Topic 1: Using expert advice

1.1 Basic scenario and modeling

1.1.1 Expert advice scenario

Imagine you are a weather reporter for the morning news. Every morning, you must predict the weather, which for simplicity is a binary outcome: either "rain" or "shine". At the end of each day, you become aware of the true outcome, and hence also whether you were correct or you made a mistake with your prediction. Because you are a television personality rather than a professional meteorologist, you consult some experts (e.g., Claude, Gemini, Kimi, Qwen) to get their predictions before you make your own. Can you learn which experts to trust, which are unreliable, etc.?

The scenario, which we call the *expert advice scenario*, is an example of *online learning*, where the "learning" process is on-going rather than something that only happens "offline".

1.1.2 Modeling

A typical modeling approach from statistics would posit that the outcome is governed by some simple mechanism, such as a small Markov chain. The experts may use different types of statistical methods to generate their predictions.

Our modeling approach, for now, is rather different. We make no assumption about the experts or outcomes: they could be controlled by an adversary seeking to make the weather reporter look foolish. To counterbalance the adversary (for now), we assume that (at least) one expert makes no mistakes. This is called the *realizability assumption*.

1.2 Basic learning algorithms for using expert advice

A *learning algorithm* for the expert advice scenario is a protocol for the weather reporter—a.k.a. the *learner*—to follow everyday to make predictions. For now, we only consider deterministic protocols.

1.2.1 Follow the Leader

One of the simplest learning algorithms is the *Follow the Leader (FTL)* algorithm:

> Sort the experts (e.g., alphabetically by name); designate the first expert as the leader. Each day, copy the prediction of the leader; upon a mistake, designate as leader the first expert who has not yet made a mistake.

Theorem 1.1. *In the expert advice scenario with N experts and the realizability assumption, FTL makes at most $N - 1$ mistakes.*

Proof. Say an expert is "alive" at a point in time if they have not yet made a mistake, and "dead" otherwise (yikes!). If FTL makes a mistake, then the expert it followed was alive at the start of the day but dies by the end of the day. With the realizability assumption, at least one expert never makes a mistake, so at most $N - 1$ experts can "die". So FTL makes at most $N - 1$ mistakes. □

Theorem 1.1 cannot generally be improved: assuming FTL picks an (alive) expert to follow without considering the expert predictions, it can be forced to make $N-1$ mistakes. To see this, imagine that the weather and the experts are controlled by an adversary who knows the details of how FTL is implemented. The adversary coordinates the weather and the expert predictions so that only the expert chosen by FTL makes a mistake (and hence so does FTL). The adversary can do this on $N-1$ days until only one expert remains. So the analysis of FTL in Theorem 1.1 is said to be "tight".

The *mistake bound* of a learning algorithm is an upper bound on the worst-case number of mistakes that it ever makes. So FTL has a mistake bound of $N-1$ with the realizability assumption.

1.2.2 Halving

One way to improve on the FTL algorithm is to improve on its mistake bound. The *Halving* (or *Majority*) algorithm of Barzdin and Freivald (1972), described below, does just that:

> Each day, copy the most popular prediction made by the "alive" experts.

(Here, we adopt the "alive"/"dead" terminology from the proof of Theorem 1.1.)

Theorem 1.2. *In the expert advice scenario with N experts and the realizability assumption, Halving makes at most $\lfloor \log_2(N) \rfloor$ mistakes.*

Proof. When Halving makes a mistake in a certain day, then at least half of the experts who are alive at the start of the day—specifically, those who also make the more popular prediction—also make a mistake and hence die at the end of the day. So the number of experts who are alive is at most half of what it was at the start of the day. All N experts are initially alive, and this number can be reduced this way at most $\lfloor \log_2(N) \rfloor$ times before it is 1. With the realizability assumption, one expert must always remain alive. Therefore Halving can make at most $\lfloor \log_2(N) \rfloor$ mistakes. □

Exercise 1.1. *Show that the bound in Theorem 1.2 is tight.*

1.2.3 Barrier for all learning algorithms

Can any learning algorithm improve on the mistake bound of Halving? In general, the answer is no: every learner can be forced to make $\lfloor \log_2(N) \rfloor$ mistakes for a worst-case sequence of the N expert predictions and outcomes.

Theorem 1.3. *In the expert advice scenario with N experts and the realizability assumption, for each learning algorithm, there is some sequence of expert predictions and outcomes on which the learner makes at least $\lfloor \log_2(N) \rfloor$ mistakes.*

Proof. Consider $N' := 2^{\lfloor \log_2(N) \rfloor}$ of the N experts and any learning algorithm. We show that among these N' experts, for there is a sequence of the N' expert predictions and outcomes such that the learner makes $\log_2(N') = \lfloor \log_2(N) \rfloor$ mistakes, and yet one of the N' experts makes no mistakes. To see this, consider the rooted perfect binary tree of depth $\log_2(N')$, with leaves corresponding to the N' experts, and each edge annotated with either "rain" or "shine" so that each binary sequence in $\{\text{rain}, \text{shine}\}^{N'}$ appears along the edges of some root-to-leaf path. The binary sequence leading to a leaf corresponding to an expert gives the predictions of that expert for the first $\log_2(N')$ days. Simulate the learning algorithm for $\log_2(N')$ days on a sequence of outcomes that always differs from the learner's predictions. The learner thus makes $\log_2(N')$ mistakes on this sequence of outcomes, but the expert corresponding to the leaf at the end of this root-to-leaf path makes no mistakes. □

1.3 Modeling and coping with imperfection

1.3.1 Experts with bounded imperfection

Another way to improve on the learning algorithms described above is to relax the assumptions under which it can reliably operate.

So far we have considered learning algorithms that crucially rely on the realizability assumption, so that assumption is a prime target for relaxation. Note that without any assumption about the experts, it could be that every learner makes a mistake in every round[1]. A less drastic relaxation is to assume that at least one expert makes at most K mistakes, for some non-negative integer K. Call this the *K-mistake assumption*. The strength of the K-mistake assumption is controlled by the modeling parameter K.

1.3.2 Halving with Resurrection

Neither FTL and Halving is well-defined when all experts are dead, and that is a possibility even with the 1-mistake assumption. A simple fix is to "resurrect" all experts whenever all of them are dead. Let *Halving with Resurrection* be the variant of Halving with this change:

> Proceed as in Halving, except: whenever all experts are dead at the end of a round, immediately "resurrect" them (i.e., declare all of them to be alive).

Theorem 1.4. *In the expert advice scenario with N experts and the K-mistake assumption, Halving with Resurrection makes at most $(\lfloor \log_2(N) \rfloor + 1) \times K + \lfloor \log_2(N) \rfloor$ mistakes.*

Proof. Partition all rounds into contiguous blocks of rounds between resurrections. If a resurrection occurs, then every expert has made at least one mistake since the last resurrection (or since before the first round). So the number of resurrections is at most K.

Within a block of rounds before a resurrection, the Halving with Resurrection can make at most $\lfloor \log_2(N) \rfloor + 1$ mistakes. This is because every mistake corresponds to at least half of the experts who are alive having made a mistake; this is the same argument as in the proof of Theorem 1.2, with the extra +1 coming from the one additional mistake that results in all experts being dead.

There can be one additional block of rounds after the last resurrection, and in this block, Halving with Resurrection makes at most $\lfloor \log_2(N) \rfloor$ mistakes, by the same argument as in the proof of Theorem 1.2.

So the total number of mistakes is bounded by

$$\begin{array}{lll} (\#\text{ mistakes in block before resurrection}) & \times (\#\text{ resurrections}) & + (\#\text{ mistakes after last resurrection}) \\ \leq (\lfloor \log_2(N) \rfloor + 1) & \times K & + \lfloor \log_2(N) \rfloor \end{array}$$

as claimed. □

Exercise 1.2. *Show that the bound in Theorem 1.4 is tight.*

Exercise 1.3. *Analyze the following two learning algorithms in the expert advice scenario with N experts and the K-mistake assumption:*

[1] It is more common to use the term "round" or "trial" instead of "day", so we will henceforth adopt this terminology.

- *Majority-of-Leaders: Let P_m be the set of experts who have made exactly m mistakes. Each day, copy the most popular prediction made by the experts in $P_{m^\star}$, where $m^\star = \min\{m \mid P_m \neq \emptyset\}$.*
- *Majority-of-Survivors: Each day, copy the most popular prediction made by experts who have made at most K mistakes.*

1.3.3 Barrier for all learning algorithms

The K-mistake assumption can be thought of as a promise about the "best expert", i.e., the expert who ultimately makes the fewest mistakes. The mistake bound in Theorem 1.4 shows a bound on the number of mistakes Halving with Resurrection can make in terms of the number of mistakes made by the best expert: roughly $\log(N) \times K$.

Is there a learning algorithm that improves on this mistake bound? The following theorem establishes a barrier on how much improvement is possible: roughly speaking, the $\log(N)$ factor cannot be reduced below two.

Theorem 1.5 (Cover, 1965)**.** *In the expert advice scenario with $N \geq 2$ experts and the K-mistake assumption, for each learning algorithm, there is some sequence of expert predictions and outcomes on which the learner makes at least $2K + 1$ mistakes.*

Proof. Consider two experts, one that always predicts "rain", and another that always predicts "shine". Now consider any learning algorithm, and simulate it for $2K + 1$ rounds on a sequence of outcomes that always differs from the learner's predictions. The learner thus makes $2K + 1$ mistakes on this sequence of outcomes, but one of the two experts makes at most K mistakes. □

Exercise 1.4. *Improve Theorem 1.5 to $2K + \lfloor \log_2(N) \rfloor$ mistakes.*

There is still a large gap between $2K + \lfloor \log_2(N) \rfloor$ and $(\lfloor \log_2(N) \rfloor + 1) \times K + \lfloor \log_2(N) \rfloor$.

1.3.4 Weighted Majority

We now describe a gentler version of Halving that has a mistake bound with the K-mistake assumption that is much closer to the lower bound from Theorem 1.5. The algorithm is called *Weighted Majority (WM)*, due to Littlestone and Warmuth (1994). The description of the algorithm uses the following notation for the expert advice scenario with N experts.

- The N experts are denoted by $[N] := \{1, 2, \ldots, N\}$.
- The possible outcomes are denoted by $\{0, 1\}$.
- The rounds are indexed by $t \in \mathbb{N} := \{1, 2, 3, \ldots\}$.
- In round t, expert i's prediction is denoted by $b_{t,i} \in \{0, 1\}$, the learner prediction is denoted by $a_t \in \{0, 1\}$, and the actual outcome is denoted by $y_t \in \{0, 1\}$.

The algorithm begins by initializing "weights" associated with the experts. In a given round, the current weights are used along with the experts' predictions to form the learner's own prediction; the weights are updated at the end of the round after observing the outcome.

Algorithm 1.1: Weighted Majority for expert advice scenario with N experts.
Hyperparameter: $\beta \in (0,1)$.

- Initially, set $w_{1,i} := 1$ for all $i \in [N]$. This is the "weight" of expert i.
- In round t:
  - Observe experts' predictions $b_{t,1}, \ldots, b_{t,N} \in \{0,1\}$.
  - For each possible outcome $y \in \{0,1\}$, let

$$Z_{t,y} := \sum_{i=1}^{N} \mathbb{1}\{b_{t,i} = y\} w_{t,i}$$

    be the total weight of experts that predict y.
  - Predict

$$a_t := \begin{cases} 1 & \text{if } Z_{t,1} > Z_{t,0}, \\ 0 & \text{otherwise.} \end{cases}$$

  - Observe outcome $y_t \in \{0,1\}$.
  - Update experts' weights: for each $i \in [N]$,

$$w_{t+1,i} := \begin{cases} w_{t,i} & \text{if } y_t = b_{t,i}, \\ \beta w_{t,i} & \text{if } y_t \neq b_{t,i}. \end{cases}$$

We can think of the weight of an expert as a measure of its reliability. We consider all experts to be equally reliable at the start. And experts who make a mistake have their reliability score reduced by multiplying it by the hyperparameter $\beta \in (0,1)$.

For simplicity, we first analyze Weighted Majority with the concrete hyperparameter $\beta = 1/2$.

Theorem 1.6 (Littlestone and Warmuth, 1994)**.** *In the expert advice scenario with N experts and the K-mistake assumption, Weighted Majority with hyperparameter $\beta = 1/2$ makes at most $\frac{1}{\log_2(4/3)}(K + \log_2(N)) \approx 2.41(K + \log_2(N))$ mistakes.*

The idea behind the proof is similar to the analyses of FTL and Halving. Each mistake of the learner is charged to the weighted fraction of experts that also make a mistake. That weighted fraction is at least half the total weight. The proof of Theorem 1.6 will track the total weight of experts as it evolves over time. (We regard the total weight of experts as a "potential function".)

Proof of Theorem 1.6. For any round t, define

$$Z_t := \sum_{i=1}^{N} w_{t,i} = Z_{t,0} + Z_{t,1}$$

to be the total weight of experts at the start of round t. At the start of round 1, we have $Z_1 = N$.

Suppose Weighted Majority makes a mistake in round t. Let

$$\alpha_t := \frac{Z_{t,1-y_t}}{Z_t}$$

be the fraction (or proportion) of the total weight of experts that were mistaken. Since $\beta = 1/2$, the weight of these experts are halved in the update; the weight of the remaining experts are unchanged. We know $\alpha_t \geq 1/2$ by the way Weighted Majority chooses its (mistaken) prediction. Therefore

$$\begin{aligned} Z_{t+1} &= Z_{t,y_t} + \frac{1}{2} Z_{t,1-y_t} \\ &= (1-\alpha_t) Z_t + \frac{1}{2}\alpha_t Z_t \\ &= Z_t - \frac{1}{2}\alpha_t Z_t \\ &\leq Z_t - \frac{1}{4} Z_t \quad (\text{since } \alpha_t \geq 1/2) \\ &= \frac{3}{4} Z_t. \end{aligned}$$

Therefore, by induction, if Weighted Majority makes M mistakes after t rounds, then

$$Z_{t+1} \leq \left(\frac{3}{4}\right)^M Z_1 = \left(\frac{3}{4}\right)^M N.$$

The weight $w_{t+1,i}$ of expert i at the start of round $t+1$ is determined by the number of mistakes expert i has made after t rounds: if expert i has made M_i mistakes, then $w_{t+1,i} = 2^{-M_i}$. Therefore, if there is an expert who has made at most K mistakes after t rounds, then

$$Z_{t+1} \geq 2^{-K}.$$

We now have upper- and lower-bounds on Z_{t+1}, so we can combine the bounds

$$2^{-K} \leq Z_{t+1} \leq \left(\frac{3}{4}\right)^M N$$

and rearrange to get $M \leq (K + \log_2 N)/\log_2(4/3)$. □

For general $\beta \in (0,1)$, the same analysis from the proof of Theorem 1.6 gives the following.

Theorem 1.7 (Littlestone and Warmuth, 1994)**.** *In the expert advice scenario with N experts and the K-mistake assumption, Weighted Majority with hyperparameter $\beta \in (0,1)$ makes at most*

$$\frac{\log(1/\beta)K + \log N}{\log(2/(1+\beta))}. \tag{1.1}$$

mistakes.

The base of the logarithm clearly does not matter in (1.1), but to make it concrete, let us always assume log uses base e. Here is what the mistake bound in (1.1) looks like for different values of β.

β	Equation (1.1)
1/2	$\approx 2.41K + 3.48\ln(N)$
2/3	$\approx 2.22K + 5.48\ln(N)$
3/4	$\approx 2.15K + 7.49\ln(N)$
4/5	$\approx 2.12K + 9.49\ln(N)$

The limit of the pre-factor on K is 2 as $\beta \to 1$ (as can be checked by L'Hôpital's rule), but the pre-factor on $\ln(N)$ diverges to $+\infty$ as $\beta \to 1$.