better? (Randomization) Random WM [Littlestone Warmuth]

~~weights = 1~~ $\forall i$ experts

Thm: $E[\text{mistakes of RWM}] \leq \underbrace{(1+\varepsilon)}\, m_i + O\left(\frac{\log N}{\varepsilon}\right)$

Randomness helps !!



$$A \leq (1+\varepsilon)m^* + \frac{\log N}{\varepsilon}$$

$$= m^* + \varepsilon m^* + \frac{\log N}{\varepsilon}$$

$$\leq m^* + \varepsilon \cdot T + \frac{\log N}{\varepsilon} =$$

sps $\exists\, \varepsilon$ such that $\varepsilon T = \frac{\log N}{\varepsilon} \Rightarrow \varepsilon^2 = \frac{\log N}{T} \Rightarrow \varepsilon = \sqrt{\frac{\log N}{T}}$

$$= m^* + 2\varepsilon T = m^* + 2\sqrt{T \log N}$$

$\varepsilon_t = \sqrt{\frac{\log N}{t}}$ ← learning rate

Repackaging

① $\forall$ time t, "loss of expert" $(\ell_1^t \quad \ell_2^t \quad \ldots \quad \ell_N^t) = \vec{\ell}^t \in [0,1]^N$

ℓ_1^t: loss of expert 1 at time t

~~prediction~~ action $(p_1^t \quad p_2^t \quad \ldots \quad p_N^t) = \vec{p}^t$

p_1^t: prob of choosing expert 1 @ t

$\sum_1^N p_i^t = 1$, $p_i^t \geq 0$

expected loss $= \sum_i^N p_i^t \cdot \ell_i^t = \langle p^t, \ell_i^t \rangle$

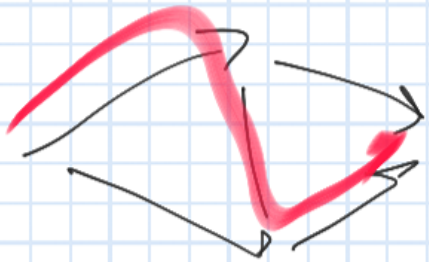
Algo: **Rand WM repackaged**

- $w_i = 1 \quad \forall i$
- at time t: action $p_i^t = \frac{w_i}{\sum_i w_i}$ $\quad p^t$

see loss vector ℓ^t — actual loss $= \langle p^t, \ell^t \rangle$

update weights: $w_i^{new} = w_i^{old}(1 - \varepsilon \ell_i^t)$

Thm: The algo RWM above satisfies $\forall i$

$$\sum_t \langle p^t, \ell^t \rangle \le (1+\varepsilon) \sum_t \ell_i^t + O\left(\frac{\log N}{\varepsilon}\right)$$

Max Flow :