Lecture 10: Agnostic Online Learning

Han Shao

Section 2.2, *Prediction, Learning, and Games*

We now remove the realizability assumption from the online learning setting. The goal is no longer to make only finitely many mistakes, but to perform nearly as well as the best hypothesis in hindsight.

1 Weighted Majority

Let $\mathcal{H}$ be a finite, nonempty hypothesis class, and use the label space $\mathcal{Y} = \{0, 1\}$. For a sequence $(x_1, y_1), \ldots, (x_T, y_T)$, define

$$\mathrm{OPT} = \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{1}\{h(x_t) \neq y_t\}, \qquad M = \sum_{t=1}^{T} \mathbb{1}\{\widehat{y}_t \neq y_t\}.$$

Thus OPT is the number of mistakes made by the best fixed hypothesis, and M is the number of mistakes made by the learner. We would like a guarantee of the form

$$M \leq \alpha \, \mathrm{OPT} + \beta$$

for desirable values of α and β.

The Weighted Majority Algorithm

Rather than removing a hypothesis after a mistake, we reduce its weight. The learner predicts by taking a weighted majority vote.

Algorithm 1 Weighted Majority

Initialize $w_h^1 = 1$ for every $h \in \mathcal{H}$.
for $t = 1, \ldots, T$ **do**
  Receive x_t.
  if $\sum_{h:h(x_t)=1} w_h^t > \sum_{h:h(x_t)=0} w_h^t$ **then**
    Predict $\widehat{y}_t = 1$.
  else
    Predict $\widehat{y}_t = 0$.
  end if
  Observe y_t and update, for every $h \in \mathcal{H}$,

$$w_h^{t+1} = \begin{cases} \frac{1}{2} w_h^t & \text{if } h(x_t) \neq y_t, \\ w_h^t & \text{otherwise.} \end{cases}$$

end for

Theorem 1 (Weighted Majority Mistake Bound)**.** *For every labeled sequence, Weighted Majority makes at most*

$$M \le \frac{\mathrm{OPT} + \log_2 |\mathcal{H}|}{\log_2(4/3)} \approx 2.4(\mathrm{OPT} + \log_2 |\mathcal{H}|)$$

mistakes.

Proof. Let $h^\star$ be a hypothesis that makes OPT mistakes in the first T rounds, and let

$$W^t = \sum_{h \in \mathcal{H}} w_h^t$$

be the total weight at the beginning of round t. Since $h^\star$ loses half its weight each time it makes a mistake,

$$W^{T+1} \ge w_{h^\star}^{T+1} = 2^{-\mathrm{OPT}}. \tag{1}$$

On the other hand, whenever the learner makes a mistake, the hypotheses that predicted incorrectly have at least half the total weight. Hence, on such a round,

$$\begin{aligned} W^{t+1} &= W^t - \frac{1}{2} \sum_{h : h(x_t) \neq y_t} w_h^t \\ &\le W^t - \frac{1}{2} \cdot \frac{1}{2} W^t = \frac{3}{4} W^t. \end{aligned}$$

The total weight never increases, including on rounds when the learner is correct. Since $W^1 = |\mathcal{H}|$, after M mistakes,

$$W^{T+1} \le \left(\frac{3}{4}\right)^M |\mathcal{H}|.$$

Combining this with (1) gives

$$2^{-\mathrm{OPT}} \le W^{T+1} \le \left(\frac{3}{4}\right)^M |\mathcal{H}|.$$

Taking base-2 logarithms and rearranging,

$$M \log_2 \frac{4}{3} \le \mathrm{OPT} + \log_2 |\mathcal{H}|,$$

which proves the theorem. □

The multiplicative factor of about 2.4 is not very satisfactory. For example, if the best hypothesis makes a mistake on 25% of the rounds, this bound permits a mistake rate of about 60%, even before the additive term. Random guessing makes a mistake with probability 1/2 on each round, provided the current label is chosen without seeing the learner's current random prediction.
Ideally, we would like $\alpha = 1$ and an additive term that is sublinear in T. Randomization is necessary for such a worst-case guarantee; the limitation of deterministic algorithms is left for homework.

2 Hedge: Randomized Weighted Majority

In online learning, hypotheses are often called *experts*, and their predictions are called *expert advice*. We now extend the setting from the 0–1 loss to arbitrary losses in $[0, 1]$. The algorithm is called *Hedge*, *Randomized Weighted Majority*, or *Multiplicative Weights*.

Prediction with Expert Advice

There are N experts, indexed by $[N]$. In every round t:

1. The environment chooses a loss vector $\ell_t \in [0,1]^N$, where $\ell_t(i)$ is the loss of following expert i.

2. Without seeing ℓ_t, the learner chooses an expert $i_t \in [N]$ and incurs loss $\ell_t(i_t)$.

3. The entire loss vector ℓ_t is revealed to the learner.

The environment may know the algorithm and adapt to the previous history, including $i_1, \dots, i_{t-1}$ and $\ell_1, \dots, \ell_{t-1}$. It cannot choose ℓ_t based on the learner's current random choice i_t.
We play for T rounds and compare the learner's cumulative loss with that of the best fixed expert, $\min_{i\in[N]} \sum_{t=1}^{T} \ell_t(i)$. We write $\mathbb{E}_{\mathcal{A}}$ for expectation over the learner's internal randomness against a fixed adversary strategy.

The Algorithm and Its Guarantee

Algorithm 2 Hedge

Require: Learning rate $\eta > 0$.

Initialize $w_1(i) = 1$ for every $i \in [N]$.
for $t = 1, \dots, T$ **do**
  Set

$$p_t(i) = \frac{w_t(i)}{\sum_{j=1}^{N} w_t(j)} \qquad \text{for every } i \in [N].$$

  Sample $i_t \sim p_t$ and incur loss $\ell_t(i_t)$.
  Observe the entire loss vector ℓ_t.
  Update $w_{t+1}(i) = w_t(i)\exp(-\eta \ell_t(i))$ for every $i \in [N]$.
end for

Theorem 2 (Hedge Guarantee)**.** *For every horizon $T \geq 1$, learning rate $\eta > 0$, and adaptive adversary satisfying the timing restriction above, Hedge satisfies*

$$\mathbb{E}_{\mathcal{A}}\left[\sum_{t=1}^{T} \ell_t(i_t) - \min_{i\in[N]} \sum_{t=1}^{T} \ell_t(i)\right] \leq \frac{\ln N}{\eta} + \frac{\eta T}{8}. \tag{2}$$

In particular, for $N \geq 2$, setting $\eta = \sqrt{8 \ln N / T}$ gives

$$\mathbb{E}_{\mathcal{A}}\left[\sum_{t=1}^{T} \ell_t(i_t) - \min_{i\in[N]} \sum_{t=1}^{T} \ell_t(i)\right] \leq \sqrt{\ln(N) T / 2}.$$

Proof. Let

$$L_t(i) = \sum_{s=1}^{t} \ell_s(i), \qquad L_0(i) = 0, \qquad W_t = \sum_{i=1}^{N} w_t(i).$$

The update rule gives

$$w_t(i) = \exp(-\eta L_{t-1}(i)), \qquad W_t = \sum_{i=1}^{N} \exp(-\eta L_{t-1}(i)).$$

We compare a lower and an upper bound on $\ln(W_{T+1}/W_1)$.

Lower bound on the final weight. For a realized loss sequence, choose $i^\star \in \operatorname{argmin}_{i\in[N]} L_T(i)$. Since $W_1 = N$,

$$\begin{aligned}\ln \frac{W_{T+1}}{W_1} &= \ln\left(\sum_{i=1}^{N} e^{-\eta L_T(i)}\right) - \ln N \\ &\geq \ln\left(e^{-\eta L_T(i^\star)}\right) - \ln N \\ &= -\eta L_T(i^\star) - \ln N. \end{aligned} \tag{3}$$

Upper bound on each weight ratio. For every t,

$$\begin{aligned}\ln \frac{W_{t+1}}{W_t} &= \ln\left(\frac{\sum_{i=1}^{N} w_t(i) e^{-\eta \ell_t(i)}}{\sum_{i=1}^{N} w_t(i)}\right) \\ &= \ln\left(\sum_{i=1}^{N} p_t(i) e^{-\eta \ell_t(i)}\right) \\ &= \ln \mathbb{E}_{i\sim p_t}\left[e^{-\eta \ell_t(i)}\right] \\ &\leq -\eta \mathbb{E}_{i\sim p_t}[\ell_t(i)] + \frac{\eta^2}{8}. \end{aligned}$$

The final inequality uses the following lemma, stated without proof.

Lemma 1 (Hoeffding's Lemma)**.** *Let X be a random variable with $a \leq X \leq b$ almost surely. Then, for every $s \in \mathbb{R}$,*

$$\ln \mathbb{E}[e^{sX}] \leq s\mathbb{E}[X] + \frac{s^2(b-a)^2}{8}.$$

Here we apply the lemma with $X = \ell_t(i)$ for $i \sim p_t$, $a = 0$, $b = 1$, and $s = -\eta$. Define the cumulative mixture loss

$$L_T(\mathcal{A}) = \sum_{t=1}^{T} \sum_{i=1}^{N} p_t(i)\ell_t(i).$$

Summing the weight-ratio bound over the T rounds gives

$$\ln \frac{W_{T+1}}{W_1} \leq -\eta L_T(\mathcal{A}) + \frac{\eta^2 T}{8}. \tag{4}$$

Combining the two bounds. By (3) and (4),

$$-\eta L_T(i^\star) - \ln N \leq -\eta L_T(\mathcal{A}) + \frac{\eta^2 T}{8}.$$

Rearranging yields the following inequality:

$$L_T(\mathcal{A}) \leq \min_{i\in[N]} L_T(i) + \frac{\ln N}{\eta} + \frac{\eta T}{8}. \tag{5}$$

□

3 Regret and an Unknown Time Horizon

Definition 1 (Regret)**.** *For an algorithm $\mathcal{A}$ and a fixed sequence of loss vectors $\ell_{1:T} = (\ell_1, \dots, \ell_T)$, define*

$$\mathrm{Reg}_T(\mathcal{A}, \ell_{1:T}) = \mathbb{E}_{\mathcal{A}}\left[\sum_{t=1}^{T} \ell_t(i_t)\right] - \min_{i \in [N]} \sum_{t=1}^{T} \ell_t(i).$$

An algorithm has regret bound R_T *if its expected regret is at most R_T against every allowed adversary.*

With the tuned learning rate, Hedge has regret bound $O(\sqrt{T \ln N})$. The tuning uses the horizon T in advance. When T is unknown, we can use the *doubling trick*.

The Doubling Trick Run Hedge in successive epochs of planned lengths

$$T_0 = 1, \quad T_1 = 2, \quad T_2 = 4, \quad \dots, \quad T_k = 2^k.$$

At the beginning of epoch k, reset all weights to 1 and use

$$\eta_k = \sqrt{\frac{8 \ln N}{2^k}}.$$

When the epoch ends, double the planned length and restart.
If an epoch stops after $u \le 2^k$ rounds, the same potential bound gives

$$\frac{\ln N}{\eta_k} + \frac{\eta_k u}{8} \le \frac{\ln N}{\eta_k} + \frac{\eta_k 2^k}{8} = \sqrt{\frac{2^k \ln N}{2}}.$$

Thus the bound also covers the last, possibly incomplete epoch. Comparing each epoch with the same fixed expert and then taking the best expert over all T rounds gives a total regret bound at most

$$\sum_{k=0}^{\lfloor \log_2 T \rfloor} \sqrt{\frac{2^k \ln N}{2}} = O(\sqrt{T \ln N}).$$

The equality in order follows by summing a geometric series with ratio $\sqrt{2}$. Although there are $O(\log T)$ epochs, this does not introduce an additional logarithmic factor.

Definition 2 (No Regret)**.** *An algorithm has* no regret *if it admits a regret bound R_T such that*

$$\frac{R_T}{T} \longrightarrow 0 \qquad \text{as } T \to \infty,$$

that is, $R_T = o(T)$.

For fixed N, Hedge with the doubling trick has no regret, since $\sqrt{T \ln N}/T \to 0$. Here sublinearity refers to a uniform upper bound on regret; on a particular sequence the learner can outperform the best fixed expert, so its regret can be negative.