CS272 - Theoretical Foundations of Learning, Decisions, and Games

Lecture 6: Online Learning II

September 16, 2025

Lecturer: Nika Haghtalab *Readings: UML Chapter 21*
Scribe: Kevin Gillespie

1 Recap

An algorithm $\mathcal{A}$ learns a concept class $\mathcal{C} : \mathcal{X} \to \mathcal{Y}$ with a **mistake bound** M *if and only if* for any sequence of pairs $(x_1, y_1), (x_2, y_2), ..., (x_t, y_t)$ s.t. $(\exists c^* \in \mathcal{C})(c^*(x_t) = y_t)$, the algorithm $\mathcal{A}$ makes **no more than** M mistakes. The last lecture concluded with a demonstration of the mistake bound of a 1D threshold classifiers using a bespoke algorithm. We now proceed to giving a general purpose algorithm, called the *Majority algorithms* that works for any concept class $\mathcal{C}$.

2 The Majority Algorithm

The **Majority algorithm**, at a time t, consults all $c \in \mathcal{C}$ that have not made a mistake, i.e.,

$$S^t = \{c \in \mathcal{C} | c(x_1) = y_1, c(x_2) = y_2, \cdots, c(x_{t-1}) = y_{t-1}\}$$

and conducts a **majority vote** on these experts (which are hypotheses or concepts).

Theorem 2.1 (Mistake Bound of the Majority Algorithm)**.** *Let $\mathcal{A}$ be the majority algorithm for the concept class $c \in \mathcal{C}$, assuming $\exists c^* \in \mathcal{C}$ s.t. $c^*(x_t) = y_t$. The mistake bound M is bounded by*

$$M \leq \log_2 |\mathcal{C}|.$$

Proof. Let W^t be the number of plausible consistent hypotheses remaining at step t, $W^t = |S^t|$. If at a step t the majority of experts is wrong, this implies that more than half of the experts made the wrong prediction. Therefore, if we make a mistake at time t, then

$$W^{t+1} \leq \frac{W^t}{2}.$$

Therefore if we make M mistakes by the end of time T, we have that $W^{T+1} \leq |\mathcal{C}|(\frac{1}{2})^M$.

Since we assume consistency, we can assert that at all steps t, $W^t \geq 1$.

Putting these together we have that $M \leq \log_2 |\mathcal{C}|$. □

2.1 A Fork in the Road

There are two directions for natural questions to proceed from here:

Direction 1: *What if $|\mathcal{C}| = \infty$?*

Direction 2: *What if you relax the consistency assumption?*

Direction 1 will be the topic of the next lecture. In this lecture, we will focus on Direction 2 where the consistency assumption is relaxed.

3 Mistake Bound Without Consistency

In this section, we relax the consistency assumption. In this case, our goal is to have a mistake bound that is not significantly worse than the number of mistakes the best experts makes. Ideally, our mistake bound is only a small additive error over this quantity. We will work towards this goal through several intuitive attempts to refine our choice of algorithm.

3.1 Try 1: Repeated Majority

Our first attempt is simple modification to the majority algorithm, we call **repeated majority**: We run majority until $S^t = \emptyset$. We then restart the majority by setting $S^{t+1} = \mathcal{C}$ and continuing to label instances based on the majority vote of the running experts. This means that every time there are no more correct experts, the algorithm enters a new epoch where the process is restarted.

Theorem 3.1 (Repeated Majority Mistake Bound)**.** *Let $\mathcal{A}$ be a repeated majority algorithm for the concept class $\mathcal{C}$. The mistake bound M of $\mathcal{A}$ at any time t is bounded by*

$$M \leq (1 + \log_2(|\mathcal{C}|))(OPT + 1),$$

where OPT is the number of mistakes of the best performing expert.

Proof. The intuition of this bound is that by the end of each epoch, we have made at most $\log_2(|\mathcal{C}|)$ mistakes in order to narrow down to a single classifier and one more mistake no experts are left. The total number of epochs is at most OPT, since in each epoch the optimal classifier makes at least 1 mistake. On the last epoch, the algorithm makes at most $\log_2(|\mathcal{C}|)$ mistakes by Theorem 3.1. This shows that $M \leq (\log_2(|\mathcal{C}|) + 1)OPT + \log_2(|\mathcal{C}|)$. □

This is a good bound if OPT is extremely small. If $OPT \geq \frac{T}{4}$ or even $OPT \geq \frac{T}{\log(|\mathcal{C}|)}$ the bound becomes trivial as it is significantly worst than random guessing!

3.2 Try 2: Weighted Majority

In the second attempt, we intuitive generalize the majority algorithm. The key idea of the majority algorithm was a "credit" system fo the experts. Experts that were so far perfect, had a 1 credit,

and those that had ever made a mistake had 0 credit. The majority then aggregated the votes of all experts that were "credible".

On a high level, the weighted majority algorithm also assigns a credit to each expert. As agents make more mistakes, this credits is shrunk instead of hitting 0 at the first mistake. Parameter ϵ decides how quickly to shrink the credit upon a mistake.

Algorithm 1 Weighted Majority (with parameter ϵ)

$\mathcal{C}$**:** Set of hypotheses $\{h_1, h_2, ..., h_n\}$
n**:** $|\mathcal{C}|$
ϵ**:** Credit penalty
$w_1^t, w_2^t, w_3^t, ..., w_n^t = 1$ ▷ Initialize credit for each expert

for $t = 1, 2, 3, ..$ **do**
    if $\sum_{i:h_i(x^t)=0} w_i^t > \sum_{i:h_i(x^t)=1} w_i^t$ **then**
        Predict $\hat{y}_t = 0$
    else
        Predict $\hat{y}_t = 1$

    $E^t = \{i | h_i(x^t) \neq y^t\}$ ▷ Set of erring h's

    for $w_1^t, w_2^t, w_3^t, ..., w_n^t$ **do**
        if $i \in E^t$ **then**
            $w_i^{t+1} = (1 - \epsilon) w_i^t$ ▷ Penalize the credit of wrong experts
        else
            $w_i^{t+1} = w_i^t$ ▷ Maintain credit of correct experts

Theorem 3.2 (Weighted Majority Mistake Bound)**.** *Let $\mathcal{A}$ be the weighted majority algorithm for the concept class $\mathcal{C}$. At any time T, the number of mistakes M made by the algorithms is bounded by*

$$M \leq \frac{2}{1-\epsilon} OPT + \frac{2}{\epsilon} \ln |C|.$$

In particular, when $\epsilon = 1/2$, we have that $M \leq 2.4(OPT + \log_2(|\mathcal{C}|))$.

Proof. We will prove this for the case of $= 1/2$, the general proof follows analogously.

Let W^t now be the total remaining credit belonging to experts at time t,

$$W^t = \sum_{i=1}^{n} w_i^t.$$

By definition, there exists *at least* one classifier $h_i \in \mathcal{C}$ that makes OPT mistakes by time T. So,

$$W^{T+1} \geq 2^{-OPT}$$

at all t.

Analogous to *Theorem 2.1*, when the algorithm is wrong at time t, over half of the credit belongs to the set of mistaken experts:

$$\sum_{i \in E^t} w_i^t \geq \frac{W^t}{2}$$

where E^t is the set of mistaken experts at step t. This means that

$$W^{t+1} = \frac{1}{2} \sum_{i \in E^t} w_i^t + \sum_{i \notin E^t} w_i^t \leq \frac{3}{4} W^t.$$

Therefore, after making M mistakes by the end of time T, we have that

$$W^{T+1} \leq n \times (3/4)^M.$$

Putting these together, we have

$$(\frac{1}{2})^{OPT} \leq W^{T+1} \leq n \times (\frac{3}{4})^M.$$

Taking the logarithm and re-arranging, we arrive at:

$$M \log \frac{4}{3} \leq OPT + \log_2 n$$

$$M < 2.4(OPT + \log_2 n).$$

□

While this is significantly better than Theorem 2.1, the bounds are still uninteresting if OPT is large, e.g., $T/4$. We'll resolve this in the next try.

3.3 Try 3: Randomized Weighted Majority

In the previous examples, imagine a scenario where 49% of experts suggest one label and 51% suggest the other. The *Weighted Majority* algorithm deterministically chooses to follow the 51% of the experts. An adversary who knows and anticipates this can ensure that the Weighted Majority is mistaken.

Arguably, given such an equal divide between the experts, it is strange that the Weighted Majority gives no weight to the opposing opinions! To fix this, we consider randomizing in proportion to the weight of each group, called *Randomized Weighted Majority*.

In the example above, the randomized Weighted Majority would pick labels 0 and 1 with almost equal probability and will only make a mistake with probability close to 0.5 rather than 1.

This is a powerful algorithm and quite general. For example, we can easily handle real-valued cost functions by adjusting how heavily we penalize an expert based on how much cost they incurred.

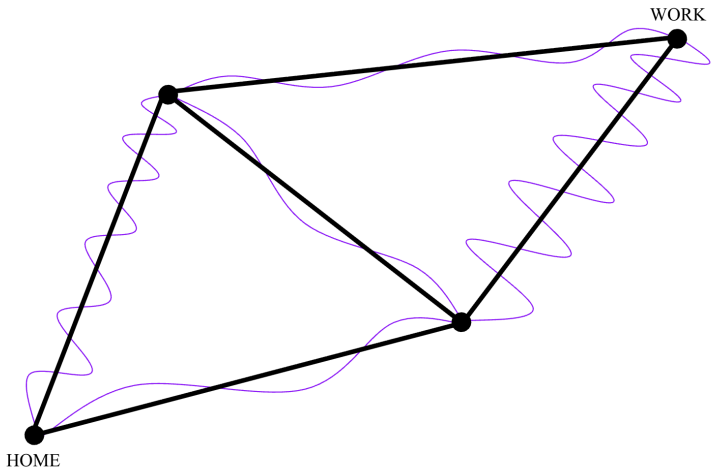

Figure 1: Routing example with randomness on every path.

Real-valued cost functions are common in practice and we will be working with them closely in the remainder of the semester. Take a routing example of a daily commit to and from work in *Figure 1*. There are many routes that you can take between home and work. Each route has a congestion on the roads consisting of it and that affects how long your commute on that road they take each day. You are interested in minimizing $\sum$(Time Traveling). Your hope is to not incur significant suboptimality after year or more of commuting. That is, not to incur significantly more travel time on average, compared to the best route in hindsight.

For randomized weighted majority, we have n **experts** $[n] = \{1, 2, ..., n\}$ that have the hypotheses $h_1, h_2, ..., h_n$. The **adversary** designs a **cost function** $c^t : [n] \to [0, 1]$. In the above example, this could be the congestion of all the various possible routes (we are working with vectors).

The algorithm chooses an expert $i^t \in [n]$ sees c^t and pays $c^t(i^t)$. The cost of the algorithm and the cost of the best expert in hindsight are respectively

$$\sum_{t=1}^{T} c^t(i^t)$$

$$OPT = \min_{i \in [n]} \sum_{t=1}^{T} c^t(i^t).$$

Theorem 3.3 (Randomized Weighted Majority)**.** *Let $\mathcal{A}$ be the randomized weighted majority algorithm for the concept class $\mathcal{C}$. For any $T > 0$, the expected cost up to that point is bounded by*

$$\mathbb{E}[\sum_{t=1}^{T} c^t(i^t)] \leq \frac{1}{1-\epsilon} \mathbb{E}[\min_{i \in [n]} c^t(i)] + \frac{1}{\epsilon} \ln n,$$

Algorithm 2 Randomized Weighted Majority

$\mathcal{C}$**:** Set of hypotheses $\{h_1, h_2, ..., h_n\}$
n**:** $|\mathcal{C}|$
ϵ**:** Credit penalty
$w_1^t, w_2^t, w_3^t, ..., w_n^t = 1$ ▷ Initialize credit for each expert

for $t = 1, 2, 3, ..$ **do**
    $p_i^t = \frac{w_i^t}{\sum_j w_j^t}$ ▷ Distribution P^t over all experts $i \in [n]$

    Pick $i^t \sim P^t$ and follow advice, incur cost $c^t(i^t)$.

    for all $i \in [n]$ **do**
        $w_i^{t+1} \leftarrow (1-\epsilon)^{c^t(i)} w_i^t$ ▷ Highest cost results in full penalty, and vice versa

where expectation is taken over both the algorithms randomness and the adversary's choice.

Note that the expectations are taken over the randomness of both the algorithm and the adversary. For example, while routing to work each day, you are randomly choosing a route to take each day as you collect information about how the other routes performed. You adjust accordingly for the following day, which might have randomness from a distribution than the day before (e.g. a football game causing unusual congestion, etc.).

We will use three mathematical fact throughout the proof. You proved these facts in HW0 already!

Fact 3.4. $(1-\epsilon)^c \leq 1 - c^\epsilon$.

Fact 3.5. *For all* $x \in (0,1)$, $1 - x \leq e^{-x}$.

Fact 3.6. $\frac{1}{\epsilon}\ln(\frac{1}{1-\epsilon}) \leq \frac{1}{1-\epsilon}$.

Proof. Let $C_i^T = \sum_{t=1}^T c^t(i)$. By the update rule,

$$w_i^{T+1} = \prod_{t=1}^{T} (1-\epsilon)^{c^t(i)} = (1-\epsilon)^{C_i^T}.$$

Let $i^\star \in \arg\min_i C_i^T$. Then

$$W^{T+1} = \sum_{i=1}^{n} w_i^{T+1} \;\geq\; w_{i^\star}^{T+1} = (1-\epsilon)^{C_{i^\star}^T}.$$

Taking $\ln$ and expectation,

$$\mathbb{E}\big[\ln(W^{T+1})\big] \;\geq\; \mathbb{E}\big[C_{i^\star}^T \cdot \ln(1-\epsilon)\big]\,. \tag{1}$$

For an upper bound, observe for each t that conditioned on the past, we have

$$\begin{aligned}
W^{t+1} &= \sum_{i=1}^{n} (1-\epsilon)^{c^t(i)}\, w_i^t \\
&\le \sum_{i=1}^{n} \big(1 - \epsilon\, c^t(i)\big)\, w_i^t \qquad \text{by Fact 3.4: } (1-\epsilon)^c \le 1 - \epsilon c \\
&= W^t \Big(1 - \epsilon \sum_{i=1}^{n} c^t(i) \tfrac{w_i^t}{W^t}\Big) \\
&= W^t \Big(1 - \epsilon\, \mathbb{E}_{i\sim p^t}[c^t(i)]\Big) \\
&\le W^t \exp\big(-\epsilon\, \mathbb{E}_{i\sim p^t}[c^t(i)]\big) \quad \text{by Fact 3.5: } 1 - x \le e^{-x}.
\end{aligned}$$

Unrolling over $t = 1, \ldots, T$ and noting $W^1 = n$,

$$W^{T+1} \le n \cdot \exp\Big(-\epsilon \sum_{t=1}^{T} \mathbb{E}_{i\sim p^t}[c^t(i)]\Big).$$

Taking expectations and applying Jensen's inequality to the concave $\ln$,

$$\mathbb{E}\big[\ln(W^{T+1})\big] \;\le\; \ln \mathbb{E}[W^{T+1}] \;\le\; \ln(n) - \epsilon \sum_{t=1}^{T} \mathbb{E}_{i\sim p^t}[c^t(i)]. \tag{2}$$

Combining (1) and (2) gives

$$\sum_{t=1}^{T} \mathbb{E}[c^t(i^t)] \;\le\; \frac{\ln n}{\epsilon} - \frac{1}{\epsilon}\, \mathbb{E}\big[C_{i^\star}^T \ln(1-\epsilon)\big]\,.$$

Using Fact 3.6, we conclude

$$\sum_{t=1}^{T} \mathbb{E}[c^t(i^t)] \;\le\; \frac{\ln n}{\epsilon} + \frac{1}{1-\epsilon}\, \mathbb{E}\big[C_{i^\star}^T\big]\,,$$

which is the claimed bound. □