Machine Learning Theory (CSC 431/531) - Lectures 15 and 16

Nishant Mehta

1 Prediction with expert advice

In the game of prediction with expert advice, there is an action space $\mathcal{A}$, an outcome space $\mathcal{Y}$, a loss function $\ell : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$ mapping each a given action $a \in \mathcal{A}$ and outcome $y \in \mathcal{Y}$ to a loss $\ell(y, a)$. At the start of each round, each of K experts provides Learner with advice in the form of a suggested action from $\mathcal{A}$. Learner then aggregates these actions in some way, producing its own action in $\mathcal{A}$. Finally, Nature selects an outcome, and Learner and each expert suffer loss according to their respective actions and the outcome. The goal of Learner is to ensure that its *regret* over T rounds is small, where the regret is defined as the amount by which Learner's cumulative loss exceeds the cumulative loss of the best action in hindsight. The game protocol is given below.

Algorithm 1: PREDICTION WITH EXPERT ADVICE

```
for t = 1 → T do
    Nature plays expert advice f_{j,t} ∈ A for each expert j ∈ [K] and reveals the advice
      (f_{1,t}, f_{2,t}, ..., f_{K,t}) to Learner
    Learner plays action a_t ∈ A
    Nature plays outcome y_t ∈ Y and reveals it to Learner
    Each expert j ∈ [K] suffers loss ℓ(y_t, f_{j,t}) and Learner suffers loss ℓ(y_t, a_t)
end
```

Note that Nature controls both the experts and the outcomes. In sequential prediction problems, the strength of the adversary (Nature) can vary; the adversary can be either:

- *oblivious* - Nature knows which algorithm Learner is using, but Nature must commit to its entire sequence of expert advice and outcomes before Learner takes its first action.
- *non-oblivious* or *adaptive* - At any point in time, Nature can make its choice (whether expert advice or outcome) based on all of Learner's previous actions.

We will make no assumptions about Nature: Nature will be a non-oblivious adversary. Anytime Nature makes a selection, it can do so using all the information revealed thus far. To be clear, the expert advice $f_{j,t}$ for any expert j can be selected with knowledge of $a_1, a_2, \ldots a_{t-1}$, while outcome y_t can be selected with the same knowledge as well as a_t.

The regret takes the form

$$\mathcal{R}_T := \sum_{t=1}^{T} \ell(y_t, a_t) - \min_{j \in [K]} \sum_{t=1}^{T} \ell(y_t, f_{j,t}).$$

That is, the regret is simply Learner's cumulative loss minus the cumulative loss of the best expert in hindsight. Intuitively, regret quantifies how much sadness Learner feels for not having always played the best action in hindsight. A basic demand in online learning is to have a learning algorithm that is *no-regret*, meaning that the time-average of the regret $\frac{1}{T}\mathcal{R}_T$ goes to zero as T approaches infinity, or, equivalently, the regret $\mathcal{R}_T$ is sublinear in T. To see why such a requirement is sensible, consider the case where the loss function takes values in some bounded range. Then if Learner has regret growing linearly in T, up to a multiplicative constant it is doing no better than constantly playing the *worst* action in hindsight! Another perspective comes from a stochastic interpretation of the data. If we were in the statistical learning setting, then at a high level we may think of the time-average of the regret as a form of risk, and the no-regret property is the analogue of the excess risk decaying to zero as the sample size n approaches infinity.

Given Nature's ability to adapt to the previous plays of Learner (and, in particular, to select y_t with knowledge of a_t), one wonders if any algorithm can always (i.e., against any strategy of Nature) obtain sublinear regret for this problem. Without further assumptions on the action space and loss function, Nature actually can ensure that any algorithm is forced to suffer linear regret. However, if we assume the action space is convex, the loss function is convex with respect to its second argument, and the losses are in some bounded range (e.g., the unit interval $[0, 1]$), then we can show that the worst-case regret is sublinear.

Two common examples satisfying these assumptions are:

- Classification with absolute loss: Here, we take $\mathcal{A} = [0, 1]$, $\mathcal{Y} = \{0, 1\}$, and $\ell(y, a) = |a - y|$.
- Classification with squared loss: We take $\mathcal{A}$ and $\mathcal{Y}$ as before and now set $\ell(y, a) = (a - y)^2$.

To simplify the presentation, we set up some notation. For any $j \in [K]$ and $t \in [T]$, let $\ell_{j,t} = \ell(y_t, f_{j,t})$ denote the loss of expert j in round t, and let $L_{j,t} = \sum_{s=1}^{t} \ell_{j,s}$ denote expert j's cumulative loss at the end of round t. Also, for any $t \in [T]$, denote Learner's loss in round t $\hat{\ell}_t = \ell(y_t, a_t)$, and denote Learner's cumulative loss at the end of round t by $\hat{L}_t = \sum_{s=1}^{t} \hat{\ell}_s$.

2 Exponentially Weighted Average Forecaster

The algorithm that we study for this setting is called the exponentially weighted average forecaster (EWA forecaster); it works as follows. In each round, the algorithm maintains weights over the experts, with $w_{j,t} = e^{-\eta L_{j,t}}$ indicating the weight of the j^{th} expert at the end of round t. In round t, the forecaster predicts according to the following weighted average of the experts' actions:

$$a_t = \frac{\sum_{j=1}^{K} w_{j,t-1} \, f_{j,t}}{\sum_{j=1}^{K} w_{j,t-1}}.$$

It will be convenient to rewrite the above using normalized weights; to this end, we introduce the probability vector $p_t \in \Delta_K$, defined as

$$p_{j,t} = \frac{w_{j,t-1}}{\sum_{i=1}^{K} w_{i,t-1}}.$$

Using p_t, we can re-express a_t as $a_t = \mathsf{E}_{j \sim p_t}[a_{j,t}]$.

In later lectures, we may consider a time-varying learning rate. However, since the learning rate η is currently constant throughout the rounds, we also can write an incremental update for the weights for all $j \in [K]$ by:

- initializing via $w_{j,0} = 1$;
- at the end of round t, incrementally updating via $w_{j,t} = w_{j,t-1} \cdot e^{-\eta \ell_{j,t}}$.

I emphasize that if the learning rate is not constant, then we *should not using the incremental form of the update*; doing so can result in linear regret.

We are now ready to see an upper bound on the worst-case regret of the EWA forecaster.

Theorem 1. *Assume $\mathcal{A}$ is convex, the loss function is convex in its second argument, and the losses are in the range $[0,1]$. For any learning rate $\eta > 0$, any sequence of expert advice $(f_{j,t})_{j \in [K], t \in [T]}$, and any sequence of outcomes $y_1, \ldots, y_T$, the regret of the EWA forecaster satisfies*

$$\hat{L}_T - \min_{j \in [K]} L_{j,T} \leq \frac{\log K}{\eta} + \frac{T\eta}{8}.$$

In particular, setting $\eta = \sqrt{\frac{8 \log K}{T}}$ makes the upper bound $\sqrt{\frac{T \log K}{2}}$.

In order to prove this result, we will use a result called Hoeffding's lemma, the key supporting lemma for proving Hoeffding's inequality.

Lemma 1 (Hoeffding's lemma)**.** *Let X be a random variable satisfying $\mathsf{E}[X] = 0$ and $a \leq X \leq b$. Then for any $\lambda \in \mathbb{R}$,*

$$\log \mathsf{E}[e^{\lambda X}] \leq \frac{\lambda^2 (b-a)^2}{8}.$$

The idea of the proof of Theorem 1 is to relate Learner's cumulative loss to the cumulative loss of a *pseudo-strategy* known as the Aggregating Pseudo-Algorithm (APA). A pseudo-strategy is a strategy that might not be implementable in reality; in short, it might be "too good to be true" (there might not be any algorithm which performs as well as the pseudo-strategy). Even so, the APA provides a convenient frame of reference for the proof. Moreover, thinking in terms of the APA will be very useful later when we consider specific types of losses for which much smaller regret bounds can be established.

Since the APA might not be implementable, we will only describe the loss it suffers in each round; that is, we will not describe an action the APA plays to actually realize that loss. In each round t, the APA suffers the *mix loss* m_t, defined as

$$m_t = -\frac{1}{\eta} \log \mathsf{E}_{j \sim p_t} \left[e^{-\eta \ell_{j,t}} \right].$$

Let $M_T = \sum_{t=1}^T m_t$ denote the cumulative mix loss up to round T. Let us get a first intuition about the mix loss. Consider some fixed values for the losses $\ell_{1,t}, \ldots, \ell_{K,t}$. Among these losses note that large losses contribute less to the expectation $\mathsf{E}_{j \sim p_t} \left[e^{-\eta \ell_{j,t}} \right]$ than do small losses due to presence of the exponential function. Consequently, the mix loss m_t can never be larger than the expected loss $\mathsf{E}_{j \sim p_t} [\ell_{j,t}]$. In fact, as η approaches zero, the expected loss and mix loss are the same.

The high-level idea of the proof of Theorem 1 is to start with the regret decomposition

$$\hat{L}_T - \min_{j \in [K]} L_{j,T} = \underbrace{\left(\hat{L}_T - M_T \right)}_{\text{regret of EWA vs APA}} + \underbrace{\left(M_T - \min_{j \in [K]} L_{j,T} \right)}_{\text{regret of APA}}. \tag{1}$$

The first term, $\hat{L}_T - M_T$, is the regret of EWA relative to the APA. We control this term a bit later, in Lemma 3. The second term, $M_T - \min_{j \in [K]} L_{j,T}$, is the regret of the APA. The next lemma shows that as long as the learning rate η is not too small, the APA's regret is small.

Lemma 2. *The regret of the APA is upper bounded as*

$$M_T - \min_{j \in [K]} L_{j,T} \leq \frac{\log K}{\eta}.$$

Proof. We show that the cumulative mix loss (the cumulative loss of the APA) is never much worse than the cumulative loss of the best expert in hindsight:

$$\begin{aligned}
M_T &= \sum_{t=1}^{T} -\frac{1}{\eta} \log \mathsf{E}_{j \sim p_t}[\exp(-\eta \ell_{j,t})] \\
&= -\frac{1}{\eta} \sum_{t=1}^{T} \log \frac{\sum_{j=1}^{K} \exp(-\eta L_{j,t-1}) \exp(-\eta \ell_{j,t})}{\sum_{j=1}^{K} \exp(-\eta L_{j,t-1})} \\
&= -\frac{1}{\eta} \sum_{t=1}^{T} \log \frac{\sum_{j=1}^{K} \exp(-\eta L_{j,t})}{\sum_{j=1}^{K} \exp(-\eta L_{j,t-1})} \\
&= -\frac{1}{\eta} \log \sum_{j=1}^{K} \exp(-\eta L_{j,T}) + \frac{1}{\eta} \log \sum_{j=1}^{K} \exp(-\eta L_{j,0}) \\
&\leq -\frac{1}{\eta} \log \max_{j \in [K]} \exp(-\eta L_{j,T}) + \frac{1}{\eta} \log \sum_{j=1}^{K} \exp(-\eta L_{j,0}) \\
&= \min_{j \in [K]} L_{j,T} + \frac{\log K}{\eta},
\end{aligned}$$

where the fourth step is from telescoping and the inequality uses the fact that the summation of nonnegative terms is at least as large as their maximum term. □

Next, in order to bound $\hat{L}_T - M_T$, it suffices to bound each term $\hat{\ell}_t - m_t$.

Lemma 3. *Under the same assumptions as Theorem 1, we have for any $t \in [T]$ that*

$$\hat{\ell}_t - m_t \leq \frac{\eta}{8}.$$

Before proving the lemma, note that its implication

$$\hat{L}_T - M_T \leq \frac{T\eta}{8},$$

together with the decomposition (1) and Lemma 2, proves Theorem 1.

Proof (of Lemma 3). The proof is in two steps. First, by assumption, for any outcome y_t, the loss function is convex in its second argument. Therefore, Jensen's inequality gives

$$\hat{\ell}_t = \ell(y_t, \mathsf{E}_{j \sim p_t}[f_{j,t}]) \leq \mathsf{E}_{j \sim p_t}[\ell(y_t, f_{j,t})] =: h_t.$$

Note that $\hat{\ell}_t$ is the loss obtained by mixing "inside the loss", i.e., by mixing the expert advice to obtain a mixed action. On the other hand, the loss h_t is the expected loss obtained by mixing "outside the loss", i.e., if Learner were to draw an expert according to distribution p_t and then play the selected expert's advice as its own action.[1]

[1] The reason for the notation h_t is that the algorithm Hedge — for an online learning setting called Decision-Theoretic Online Learning (DTOL) — is precisely the algorithm that mixes outside the loss while still using the exponential weights strategy. As we will see later, Hedge's regret bound will be effortless given Theorem 1.

Then we have

$$\begin{aligned}\hat{\ell}_t - m_t &\leq h_t - m_t \\ &= \mathsf{E}_{j\sim p_t}[\ell_{j,t}] + \frac{1}{\eta}\log \mathsf{E}_{j\sim p_t}[e^{-\eta \ell_{j,t}}] \\ &= \frac{1}{\eta}\log e^{\eta\, \mathsf{E}_{j\sim p_t}[\ell_{j,t}]} + \frac{1}{\eta}\log \mathsf{E}_{j\sim p_t}[e^{-\eta \ell_{j,t}}] \\ &= \frac{1}{\eta}\log \mathsf{E}_{j\sim p_t}[e^{-\eta(\ell_{j,t} - \mathsf{E}_{j\sim p_t}[\ell_{j,t}])}].\end{aligned}$$

The last line may seem a peculiar way to rewrite the expression from the second line. The rewrite is motivated by the fact that the random variable $\ell_{j,t} - \mathsf{E}_{j\sim p_t}[\ell_{j,t}]$ is centered, and so we may bound this expression using Hoeffding's lemma (Lemma 1). Note that the losses are in $[0, 1]$ (and shifting the losses by their expectation does not change the difference $(b-a)$ in Hoeffding's lemma). Therefore, Hoeffding's lemma implies that

$$\log \mathsf{E}_{j\sim p_t}[e^{-\eta(\ell_{j,t} - \mathsf{E}_{j\sim p_t}[\ell_{j,t}])}] \leq \frac{\eta^2}{8}.$$

Consequently,

$$\hat{\ell}_t - m_t \leq \frac{\eta}{8}.$$

□

The proof of Theorem 1 curiously revolves around the notion of a mix loss, which is the loss of the APA. Is it possible that under certain natural assumptions on the loss function that there exists an algorithm which can always achieve loss no larger than the APA? As it turns out, the answer is yes.

Definition 1 (Mixability)**.** We say a loss is η-mixable if, for all $P \in \Delta(\mathcal{A})$, there exists an action $a_P \in \mathcal{A}$ such that

$$\ell(y, a_P) \leq -\frac{1}{\eta}\log \mathsf{E}_{a\sim P}\left[e^{-\eta \ell(y,a)}\right] \quad \text{for all } y \in \mathcal{Y}.$$

Mixability is precisely the assumption that there exists a *substitution function* $\Sigma\colon \Delta(\mathcal{A}) \to \mathcal{A}$ which, given a mix loss based on mixing using a distribution P, outputs an actually playable action $\Sigma(P) = a_P$ which, no matter the outcome y, has loss at most the mix loss. Momentarily, we will see common examples of η-mixable losses for specific values of η. Before that, let us see how the regret bound in Theorem 1 can be greatly improved under the assumption of η-mixability.

Let the Aggregating Algorithm (AA) be the same as APA except in each round t, it plays the actual action $\Sigma(p_t)$. The following theorem is immediate.

Theorem 2. *For some $\eta > 0$, assume the loss is η-mixable with substitution function Σ. Then, for any sequence of expert advice $(f_{j,t})_{j\in[K],t\in[T]}$ and any sequence of outcomes $y_1, \ldots, y_T$, the AA run with substitution function Σ has regret at most*

$$\hat{L}_T - \min_{j\in[K]} L_{j,T} \leq \frac{\log K}{\eta}.$$

Remarkably, for η-mixable losses the regret is *constant* with respect to T. To let that sink in, one could play a game for an indefinitely long period, and the regret of the AA will never grow beyond the constant $\frac{\log K}{\eta}$. One might again wonder: do η-mixable losses really exist? Again, the answer is yes. We will focus on a subclass of mixable loss functions known as exp-concave loss functions, for which the substitution takes a very simple form.

3 Exp-concave losses

Exp-concave loss functions are of interest for at least two reasons. First, they encompass several well-known and widely-used loss functions, including squared loss, logistic loss, and log loss. Second, they are mixable and hence the AA enjoys a constant regret bound for these losses.

Definition 2. We say that a loss function ℓ is η-exp-concave if, for each outcome $y \in \mathcal{Y}$, the function $a \mapsto e^{-\eta\ell(y,a)}$ is concave. Equivalently, ℓ is η-exp-concave if, for all $y \in \mathcal{Y}$ and all distributions P over $\mathcal{A}$,

$$\mathsf{E}_{a\sim P}\left[e^{-\eta\ell(y,a)}\right] \le e^{-\eta\ell(y,\mathsf{E}_{a\sim P}[a])}. \tag{2}$$

Note that (2) may be rewritten as

$$-\frac{1}{\eta}\log \mathsf{E}_{a\sim P}\left[e^{-\eta\ell(y,a)}\right] \ge \ell(y, \mathsf{E}_{a\sim P}[a]),$$

which implies that $\Sigma(P) = \mathsf{E}_{a\sim P}[a]$ is a substitution function. Therefore, if a loss is η-exp-concave, it also is η-mixable; hence, Theorem 2 applies and the regret is at most $\frac{\log K}{\eta}$.

Our first and simplest example is log loss. Prediction with expert advice with log loss is specified by taking $\mathcal{A} = [0,1]$, $\mathcal{Y} = \{0,1\}$, and

$$\ell(y,a) = -y\log a - (1-y)\log(1-a).$$

Log loss is 1-exp-concave, as is readily verified by considering the two cases. For instance, if $y = 1$, then the function

$$a \mapsto e^{-\ell(1,a)} = e^{\log a} = a$$

is clearly concave.

Our second example is squared loss, with $\mathcal{A} = \mathcal{Y} = [0,1]$ and $\ell(y,a) = (a-y)^2$. In order to establish the exp-concavity of squared loss, we will use an alternate characterization of exp-concavity. For the time being, we restrict to one-dimensional actions a for simplicity. Take $\mathcal{X} \subset \mathbb{R}$; it turns out that twice-differentiable function $g\colon \mathcal{X} \to \mathbb{R}$ is concave if $g''(x) \le 0$ for all $x \in \mathcal{X}$. Now, using the definition of η-exp-concavity, we see that a function $f\colon \mathcal{X} \to \mathbb{R}$ is η-exp-concave if and only if

$$\eta^2(f'(x))^2 e^{-\eta f(x)} - \eta f'(x) e^{-\eta f(x)} \le 0 \qquad \text{for all } x \in \mathcal{X},$$

which is equivalent to the condition

$$\eta(f'(x))^2 \le f''(x) \qquad \text{for all } x \in \mathcal{X}.$$

Returning to the example of squared loss, we can verify that the squared loss is η-exp-concave if and only if

$$\eta\,(2(a-y))^2 \leq 2 \qquad \text{for all } a, y \in [0,1],$$

or equivalently,

$$(a-y)^2 \leq \frac{1}{2\eta} \qquad \text{for all } a, y \in [0,1].$$

This condition is satisfied for $\eta = \frac{1}{2}$, and so the squared loss is $\frac{1}{2}$-exp-concave.

4 Decision-theoretic online learning

We now introduce the setting of decision-theoretic online learning (DTOL). This setting is a very important special case of prediction with expert advice. The DTOL protocol unfolds as follows.

Algorithm 2: DECISION-THEORETIC ONLINE LEARNING

for $t = 1 \to T$ **do**
  Learner plays probability distribution $p_t \in \Delta_K$
  Nature plays loss vector $\ell_t = (\ell_{1,t}, \ldots, \ell_{K,t})^\top \in [0,1]^K$ and reveals it to Learner
  Each expert $j \in [K]$ suffers loss $\ell_{j,t}$ and Learner suffers loss $\langle \ell_t, p_t \rangle$
end

This setting could be thought of as "prediction with expert advice *without the expert advice*". Learner never gets to observe each expert's action (advice), but it still gets to observe each expert's loss. As such, Learner cannot mix the advice of the experts — it cannot mix inside the loss — but it *can* randomize over the experts — it can mix outside the loss. The interpretation of $\langle \ell_t, p_t \rangle = \mathsf{E}_{j \sim p_t}[\ell_{j,t}]$ is then Learner's expected loss if it randomly selects expert j with probability $p_{j,t}$. Alternatively, if Learner is allocating a fixed sum of money across the experts, then p_t specifies how Learner divides its money among the experts, and $\langle \ell_t, p_t \rangle$ is Learner's actual (not expected) loss.

Let us see how DTOL is a special case of prediction with expert advice. In prediction with expert advice, we take action space $\mathcal{A} = \Delta_K$ and outcome space $\mathcal{Y} = [0,1]^K$. Next, for given action $a_t \in \mathcal{A}$ and outcome $y_t \in \mathcal{Y}$ — which we may view as probability vector $p_t \in \Delta_K$ and loss vector $\ell_t \in [0,1]^K$ respectively — the loss is $\ell(y_t, a_t) = \langle y_t, a_t \rangle = \langle \ell_t, p_t \rangle$. That is, for a fixed outcome, the loss function is linear in the action. Finally, each expert $j \in [K]$ uses the *constant* strategy of setting (for all t) $f_{j,t}$ equal to e_j; here, e_j is the j^{th} standard basis vector in $\mathbb{R}^K$. This choice satisfies $\ell(y_t, f_{j,t}) = \ell(\ell_t, e_j) = \langle \ell_t, e_j \rangle = \ell_{j,t}$, as desired.

What regret can Learner hope to acheve in DTOL? Since the losses are in $[0,1]$, the loss function is convex in its second argument (recall that linear functions are convex), and the action space is convex, Learner can use the EWA forecaster. Our regret bound Theorem 1 applies, and so for all sequences of loss vectors, the regret is at most $\sqrt{\frac{T \log K}{2}}$.

It is a bit clunky to view the algorithm as the EWA forecaster for prediction with expert advice in the special case of DTOL. For simplicity, we directly present the algorithm, which is known as Hedge, below.

Algorithm 3: HEDGE

Input: $\eta > 0$

Set $w_{j,0} = 1$ for $j = 1, \ldots, K$

for $t = 1 \to T$ **do**

  Set $p_{j,t} = \frac{w_{j,t-1}}{\sum_{i=1}^{K} w_{i,t-1}}$ for $j = 1, \ldots, K$

  Observe loss vector $\boldsymbol{\ell}_t$ from Nature

  Suffer loss $\boldsymbol{p}_t \cdot \boldsymbol{\ell}_t$

  Set $w_{j,t} = w_{j,t-1} e^{-\eta \ell_{j,t}}$ for $j = 1, \ldots, K$

end

For convenience, we also summarize Hedge's regret guarantee.

Theorem 3. *Let Hedge be run with learning rate* $\eta = \sqrt{\frac{8 \log K}{T}}$. *Then, for all sequences of loss vectors* $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T$,

$$\hat{L}_T \leq \min_{j \in [K]} L_{j,T} + \sqrt{\frac{T \log K}{2}}.$$