15451 Spring 2023

Multiplicative Weights Algorithm

Elaine Shi

Stock Market

Stock Market

Every day the stock goes up or down

You discover it at the end of the day

You have to predict at the beginning of the day

Stock Market

Every day the stock goes up or down

You discover it at the end of the day

You have to predict at the beginning of the day

There are n experts to help you, each of them makes a prediction at the beginning of day

Goal

Design an algorithm whose performance is close to the **best expert in hindsight**

"minimize regret"

Discussion: why not compare with the best **algorithm** in hindsight

Goal

Design an algorithm whose performance is close to the **best expert in hindsight**

Warmup: you're promised that there exists an expert who is always right n : # experts

Warmup: you're promised that there exists an expert who is always right

Claim: there is an algorithm that makes at most $\log_2 n$ mistakes

Algorithm: take majority vote among experts
kick out whoever is wrong

If we make a mistake, then at least half of the remaining experts will be kicked out

Warmup: promised that the best expert makes **at most M** mistakes

Claim: there is an algorithm that makes at most **$(M+1)(\log_2 n + 1)$** mistakes

Algorithm: run the previous algorithm
once you have kicked out everyone, restart

phase: start $\longrightarrow$ kicked out everyone.

claim: in each phase, alg makes at most
$\log_2 n + 1$ mistakes

total mistakes

$$\leq M \cdot (\log_2 n + 1) + \log_2 n \leq (M+1)(\log_2 n + 1)$$

Warmup: promised that the best expert makes **at most M** mistakes

Claim: there is an algorithm that makes at most $\mathbf{2.41\,(M + \log_2 n)}$ mistakes

Deterministic Weighted Majority

- Initially, every expert has weight 1.
- When an expert makes a mistake, half its weight.
- Output the prediction of the weighted majority.

Analysis

sum of weights

Idea: relate Φ to perf of our alg / perf of the best expert

Let $\Phi := \sum_{i=1}^{n} w_i$

Analysis

Let $\Phi := \sum_{i=1}^{n} w_i$

Claim: If our alg makes a mistake, then

$$\Phi_{new} \leq \frac{3}{4}\Phi_{old}.$$

Intuition: if alg makes many mistakes, then final sum of weights is small

Analysis

Let $\Phi := \sum_{i=1}^{n} w_i$

Claim: If our alg makes a mistake, then

$$\Phi_{new} \leq \frac{3}{4}\Phi_{old}.$$

Note: Φ can never increase

Claim: If our alg makes a mistake, then

$$\Phi_{new} \leq \frac{3}{4}\Phi_{old}.$$

Proof:

Claim: If our alg makes a mistake, then

$$\Phi_{new} \le \frac{3}{4}\Phi_{old}.$$

Proof: if we make a mistake, at least half of the weight will get halved

"**Intuition:** if alg makes many mistakes, then final sum of weights is small"

Formally, if we make **m** mistakes, then

$$\Phi_{final} \leq (3/4)^m \cdot \Phi_{init} = (3/4)^m \cdot n.$$

OTOH: if best expert makes few mistakes, final sum of weights must be high

Formally, if i* makes M mistakes, then

$$\Phi_{final} \geq w_{i^*} = (1/2)^M.$$

$$\Phi_{final} \leq (3/4)^m \cdot \Phi_{init} = (3/4)^m \cdot n.$$

$$\Phi_{final} \geq w_{i^*} = (1/2)^M.$$

$$m \leq \frac{1}{\log_2(4/3)} \cdot (\log_2 n + M)$$

$$\leq 2.41(\log_2 n + M)$$

$$(1/2)^M \leq (3/4)^m \cdot n \quad \Rightarrow \quad (4/3)^m \leq n2^M.$$

$$m \log_2(4/3) \leq \log_2 n + M$$

Claim: can improve the constant 2.41 to

$$2(1+\epsilon)M + O\left(\frac{\log n}{\epsilon}\right).$$

Idea: instead of halving, multiply by 1-ε

Assume $\varepsilon \le \frac{1}{2}$

Claim: can improve the constant 2.41 to

$$2(1+\epsilon)M + O(\frac{\log n}{\epsilon}).$$

Claim: 2 is optimal for any **deterministic** algorithm!

T: total # time steps

Why?

Construct an adversarial input that makes the det. alg. wrong every time.

Consider two experts, one always UP, one always Down

at least one of them makes $\leq \frac{1}{2}T$ mistakes

How can we overcome this 2 barrier?

How can we overcome this 2 barrier?

Use randomization!

(recall online paging in the last lecture)

Randomized weighted majority

(Multiplicative Weights Algorithm)

- Initially, every expert has weight 1.
- When an expert makes a mistake, multiply its weight by 1-ε
- Predict "up" with probability $\frac{\sum_{j \text{ says Up}} w_j}{\sum_j w_j}$;

Assume: ε ≤ 1/2

Randomized weighted majority

- Initially, every expert has weight 1.
- When an expert makes a mistake, multiply its weight by 1-ε
- Go with each expert i with prob $\frac{w_i}{\sum_j w_j}$

Theorem: suppose ε ≤ ½ and best expert makes M mistakes. Then the expected number of mistakes we make is at most

$$(1+\epsilon)M + \frac{\ln n}{\epsilon}.$$

$\le \sqrt{\frac{\ln n}{T}} \cdot T + \frac{\ln n}{\sqrt{\frac{\ln n}{T}}}$

$= 2\sqrt{T \cdot \ln n}$

T: total # days

$E[\text{our \# mistakes}] - OPT \le \varepsilon M + \frac{\ln n}{\varepsilon} \le \varepsilon \cdot T + \frac{\ln n}{\varepsilon}$

OPT ↑ best expert in hindsight

$\varepsilon T = \frac{\ln n}{\varepsilon}$

$\varepsilon = \sqrt{\frac{\ln n}{T}}$

Theorem: suppose ε ≤ ½ and best expert makes M mistakes. Then the expected number of mistakes we make is at most

$$(1+\epsilon)M + \frac{\ln n}{\epsilon}.$$

ε : learning rate

ε large: punish wrong expert more

Corollary:

Our expected # errors ≤ OPT + $2\sqrt{\frac{\ln n}{T}}$

Proof:

Proof: $\Phi := \sum_{i=1}^{n} w_i$

F_t: fraction of the total weight on the t th day on experts who make a mistake

Proof: $\Phi := \sum_{i=1}^{n} w_i$

F_t: fraction of the total weight on the t th day on experts who make a mistake

Q: What is the expected # total mistakes we make?

Proof: $\Phi := \sum_{i=1}^{n} w_i$

F_t: fraction of the total weight on the t th day on experts who make a mistake

Q: What is the expected # total mistakes we make?

A: $\sum_t F_t$.

E[# mistakes on day t] = Pr[mistake on day t] = F_t

linearity of expectation

Proof: $\Phi := \sum_{i=1}^{n} w_i$

day t: wrong $F_t \cdot \Phi_{old} \Rightarrow F_t \cdot \Phi_{old} \cdot (1-\epsilon)$; right $(1-F_t) \cdot \Phi_{old} \Rightarrow (1-F_t) \cdot \Phi_{old}$

F_t: fraction of the total weight on the t th day on experts who make a mistake

Claim: on the t-th day, $\Phi_{new} = \Phi_{old} \cdot (1 - \epsilon F_t)$.

$\Phi_{new} = F_t \cdot \Phi_{old} (1-\epsilon) + (1-F_t) \cdot \Phi_{old} = \Phi_{old} (F_t - \epsilon F_t + 1 - F_t)$

Intuition: if alg makes many mistakes in expectation, then final sum of weights is small

Proof: $\Phi := \sum_{i=1}^{n} w_i$

F_t: fraction of the total weight on the t th day on experts who make a mistake

Claim: on the t-th day, $\Phi_{new} = \Phi_{old} \cdot (1 - \epsilon F_t).$

Proof:

Intuition: if alg makes many mistakes in expectation, then final sum of weights is small

Formally,

initial sum weights

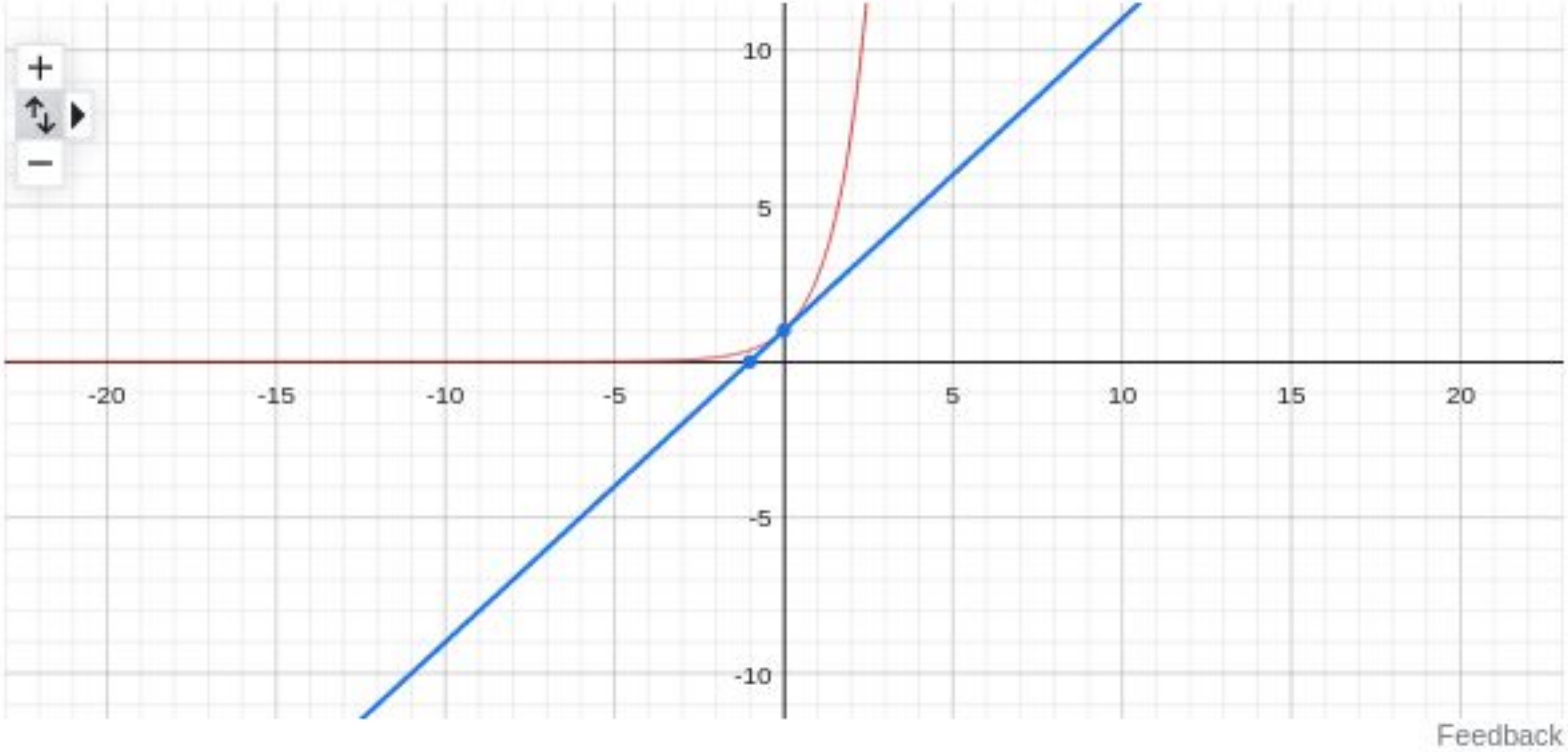
$$\Phi_{final} = n \cdot \prod_t (1 - \epsilon F_t) \leq n \cdot e^{-\epsilon \sum_t F_t}$$

$\leq n \cdot \prod_t e^{-\varepsilon F_t}$

$x = -\varepsilon F_t$

Recall: $1 + x \leq e^x$

Graph for **1+x**, e^x



OTOH: if best expert makes few mistakes, final sum of weights must be high

$$\Phi_{final} \geq (1-\epsilon)^M$$

$$\Phi_{final} \leq n \cdot e^{-\epsilon \sum_t F_t}$$

$$\Phi_{final} \geq (1-\epsilon)^M$$

$$\Phi_{final} \leq n \cdot e^{-\epsilon \sum_t F_t}$$

$$\Phi_{final} \geq (1-\epsilon)^M$$

$$(1-\epsilon)^M \leq n \cdot e^{-\epsilon \sum_t F_t}$$

$$\Phi_{final} \leq n \cdot e^{-\epsilon \sum_t F_t}$$

$$\Phi_{final} \geq (1 - \epsilon)^M$$

$$m \ln(1-\varepsilon) \leq \ln n - \varepsilon \cdot \textstyle\sum_t F_t$$

$$(1 - \epsilon)^M \leq n \cdot e^{-\epsilon \sum_t F_t} \quad \Rightarrow \quad \epsilon \sum_t F_t \leq M \ln \frac{1}{(1-\epsilon)} + \ln n.$$

$$\Phi_{final} \leq n \cdot e^{-\epsilon \sum_t F_t}$$

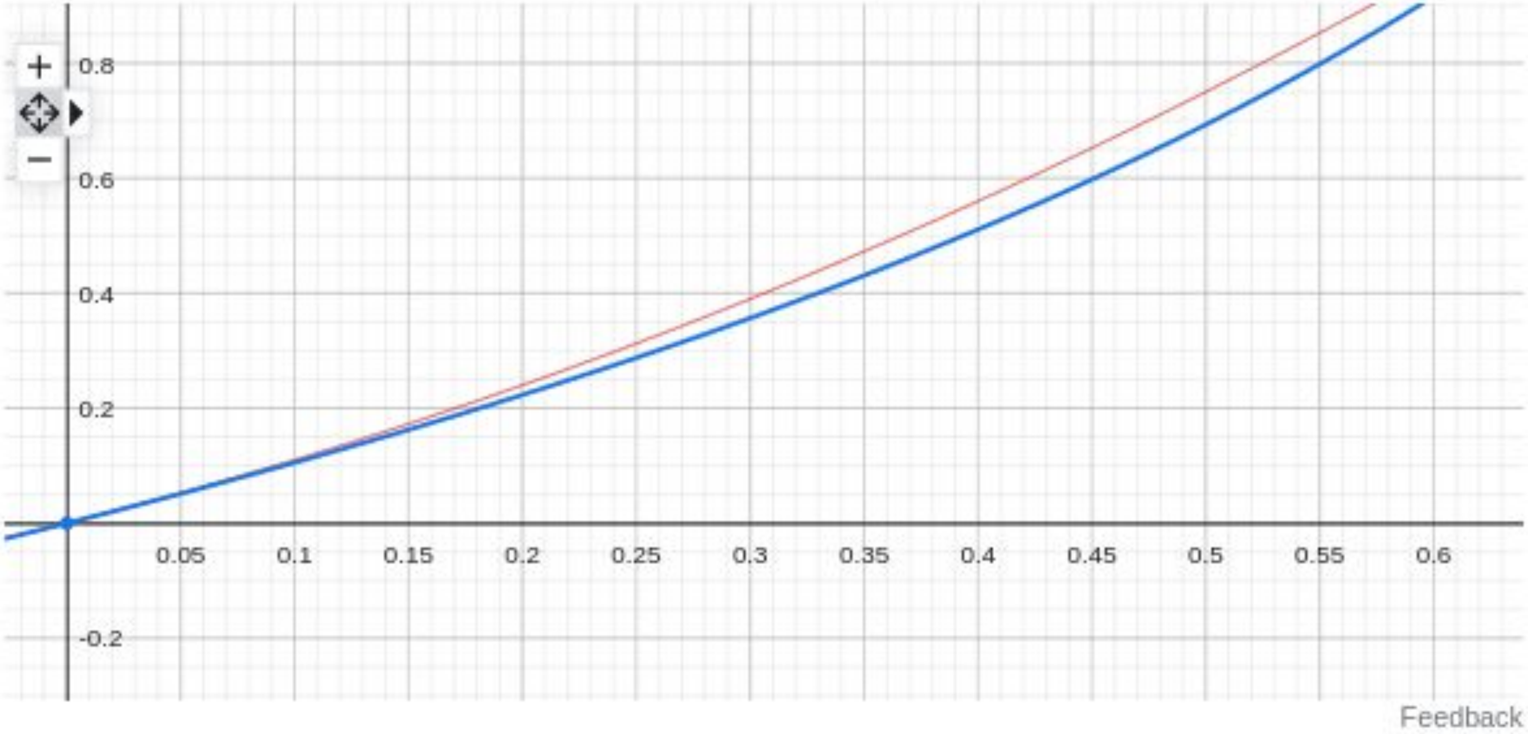
$$\Phi_{final} \geq (1-\epsilon)^M$$

$$(1-\epsilon)^M \leq n \cdot e^{-\epsilon \sum_t F_t} \quad \Rightarrow \quad \epsilon \sum_t F_t \leq M \ln \frac{1}{(1-\epsilon)} + \ln n.$$

Fact: $\ln \frac{1}{(1-\epsilon)} = -\ln(1-\epsilon) \leq \epsilon + \epsilon^2$

for $\epsilon \in [0, \frac{1}{2}]$

Graph for ln(1/(1-x)), x+x*x



$$\Phi_{final} \leq n \cdot e^{-\epsilon \sum_t F_t}$$

$\varepsilon \cdot \sum_t F_t \leq M(\varepsilon + \varepsilon^2) + \ln n$

$$\Phi_{final} \geq (1-\epsilon)^M$$

$\varepsilon + \varepsilon^2$

$$(1-\epsilon)^M \leq n \cdot e^{-\epsilon \sum_t F_t} \quad \Rightarrow \quad \epsilon \sum_t F_t \leq M \ln \frac{1}{(1-\epsilon)} + \ln n.$$

Fact: $\ln \frac{1}{(1-\epsilon)} = -\ln(1-\epsilon) \leq \epsilon + \epsilon^2$

for $\varepsilon \in [0, \frac{1}{2}]$

$$\sum_t F_t \leq M(1+\epsilon) + \frac{\ln n}{\epsilon}.$$

Extension: fractional rewards

reward $\in \{0, 1\}$
$\{-1, 1\}$

In each time step, each expert predicts some action, and at the end of the day, a reward ∈ [-1, 1] is revealed for each action

Performance of expert: sum of rewards over time

Applications of multiplicative weights

- Machine learning: AdaBoost, Winnow, Hedge
- Optimization (solving LP)
- Game theory
  - **see another proof of mini-max theorem in lecture notes**
- Operations research and online statistical decision-making
- Computational geometry
- Complexity theory
- Approximation algorithms
- Differential privacy