# Lecture 15: Multiplicative weights and regret minimization.

**Online learning**: how to make decisions "on the fly" as data comes in.

e.g. Learning from Experts: predicting rain.

| | day 1 | 2 | 3 | ... | T |
|---|---|---|---|---|---|
| : | YES ✓ | NO ✓ | NO ✗ | | |
| : | NO ✗ | YES ✗ | NO ✗ | | |
| | NO ✗ | NO ✓ | YES ✓ | | |
| ⋮ | ⋮ | ⋮ | | | |
| : | YES ✓ | YES ✗ | YES ✓ | | |
| | ↓ | ↓ | ↓ | | |
| | YES ✓ | YES ✗ | NO ✓ | ... | |

$n$

Q: Can you do as well as the best expert in hindsight?

**Def:** For any algorithm, the regret of that algorithm is defined to be

$$\mathrm{Reg}(T) = \#\text{ mistakes by alg} - \#\text{ mistakes by best expert}$$

expected mistakes

e.g. one expert is always correct, rest are random

→ after $C\log n$ rounds, for every random expert:

$$\Pr[\text{perfect so far}] = \left(\tfrac{1}{2}\right)^{C\log n} = n^{-c}.$$

So by a union bound, $\Pr[\text{any random is perfect after } 2\log n \text{ rounds}]$
$$\leq (n-1)\cdot n^{-2} \leq \frac{1}{n}.$$

So you can identify best expert after $\log n$ rounds, and you pay total regret $O(\log n)$.

Can achieve something in general via multiplicative weights.

## Learning from Experts:

For $t = 1, \dots, T$:

1. All experts are presented w/ Y/N question.
2. Each expert predicts Y/N.
3. Alg chooses distribution over experts $p_t$
4. Alg samples expert $i$ w.p. $p_t(i)$.
5. Adversary reveals correct answer

both expert predictions + adversarial responses can be adaptively + adversarially chosen!

<u>Thm:</u> Multiplicative weights obtains

$$\mathrm{Reg}(T) \leq 2\sqrt{T \log n}.$$

sublinear in T!

## More general:

For $t = 1, \dots, T$

1. Alg chooses distribution $p_t(i)$.
2. Adversary reveals loss function $\ell_t(i) : [n] \to [-1, 1]$ important!
3. Algorithm incurs loss

$$\mathbb{E}_{i \sim p_t}[\ell_t(i)] = \sum_{i=1}^{n} p_t(i)\, \ell_t(i) = \langle p_t, \ell_t \rangle$$

total regret is defined as

$$\sum_{t=1}^{T} \mathbb{E}_{i \sim p_t}[\ell_t(i)] - \min_{i \in [n]} \sum \ell_t(i).$$

Learning from experts: $\ell_t(i) = \begin{cases} 1 & \text{if expert is wrong} \\ 0 & \text{if correct} \end{cases}$

Simple algorithms don't work:

e.g. Follow-the-Leader: pick the expert w/ fewest mistakes so far, breaking ties deterministically

| experts | 1 | 2 | ... | T |
|---|---|---|---|---|
| 1 | ✗ | ✓ | ✓ | |
| ⋮ | ✓ | ✗ | ✓ | ... |
| | ✓ | ✓ | ✗ | |
| ⋮ | ⋮ | ⋮ | ⋮ | |
| n | ✓ | ✓ | ✓ | |

Algo is <u>always</u> wrong, best possible in retrospect is $N/T$.

# Multiplicative Weights.

**Idea:** maintain "potential" vector $w_t \in \mathbb{R}^n$ that tracks our confidence in experts.

$w_t(i)$ is high $\rightarrow$ we think expert is good.

Multiplicatively downweight by loss.

Let $\varepsilon > 0$ be some decay (learning rate) parameter.

Intially let $w_1(i) = 1 \quad \forall\ i = 1, \dots, n$.

For $t = 1, \dots, T$

1. Let $\displaystyle p_t(i) = \frac{w_t(i)}{\Phi_t}$, $\displaystyle \Phi_t = \sum_{i=1}^{n} w_t(i)$.
2. Observe losses $\ell_t$
3. Update: for all $i = 1, \dots, n$, let
$$w_{t+1}(i) = w_t(i) \cdot e^{-\varepsilon \ell_t(i)}$$

For learning from experts:

Recall $\ell_t(i) = \begin{cases} 1 & \text{if expert } i \text{ is incorrect at time } t \\ 0 & \text{o.w.} \end{cases}$

So $\quad w_t(i) = e^{-\varepsilon(\#\text{ mistakes made by expert } i)}$

Set $\varepsilon = \ln 2$, so
$$w_t(i) = 2^{-(\#\text{ mistakes})}$$

Weight is halved for every mistake.

Consider the following, simpler (but worse) update:

At time $t$, output YES iff
$$\sum_{\substack{\text{experts who} \\ \text{said YES at } t}} \frac{w_t}{\cancel{\Phi(t)}} \geq \sum_{\substack{\text{experts who} \\ \text{said no}}} \frac{w_t}{\cancel{\Phi(t)}}$$

**Thm:** For learning from experts, this update achieves
$$\mathbb{E}\,[\#\text{ mistakes}] \leq (2.41) \cdot (\#\text{ mistakes of expert } i + C \cdot \log n)$$

should be 1!

**Pf:** We will use a kind of argument known as a potential argument.

Potential argument

$$\Phi_t = \sum_{i=1}^{n} w_t(i).$$

Idea: If we make a lot of mistakes, $\Phi_t$ is small.

But $\Phi_t \geq w_t$ so it can't be too small

$2^{(\# \text{ mistakes})}$

Initially: $\Phi_1 = n$, since $w_1(i) = 1 \ \forall i$.

At $t = T$,

$$\Phi_T \geq w_T(i) = 2^{-\# \text{ total mistakes of } i}$$

Claim: If weighted average was incorrect, then

$$\Phi_{t+1} \leq \frac{3}{4}\Phi_t.$$

pf: Suppose we said YES but it was actually NO.

we said YES $\Rightarrow$

$$\sum_{\text{YES experts}} w_t(i) \geq \sum_{\text{NO experts}} w_t(i).$$

$$\Rightarrow \sum_{\text{YES}} w_t(i) \geq \frac{1}{2}\Phi_t, \text{ or } \sum_{\text{NO}} w_t(i) \leq \frac{1}{2}\Phi_t.$$

at time $t+1$, all YES experts are halved.

time $t+1$:

$$\Phi_{t+1} = \frac{1}{2}\sum_{\text{YES at } t} w_t(i) + \sum_{\text{NO}} w_t(i)$$

$$= \frac{1}{2}\Phi_t + \frac{1}{2}\sum_{\text{NO}} w_t \leq \frac{1}{2}\Phi_t + \frac{1}{4}\Phi_t$$

$$\leq \frac{3}{4}\Phi_t$$

So:

$$\Phi_T \leq \Phi_1 \left(\frac{3}{4}\right)^{\# \text{ our mistakes}} = n \cdot \left(\frac{3}{4}\right)^{\# \text{ our mistakes}}$$

But $\Phi_T \geq w_T(i) \ \forall i$, so for any $i$:

$$w_T(i) = \left(\frac{1}{2}\right)^{\# \text{ mistakes of } i} \leq \Phi_T \leq n \cdot \left(\frac{3}{4}\right)^{\# \text{ our mistakes}}$$

taking logs:

$$(\# \text{ mistakes of } i) \cdot (-\log 2) \leq \log n + (\# \text{ ours}) \cdot (-\log 4/3)$$

$$\Rightarrow \# \text{ ours} \leq \frac{1}{\log 4/3} \cdot \left( (\# \text{ mistakes of } i) \cdot \log 2 + \log n \right).$$

To get constant 1:

1. Need to analyze randomized version.

2. Need to choose good step size.

**Thm:** For multiplicative weights update for general learning from experts achieves:

$$\sum_{i=1}^{T} \mathbb{E}_{i \sim p_t}[\ell_t(i)] - \min_{i \in [n]} \sum_{i=1}^{T} \ell_t(i) \le T\varepsilon + \frac{\log n}{\varepsilon},$$

If $|\ell_t| \le R$, pay an $R$ in both terms

$\rightarrow$ regret is multiplied by $R$

for learning rate $\varepsilon$.

To optimize $\varepsilon$: set two terms equal.

$$T\varepsilon = \frac{\log n}{\varepsilon} \Rightarrow \varepsilon^2 = \frac{\log N}{T}, \quad \varepsilon = \sqrt{\frac{\log N}{T}}.$$

Then $T\varepsilon = \frac{\log n}{\varepsilon} = \sqrt{T \log n}$.

Same basic argument structure!

$$\Phi_t = \sum_{i=1}^{n} w_t(i)$$

Main inductive invariant:

our expected regret.

$$\Phi_{t+1} \le \Phi_t \cdot \exp\left(\varepsilon^2 - \varepsilon \cdot \mathbb{E}_{i \sim p_t}[\ell_t(i)]\right)$$

$$\Phi_T \le n \cdot \exp\left(\varepsilon^2 T - \varepsilon \cdot \sum_{t=1}^{T} \mathbb{E}_{i \sim p_t}[\ell_t(i)]\right).$$

For any expert $i$,

$$w_T(i) = \prod_{t=1}^{T} e^{(-\varepsilon \ell_t(i))}$$

how much expert is total loss is.

$$= \exp\left(-\varepsilon \cdot \sum_{t=1}^{T} \ell_t(i)\right).$$

and $\Phi_T \ge w_T(i) \; \forall i$. So,

$$\exp(-\varepsilon \cdot (\text{expert } i \text{ loss})) \le n \cdot \exp(\varepsilon^2 T - \varepsilon(\text{our loss}))$$

$\Rightarrow$

$$-\varepsilon \cdot (\text{expert } i \text{ loss}) \le \log n + \varepsilon^2 T - \varepsilon(\text{our loss})).$$

$\Rightarrow \quad \varepsilon \cdot \text{Regret} \le \log n + \varepsilon^2 T.$

$\Rightarrow \quad \text{Regret} \le \frac{1}{\varepsilon} \log n + \varepsilon T.$

**Applications:** Many beyond online learning!

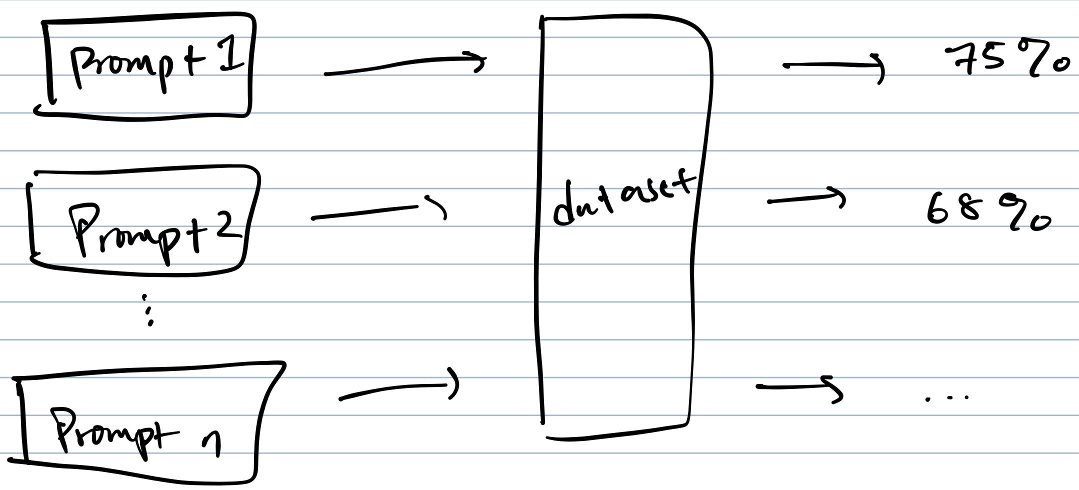
1 Finding Nash equilibria

2. "Boosting": Suppose I have a bunch of classifiers, each has accuracy 51%. But, they're "independent". Boosting → achives 99% accuracy via MW.

3. Optimization → see next lecture.

4. "Bandits": what if you only see reward of the expert you followed? MW variants (see e.g. Exp3) still work here.

e.g. Prompt optimization: Find the best prompt for come classification task.




Problem: evaluating a prompt on a data point requires an API call, which is expensive

→ can't query full data set!

Idea: treat each prompt as an expert.

At round $t$, choose prompt $i$

Then, to get loss, sample random data point, and

loss is $\begin{cases} 1 & \text{if prompt } i \text{ classifies correctly} \\ 0 & \text{o.w.} \end{cases}$

$\mathbb{E}_{\text{data}}[\text{loss}] = \text{loss of expert } i.$

So $\mathbb{E}[\text{Regret}] = \mathbb{E}[\text{our loss}] - \mathbb{E}[\text{best loss}]$.

to find best prompt: "best arm identification".

Heuristic: choose $i$ w/ largest $w_T(i)$.