CS272 - Theoretical Foundations of Learning, Decisions, and Games

Lecture 9: Efficient Algorithms for Online Learning

September 25, 2025

Lecturer: Nika Haghtalab *Readings: UML Chapter 21*
Scribe: -

Background and motivation

A central storyline in learning theory contrasts the *offline* (statistical) world with the *online* (adversarial) world. In the previous lecture, we addressed the *statistical* gap between these two paradigms: offline learnability is characterized by the finiteness of VC dimension, online learnability by the finiteness of Littlestone dimension, and smoothed analysis yields regret bounds that depend only on the VC dimension even against smoothed adaptive adversaries.

In this lecture, we turn to the *algorithmic* gap between online and offline learning. Early in the semester we observed that ERM is a universally good algorithm for offline learning: any class of VC dimension d is PAC-learnable by ERM from $O\big(\epsilon^{-2}(d + \ln(1/\delta))\big)$ samples. In contrast, ERM is not a good algorithm for online learning; in particular, we saw in previous lectures that repeated ERM (and any other deterministic algorithm) can be forced to incur a regret of $\Omega(T)$.

On the other hand, the Randomized Weighted Majority (RWM) algorithm is a natural online choice, but its running time scales linearly with the number of experts. This is unfortunate in settings with exponentially many (or even infinitely many) experts, where RWM becomes computationally intractable — even when the corresponding offline ERM problem can still be solved efficiently by exploiting structure.

Our goal in this lecture is to design online algorithms that build up on ERM. We will study *Follow-the-Regularized-Leader (FTRL)* and *Follow-the-Perturbed-Leader (FTPL)*, two general frameworks for online algorithm design that achieve low regret while prioritizing computational efficiency by using the same or similar computation as ERM does.

Up to now, we have discussed the expert setting where on round t we choose an expert $i^t \in [n]$ and analyze the expected loss $\mathbb{E}[c^t(i^t)]$. Today we move to a more general linear model. In the adversarial online setting, on each round $t = 1, \ldots, T$, the learner chooses $x^t \in \mathcal{X}$; then a linear loss $c^t : \mathcal{X} \to \mathbb{R}$ is revealed and the learner suffers

$$c^t(x^t) \;=\; \langle c^t, x^t \rangle.$$

The regret against a comparator $x^\star \in \mathcal{X}$ is

$$\mathrm{Regret}_T(x^\star) \;=\; \sum_{t=1}^{T} c^t(x^t) \;-\; \sum_{t=1}^{T} c^t(x^\star).$$

The learning-with-expert-advice setting is a special case with $\mathcal{X} = \Delta_n$ (the probability simplex), $\|c^t\|_\infty \le 1$, and $x^t \ge 0$, $\|x^t\|_1 = 1$, in which case $\mathbb{E}_{i^t \sim x^t}[c^t(i^t)] = \langle c^t, x^t \rangle$.

1 Role of Stability in Online learning

Let us start by recalling the lower bound on repeated ERM from prior lectures.

Theorem 1.1. *For any deterministic algorithm (including ERM) $\mathcal{A}$, there is an online sequence such that* $\textsc{Regret}(\mathcal{A}) \geq \left(1 - \frac{1}{n}\right) T$*, where n is the number of experts in the problem.*

Proof. At each iteration t, since $\mathcal{A}$ is deterministic, the adversary can design his cost function as follows: $c^t(i^t) = 1$ if i^t is the expert that the player will choose by $\mathcal{A}$ and $c^t(i) = 0$ for all $i \neq i^t$. In this case, $\sum_{t=1}^{T} c^t(i^t) = T$, while $\min_{i \in [n]} \sum_{t=1}^{T} c^t(i) \leq \frac{T}{n}$ because at least one of the n experts appeared in less than T/n time steps. Thus $\textsc{Regret}(\mathcal{A}) \geq T - \frac{T}{n} = \left(1 - \frac{1}{n}\right) T$. □

This theorem shows the performance of ERM against an worst-case adversary. What made this sequence very challenging to learn was that the best expert so far, i.e., ERM's outcome, kept changing at most time steps. That is, the algorithm was running behind the ERM at the next step. Indeed, this is the main challenge when it comes to online learning. To begin, we will prove the following lemma, often referred to as the "Be the Leader" lemma.

Lemma 1.2 (Be the Leader)**.** *If the strategy of the player is ERM, i.e., $x^t = \arg\min_x \sum_{\tau=1}^{t-1} c^\tau(x)$, then*

$$\textsc{Regret}(\mathsf{ERM}) = \sum_{t=1}^{T} c^t\left(x^t\right) - \min_x \sum_{t=1}^{T} c^t\left(x\right) \leq \sum_{t=1}^{T} \left(c^t\left(x^t\right) - c^t\left(x^{t+1}\right)\right).$$

Proof. Note that, it suffices to prove that

$$\sum_{t=1}^{T} c^t\left(x^{t+1}\right) \leq \min_x \sum_{t=1}^{T} c^t\left(x\right) = \sum_{t=1}^{T} c^t\left(x^{T+1}\right),$$

where the second equality holds by the definition of x^{T+1}. We prove this new inequality by induction. When $T = 1$, the claim follows by definition. Assume

$$\sum_{t=1}^{T-1} c^t\left(x^{t+1}\right) \leq \min_x \sum_{t=1}^{T-1} c^t(x).$$

Then,

$$\begin{aligned}
\sum_{t=1}^{T} c^t\left(x^{t+1}\right) &= \sum_{t=1}^{T-1} c^t\left(x^{t+1}\right) + c^T\left(x^{T+1}\right) \\
&\leq \min_x \sum_{t=1}^{T-1} c^t(x) + c^T\left(x^{T+1}\right) && \text{(Induction Hypothesis)} \\
&\leq \sum_{t=1}^{T-1} c^t(x^{T+1}) + c^T\left(x^{T+1}\right) && \text{(replacing } x \leftarrow x^{T+1}\text{)}
\end{aligned}$$

$$= \sum_{t=1}^{T} c^t\left(x^{T+1}\right)$$
$$= \min_x \sum_{t=1}^{T} c^t(x).$$

as required. □

Now let us see what this lemma implies. If the algorithm's decisions change on only k rounds—namely, if $x^t \neq x^{t+1}$ for at most k time steps—then $\mathrm{Regret}_T(\mathsf{ERM}) \leq k$. More generally, even when $x^t \neq x^{t+1}$ but the loss functions c^t are Lipschitz and the successive decisions x^{t+1} and x^t remain close in distance, the same lemma yields a meaningful (and typically small) regret bound.

2 Follow the Regularized Leader

In this section, we see how we can stabilize ERM so as to achieve a no-regret algorithm. We start by considering ERM with one small change, we add to the history a new cost function $c^0(\cdot)$. This cost function is often called a *regularizer*. We refer to ERM with a regularizer as Follow the Regularized Leader (FTRL) algorithm. See Algorithm 1 for a description.

Algorithm 1 Follow the Regularized Leader (FTRL)

Input: regularizer $R(\cdot)$
$x^1 = \arg\min_{x\in\mathcal{X}} R(x)$
for $t = 2, 3, \ldots, T$ **do**
  $x^t = \arg\min_{x\in\mathcal{X}} \left(R(x) + \sum_{\tau=1}^{t-1} c^\tau(x)\right)$
end for

Theorem 2.1 (FTRL/FTPL Theorem)**.** *Let x^t be the action played by the FTRL algorithm (Algorithm 1) at time t. Then, for any $x^* \in \mathcal{X}$,*

$$\sum_{t=1}^{T} c^t(x^t) - \sum_{t=1}^{T} c^t(x^*) \leq \sum_{t=1}^{T} \left[c^t(x^t) - c^t(x^{t+1})\right] + \left[R(x^*) - R(x^1)\right]$$

In other words, if you regularize your ERM, the regret is the same as in Lemma 1.2, but includes the difference term for the regularizer. The proof is a simple edit to thgeec proof of Lemma 1.2 when adding day 0's cost function.

To see the implications of the FTRL theorem, let's consider the linear cost function again. That is, $c^t(x) = c^t \cdot x$. Then our hope is that the choice of regularizer induces large enough cost so that the choice of x^t and its loss changes slowly, i.e., $c^t(x^t) - c^t(x^{t+1}) \leq \epsilon$, but also the regularizer is small enough so that $R(x^*) - R(x^1)$ is also small. In that case the regret $\leq \epsilon \cdot T + \max_x R(x)$ is hopefully small. So, we need to choose a regularizer that has these properties.

Theorem 2.2. *Let $\mathcal{X} \subseteq \mathbb{R}^n$ be $\mathcal{X} = \{x \mid \|x\|_2 \leq 1\}$ and assume that the cost functions on each step are linear and $\|c\|_2 \leq 1$. Consider the FTRL algorithm with the following regularizer*

$$R(x) = \sqrt{\frac{T}{2}}\|x\|_2^2.$$

Then the regret of FTRL is at most $\sqrt{2T}$.

Proof. Note that $x^1 = 0$ is the minimizer of the regularizer. For all other time steps we have

$$x^{t+1} = \text{argmin}_x \sqrt{\frac{T}{2}}\|x\|_2^2 + \sum_{\tau=1}^{t} c^\tau \cdot x.$$

We find this minimum by taking the gradient and setting it to be equal to 0 as follows.

$$2\sqrt{\frac{T}{2}} \cdot x + \sum_{\tau=1}^{t} c^\tau = 0$$

$$x^{t+1} = -\frac{1}{\sqrt{2T}} \sum_{\tau=1}^{t} c^\tau$$

If we had done this at the time step before, we would have gotten:

$$x^t = -\frac{1}{\sqrt{2T}} \sum_{\tau=1}^{t-1} c^\tau.$$

So

$$x^{t+1} = x^t - \frac{1}{\sqrt{2T}} c^t.$$

This is online gradient descent! It also shows that $c^t \cdot x^t - c^t \cdot x^{t+1}$ changes slowly. Then,

$$\begin{aligned}
\text{Regret} &\leq \sum_{t=1}^{T} c^t \cdot (x^t - x^{t+1}) + R(x^*) - R(0) \\
&\leq \sum_{t=1}^{T} \left(c^t \cdot \frac{1}{\sqrt{2T}} \cdot c^t \right) + \sqrt{\frac{T}{2}}\|x^*\|_2^2 \\
&\leq \sum_{t=1}^{T} \frac{1}{\sqrt{2T}} + \sqrt{\frac{T}{2}} \\
&= \sqrt{\frac{T}{2}} + \sqrt{\frac{T}{2}} \leq \sqrt{2T}
\end{aligned}$$

□

Note that in the above application of FTRL, it's possible that we play $x^t \notin \mathcal{X}$, i.e., it is possible that $\|x^t\| > 1$. Can we get similar guarantees as in Theorem 2.2 if we limited FTRL to play within $\mathcal{X}$? Yes, when $\mathcal{X}$ is a convex body as is in Theorem 2.2. In that case, at every step of the optimization, let $\hat{x}^t$ be what FTRL suggests, and let $x^t \in \mathcal{X}$ be the closest point to $\hat{x}^t$. That is $x^t \in \mathcal{X}$ is a projection of $\hat{x}^t$ on the convex set $\mathcal{X}$. Note that the distance between two points after projection on a convex body is only smaller than before. That is,

$$\|x^t - x^{t+1}\| \leq \|\hat{x}^t - \hat{x}^{t+1}\|.$$

So the stability property maintained by FTRL still holds here after projection.

3 Follow the Perturbed Leader

Let's consider an online routing game, where every day we decide what route to take from home to work. There is a graph $G = (V, E)$ where the domain of our actions are valid paths in G, shown by the set $\mathcal{X} \subseteq \{0, 1\}^E$. Our cost function is a vector c where entry c_i is the traffic or delay on edge i. When taking route x our total delay is $c \cdot x$. While this is a linear cost function, our domain set $\mathcal{X}$ is not convex. Even though this problem is not a convex optimization problem, we can still solve the ERM efficiently in time $poly(|E|)$ by using Dijkstra's algorithm. In this section, we ask whether for linear functions we can turn any efficient ERM into a no-regret algorithm that is also efficient, even if the problem is non-convex?

Let us consider the case of learning with experts. We have $\mathcal{K}$ be the simplex of n dimensions (all probability distributions over all experts): $\{x : \ x_i \geq 0 \ \forall i, \ \sum_{j=1}^n x_j = 1\}$. Set c to be such that c_i is the cost of expert i, which is between 0 and 1. Theorem 2.2 suggests that if we pick a strongly convex cost of $\sqrt{T/2}\|x\|_2^2$ then we can get a no-regret algorithm. While this is true, note that $\sqrt{T/2}\|x\|_2^2$ cannot be interpreted as cost of experts at a time step, simply because it is not linear. So, we ask whether we can use other methods of providing regularization that lead to $c^0(x) = R(x)$ referring to the cost of the experts on time step 0?

The following algorithm, called Follow the Perturbed Leader (FTPL), achieves this exactly. It takes a cost function c^0 that assigns random costs to each expert at time 0. Then, at every time step it picks the expert whose cumulative cost including step 0 is minimized.

Algorithm 2 Follow the Perturbed Leader

for $t = 1, 2, 3, 4, ...T$ **do**
  $c^0 \sim$ distribution
  $x^t = \text{argmin}_x \left(c^0(x) + \sum_{\tau=1}^{t-1} c^\tau(x)\right)$
end for

For an adaptive adversary, it is important that we re-draw c^0 at every step to preserve the randomness of our algorithm. But for the oblivious adversary we could take c^0 once at the beginning and reuse the same cost throughout. The expected regret of both algorithms is the same.

Theorem 3.1. *Let $\mathcal{X} \subset \mathbb{R}^n$ be any set (not-necessarily convex) such that $\max_{x,x'} \|x - x'\|_1 \leq D$. Assume that the cost functions are such that $\|c^t\|_1 \leq A$. Furthermore, assume that for any x and x, $|c \cdot x| \leq R$. Then, Follow the Perturbed Leader with $c^0 \sim \text{Unif}\left[0, \frac{2}{\epsilon}\right]^n$ has*

$$\mathbb{E}\left[\textit{Regret}\right] \leq \frac{2D}{\epsilon} + TRA\epsilon.$$

Note that for $\epsilon = \sqrt{\frac{2D}{ART}}$, this leads to $\mathbb{E}\left[\textit{Regret}\right] \leq \sqrt{2ARDT}$.

Proof. Recall from Theorem 2.2 that

$$\mathbb{E}\left[\text{Regret}\right] \leq \mathbb{E}\left[c^0 \cdot (x^* - x^1) + \sum_{t=1}^{T} c^t \cdot (x^t - x^{t+1})\right] \tag{1}$$

For the first term in the above inequality we have

$$c^0 \cdot (x^* - x^1) \leq \|c^0\|_\infty \|x^* - x^1\|_1 \leq \frac{2}{\epsilon} D.$$

Next, we will analyze the second term in Equation 1. By linearity of expectation, we only need to show that $\mathbb{E}[c^t \cdot x^t] - \mathbb{E}[c^t \cdot x^{t+1}] \leq RA\epsilon$ for a fixed time step t. To help us with the notation, we'll define $\text{Box}(a, b) = x \mid a_i \leq x_i \leq b_i$. Note that FTPL is choosing a random $c^0 \sim \text{Box}(0, \frac{2}{\epsilon})$.

We will complete this proof in the next lecture. For now, we just define a useful quantity that will be used in the proof in the next time.

$$p := \Pr_{c^0 \sim \text{Box}(0, 2/\epsilon)}\left[c^0 \notin \text{Box}(c^t, \frac{2}{\epsilon})\right] = \frac{\text{Box}(0, \frac{2}{\epsilon}) \setminus \text{Box}(c^t, \frac{2}{\epsilon})}{\text{Box}(0, \frac{2}{\epsilon})} = \frac{\text{Box}(0, \frac{2}{\epsilon}) \setminus \text{Box}(0, \frac{2}{\epsilon} - c^t)}{\text{Box}(0, \frac{2}{\epsilon})}$$

where the last equality is by the symmetric nature of the box definitions. □