Lecture 12: Learning from Experts

We consider the following setting: there are T time steps, and n "experts" $A_1, \dots, A_n$: at each time step t, the algorithm A

T might even be infinite.

- receives advice $v_{1,t}, \dots, v_{n,t} \in \{0,1\}$ from the experts, where $v_{i,t}$ comes from A_i;
- outputs a prediction $\widehat{u}_t \in \{0,1\}$;
- after the prediction is made, gets the ground truth $u_t \in \{0,1\}$, and pays cost

$$c_t := \mathbb{1}_{\widehat{u}_t \neq u_t}$$

There is no assumption on the true values: they could be correlated, independent, adversarial. There is no assumption on the experts either: they could collude, be randomised, be adversarial, be omniscient. And there is no constraint on the algorithm itself: it can use as much memory as needed, be computationally inefficient, etc. But it *cannot see the future*: all the information it has, at each time step t, is what happened in previous time steps, along with the current advice $v_{1,t}, \dots, v_{n,t}$ from the experts.

> How to minimise the total cost $C(T) = \sum_{t=1}^{T} c_t$?

First, what does it even mean to minimise the total cost? How to formulate what this means? Can we get total cost, say, $C(T) = o(T)$? $C(T) = O(\log T)$?

Some bad news.

Fact 56.1. *For any deterministic algorithm A, and for any set of n experts, there is a sequence $u_1, \dots, u_T$ such that A must have cost $C(T) = T$.*

Proof. $\widehat{u}_t$ is fully determined by the past, and the advice received: set $u_t = 1 - \widehat{u}_t$. □

Of course, it is for *deterministic* algorithms, these weaklings. Unfortunately, randomised algorithms do not do much better:

Fact 56.2. *For any algorithm A, and for any set of n experts, there is a distribution over sequences $u_1, \dots, u_T$ such that A must have expected cost $\mathbb{E}[C(T)] \geq \frac{T}{2}$.*

Proof. Uniformly random sequence. □

Changing the goal. In view of this seriously underwhelming state of affairs, we need to reconsider either the *setting*, or the *objective*. We will do the second: in particular, one observation is that while in these bad examples the algorithm A does very poorly, *so do all the* n *experts*. This suggests that the right thing to try to achieve is not a small *absolute* error, but an small error compared to that of *the best expert in hindsight*. Namely, after T steps, let

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

denote the minimum cost achieved by the best of the n experts.

How to minimise the cost $C(T)$ compared to $C^*(T)$?

Still some bad news. Even then, we cannot do *arbitrarily* close to $C^*(T)$, at least not with a deterministic algorithm: a multiplicative factor at least 2 is necessary.

Fact 56.3. *For any deterministic algorithm A, and for any set of n experts, there is a sequence $u_1, \dots, u_T$ such that A must have regret $C(T) = T$, but $C^*(T) \leq \frac{T}{2}$.*

Proof. In the tutorial. □

Warmup: one perfect expert But there is some good news, too! Imagine one of the n experts makes *no mistakes*. Of course, we do not know which one in advance: yet, we *can* leverage this.

Consistent Expert

Theorem 57. *There is a (deterministic) algorithm (Algorithm 22) such that, if one of the n experts makes* zero *mistakes, i.e., $C^*(T) = 0$, then*

$$C(T) \leq n - 1\,.$$

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

Algorithm 22: Consistent Expert algorithm

Set $S \leftarrow [n]$
for all $1 \leq t \leq T$ **do**
  Receive $v_{1,t}, \dots, v_{n,t}$
  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

Proof. The proof uses what is known as a *potential argument*, where we define a suitable quantity Φ such that (1) initially, $\Phi \leq \Phi_0$, (2) at

Potential argument

the end, $\Phi \geq \Phi_\infty$, and (3) every time a "bad event" happens, Φ decreases by a quantifiable amount (typically, either decreases by at least some quantity $\Delta > 0$ or by a constant factor $\gamma > 1$). By putting all 3 together, we are able to argue that the number of "bad events" is bounded by some values (which depends on Φ_0, Φ_∞, and Δ or γ).

Here, our potential function Φ is simply $\Phi = |S_t|$, where S_t is the set S at the end of step $1 \leq t \leq T$. We have, at the beginning, $\Phi = \Phi_0 := |[n]| = n$; and, at the end, since by assumption at least *one* expert never makes any mistake and thus is never removed from S, $\Phi_\infty = |S_T| \geq 1$.

The "bad event" is when the algorithm makes a mistake: if this happens at time t, it is because the expert chosen from $S = S_{t-1}$ in Step 5 was wrong, and so it will be removed from S_{t-1}: which means $|S_t|$ decreases by (at least) $\Delta = 1$: $\Phi_t \leq \Phi_{t-1} - \Delta$.

Putting it all together, if we make C mistakes then our potential Φ decreases by at least $\Delta \cdot C$:

$$1 \leq \Phi_\infty \leq \Phi_0 - \Delta \cdot C = n - 1 \cdot C$$

and so $C \leq n - 1$. □

However, we can do even better! The main insight in the previous algorithm was that, every time we made a mistake, we could remove at least *one* expert from the pool S. What if we could remove at least *a constant fraction* of them?

Halving Algorithm

Theorem 58. *There is a (deterministic) algorithm (Algorithm 23) such that, if one of the n experts makes* zero *mistakes, i.e., $C^*(T) = 0$, then*

$$C(T) \leq \log_2 n\,.$$

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

(As a side note: there is "no free lunch." If we create 2^T *fake experts*, one for each possible sequence $u_1, \ldots, u_T$, then of course one of them will make no mistake: but then, the RHS in the theorem above becomes T.)

Algorithm 23: Halving algorithm

```
Set S ← [n]
for all 1 ≤ t ≤ T do
    Receive v_{1,t}, ..., v_{n,t}
    if |S| ≥ 1 then
        Choose û_t ← maj_{i∈S} v_{i,t}          ▷ Take the majority advice
    else
        Choose û_t ← 0                          ▷ Arbitrary
    Receive u_t                                 ▷ Observe the truth
    S ← S \ {i ∈ S : v_{i,t} ≠ u_t}             ▷ Remove all mistaken experts
```

Proof. Again, we will use a potential function argument, taking $\Phi = |S_t|$ as our potential. As before, $\Phi_0 = n$, while $\Phi_\infty = |S_T| \geq 1$; the main difference is that, each time we make a mistake, we know that this is before *at least half* of the current experts in S_t were wrong (as we took a majority vote among them), and so every time we make a mistake ("bad event") Step 9 will remove at least a $\gamma = 1/2$ fraction of Φ.[50]

As a result, if we make C mistakes then our potential Φ decreases by at least $(1-\gamma)^C$:

$$1 \leq \Phi_\infty \leq \Phi_0 \cdot (1-\gamma)^C = \frac{n}{2^C}$$

and so $C \leq \log_2 n$. □

[50] The algorithm might also decrease S in Step 9, and so Φ, when we do *not* make a mistake: but we cannot prove anything about this, when or if it happens, and by how much.

The theorem above is very good if at least one of the n experts is perfect. Unfortunately, this is very seldom the case: what can we say when all experts make some mistake, sometimes? Can we do anything?

Making this robust. Here is an alternative view of the Halving algorithm:

- We start with n weights, $w_1 = \cdots = w_n = 1$.
- We answer according to the *weighted majority*

$$\mathrm{maj}_{1 \leq i \leq n}\, w_i v_{i,t}$$

- If an expert i is wrong at some step t, then their weight is set to 0: $w_i \leftarrow 0 \cdot w_i$.

This may sound a little extreme: "one strike and you're out." Instead of setting a weight to *zero* when a mistake is made, we could, instead, *decrease* it.

Algorithm 24: (Basic) Multiplicative Weights Updates algorithm

```
Set w_1, ..., w_n ← 1
for all 1 ≤ t ≤ T do
    Receive v_{1,t}, ..., v_{n,t}
    Choose û_t ← sign( Σ_{i=1}^n w_i v_{i,t} ≥ ½ Σ_{i=1}^n w_i )      ▷ Weighted majority
    Receive u_t                                                       ▷ Observe the truth
    for all 1 ≤ i ≤ n do                                              ▷ Penalise all mistaken experts
        w_i ← { ½ w_i   if v_{i,t} ≠ u_t
              { w_i     otherwise.
```

Basic MWU Algorithm

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

$$C(T) \leq \frac{C^*(T) + \log_2 n}{\log_2 \frac{4}{3}} \leq 2.41\left(C^*(T) + \log_2 n\right).$$

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

Proof. Again (again), we will use a potential function argument, but this time defining our potential function Φ as

$$\Phi = W_t$$

where $W_t = \sum_{i=1}^{n} w_{i,t}$ (introducing new notation to get the dependence on t) is the sum of the weights of the n experts at the beginning of time t.

We get $\Phi_0 = n$, but now we can no longer bound Φ_∞ by 1; however, we know that, by definition, there will be at least one expert who makes only C^* mistakes in total. That expert will be penalised and see its weight scaled by $1/2$ C^* times, and so at the end will still have weight $1/2^{C^*}$. But the total weight at the end is at least the weight of that one expert, and so

$$\Phi_\infty \geq \frac{1}{2^{C^*}} .$$

Moreover, since we are taking a (weighted) majority we know, as in the proof of Theorem 58, that each time we make a mistake *at least half of the total weight* of the experts was on wrong experts, and so every time we make a mistake ("bad event") Step 7 will remove at least a $1/2$ fraction of the "wrong experts" weight: call this W_t^{wrong}. But our potential function is the *total* weight, so *what decrease does that imply for* W_t*?*

Writing

$$W_t = \underbrace{W_t^{\text{wrong}}}_{\substack{\text{weight on experts wrong} \\ \text{at time } t}} + \underbrace{(W_t - W_t^{\text{wrong}})}_{\substack{\text{weight on experts correct} \\ \text{at time } t}}$$

we get that, if a mistake is made by the algorithm at time t, then

$$\begin{aligned} W_{t+1} &= \frac{1}{2} \cdot W_t^{\text{wrong}} + (W_t - W_t^{\text{wrong}}) \\ &= W_t - \frac{1}{2} W_t^{\text{wrong}} \\ &\leq W_t - \frac{1}{2} \cdot \frac{1}{2} W_t \quad \text{(The majority was wrong: } W_t^{\text{wrong}} \geq \tfrac{1}{2} W_t) \\ &= \frac{3}{4} W_t \end{aligned}$$

That is, at every mistake Step 9 will remove at least a $\gamma = 1/4$ fraction of the total weight. As a result, if we make C mistakes then our potential Φ decreases by at least $(1-\gamma)^C$:

$$\frac{1}{2^{C^*}} \leq \Phi_\infty \leq \Phi_0 \cdot (1-\gamma)^C = \left(\frac{3}{4}\right)^C \cdot n$$

and so $C \leq \frac{C^* + \log_2 n}{\log_2(4/3)}$. □

Now, there is nothing too special about $1/2$, except that it is a convenient number. We could instead keep it a "penalty parameter" $\beta \in (0,1)$. This gives the following variant of the MWU:

MWU Algorithm

Theorem 60. *There is a (deterministic) algorithm (Algorithm 25) such that*

$$C(T) \leq \frac{C^*(T) \log_2(1/\beta) + \log_2 n}{\log_2 \frac{2}{1+\beta}} .$$

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

Algorithm 25: Multiplicative Weights Updates algorithm

Input: Penalty parameter $\beta \in (0,1)$
Set $w_1, \dots, w_n \leftarrow 1$
for all $1 \leq t \leq T$ **do**
  Receive $v_{1,t}, \dots, v_{n,t}$
  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
    $w_i \leftarrow \begin{cases} \beta w_i & \text{if } v_{i,t} \neq u_t \\ w_i & \text{otherwise.} \end{cases}$

Proof. Same proof as Theorem 59, but replacing $1/2$ by β in the appropriate locations. □

Good practice: go through the details!

In this sense, β lets us trade-off robustness ($\beta \to 1$) for accuracy ($\beta \to 0$). One can check that for $\beta = 1/2$, we get back the guarantees of Theorem 59, and that

Check it!

- When $\beta \to 0$ and , we get

$$\frac{C^*(T)\log_2(1/\beta) + \log_2 n}{\log_2 \frac{2}{1+\beta}} \underset{\beta \to 0^+}{\sim} \log_2 \frac{1}{\beta} \cdot C^*(T) + \log_2 n$$

  retrieving the guarantees of Theorem 58 when $C^*(T) = 0$;

- if $\beta = 1 - \varepsilon$ with $\varepsilon \to 0^+$, then

$$\frac{C^*(T)\log_2(1/\beta) + \log_2 n}{\log_2 \frac{2}{1+\beta}} \underset{\varepsilon \to 0^+}{\sim} 2 \cdot C^*(T) + \frac{2}{\varepsilon}\ln n$$

  much better in terms of dependence[51] on C^*, but *much* worse with respect to the additive $\log_2 n$.

[51] Recall that a factor 2 there is optimal for deterministic algorithms, by Fact 56.3.

But can we do better? Our impossibility result (lower bound) from Fact 56.3 applies to deterministic algorithms. The MWU algorithm, which achieves this bound, is deterministic. If we allow *randomisation* (and relax our goal a little to allow *expected* error, can we circumvent this lower bound?

The answer is (of course?) yes. What is even better, this leads to a simple and very natural algorithm!

Here again, the crucial observation is that Algorithm 25 takes a "hard" majority: the algorithm will predict the same thing if the weighted majority is 50.1% or if it is 100%. This sounds a little silly: if our weighting of experts predicts essentially a coin toss, maybe we should not treat it too confidently?

Maybe we should... toss a coin?

Observe that, in our binary setting where $u_t \in \{0,1\}$, what the algorithm does is equivalent to computing

$$\tilde{p}_t := \sum_{i=1}^n \frac{w_i}{\sum_{j=1}^n w_j} \mathbb{1}_{\{v_{i,t}=1\}}$$

Algorithm 26: Randomised Multiplicative Weights Updates algorithm

```
Input: Penalty parameter β ∈ (0,1)
1: Set w_1, ..., w_n ← 1
2: for all 1 ≤ t ≤ T do
3:     Receive v_{1,t}, ..., v_{n,t}
4:     Draw I ∈ [n] according to the weights:
           Pr[ I = i ] = w_i / (Σ_{i=1}^n w_i),    i ∈ [n]
5:     Choose û_t ← v_{I,t}                 ▷ One expert gets the vote
6:     Receive u_t                          ▷ Observe the truth
7:     for all 1 ≤ i ≤ n do                 ▷ Penalise all mistaken experts
8:         w_i ← { βw_i   if v_{i,t} ≠ u_t
                 { w_i    otherwise.
```

and setting $\hat{u}_t \sim \text{Bern}(\tilde{p}_t)$. But phrasing it this way makes it easier to generalise to more complicated predictions than binary, and also can be more efficient to implement.

Can you see why?

MWU Algorithm

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 $T = \infty$.

Proof. Denote by F_t the fraction of the total weight that is on "wrong experts" at time step t; that is, with the notation of the proof of Theorem 59,

$$F_t = \frac{W_t^{\text{wrong}}}{W_t} \in [0,1], \qquad 1 \leq t \leq T$$

Since we are choosing which decision to make by sampling an expert according to the weights, the probability to make a mistake at time t is exactly F_t; and so, by linearity of expectation,

$$\mathbb{E}[C(T)] = \sum_{t=1}^{T} F_t\,. \tag{78}$$

Following as in the previous proofs a potential argument, we again choose as potential function Φ the total weight of the experts:

$$\Phi_t := W_t = \sum_{i=1}^{n} w_{i,t}\,.$$

We again have $\Phi_0 = n$, and $\Phi_\infty \geq \beta^{C^*}$ (the best expert makes only C^* mistakes, and so its weight at the end is β^{C^*}). Now, Step 8 penalises the wrong experts at every step: in the previous theorems, we only kept track of this when our algorithm made a mistake, since this is all we had a handle on[52] But now, *we can*: regardless of whether our algorithm did make an actual mistake or not, at *every* time step the weight on wrong experts is directly related to the probability to have made a mistake.

[52] Namely, all we could say is that "at least half of the weight was on wrong experts" *when we made a mistake*. The rest of the time, we had no way to relate changes in the total weight to the total number of mistakes C.

That is, at *every* time step $1 \leq t \leq T$, we have

$$\begin{aligned} W_{t+1} &= \beta \cdot F_t W_t + (1 - F_t) W_t \\ &= (1 - (1-\beta) F_t) \cdot W_t \end{aligned}$$

and so we have

$$\Phi_\infty = W_T = W_0 \prod_{t=1}^{T} (1 - (1-\beta)F_t) = \prod_{t=1}^{T} (1 - (1-\beta)F_t) \tag{79}$$

Viewed under the lens of our potential function argument, this corresponds to a decrease by a factor $\gamma = \gamma_t = (1-\beta)F_t$ which is not a constant, but depends on t.

of, taking logarithms and recalling that $W_0 = \Phi_0 = n$,

$$\ln \Phi_\infty = \ln n + \sum_{t=1}^{T} \ln(1 - (1-\beta)F_t) \tag{80}$$

This is promising, but we need to relate this to the expected number of errors $\mathbb{E}[C(T)]$, which by Eq. (78) is $\sum_{t=1}^{T} F_t$ – not the much worse-looking expression above. Recalling the "life-saver" inequality

$$\ln(1+x) \leq x, \qquad x > -1$$

along with $\ln \Phi_\infty \geq C^* \ln \beta$, we obtain

$$C^* \ln \beta \leq \ln \Phi_\infty \leq \ln n - (1-\beta) \sum_{t=1}^{T} F_t \tag{81}$$

Since $\sum_{t=1}^{T} F_t = \mathbb{E}[C(T)]$, reorganising the inequality above gives

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

as claimed. □

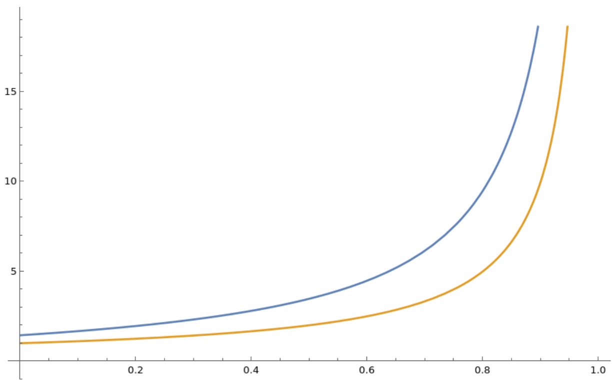

Figure 21: Comparison of the two terms from Theorems 60 and 61: in orange, $\frac{1}{1-\beta}$, and in blue, $\frac{1}{\ln \frac{2}{1+\beta}}$.

Don't let yourself get fooled by the change of logarithm basis between Theorems 60 and 61: the new bound is always better – up to exactly that factor 2, as $\beta \to 1$! (Except, of course, that it is only in expectation).

> *Going further:* for more on this, and connections to online learning and learning theory, see the (excellent) lecture notes by Daniel Hsu, available at `https://www.cs.columbia.edu/~djhsu/coms6998-f17/notes.pdf`.