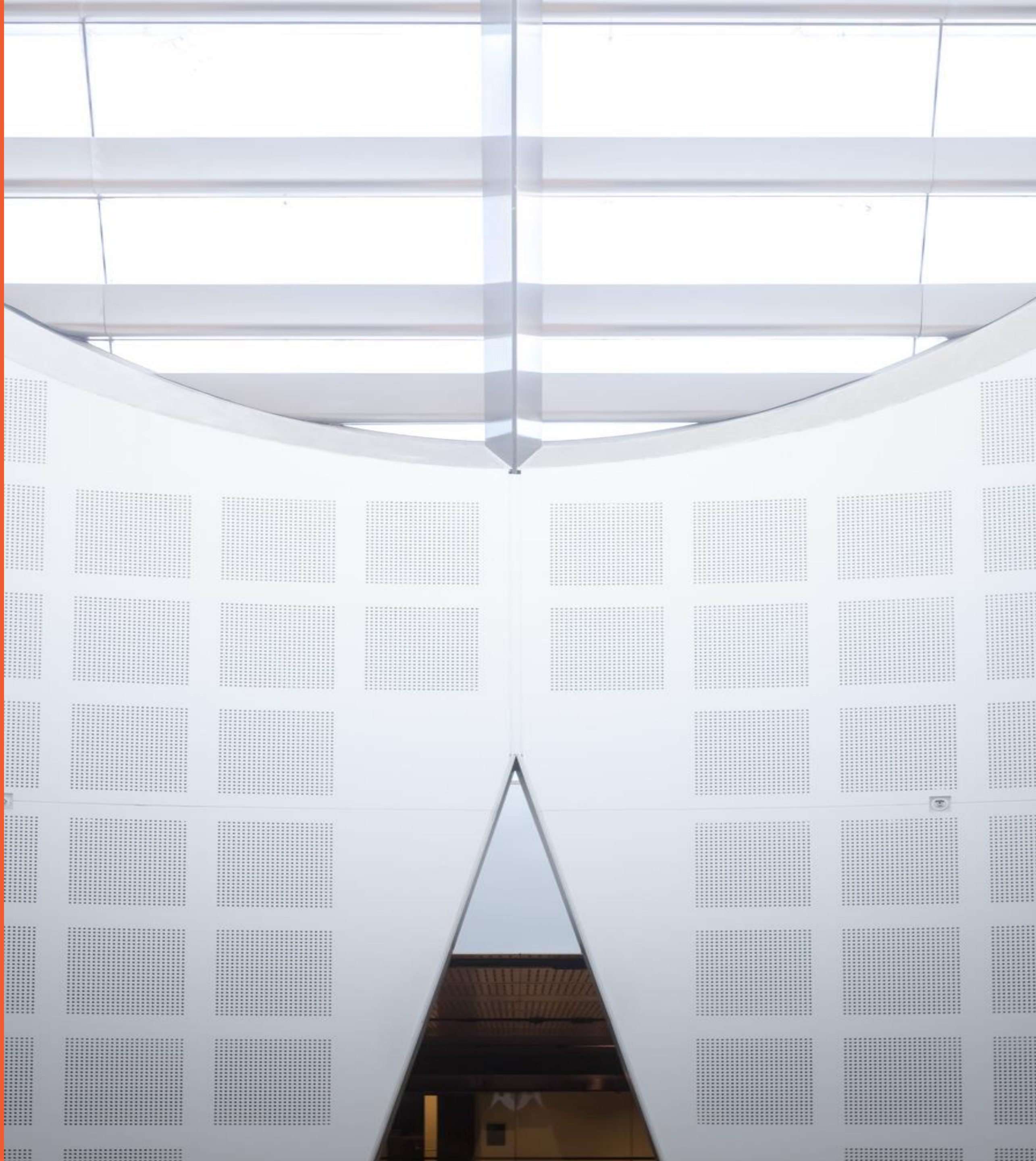
COMPx270: Randomised and Advanced Algorithms
Lecture 12: Learning from experts

Clément Canonne

School of Computer Science

THE UNIVERSITY OF SYDNEY



A question 📈

You invest in the stock market. Each morning, you have to decide whether to sell or buy.

A question 📈

You invest in the stock market. Each morning, you have to decide whether to sell or buy. At the end of the day, you see if you made the right decision. If you did, great: that day, you made money.

A question 📈

You invest in the stock market. Each morning, you have to decide whether to sell or buy. At the end of the day, you see if you made the right decision. If you did, great: that day, you made money.

However, you don't know anything about the stock market.

A question 📈

You invest in the stock market. Each morning, you have to decide whether to sell or buy. At the end of the day, you see if you made the right decision. If you did, great: that day, you made money.

However, you don’t know anything about the stock market.

But you have many friends who do: they’re all “experts.”

A question 📈

So every morning, before you make your decision, all those friends will give you their advice.

A question 📈

So every morning, before you make your decision, all those friends will give you their advice.

Some might collude, or be completely wrong, or even try to make you lose money. But each of them will tell you to either sell or buy.

A question 📈

Then, based on those many pieces of advice, **you** decide.

(And you do that again, every day.)

A question

Then, based on those many pieces of advice, **you** decide.

(And you do that again, every day.)

What is a good strategy to make money?

A question 📈

Then, based on those many pieces of advice, **you** decide.

(And you do that again, every day.)

What is a **provably** good strategy to make money?

Let’s make this formal

- There are n experts.
- Each day, $t = 1, \ldots, T$, each of them makes a prediction $v_{i,t} \in \{0,1\}$
- Based on those, you make your own prediction $\hat{u}_t \in \{0,1\}$
- Then the "true" value $u_t \in \{0,1\}$ is revealed
- If $\hat{u}_t \neq u_t$, this counts as a mistake (mistakes are bad)

Goal: minimise total number of mistakes $M = \sum_{t=1}^{T} 1_{\hat{u}_t \neq u_t}$

- There are n experts.
- Each day, $t = 1, \dots, T$, each of them makes a prediction $v_{i,t} \in \{0,1\}$
- Based on those, you make your own prediction $\hat{u}_t \in \{0,1\}$
- Then the "true" value $u_t \in \{0,1\}$ is revealed
- If $\hat{u}_t \neq u_t$, this counts as a mistake (mistakes are bad)

Goal: minimise total number of mistakes $M = \sum_{t=1}^{T} 1_{\hat{u}_t \neq u_t}$

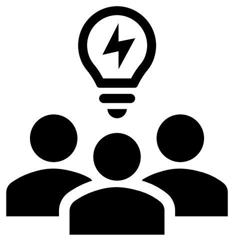
But what do we mean by this? We don't assume **anything** on the experts or on the true values. **They could even all be adversarial!**

- There are n experts.
- Each day, $t = 1, \ldots, T$, each of them makes a prediction $v_{i,t} \in \{0,1\}$
- Based on those, you make your own prediction $\hat{u}_t \in \{0,1\}$
- Then the "true" value $u_t \in \{0,1\}$ is revealed
- If $\hat{u}_t \neq u_t$, this counts as a mistake (mistakes are bad)

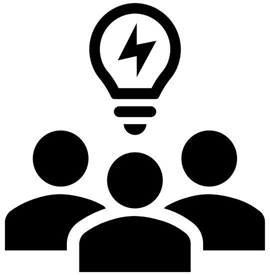
Goal: minimise total number of mistakes $M = \sum_{t=1}^{T} 1_{\hat{u}_t \neq u_t}$ compared to the best expert (whoever that is).

Not make much more mistakes than the **best advice in hindsight.**

Warmup: a Perfect Expert

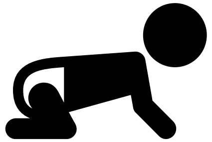


- There are n experts. **Suppose one of them (unknown) is always right.**
- Each day, $t = 1, \ldots, T$, each of them makes a prediction $v_{i,t} \in \{0,1\}$
- Based on those, you make your own prediction $\hat{u}_t \in \{0,1\}$
- Then the "true" value $u_t \in \{0,1\}$ is revealed
- If $\hat{u}_t \neq u_t$, this counts as a mistake (mistakes are bad)



Goal: minimise total number of mistakes $M = \sum_{t=1}^{T} 1_{\hat{u}_t \neq u_t}$

Theorem. There is a strategy guaranteeing $M \leq \; n-1$, regardless of T (even for $T = \infty$).



Set $S \leftarrow [n]$
for all $1 \leq t \leq T$ **do**
    Receive $v_{1,t}, \ldots, v_{n,t}$
    if $|S| \geq 1$ **then**
        Pick any $i \in S$ ▷ Lexicographically, for instance
        Choose $\widehat{u}_t \leftarrow v_{i,t}$
    else
        Choose $\widehat{u}_t \leftarrow 0$ ▷ Arbitrary
    Receive u_t ▷ Observe the truth
    $S \leftarrow S \setminus \{i \in S : v_{i,t} \neq u_t\}$ ▷ Remove all mistaken experts

Theorem. There is a strategy guaranteeing $M \leq n - 1$, regardless of T (even for $T = \infty$).

Proof.

1. Every time I make a mistake, $|S|$ decreases by 1.
2. At first, $|S| = n$.
3. Also, $|S| \geq 1$ always (there is a perfect expert!).

Potential argument.

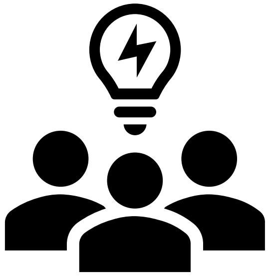
Potential function: $\phi = |S|$

$\phi_0 = n$

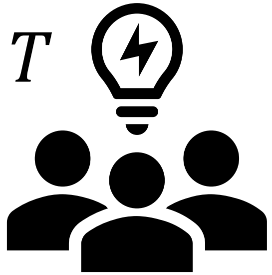
$\Delta\phi$

$\phi_t \geq 1 \quad \forall t$

Theorem. There is a strategy guaranteeing $M \leq \log_2 n$, regardless of T (even for $T = \infty$).



Claim. There is a strategy guaranteeing $M \le \log_2 n$, regardless of T (even for $T = \infty$).

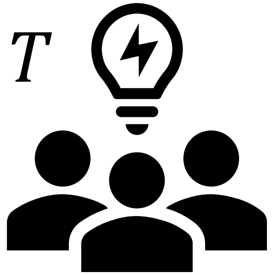


Algorithm: Start with $S = \{1,2, \dots n\}$. Each day, choose $\hat{u}_t$ to be the majority of advices from experts still in S. At the end of the day, remove from S all experts who predicted wrong.

Claim. There is a strategy guaranteeing $M \leq \log_2 n$, regardless of T (even for $T = \infty$).

Set $S \leftarrow [n]$
for all $1 \leq t \leq T$ **do**
  Receive $v_{1,t}, \ldots, v_{n,t}$
  if $|S| \geq 1$ **then**
    Choose $\widehat{u}_t \leftarrow \mathrm{maj}_{i \in S}\, v_{i,t}$ ▷ Take the majority advice
  else
    Choose $\widehat{u}_t \leftarrow 0$ ▷ Arbitrary
  Receive u_t ▷ Observe the truth
  $S \leftarrow S \setminus \{i \in S : v_{i,t} \neq u_t\}$ ▷ Remove all mistaken experts

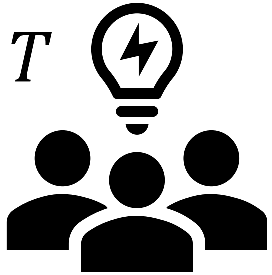
Claim. There is a strategy guaranteeing $M \leq \log_2 n$, regardless of T (even for $T = \infty$).



Algorithm: Start with $S = \{1,2, \ldots n\}$. Each day, choose $\hat{u}_t$ to be the majority of advices from experts still in S. At the end of the day, remove from S all experts who predicted wrong.

Proof of correctness. Every time we make a mistake, at least half the experts in S must have been wrong (we took the majority vote). So after each mistake the size of S is at least halved.

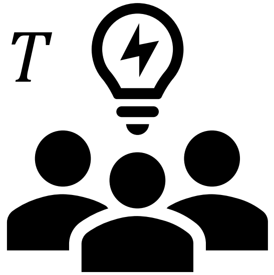
Claim. There is a strategy guaranteeing $M \leq \log_2 n$, regardless of T (even for $T = \infty$).



Algorithm: Start with $S = \{1,2, \dots n\}$. Each day, choose $\hat{u}_t$ to be the majority of advices from experts still in S. At the end of the day, remove from S all experts who predicted wrong.

Proof of correctness. Every time we make a mistake, at least half the experts in S must have been wrong (we took the majority vote). So after each mistake the size of S is at least halved. But we always have $|S| \geq 1$, since (by assumption) there exists an expert who is always right (and therefore never gets removed).

Claim. There is a strategy guaranteeing $M \leq \log_2 n$, regardless of T (even for $T = \infty$).



Algorithm: Start with $S = \{1,2, \dots n\}$. Each day, choose $\hat{u}_t$ to be the majority of advices from experts still in S. At the end of the day, remove from S all experts who predicted wrong.

Proof of correctness. Since we started with $|S| = n$, our total number M of mistakes must then satisfy

$$\frac{n}{2^M} \geq 1$$

HALVING ALGORITHM

that is, $M \leq \log_2 n$.

Nobody’s Perfect

This is great! But… things completely fail if there is no “perfect expert.”

What if even the best expert made some mistakes? Can we make things **robust**?

This is great! But… things completely fail if there is no "perfect expert."

What if even the best expert made some mistakes? Can we make things **robust?**

Let's revisit the algorithm.

$S = \{ i \in [n] : w_i = 1 \}$

We had n weights $w_1, \ldots\ w_n$ initialised to 1.

At day t, our prediction was $\hat{u}_t \leftarrow \mathrm{Maj}(w_1 v_{1,t} + \cdots + w_n v_{n,t})$

Whenever expert i made a mistake, we set $w_i \leftarrow 0 \cdot w_i$.

This is great! But… things completely fail if there is no "perfect expert."

What if even the best expert made some mistakes? Can we make things **robust?**

Let's revisit the algorithm.

We had n weights $w_1, \ldots\ w_n$ initialised to 1.

At day t, our prediction was $\hat{u}_t \leftarrow \mathrm{Maj}(w_1 v_{1,t} + \cdots + w_n v_{n,t})$

Whenever expert i made a mistake, we set $w_i \leftarrow \mathbf{0} \cdot w_i$.

This is great! But… things completely fail if there is no “perfect expert.”

What if even the best expert made some mistakes? Can we make things **robust?**

Let’s revisit the algorithm.

We have n weights $w_1, \ldots\ w_n$ initialised to 1.

At day t, our prediction is $\hat{u}_t \leftarrow \mathrm{Maj}(w_1 v_{1,t} + \cdots + w_n v_{n,t})$

Whenever expert i made a mistake, we set $w_i \leftarrow$ ½ $\cdot\ w_i$.

Algorithm (Multiplicative Weights Update).

Start with n weights $w_1, \ldots\ w_n$ initialised to 1.

Each day, choose the weighted majority $\hat{u}_t \leftarrow \mathrm{Maj}(w_1 v_{1,t} + \cdots + w_n v_{n,t})$

At the end of the day, set $w_i \leftarrow ½ \cdot w_i$ for expert i made a mistake.

Set $w_1, \ldots, w_n \leftarrow 1$

for all $1 \leq t \leq T$ **do**

  Receive $v_{1,t}, \ldots, v_{n,t}$

  Choose $\hat{u}_t \leftarrow \mathrm{sign}\left(\sum_{i=1}^{n} w_i v_{i,t} \geq \frac{1}{2} \sum_{i=1}^{n} w_i\right)$ ▷ Weighted majority

  Receive u_t ▷ Observe the truth

  for all $1 \leq i \leq n$ **do** ▷ Penalise all mistaken experts

$$w_i \leftarrow \begin{cases} \frac{1}{2} w_i & \text{if } v_{i,t} \neq u_t \\ w_i & \text{otherwise.} \end{cases}$$

$$\mathrm{sign}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

Set $w_1, \ldots, w_n \leftarrow 1$
for all $1 \leq t \leq T$ **do**
  Receive $v_{1,t}, \ldots, v_{n,t}$
  Choose $\hat{u}_t \leftarrow \operatorname{sign}\left(\sum_{i=1}^{n} w_i v_{i,t} \geq \frac{1}{2} \sum_{i=1}^{n} w_i\right)$ ▷ Weighted majority
  Receive u_t ▷ Observe the truth
  for all $1 \leq i \leq n$ **do** ▷ Penalise all mistaken experts
    $w_i \leftarrow \begin{cases} \frac{1}{2} w_i & \text{if } v_{i,t} \neq u_t \\ w_i & \text{otherwise.} \end{cases}$

Theorem 59. *There is a (deterministic) algorithm (Algorithm 24) such that*

$$C(T) \leq \frac{C^*(T) + \log_2 n}{\log_2 \frac{4}{3}} \leq 2.41(\underbrace{C^*(T)}_{\text{\# errors of the best expert in hindsight}}) + \log_2 n) .$$

Moreover, this holds even when $T = \infty$.

$$C^*(T) = \min_{1 \leq i \leq n} \sum_{t=1}^{T} \mathbb{1}_{v_{i,t} \neq \hat{u}_t}$$

Theorem. The MWU algorithm guarantees $M \leq 2.41(M^* + \log_2 n)$, where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Potential

Proof. Let W_t be the total weights of experts on day t. Initially, $W_0 = n$. Every time we make a mistake, this means at least half the weight was on experts who did a mistake (since we took the weighted majority). So if we made a mistake at day t,

$$W_{t+1} = W_t^{\text{good}} + \frac{1}{2} W_t^{\text{bad}} \overset{\circledast}{\leq} \frac{1}{2} W_t + \frac{1}{2} \cdot \frac{1}{2} W_t = \frac{3}{4} W_t$$

Now, look at the best expert (in hindsight). They made M^* mistakes, so their final weight is $(1/2)^{M^*}$.

$\circledast$ $W_t = p \cdot W_t^{good} + (1-p) W_t^{bad}$ $\quad (p \leq \frac{1}{2})$

$$W_t^{good} + \frac{1}{2} W_t^{bad} = W_t^{good} + \frac{1}{2}(W_t - W_t^{good})$$
$$= \frac{1}{2} W_t^{good} + \frac{1}{2} V_t \leq \frac{1}{4} W_t + \frac{1}{2} W_t$$

Theorem. The MWU algorithm guarantees $M \leq 2.41(M^* + \log_2 n)$, where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Proof. Let W_t be the total weights of experts on day t. Initially, $W_0 = n$. Every time we make a mistake, this means at least half the weight was on experts who did a mistake (since we took the weighted majority). So if we made a mistake at day t,

$$W_{t+1} = W_t^{\text{good}} + \frac{1}{2} W_t^{\text{bad}} \leq \frac{1}{2} W_t + \frac{1}{2} \cdot \frac{1}{2} W_t = \frac{3}{4} W_t$$

Now, look at the best expert (in hindsight). They made M^* mistakes, so their final weight is $(1/2)^{M^*}$. So the final total weight is $W_T \geq (1/2)^{M^*}$.

Theorem. The MWU algorithm guarantees $M \leq 2.41(M^* + \log_2 n)$, where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Proof. Let W_t be the total weights of experts on day t. Initially, $W_0 = n$. Every time we make a mistake, this means at least half the weight was on experts who did a mistake (since we took the weighted majority). So if we made a mistake at day t,

$$W_{t+1} = W_t^{\text{good}} + \frac{1}{2} W_t^{\text{bad}} \leq \frac{1}{2} W_t + \frac{1}{2} \cdot \frac{1}{2} W_t = \frac{3}{4} W_t$$

Now, look at the best expert (in hindsight). They made M^* mistakes, so their final weight is $(1/2)^{M^*}$. So the final total weight is $W_T \geq (1/2)^{M^*}$.

Theorem. The MWU algorithm guarantees $M \leq 2.41(M^* + \log_2 n)$, where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Proof. Putting it all together:

$$\left(\frac{1}{2}\right)^{M^*} \leq W_T \leq \left(\frac{3}{4}\right)^{M} W_0 = \left(\frac{3}{4}\right)^{M} n$$

Theorem. The MWU algorithm guarantees $M \leq 2.41(M^* + \log_2 n)$, where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Proof. Putting it all together:

$$\left(\frac{1}{2}\right)^{M^*} \leq W_T \leq \left(\frac{3}{4}\right)^{M} W_0 = \left(\frac{3}{4}\right)^{M} n$$

Now, we take the logarithm:

$$-M^* \leq M \underbrace{\log_2\left(\frac{3}{4}\right)}_{<0} + \log_2 n$$

Theorem. The MWU algorithm guarantees $M \leq 2.41(M^* + \log_2 n)$, where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Proof. Putting it all together:

$$\left(\frac{1}{2}\right)^{M^*} \leq W_T \leq \left(\frac{3}{4}\right)^{M} W_0 = \left(\frac{3}{4}\right)^{M} n$$

Now, we take the logarithm:

$$-M^* \leq M \log_2\left(\frac{3}{4}\right) + \log_2 n$$

and get

$$M \leq \frac{M^* + \log_2 n}{\text{-}\log_2\left(\frac{3}{4}\right)}$$

Theorem. The MWU algorithm guarantees $M \leq 2.41(M^* + \log_2 n)$, where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Proof. Putting it all together:

$$\left(\frac{1}{2}\right)^{M^*} \leq W_T \leq \left(\frac{3}{4}\right)^{M} W_0 = \left(\frac{3}{4}\right)^{M} n$$

Now, we take the logarithm:

$$-M^* \leq M \log_2\left(\frac{3}{4}\right) + \log_2 n$$

and get

$$M \leq \frac{M^* + \log_2 n}{-\log_2\left(\frac{3}{4}\right)} \leq 2.41(M^* + \log_2 n)$$

$\square$

Let’s go further!

This is what we proved:

Algorithm (Multiplicative Weights Update).

Start with n weights $w_1, \ldots\ w_n$ initialised to 1.

Each day, choose the weighted majority $\hat{u}_t \leftarrow \mathrm{Maj}(w_1 v_{1,t} + \cdots + w_n v_{n,t})$

At the end of the day, set $w_i \leftarrow$ ½ $\cdot\ w_i$ for expert i made a mistake.

Theorem. The MWU algorithm guarantees $M \leq 2.41(M^* +\ \log_2 n)$, where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

This is what we proved:

Algorithm (Multiplicative Weights Update).

Start with n weights $w_1, \ldots\ w_n$ initialised to 1.

Each day, choose the weighted majority $\hat{u}_t \leftarrow \mathrm{Maj}(w_1 v_{1,t} + \cdots + w_n v_{n,t})$

At the end of the day, set $w_i \leftarrow$ ½ $\cdot\, w_i$ for expert i made a mistake.

Theorem. The MWU algorithm guarantees $M \leq 2.41(M^* +\ \log_2 n)$, where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Using exactly the same argument *(try it!)*, we get, for any $\beta \in (0,1)$:

Algorithm (Multiplicative Weights Update).

Start with n weights $w_1, \dots w_n$ initialised to 1.

Each day, choose the weighted majority $\hat{u}_t \leftarrow \mathrm{Maj}(w_1 v_{1,t} + \cdots + w_n v_{n,t})$

At the end of the day, set $w_i \leftarrow \beta \cdot w_i$ for expert i made a mistake.

Theorem. The MWU algorithm guarantees $M \leq \frac{M^* \log_2(1/\beta) + \log_2 n}{\log_2(\frac{2}{1+\beta})}$,

where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Using exactly the same argument we get, for any $\beta \in (0,1)$:

Theorem. The MWU algorithm guarantees

$$M \leq \frac{M^* \log_2\left(\frac{1}{\beta}\right) + \log_2 n}{\log_2\left(\frac{2}{1+\beta}\right)}$$

where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

$\beta \to 0$: $\sim M^* \underbrace{\log \frac{1}{\beta}}_{\to\infty} + \log_2 n$

$\beta = \frac{1}{2}$: $\dfrac{M^* \cdot 1 + \log_2 n}{\log_2 \frac{4}{3}}$

$\beta = 1-\varepsilon$, $\varepsilon \to 0$: $\dfrac{M^* \ln \frac{1}{1-\varepsilon} + \ln n}{\ln\left(\frac{2}{2-\varepsilon}\right)} \simeq \dfrac{M^* \varepsilon + \ln n}{\varepsilon/2} = 2M^* + O\left(\frac{\ln n}{\varepsilon}\right)$

Using exactly the same argument we get, for any $\beta = 1 - \varepsilon \in (0,1)$:

Theorem. The MWU algorithm guarantees

$$M \leq \frac{M^* \log_2\left(\frac{1}{\beta}\right) + \log_2 n}{\log_2\left(\frac{2}{1+\beta}\right)} \approx 2\left(M^* + \frac{\ln n}{\varepsilon}\right)$$

where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

① $W_0 = n$

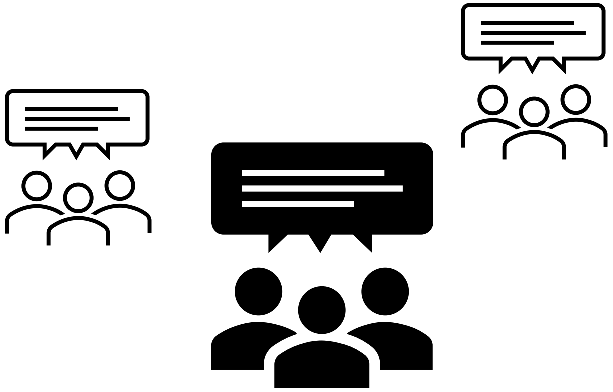
③ $W_T \geq \beta^{M^*}$

② $W_{t+1} = W_t^{good} + \beta W_t^{bad} = (1-\beta) W_t^{good} + \beta W_t$

$\leq \frac{1+\beta}{2} W_t$

$\rightarrow \beta^{M^*} \leq W_T \leq \left(\frac{1+\beta}{2}\right)^M n$

Is that tight?



Theorem. The MWU algorithm guarantees

$$M \approx 2\left(M^* + \frac{\ln n}{\varepsilon}\right)$$

where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Theorem. The MWU algorithm guarantees

$$M \approx 2\left(M^* + \frac{\ln n}{\varepsilon}\right)$$

where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Can we improve that factor 2?

Theorem. The MWU algorithm guarantees

$$M \approx 2\left(M^* + \frac{\ln n}{\varepsilon}\right)$$

where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Can we improve that factor 2? **No.**

Theorem. The MWU algorithm guarantees

$$M \approx 2\left(M^* + \frac{\ln n}{\varepsilon}\right)$$

where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Can we improve that factor 2? **No.** Consider two sets of $n/2$ experts, where experts in the first set are wrong on odd-numbered days, and those in the second set are wrong on even days. That will force T mistakes (while the best experts make $T/2$).

Theorem. The MWU algorithm guarantees

$$M \approx 2\left(M^* + \frac{\ln n}{\varepsilon}\right)$$

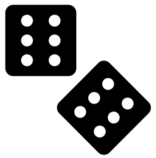
where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

Can we improve that factor 2? **Yes.**

Theorem. The MWU algorithm guarantees

$$M \approx 2\left(M^* + \frac{\ln n}{\varepsilon}\right)$$

where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

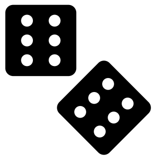


Can we improve that factor 2? **Yes.** With randomisation! Instead of deterministically choosing the weighted majority, pick the answer at random according to the weights.

Theorem. The MWU algorithm guarantees

$$M \approx 2\left(M^* + \frac{\ln n}{\varepsilon}\right)$$

where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).

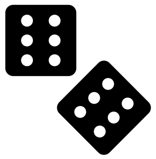


Can we improve that factor 2? **Yes.** With randomisation! Instead of deterministically choosing the weighted majority, pick the answer at random according to the weights. Improves the constant 2 to some $c < 2$.

Theorem. The MWU algorithm guarantees

$$M \approx 2\left(M^* + \frac{\ln n}{\varepsilon}\right)$$

where M^* is the # of mistakes made by the best expert. This holds regardless of T (even for $T = \infty$).



Can we improve that factor 2? **Yes.** With randomisation! Instead of deterministically choosing the weighted majority, pick the answer at random according to the weights. Improves the constant 2 to some $c < 2$. (But only guarantee on **expected** number of mistakes).

Input: Penalty parameter $\beta \in (0,1)$

Set $w_1, \dots, w_n \leftarrow 1$

for all $1 \le t \le T$ **do**

Receive $v_{1,t}, \dots, v_{n,t}$

Draw $I \in [n]$ according to the weights:

$$\Pr[\, I = i \,] = \frac{w_i}{\sum_{i=1}^{n} w_i}, \qquad i \in [n]$$

Choose $\widehat{u}_t \leftarrow v_{I,t}$ ▷ One expert gets the vote

Receive u_t ▷ Observe the truth

for all $1 \le i \le n$ **do** ▷ Penalise all mistaken experts

$$w_i \leftarrow \begin{cases} \beta w_i & \text{if } v_{i,t} \neq u_t \\ w_i & \text{otherwise.} \end{cases}$$

Input: Penalty parameter $\beta \in (0,1)$
Set $w_1, \ldots, w_n \leftarrow 1$
for all $1 \leq t \leq T$ **do**
  Receive $v_{1,t}, \ldots, v_{n,t}$
  Draw $I \in [n]$ according to the weights:

$$\Pr[I = i] = \frac{w_i}{\sum_{i=1}^{n} w_i}, \qquad i \in [n]$$

  Choose $\hat{u}_t \leftarrow v_{I,t}$ ▷ One expert gets the vote
  Receive u_t ▷ Observe the truth
  for all $1 \leq i \leq n$ **do** ▷ Penalise all mistaken experts

$$w_i \leftarrow \begin{cases} \beta w_i & \text{if } v_{i,t} \neq u_t \\ w_i & \text{otherwise.} \end{cases}$$

Theorem 61. *There is a (randomised) algorithm (Algorithm 26) such that*

$$\mathbb{E}[C(T)] \leq \frac{C^*(T)\ln(1/\beta) + \ln n}{1-\beta}.$$

Moreover, this holds even when $T = \infty$.

Before:

$$C(T) \leq \frac{C^*(T)\ln\left(\frac{1}{\beta}\right) + \ln n}{\ln\left(\frac{2}{1+\beta}\right)}$$

Claim. $1-\beta > \ln \frac{2}{1+\beta}$ so the new bound is better!

Input: Penalty parameter $\beta \in (0,1)$
Set $w_1, \ldots, w_n \leftarrow 1$
for all $1 \le t \le T$ **do**
Receive $v_{1,t}, \ldots, v_{n,t}$
Draw $I \in [n]$ according to the weights:

$$\Pr[I = i] = \frac{w_i}{\sum_{i=1}^{n} w_i}, \qquad i \in [n]$$

Choose $\hat{u}_t \leftarrow v_{I,t}$ ▷ One expert gets the vote
Receive u_t ▷ Observe the truth
for all $1 \le i \le n$ **do** ▷ Penalise all mistaken experts

$$w_i \leftarrow \begin{cases} \beta w_i & \text{if } v_{i,t} \neq u_t \\ w_i & \text{otherwise.} \end{cases}$$

Theorem 61. *There is a (randomised) algorithm (Algorithm 26) such that*

$$\mathbb{E}[C(T)] \le \frac{C^*(T)\ln(1/\beta) + \ln n}{1-\beta}.$$

Moreover, this holds even when $T = \infty$.

Pf: $\mathbb{E}[C(t)] = \sum_{s=1}^{t} \Pr[\hat{u}_s \neq u_s] = \sum_{s=1}^{t} \frac{W_s^{\text{bad}}}{W_s} \triangleq F_s$

Fraction of the weight on wrong advice

At every time step

$W_t = (1-F_t)W_t + F_t W_t$

② $W_{t+1} = (1-F_t)W_t + \beta F_t W_t$

$= W_t \cdot (1-(1-\beta)F_t)$

① $W_0 = n$

③ $W_T \ge \beta^{C^*}$

$\beta^{C^*} \le W_T \le n \cdot \prod_{t=1}^{T} (1-(1-\beta)F_t)$

"Take logarithms"

Recall

* $\mathbb{E}[C(T)] = \sum_{t=1}^{T} F_t$

* $\ln(1+x) \le x$

$-\ln(1-x) \ge x$

"Convexity"

Theorem 61. *There is a (randomised) algorithm (Algorithm 26) such that*

$$\mathbb{E}[C(T)] \le \frac{C^*(T)\ln(1/\beta) + \ln n}{1-\beta}.$$

Moreover, this holds even when $T = \infty$.

Input: Penalty parameter $\beta \in (0,1)$

Set $w_1, \ldots, w_n \leftarrow 1$

for all $1 \le t \le T$ **do**

  Receive $v_{1,t}, \ldots, v_{n,t}$

  Draw $I \in [n]$ according to the weights:

$$\Pr[I = i] = \frac{w_i}{\sum_{i=1}^{n} w_i}, \qquad i \in [n]$$

  Choose $\hat{u}_t \leftarrow v_{I,t}$ ▷ One expert gets the vote

  Receive u_t ▷ Observe the truth

  for all $1 \le i \le n$ **do** ▷ Penalise all mistaken experts

$$w_i \leftarrow \begin{cases} \beta w_i & \text{if } v_{i,t} \ne u_t \\ w_i & \text{otherwise.} \end{cases}$$

$$\underbrace{\ln\left(\beta^{C^*}\right)}_{C^* \ln \beta} \le \ln n + \sum_{t=1}^{T} \ln(1 - (1-\beta)F_t)$$

$$C^* \ln \frac{1}{\beta} \ge -\ln n + \sum_{t=1}^{T} \left(-\ln(1-(1-\beta)F_t)\right)$$

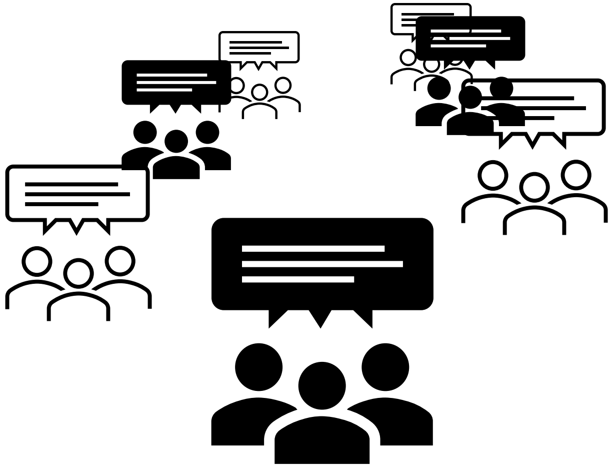
$$\ge -\ln n + (1-\beta)\underbrace{\sum_{t=1}^{T} F_t}_{\mathbb{E}[C(T)]}$$

$$\boxed{-\ln(1-x) \ge x}$$

$$(1-\beta)\,\mathbb{E}[C(T)] \le \ln n + C^* \ln \frac{1}{\beta}$$

□

Concluding remarks



- This was a **short** intro to the Multiplicative Weights Update Algorithms. Much more to say!
    - Different predictions (not only binary)
    - Different payoffs (not just 0-1 loss: correct/incorrect)
    - Randomised version!

- Discovered/rediscovered in many areas: learning theory, game theory/economics, computational geometry, convex optimisation…

- Many (sometimes unexpected) applications: online learning/bandits, semidefinite programming, flow algorithms, zero-sum games, algorithmic takes on evolution (!)

Some pointers if you have questions or want to know more about any of those (or connections to some of those topics):

- *The Multiplicative Weights Update Method: a Meta-Algorithm and Applications.* Arora, Hazan, Kale (2012): https://theoryofcomputing.org/articles/v008a006/
- Lecture notes by Daniel Hsu (2017), Chapter 1: https://www.cs.columbia.edu/~djhsu/coms6998-f17/notes.pdf