Notes for COMS 6998-4 Fall 2017

Daniel Hsu

November 1, 2017

Contents

About these notes

These are lecture notes for the "Interactive Learning" seminar course taught in Fall 2017 at Columbia University as COMS 6998-4. They are being written and revised throughout the course. Borrowing a line from Sasha Rakhlin:

> These lecture notes are constantly evolving, so if your version says $x < y$ today, it might say $x > y$ tomorrow.

Chapter 1

Experts and bandits

1.1 Expert advice

1.1.1 The setting

Consider a scenario where an agent (the "learner") must repeatedly make a decision on some important matter, and each decision has some quantifiable consequence that ought to be optimized.

Formally, suppose there are several rounds of this decision-making game. In each round, the learner must choose between two actions, $\{\pm 1\}$. The opponent (the "environment" or "Nature") picks one of the actions for that round to be correct, and the other action is a mistake. The goal is to minimize the number of mistakes made over the rounds of the game.

So far, this setting is impossibly abstract. Whatever actions the learner chooses, they could all be correct, or they could all be mistakes, ...

Let's make things more tractable. Suppose that before choosing the action in each round, the learner receives a recommended action from each of N experts. Why should this help? The *hope* is that some (maybe just one!) of the experts make good recommendations over the rounds of the game—we will formalize this shortly. If this is the case, the learner can then try to just do (almost) as well as the best of these experts. The protocol is given in Algorithm 1.

Algorithm 1 Protocol for online decision-making using expert advice (binary actions)

1: **for** $t = 1, 2, \ldots$ **do**
2: Receive experts' actions: $b_{t,i} \in \{\pm 1\}$ for $i \in [N]$.
3: Learner chooses action: $a_t \in \{\pm 1\}$.
4: Learner receives correct action: $y_t \in \{\pm 1\}$.
5: **end for**

We use the notation $[n] := \{1, \ldots, n\}$ to denote the first n positive integers.

1.1.2 If one expert is perfect . . .

Suppose we are promised that one of the experts will *always* recommend the correct action in all rounds. We would like to design an algorithm for the learner that minimizes the number of mistakes.

The following algorithm keeps track of which experts are perfect-so-far, and chooses a_t to agree with any such expert. This is called the *Consistent Expert* algorithm.

Algorithm 2 Consistent Expert algorithm

Let $V_0 := [N]$.
for $t = 1, 2, \ldots$ **do**
  Receive experts' actions: $b_{t,i} \in \{\pm 1\}$ for $i \in [N]$.
  Choose action: pick any $i \in V_{t-1}$, and set $a_t := b_{t,i}$.
  Receive correct action: $y_t \in \{\pm 1\}$.
  Update: $V_t := \{i \in V_{t-1} : b_{t,i} = y_t\}$.
end for

Observe that by round t, each "surviving" expert in V_{t-1} has predicted the correct action *perfectly* in the first $t - 1$ rounds.

Theorem 1.1 (Consistent Expert)**.** *If one of the N experts makes no mistakes, then a learner using Algorithm 2 (Consistent Expert) makes at most $N - 1$ mistakes.*

Can we do better? Here is one way: choose a_t to agree with the majority of perfect-so-far experts. This is called the *Halving* algorithm.

Algorithm 3 Halving algorithm

Let $V_0 := [N]$.
for $t = 1, 2, \ldots$ **do**
  Receive experts' actions: $b_{t,i} \in \{\pm 1\}$ for $i \in [N]$.
  Choose action: $a_t := \text{sign}\left(\sum_{i \in V_{t-1}} b_{t,i}\right)$.
  Receive correct action: $y_t \in \{\pm 1\}$.
  Update: $V_t := \{i \in V_{t-1} : b_{t,i} = y_t\}$.
end for

In Algorithm 3, we use the notation $\text{sign}(z) := +1$ if $z > 0$, and $\text{sign}(z) := -1$ if $z \leq 0$.

Whenever the learner makes a mistake, the number of perfect-so-far experts decreases by at least a factor of two. This can happen at most $\log_2(N)$ times before the number of perfect-so-far experts is one. Thus we have proven the following.

Theorem 1.2 (Halving)**.** *If one of the N experts makes no mistakes, then a learner using Algorithm 3 (Halving) makes at most $\log_2(N)$ mistakes.*

This is exponentially better than the Consistent Expert algorithm!

1.1.3 If one expert is nearly perfect . . .

We now generalize to the case where we only assume that there is an expert that makes at most K mistakes, for some non-negative integer K. Neither of the previous algorithms make sense anymore, because it is possible that there are no perfect-so-far experts even after the first round. So we shouldn't throw out experts that make just a single mistake.

A simple idea is to use a weighted majority over all the experts (instead of a simple majority over the perfect-so-far experts). Instead of throwing out an expert who makes a mistake, we simply cut its weight in half. Therefore, the weight of an expert is small if it makes many mistakes. This is the *Weighted Majority* algorithm.

Algorithm 4 Weighted Majority algorithm

Let $w_{1,i} := 1$ for $i \in [N]$.
for $t = 1, 2, \ldots$ **do**
  Receive experts' actions: $b_{t,i} \in \{\pm 1\}$ for $i \in [N]$.
  Choose action: $a_t := \text{sign}\left(\sum_{i \in [N]} w_{t,i} \cdot b_{t,i}\right)$.
  Receive correct action: $y_t \in \{\pm 1\}$.
  Update: for each $i \in [N]$, $w_{t+1,i} := w_{t,i}/2$ if $b_{t,i} \neq y_t$, and $w_{t+1,i} := w_{t,i}$ otherwise.
end for

Theorem 1.3 (Weighted Majority)**.** *If one of the N experts makes at most K mistakes, then a learner Algorithm 4 (Weighted Majority) makes at most $(K + \log_2(N))/\log_2(4/3)$ mistakes.*

Proof. Let $Z_t := \sum_{i=1}^{N} w_{t,i}$ denote the *total weight* of all experts at the start of round t (so, e.g., $Z_1 = N$). Suppose the learner makes a mistake on round t. Let $\alpha_t \geq 1/2$ denote the fraction of the total weight corresponding to experts i that choose $b_{t,i} = a_t$. Half of that weight will be cut by the learner by the end of round t. So the remaining weight Z_{t+1} in such a round satisfies

$$Z_{t+1} = (1 - \alpha_t)Z_t + \frac{1}{2} \cdot \alpha_t Z_t \leq \frac{3}{4} Z_t.$$

So if a total of M mistakes have been made through round t, then

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

On the other hand, if there is an expert i who makes at most K mistakes after t rounds, then its weight satisfies $w_{t+1,i} \geq 2^{-K}$. The total weight after t rounds must be at least the weight of this expert:

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

Therefore

$$2^{-K} \leq \left(\frac{3}{4}\right)^M N.$$

Taking logarithm of both sides and rearranging gives

$$M \leq \frac{K + \log_2(N)}{\log_2(4/3)},$$

which finishes the proof. □

1.1.4 Regret

As the number of rounds T grows to infinity, it may be unreasonable to assume that there is an expert that makes at most a constant number of mistakes (independent of T). Perhaps a more realistic assumption is that there is an expert with a low rate of making mistakes, in the sense that the number of mistakes after T rounds is at most ρT, for some fraction ρ.

Under this relaxed assumption, the Weighted Majority algorithm guarantees the learner makes at most

$$\frac{\rho T + \log_2(N)}{\log_2(4/3)} \leq 2.41(\rho T + \log_2(N))$$

mistakes after T rounds. Note that this bound is not interesting if $\rho \geq \log_2(4/3)/2 \approx 0.21$, because in that case, the bound is larger than $T/2$ (which is what one gets, in expectation, by guessing uniformly at random . . .).

We would instead like an algorithm that performs almost as well as the best expert, with a mistake bound of the form

$$M_T - \min_{i \in [N]} M_{T,i} \leq o(T),$$

where M_T is the number of mistakes of the learner after T rounds, $M_{T,i}$ is the number of mistakes of expert i after T rounds. We call the left-hand side quantity the *regret* of the learner to the best expert after T rounds.

If such a sublinear regret bound is possible, and the best expert makes at most ρT mistakes after T rounds, then the learner makes at most $(\rho + o(1))T$ mistakes after T rounds. This would mean that the learner's performance approaches that of the best expert as T increases to infinity.

Unfortunately, a sublinear regret bound is impossible in general.

Theorem 1.4 (Impossibility of sublinear regret)**.** *Let $b_{t,1} = -1$ and $b_{t,2} = +1$ for all rounds t. For any learning algorithm, there is a sequence $y_1, \ldots, y_T \in \{\pm 1\}$ such that the regret of the learner is at least $T/2$.*

Proof. For any $y_1, \ldots, y_T \in \{\pm 1\}$, the better of the two experts makes at most $T/2$ mistakes. Therefore there is some sequence for which the learner makes T mistakes while the best expert makes only $T/2$ mistakes. □

Exercise 1.1. Show how to modify Algorithm 4 so that its mistake bound is

$$2K + O\left(\sqrt{K\log(N)} + \log(N)\right)$$

whenever there is an expert that makes at most K mistakes.

1.1.5 Randomization

It turns out we can get around this lower bound by allowing *randomization* and only considering the number of mistakes made by the learner *in expectation*. Here, we are limiting the power of the opponent and assuming that their decisions are made before learner's random coins are tossed. In fact, we shall simply think of the sequence of correct and incorrect actions as being set once and for all before the start of the game.

We modify the Weighted Majority algorithm to use randomness in a natural way. This is the *Randomized Weighted Majority* algorithm.

Algorithm 5 Randomized Weighted Majority algorithm

Parameter: $\eta > 0$.
$w_{1,i} := 1$ for $i \in [N]$.
for $t = 1, 2, \ldots$ **do**
  Receive experts' actions: $b_{t,i} \in \{\pm 1\}$ for $i \in [N]$.
  Choose action: randomly draw $i_t \sim (w_{t,1}, \ldots, w_{t,N})/Z_t$, and set $a_t := b_{t,i_t}$.
  Receive correct action: $y_t \in \{\pm 1\}$.
  Update: for each $i \in [N]$, $w_{t+1,i} := e^{-\eta} w_{t,i}$ if $b_{t,i} \neq y_t$, and $w_{t+1,i} := w_{t,i}$ otherwise.
end for

There are two differences in Randomized Weighted Majority relative to Weighted Majority. First, we have replaced the value of $1/2$ in the update step with an arbitrary fraction $e^{-\eta}$. This is simply because there is nothing special about the factor of $1/2$, so we may as well pick the η that gives the best guarantee—think of it as a "tuning parameter".[1]

Second, we now choose the action by randomly picking an expert with probability proportional to the weight of the expert, and then following the action of the chosen expert; we then only count the expected number of mistakes. The randomization and averaging are crucial: without them, the learner would be subject to the previous impossibility result. Essentially, they provide a way to transform a difficult discrete problem into an easier continuous problem. Let $p_t := (w_{t,1}, \ldots, w_{t,N})/Z_t$ denote the normalized vector of weights on experts in round t (where $Z_t := w_{t,1} + \cdots + w_{t,N}$). Then the probability that Randomized

[1]Tuning parameters can usually be set to optimize some criterion (such as a regret bound), but often these settings are overly conservative. Therefore, it is desirable to have algorithms that do not have tuning parameters.

Weighted Majority makes a mistake—i.e., the fraction of weight on mistaken experts—is a *linear* function of w_t:

$$\begin{aligned}\Pr(b_{t,i_t} \neq y_t) &= \sum_{i=1}^{N} \Pr(i_t = i) \cdot 1_{\{b_{t,i} \neq y_t\}} \\ &= \sum_{i=1}^{N} \frac{w_{t,i}}{Z_t} \cdot 1_{\{b_{t,i} \neq y_t\}} \\ &= \langle \ell_t, p_t \rangle,\end{aligned}$$

where $\ell_t := (1_{\{b_{t,1} \neq y_t\}}, \ldots, 1_{\{b_{t,N} \neq y_t\}})$ is the vector that indicates whether each expert makes a mistake in round t.[2] The expected number of mistakes over all T rounds is

$$\mathbb{E}(M_T) = \sum_{t=1}^{T} \Pr(a_t \neq y_t) = \sum_{t=1}^{T} \langle \ell_t, p_t \rangle.$$

It turns out with a suitable setting of the parameter η, the expected regret of Randomized Weighted Majority is sublinear in the number of rounds.

1.2 Linear loss game on the simplex

1.2.1 The setting

The Randomized Weighted Majority algorithm turns out to be a special case of an algorithm for a more general problem. In this problem, again there are N experts. In each round, the learner chooses a probability distribution $p_t = (p_{t,1}, \ldots, p_{t,N})$ over the N experts; after, each expert incurs a non-negative loss $\ell_{t,i}$, and the learner incurs the weighted loss $\sum_{i=1}^{n} p_{t,i}\ell_{t,i} = \langle \ell_t, p_t \rangle$ where $\ell_t := (\ell_{t,1}, \ldots, \ell_{t,N})$. We call this the linear loss game on the (probability) simplex. The protocol is given in Algorithm 6.

Algorithm 6 Protocol for the linear loss game on the simplex

1: **for** $t = 1, 2, \ldots$ **do**
2: Learner chooses probability vector: $p_t = (p_{t,1}, \ldots, p_{t,N}) \in \Delta^{N-1}$.
3: Learner receives loss vector: $\ell_t = (\ell_{t,1}, \ldots, \ell_{t,N}) \in \mathbb{R}_+^N$.
4: Learner incurs loss: $\langle \ell_t, p_t \rangle$.
5: **end for**

The following notations are used above:

$$\begin{aligned}\mathbb{R}_+^N &:= \left\{(p_1, \ldots, p_N) \in \mathbb{R}^N : p_i \geq 0 \, \forall i \in [N]\right\}, \\ \Delta^{N-1} &:= \left\{(p_1, \ldots, p_N) \in \mathbb{R}_+^N : p_1 + \cdots + p_N = 1\right\}.\end{aligned}$$

[2] We use the notation $\langle u, v \rangle := \sum_{i=1}^{N} u_i v_i$ for the standard inner product between vectors $u, v \in \mathbb{R}^N$; and we use $1_{\{P\}}$ for the indicator of a predicate P, which takes value 1 if P is true, and value 0 if P is false.

The total loss of expert i after T rounds is

$$L_{T,i} := \sum_{t=1}^{T} \ell_{t,i},$$

and the total loss of the learner after T rounds is

$$L_T := \sum_{t=1}^{T} \langle \ell_t, p_t \rangle.$$

The goal of the learner is to minimize the regret after T rounds:

$$L_T - \min_{i \in [N]} L_{T,i}.$$

Exercise 1.2. Let $L_{T,q} := \sum_{i=1}^{T} q_i L_{T,i}$ be the total loss for a fixed probability distribution $q \in \Delta^{N-1}$. Show that

$$\min_{i \in [N]} L_{T,i} = \min_{q \in \Delta^{N-1}} L_{T,q}.$$

1.2.2 Hedge

The algorithm that generalizes the Randomized Weighted Majority algorithm is called the *Hedge* algorithm.

Algorithm 7 Hedge algorithm

Require: $\eta > 0$.

1: Let $w_1 = (w_{1,1}, \ldots, w_{1,N}) := (1, \ldots, 1)$.
2: **for** $t = 1, 2, \ldots$ **do**
3: Choose probability vector: $p_t = w_t / Z_t \in \Delta^{N-1}$, where $Z_t := \sum_{i=1}^{N} w_{t,i}$.
4: Receive loss vector: $\ell_t = (\ell_{t,1}, \ldots, \ell_{t,N}) \in \mathbb{R}_+^N$.
5: Update: $w_{t+1,i} := w_{t,i} \exp(-\eta \ell_{t,i})$ for all $i \in [N]$.
6: **end for**

Theorem 1.5 (Hedge). *For any sequence of loss vectors $\ell_1, \ldots, \ell_T \in \mathbb{R}_+^N$, the total loss L_T incurred by a learner using Algorithm 7 (Hedge) with parameter $\eta > 0$ after T rounds satisfies*

$$L_T - L_{T,i} = \sum_{t=1}^{T} \langle \ell_t, p_t \rangle - \sum_{t=1}^{T} \ell_{t,i} \leq \frac{\ln(N)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \langle \ell_t^2, p_t \rangle, \quad i \in [N]$$

where $\ell_t^2 := (\ell_{t,1}^2, \ldots, \ell_{t,N}^2)$.

The proof is very similar to the analysis of the Weighted Majority algorithm. The main idea is to bound the total weight Z_{T+1} after T rounds from above and below. The total weight shrinks from round to round as the different experts incur losses, and this amount can be related to the loss incurred by the learner. On the other hand, the total weight Z_{T+1} can also be related to the total loss of any single expert after all T rounds.

Proof. Below, we use the following approximations of the exponential:

$$\exp(z) \le 1 + z + z^2/2, \quad z \le 0; \tag{1.1}$$

$$\exp(z) \ge 1 + z, \quad z \in \mathbb{R}. \tag{1.2}$$

We consider the relative change in total weight after round t:

$$\begin{aligned}
\frac{Z_{t+1}}{Z_t} = \sum_{i=1}^{N} \frac{w_{t+1,i}}{Z_t} &= \sum_{i=1}^{N} p_{t,i} \exp(-\eta \ell_{t,i}) \\
&\le \sum_{i=1}^{N} p_{t,i} \left(1 - \eta \ell_{t,i} + \frac{\eta^2}{2} \ell_{t,i}^2\right) \\
&= 1 - \eta \langle \ell_t, p_t \rangle + \frac{\eta^2}{2} \langle \ell_t^2, p_t \rangle \\
&\le \exp\left(-\eta \langle \ell_t, p_t \rangle + \frac{\eta^2}{2} \langle \ell_t^2, p_t \rangle\right).
\end{aligned}$$

Above, the first inequality uses Equation (1.1), and the second inequality uses Equation (1.2). Therefore,

$$\ln\left(\frac{Z_{t+1}}{Z_t}\right) \le -\eta \langle \ell_t, p_t \rangle + \frac{\eta^2}{2} \langle \ell_t^2, p_t \rangle. \tag{1.3}$$

Summing up Equation (1.3) over all $t = 1, \ldots, T$,

$$\begin{aligned}
\ln(Z_{T+1}) - \ln(Z_1) &= \ln(Z_{T+1}) - \ln(N) \\
&\le -\eta \sum_{t=1}^{T} \langle \ell_t, p_t \rangle + \frac{\eta^2}{2} \sum_{t=1}^{T} \langle \ell_t^2, p_t \rangle.
\end{aligned}$$

On the other hand, for any $i \in [N]$,

$$\ln(Z_{T+1}) \ge \ln(w_{T+1,i}) = -\eta \sum_{t=1}^{T} \ell_{t,i}.$$

Therefore, for any $i \in [N]$,

$$\eta \sum_{t=1}^{T} \langle \ell_t, p_t \rangle \le \eta \sum_{t=1}^{T} \ell_{t,i} + \ln(N) + \frac{\eta^2}{2} \sum_{t=1}^{T} \langle \ell_t^2, p_t \rangle,$$

which concludes the proof. □

Exercise 1.3. Show that if $\ell_{t,i} \in [0,1]$ for each i and t, then there is some setting of η (in terms of N and T) that guarantees the regret of the learner using Algorithm 7 after T rounds satisfies

$$L_T - \min_{i\in[N]} L_{T,i} \leq \sqrt{2T\ln(N)}.$$

Exercise 1.4. Deduce a bound on the expected regret for a learner using Algorithm 5 (Randomized Weighted Majority) after T rounds for the setting from Section 1.1.1.

Exercise 1.5. Suppose it is promised that $\min_{i\in[N]} L_{T,i} \leq L^*$. Show that if $\ell_{t,i} \in [0,1]$ for each i and t, then there is some setting of η (in terms of N and L^*) that guarantees the regret of the learner using Algorithm 7 after T rounds satisfies

$$L_T - \min_{i\in[N]} L_{T,i} \leq O\left(\sqrt{L^*\log(N)} + \log(N)\right).$$

1.2.3 Entropy maximization

The probability vector chosen by Hedge in round t turns out to be the solution to the following optimization problem:

$$\min_{p\in\Delta^{N-1}} \sum_{s=1}^{t-1}\langle \ell_s, p\rangle - \frac{1}{\eta}\sum_{i=1}^{N} p_i \ln\frac{1}{p_i}.$$

The objective balances between minimizing the total loss in previous rounds using p and maximizing the *entropy* of p. This motivates the following alternative proof of Theorem 1.5.

Alternative proof of Theorem 1.5. The proof uses the fact that the excess total loss $L_T - L_{T,q}$ compared to any fixed $q \in \Delta^{N-1}$ can be related to the decreases in *relative entropy* $\mathrm{RE}(q,p_t) - \mathrm{RE}(q,p_{t+1})$ to q from round to round, where

$$\mathrm{RE}(q,p) := \sum_{i=1}^{N} q_i \ln\frac{q_i}{p_i}.$$

Indeed,

$$\begin{aligned}
\mathrm{RE}(q,p_{t+1}) - \mathrm{RE}(q,p_t) &= \sum_{i=1}^{n} q_i \ln\frac{p_{t,i}}{p_{t+1,i}} \\
&= \eta\sum_{i=1}^{N} q_i\ell_{t,i} + \ln\left(\sum_{i=1}^{N} p_{t,i}\exp(-\eta\ell_{t,i})\right) \\
&\leq \eta\left(\langle \ell_t, q\rangle - \langle \ell_t, p_t\rangle\right) + \frac{\eta^2}{2}\langle \ell_t^2, p_t\rangle.
\end{aligned}$$

This shows that in each round, if the learner incurs a significantly greater loss than the fixed distribution q, then the update of the learner's distribution moves it "closer" to q in

the sense of relative entropy. Now, summing this up from for $t = 1, \ldots, T$ gives

$$\mathrm{RE}(q, p_{T+1}) - \mathrm{RE}(q, p_1) \leq \eta \left(L_{T,q} - L_T \right) + \frac{\eta^2}{2} \sum_{t=1}^{T} \langle \ell_t^2, p_t \rangle.$$

The claim follows because $\mathrm{RE}(q, p_1) \leq \ln(N)$ and $\mathrm{RE}(q, p_{T+1}) \geq 0$. □

1.3 Prediction with partial feedback

1.3.1 The setting

We return to the original online decision-making problem. The setting from Section 1.1.1 considers just two actions, but it is natural to generalize to the case where there are $K \geq 2$ possible actions $\{1, \ldots, K\}$. However, we also need to consider the nature of the feedback. Is the learner told the "correct" action? What if there are multiple correct actions? What if each action has a different "cost" associated with it in each round? The possibilities are seemingly endless.

We focus on a specific but canonical case here: every action $a \in [K]$ incurs some cost $c_t(a) \geq 0$ in round t, but the learner only observes the value of the loss for the chosen action a_t. For simplicity, we begin with the version of the problem where there are no experts offering their advice. This is called the "multi-armed bandits" problem or online decision-making with "bandit feedback" (for historical reasons). The protocol is given in Algorithm 8.

Algorithm 8 Protocol for online decision-making with bandit feedback

1: **for** $t = 1, 2, \ldots$ **do**
2: Learner chooses action: $a_t \in [K]$.
3: Learner receives cost for chosen action: $c_t(a_t) \geq 0$.
4: **end for**

The total loss of picking a single action $a \in [K]$ in all T rounds is $L_{T,a} := \sum_{t=1}^{T} c_t(a)$, and the total loss of the learner after T rounds is $L_T := \sum_{t=1}^{T} c_t(a_t)$. The regret of the learner after T rounds is

$$L_T - \min_{a \in [K]} L_{T,a}.$$

We would like an algorithm that guarantees low regret in expectation.

1.3.2 Inverse probability weighting

A simple approach to the multi-armed bandits problem is to try to reduce it back to the "full-information" setting of the linear loss game on the simplex from Section 1.2.1, where the entire loss vector is revealed after each round.

We first consider a simpler problem. Suppose we only want to know the average cost of a *particular* action, say, action 1, over T rounds. How can we do this? Easy: pick $a_t = 1$ in each round; we are then guaranteed to observe $c_t(1)$ in every round.

Now suppose we only want to know the average costs of two particular actions, say, actions 1 and 2. You can't just pick $a_t = 1$, since then we won't see $c_t(2)$. However, if we let a fair coin toss determine whether we pick either action 1 or action 2, then we have a (50%, 50%) chance of seeing $c_t(1)$ and $c_t(2)$. The observed cost $c_t(a_t)$ is, in expectation,

$$\mathbb{E}(c_t(a_t)) = \frac{c_t(1) + c_t(2)}{2}.$$

Over T rounds, we have

$$\mathbb{E}\left(\frac{1}{T}\sum_{t=1}^{T} c_t(a_t)\right) = \frac{1}{T}\sum_{t=1}^{T} \frac{c_t(1) + c_t(2)}{2},$$

and by the law of large numbers, $\sum_{t=1}^{T} c_t(a_t)/T$ will be close to this expected value with high probability as T becomes large. This is interesting—it at least has some information about both actions—but it is not quite what we wanted.[3]

Here is a simple trick for isolating the individual costs. Consider the following estimators of $c_t(1)$ and $c_t(2)$:

$$\hat{c}_t(1) := \frac{1_{\{a_t=1\}}}{1/2} c_t(a_t);$$
$$\hat{c}_t(2) := \frac{1_{\{a_t=2\}}}{1/2} c_t(a_t).$$

Observe that one of these estimators is guaranteed to be zero (since we cannot have $a_t = 1$ and $a_t = 2$ both be true). If $a_t = 1$, then $\hat{c}_t(1)$ is *twice* the observed cost while $\hat{c}_t(2) = 0$ (and vice versa if $a_t = 2$). However, in expectation, *both* estimates are equal to the costs of the respective actions:

$$\mathbb{E}(\hat{c}_t(1)) = \Pr(a_t = 1) \cdot \frac{1_{\{1=1\}}}{1/2} c_t(1) + \Pr(a_t = 2) \cdot \frac{1_{\{2=1\}}}{1/2} c_t(2) = c_t(1);$$
$$\mathbb{E}(\hat{c}_t(2)) = \Pr(a_t = 1) \cdot \frac{1_{\{1=2\}}}{1/2} c_t(1) + \Pr(a_t = 2) \cdot \frac{1_{\{2=2\}}}{1/2} c_t(2) = c_t(2).$$

[3]One option that sounds plausible is to use action 1 in the first $T/2$ rounds, and action 2 in the last $T/2$ rounds. But this is reasonable only if

$$\frac{2}{T}\sum_{t=1}^{T/2} c_t(1) = \frac{2}{T}\sum_{t=T/2+1}^{T} c_t(1),$$

and an analogous statement holds for action 2. Such assumptions can be avoided using randomness.

Again, over T rounds, we have

$$\mathbb{E}\left(\frac{1}{T}\sum_{t=1}^{T}\hat{c}_t(1)\right) = \frac{1}{T}\sum_{t=1}^{T}c_t(1)$$

and by the law of large numbers, $\sum_{t=1}^{T}\hat{c}_t(1)/T$ will be close to this expected value with high probability as T becomes large; an analogous statement holds for action 2.

More generally, suppose we fix a probability distribution $p_t := (p_t(1), \ldots, p_t(K)) \in \Delta^{K-1}$ over the K actions, and then randomly draw $a_t \sim p_t$. For each particular action $a \in [K]$, define the estimator

$$\hat{c}_t(a) := \frac{1_{\{a_t=a\}}}{p_t(a)}c_t(a_t). \tag{1.4}$$

Again, in expectation, for *all* actions $a \in [K]$,

$$\begin{aligned}\mathbb{E}(\hat{c}_t(a)) &= \sum_{a'=1}^{K}\Pr(a_t = a')\cdot\frac{1_{\{a'=a\}}}{p_t(a)}c_t(a') \\ &= \Pr(a_t = a)\cdot\frac{1_{\{a=a\}}}{p_t(a)}c_t(a) \\ &= c_t(a).\end{aligned}$$

Therefore, the vector of cost estimates $\hat{c}_t$ is an *unbiased estimator* of the vector of (true) costs c_t. This technique is called *inverse probability weighting*.

We see that randomization provides a simple means to enable *counterfactual inference*: for any $a \in [K]$, we are able to estimate (in an unbiased manner) the loss that we would have incurred if we had chosen action a.

Although unbiasedness is a useful property of an estimator, it is not the only property that is relevant in applications. In particular, besides the mean of $\hat{c}_t(a)$, we may also care about, say, higher-order moments $\hat{c}_t(a)$, or other properties of its distribution. For instance, the variance of $\hat{c}_t(a)$ is

$$\begin{aligned}\mathrm{var}(\hat{c}_t(a)) &= \mathbb{E}(\hat{c}_t(a)^2) - \mathbb{E}(\hat{c}_t(a))^2 \\ &= \sum_{a'=1}^{K}\Pr(a_t = a')\cdot\frac{1^2_{\{a'=a\}}}{p_t(a)^2}c_t(a')^2 - c_t(a)^2 \\ &= \Pr(a_t = a)\cdot\frac{1_{\{a=a\}}}{p_t(a)^2}c_t(a)^2 - c_t(a)^2 \\ &= \left(\frac{1}{p_t(a)} - 1\right)c_t(a)^2.\end{aligned}$$

This can be large if $p_t(a)$ is small. This means that the estimates $\hat{c}_t$ may only be reliable if the probabilities p_t are not too close to zero.

There is a trade-off involved in the choice of the probability distribution p_t. The most "balanced" choice p_t is simply the uniform distribution: all actions are tried equally often, and the losses in each round are estimated with reasonably small variance ($\text{var}(\hat{c}_t(a)) \leq K-1$ for all $a \in [K]$ if $c_t(a) \in [0,1]$). However, choosing actions in this way may yield poor performance (i.e., high regret); it is instead preferred to put lower probability on actions that seem to incur high loss, and higher probability on actions that seem to incur low loss. This is called the "exploration vs. exploitation" dilemma.

1.3.3 Hedging over actions with bandit feedback

A natural reduction from the bandit feedback setting to the full-information setting is to simply run an algorithm (like Hedge) designed for the full-information setting using the cost estimates $\hat{c}_t$ described above in Equation (1.4) as the loss vectors ℓ_t. This is exactly the approach taken by the *Exp3* algorithm (short for *Exponential-weight algorithm for Exploration and Exploitation*).

Algorithm 9 Exp3 algorithm

Require: $\eta > 0$.

Let $w_1 = (w_{1,1}, \ldots, w_{1,K}) := (1, \ldots, 1)$.
for $t = 1, 2, \ldots$ **do**
  Form vector $p_t := w_t / Z_t \in \Delta^{K-1}$, where $Z_t := \sum_{a=1}^{K} w_{t,a}$.
  Choose action: randomly draw $a_t \sim p_t$.
  Receive cost for chosen action: $c_t(a_t) \geq 0$.
  Form cost estimates: $\hat{c}_t(a) = 1_{\{a_t = a\}} \, c_t(a_t)/p_t(a)$ for all $a \in [K]$.
  Update: $w_{t+1,a} := w_{t,a} \exp(-\eta \hat{c}_t(a))$ for all $a \in [K]$.
end for

Theorem 1.6 (Exp3). *For any sequence of cost vectors $c_1, \ldots, c_T \in \mathbb{R}_+^K$, the total loss $L_T := \sum_{t=1}^{T} c_t(a_t)$ incurred by a learner using Algorithm 9 (Exp3) with parameter $\eta > 0$ after T rounds satisfies*

$$\mathbb{E}(L_T) \leq L_{T,a} + \frac{\ln(K)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{a'=1}^{K} c_t(a')^2, \quad a \in [K].$$

Proof. Let $\mathbb{E}_t$ denote the conditional expectation operator given all information through the first $t-1$ rounds. We use the following properties about the estimated costs from Section 1.3.2:

$$\mathbb{E}(\hat{c}_t(a)) = c_t(a);$$
$$\mathbb{E}_t(\hat{c}_t(a)^2) = \frac{c_t(a)^2}{p_t(a)}.$$

Observe that the evolution of the weights w_t in Exp3 is the same as that of Hedge using loss vectors $\ell_t = \hat{c}_t$. Therefore, we have from Theorem 1.5 that, for any $a \in [K]$,

$$\sum_{t=1}^{T} \langle \hat{c}_t, p_t \rangle \leq \sum_{t=1}^{T} \hat{c}_t(a) + \frac{\ln(K)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \langle \hat{c}_t^2, p_t \rangle. \tag{1.5}$$

Now we take expectations of both sides of Equation (1.5). First, observe that each term on the left-hand side of Equation (1.5) has conditional expectation

$$\mathbb{E}_t(\langle \hat{c}_t, p_t \rangle) = \sum_{a=1}^{K} p_t(a)\, \mathbb{E}_t(\hat{c}_t(a)) = \sum_{a=1}^{K} p_t(a) c_t(a) = \mathbb{E}(c_t(a_t)).$$

This shows that the left-hand side of Equation (1.5) is the expected loss of the learner $\mathbb{E}(L_T)$.

Next, the first summation on the right-hand side of Equation (1.5) has expectation

$$\sum_{t=1}^{T} \mathbb{E}(\hat{c}_t(a)) = \sum_{t=1}^{T} c_t(a) = L_{T,a}.$$

Finally, each term in the second summation on the right-hand side of Equation (1.5) has expectation

$$\begin{aligned}
\mathbb{E}(\langle \hat{c}_t^2, p_t \rangle) &= \sum_{a'=1}^{K} \mathbb{E}(p_t(a') \hat{c}_t(a')^2) \\
&= \sum_{a'=1}^{K} \mathbb{E}(p_t(a')\, \mathbb{E}_t(\hat{c}_t(a')^2)) \\
&= \sum_{a'=1}^{K} \mathbb{E}\left(p_t(a') \frac{c_t(a')^2}{p_t(a')} \right) \\
&= \sum_{a'=1}^{K} c_t(a')^2.
\end{aligned}$$

So, the inequality relating the expectations of each side of Equation (1.5) finally becomes

$$\mathbb{E}(L_T) \leq L_{T,a} + \frac{\ln(K)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{a'=1}^{K} c_t(a')^2,$$

which completes the proof. □

Exercise 1.6. Show that if $c_t(a) \in [0, 1]$ for each a and t, then there is some setting of η (in terms of K and T) that guarantees the expected regret of the learner using Algorithm 9[4] after T rounds satisfies

$$\mathbb{E}(L_T) - \min_{a \in [K]} L_{T,a} \leq \sqrt{2KT \ln(K)}.$$

[4]It turns out the expected regret bound of Exp3 is suboptimal. There is a different algorithm with expected regret bound $O(\sqrt{KT})$, and this is optimal.

1.3.4 Hedging over experts with bandit feedback

Now we bring experts back into the picture. The protocol is given in Algorithm 10.

Algorithm 10 Protocol for online decision-making with bandit feedback and expert advice

1: **for** $t = 1, 2, \ldots$ **do**
2: Learner receives experts' actions: $b_{t,i} \in [K]$ for $i \in [N]$.
3: Learner chooses action: $a_t \in [K]$.
4: Learner receives cost for chosen action: $c_t(a_t) \geq 0$.
5: **end for**

The total loss of expert $i \in [N]$ in all T rounds is $L_{T,i} := \sum_{t=1}^{T} c_t(b_{t,i})$, and the total loss of the learner after T rounds is $L_T := \sum_{t=1}^{T} c_t(a_t)$. Hence, the regret of the learner after T rounds is

$$L_T - \min_{i \in [K]} L_{T,i}.$$

We would like an algorithm that guarantees low regret in expectation. Note that this is a very different notion of regret than what was considered in Section 1.3.1 and Section 1.3.3. This is because we are comparing the performance of the learner to that of the best expert, rather than the best action. This is an important difference because there may not be any single action that has low cost over all T rounds. However, there may be an expert who is good at picking different actions in different rounds. This setting is often called the *contextual bandit* setting, because the experts can be viewed as making use of context information in each round to recommend actions, which is natural in real-world applications.

The main idea of the following algorithm is similar to that of Exp3. We use a full-information algorithm (Hedge) with estimated costs for the experts. In this case, we need to estimate the cost incurred by an expert i in round t: since expert i recommends action $b_{t,i} \in [K]$ in round t, we use is

$$\hat{c}_t(b_{t,i}) = \frac{1_{\{a_t = b_{t,i}\}}}{p_t(a)} c_t(a_t)$$

just as in Equation (1.4). The vector of cost estimates for experts can then be used to define weights over experts, which in turn are used to define a probability distribution over actions. The resulting algorithm is called *Exp4* (short for *Exponential-weight algorithm for Exploration and Exploitation using Expert advice*).

Theorem 1.7 (Exp4)**.** *For any sequence of cost vectors $c_1, \ldots, c_T \in \mathbb{R}_+^K$, the total loss $L_T := \sum_{t=1}^{T} c_t(a_t)$ incurred by a learner using Algorithm 11 (Exp4) with parameter $\eta > 0$ after T rounds satisfies*

$$\mathbb{E}(L_T) \leq L_{T,i} + \frac{\ln(N)}{\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \sum_{a=1}^{K} c_t(a)^2, \quad i \in [N].$$

Algorithm 11 Exp4 algorithm

Require: $\eta > 0$.

Let $w_1 = (w_{1,1}, \ldots, w_{1,N}) := (1, \ldots, 1)$.
for $t = 1, 2, \ldots$ **do**
  Receive experts' actions: $b_{t,i} \in [K]$ for $i \in [N]$.
  Form vector $p_t \in \Delta^{K-1}$ given by $p_t(a) := \sum_{i=1}^{N} w_{t,i} 1_{\{b_{t,i}=a\}} / Z_t$ and $Z_t := \sum_{i=1}^{N} w_{t,i}$.
  Choose action: randomly draw $a_t \sim p_t$.
  Receive cost for chosen action: $c_t(a_t) \geq 0$.
  Form cost estimates: $\hat{c}_t(a) = 1_{\{a_t=a\}} c_t(a_t)/p_t(a)$ for all $a \in [K]$.
  Update: $w_{t+1,i} := w_{t,i} \exp(-\eta \hat{c}_t(b_{t,i}))$ for all $i \in [N]$.
end for

The proof of Theorem 1.7 is very similar to that of Theorem 1.6.

Exercise 1.7. Prove Theorem 1.7.

Exercise 1.8. Show that if $c_t(a) \in [0, 1]$ for each a and t, then there is some setting of η (in terms of K, N, and T) that guarantees the expected regret of the learner Algorithm 11 satisfies

$$\mathbb{E}(L_T) - \min_{i \in [N]} L_{T,i} \leq \sqrt{2KT \ln(N)}.$$

1.4 Bibliographic references

The Weighted Majority and Randomized Weighted Majority algorithms are due to Littlestone and Warmuth (1994). The impossibility of sublinear regret is due to Cover (1965). The Hedge algorithm is due to Freund and Schapire (1997), and the analysis based on relative entropy is from Freund and Schapire (1999). The importance weighting trick is attributed to Horvitz and Thompson (1952). The Exp3 and Exp4 algorithms are due to Auer *et al.* (2002).

Chapter 2

Concept learning

2.1 Online classification

2.1.1 Learning concepts from examples

A basic problem in learning is that of classification (or concept learning). Abstractly, a *concept* c is a subset of the objects from some domain $\mathcal{X}$. For example, if $\mathcal{X}$ is all present-day animals on Earth, then a possible concept c is the set of mammals currently living in Africa, $\{\text{giraffe}, \text{lion}, \dots\}$. Or, if $\mathcal{X}$ is all 28×28 arrays of binary pixels, then a possible concept c is the arrays that depict the Arabic numeral "4". We shall overload notation and also let $c : \mathcal{X} \to \{0, 1\}$ denote the *characteristic function* for the concept it represents: $c(x) = 1$ if and only if $x \in c$.

Suppose a learner would like to learn some *target concept* c. How could they go about this task? One way to do this is to have a teacher provide some positive and negative examples of the concept. More precisely, the teacher provides pairs of the form $(x, c(x))$ where $x \in \mathcal{X}$. How can the teacher assess if the learner has actually learned (something non-trivial about) the target concept? One way to is to test the learner's ability to predict the label (i.e., $c(x)$) of the objects $x \in \mathcal{X}$ provided to the learner (before providing the correct answer $c(x)$, of course). This resembles the online decision-making setting from Section 1.1.1. Another way is to have the learner commit to a hypothesis $h : \mathcal{X} \to \{0, 1\}$ after having seen some number of labeled examples, upon which h is evaluated by the teacher (or some other party) in terms of its distance to c (under some suitable metric).

2.1.2 Basic setting

We start with the basic online classification setting, in which the teacher repeatedly quizzes the learner about a target concept, and the goal of the learner is to make as few mistakes as possible. The protocol is given in Algorithm 12.

Suppose the learner has in mind a *hypothesis class* H, i.e., a set of functions $h : \mathcal{X} \to \{0, 1\}$. The learner uses H as candidates for the unknown target concept c. For simplicity, we only

Algorithm 12 Protocol for online classification

for $n = 1, 2, \ldots$ **do**
  Teacher reveals object: $x_n \in \mathcal{X}$.
  Learner makes prediction: $a_n \in \{0, 1\}$.
  Teacher reveals correct answer (label): $y_n \in \{0, 1\}$.
end for

consider the *realizable setting*, where we assume that $c \in H$. Two natural algorithms to use in this case are the analogues of the Consistent Expert and Halving algorithms.

Algorithm 13 Consistent Hypothesis algorithm for online classification

Let $V_0 := H$.
for $n = 1, 2, \ldots$ **do**
  Receive object: $x_n \in \mathcal{X}$.
  Make prediction: pick any $h \in V_{n-1}$, and set $a_n := h(x_n)$.
  Receive correct answer (label): $y_n \in \{0, 1\}$.
  Update: $V_n := \{h \in V_{n-1} : h(x_n) = y_n\}$.
end for

Algorithm 14 Halving algorithm for online classification

Let $V_0 := H$.
for $n = 1, 2, \ldots$ **do**
  Receive object: $x_n \in \mathcal{X}$.
  Make prediction: $a_n := \arg\max_{y \in \{0,1\}} |\{h \in V_{n-1} : h(x) = y\}|$.
  Receive correct answer (label): $y_n \in \{0, 1\}$.
  Update: $V_n := \{h \in V_{n-1} : h(x_n) = y_n\}$.
end for

Theorem 2.1 (Consistent Hypothesis for online classification)**.** *In the realizable setting, the learner using Algorithm 13 (Consistent Hypothesis) makes at most* $|H| - 1$ *mistakes.*

Observe that the Consistent Hypothesis algorithm need not explicitly maintain the set V_{n-1}, as long as there is a procedure for finding some $h \in H$ that is consistent with a sequence of labeled examples.

Theorem 2.2 (Halving for online classification)**.** *In the realizable setting, the learner using Algorithm 14 (Halving) makes at most* $\log_2 |H|$ *mistakes.*

Is this the best we can do? The answer depends on the hypothesis class H. Specifically, the optimal mistake bound depends on a complexity measure called the *Littlestone dimension* of H.

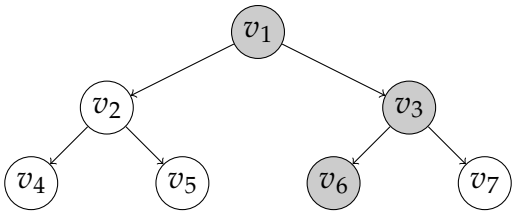

Figure 2.1: A perfect binary tree labeled by $v_1, \ldots, v_7$, shading the path corresponding to $(y_1, y_2) = (1, 0)$.

The definition of the Littlestone dimension uses the concept of *shattering* a perfect binary tree labeled by points in $\mathcal{X}$. Recall that in a perfect binary tree, every non-leaf vertex has exactly two children, and every leaf is at the same depth. We label the vertices of such a depth d tree with points $v_1, \ldots, v_{2^d-1}$ from $\mathcal{X}$ in breadth-first order (see Figure 2.1 for an example). Every $(y_1, \ldots, y_{d-1}) \in \{0,1\}^{d-1}$ specifies a root-to-leaf path $(i_1, \ldots, i_d) \in [2^d - 1]^d$, where $i_1 = 1$ and, for $n = 1, \ldots, d-1$,

$$i_{n+1} = \begin{cases} 2i_n & \text{if } y_n = 0 \text{ (left child of } i_n), \\ 2i_n + 1 & \text{if } y_n = 1 \text{ (right child of } i_n). \end{cases}$$

We say H *shatters* such a perfect binary tree labeled by $v_1, \ldots, v_{2^d-1}$ if, for every $(y_1, \ldots, y_d) \in \{0,1\}^d$, there exists $h \in H$ such that $h(v_{i_n}) = y_n$ for all $n \in [d]$, where $(i_1, \ldots, i_d)$ is the root-to-leaf path specified by $(y_1, \ldots, y_{d-1})$.

For example, if the tree depicted in Figure 2.1 is shattered by H, then for the $\{0,1\}$ sequence $(y_1, y_2, y_3) = (1, 0, 0)$, then is a hypothesis $h \in H$ such that $h(v_1) = 1$, $h(v_3) = 0$, and $h(v_6) = 0$.

The *Littlestone dimension* of H, denoted $\mathrm{Ldim}(H)$, is defined to be the largest integer d such that there is a labeled perfect binary tree of depth d shattered by H.

One can show that $\mathrm{Ldim}(H) \leq \log_2 |H|$, but sometimes $\mathrm{Ldim}(H)$ is much smaller. For example, consider the hypothesis class $H := \{h_v : v \in \mathcal{X}\}$, where $h_v(x) := 1_{\{x=v\}}$. Then $\mathrm{Ldim}(H) = 1$, which may be arbitrarily smaller than $\log_2 |H| = \log_2 |\mathcal{X}|$.

Theorem 2.3 (Lower bound for online classification)**.** *For every $d \leq \mathrm{Ldim}(H)$ and every learning algorithm $\mathcal{A}$, there exists $h \in H$ and a sequence $x_1, \ldots, x_d \in \mathcal{X}$ such that a learner using $\mathcal{A}$ predicts $a_n \neq h(x_n)$ for all $n \in [d]$.*

Proof. Let $d := \mathrm{Ldim}(H)$, and fix a depth d perfect binary tree labeled by $v_1, \ldots, v_{2^d-1} \in \mathcal{X}$ shattered by H. We run algorithm $\mathcal{A}$ starting with $x_1 = v_1$ and $i_1 = 1$. In round n, we set the correct label y_n to be the opposite of the learner's prediction a_n, then choose x_{n+1} to be the label $v_{i_{n+1}}$ where i_{n+1} is the left child (respectively, right child) of i_n in the tree if $y_n = 0$ (respectively, if $y_n = 1$). We are guaranteed that there exists $h \in H$ such that $h(x_n) = y_n$ for all $n \in [d]$. □

A variant of the Halving algorithm, called the *Standard Optimal Algorithm*, makes no more than $\mathrm{Ldim}(H)$ mistakes in the realizable setting. Like Halving, the algorithm tracks the set of hypotheses V_{n-1} consistent with all examples seen before round n. However, instead of predicting with the majority of hypotheses in V_{n-1}, it predicts with the hypotheses that comprise a set of higher Ldim. This way, if a mistake is made in round n, $\mathrm{Ldim}(V_n) \leq \mathrm{Ldim}(V_{n-1}) - 1$.

Algorithm 15 Standard Optimal Algorithm for online classification

Let $V_0 := H$.
for $n = 1, 2, \ldots$ **do**
  Receive object: $x_n \in \mathcal{X}$.
  Let $V_{n-1}^y := \{h \in V_{n-1} : h(x_n) = y\}$.
  Make prediction: $a_n := \arg\max_{y \in \{0,1\}} \mathrm{Ldim}(V_{n-1}^y)$.
  Receive correct answer (label): $y_n \in \{0, 1\}$.
  Update: $V_n := \{i \in V_{n-1} : h(x_n) = y_n\}$.
end for

Theorem 2.4 (Mistake bound for online classification)**.** *In the realizable setting, a learner using Algorithm 15 (Standard Optimal Algorithm) makes at most* $\mathrm{Ldim}(H)$ *mistakes.*

Proof. We just show that if the learner using the standard optimal algorithm makes a mistake in round n, then $\mathrm{Ldim}(V_n) \leq \mathrm{Ldim}(V_{n-1}) - 1$. Assume for sake of contradiction that $\mathrm{Ldim}(V_n) = \mathrm{Ldim}(V_{n-1})$. Then each of $\mathrm{Ldim}(V_{n-1}^0)$ and $\mathrm{Ldim}(V_{n-1}^1)$ is equal to $d := \mathrm{Ldim}(V_{n-1})$. We construct a labeled perfect binary tree of depth $d + 1$ as follows: take depth d trees shattered V_{n-1}^0 and V_{n-1}^1, and let their roots be the left and right children, respectively, of a new root node labeled by x_n. This new tree is shattered by V_{n-1}, a contradiction of the fact that $\mathrm{Ldim}(V_{n-1}) = d$. □

The standard optimal algorithm needs to compute the Littlestone dimension various subsets of H. There is an algorithm for deciding if $\mathrm{Ldim}(H) \leq d$ in time $O(|H|(2|\mathcal{X}|)^d)$.

Exercise 2.1. Prove that Algorithm 16 correctly decides if $\mathrm{Ldim}(H) \leq d$ in time $O(|H|(2|\mathcal{X}|)^d)$.

2.1.3 Linear threshold functions

A commonly used hypothesis class over the domain $\mathcal{X} = \mathbb{R}^d$ is the class of *linear threshold functions*. A linear threshold function $h_{w,b} \colon \mathbb{R}^d \to \{\pm 1\}$ is specified by a *weight vector* $w \in \mathbb{R}^d$ and a *threshold* $b \in \mathbb{R}$; on input $x \in \mathbb{R}^d$, it returns the value $h_{w,b}(x) = \mathrm{sign}(\langle w, x \rangle - b)$. (Here, it is more convenient to use the output value -1 in place of 0.)

It is possible to efficiently implement the Consistent Hypothesis algorithm (Algorithm 13) in this case, even though the hypothesis class is infinite. Observe that the set V_{n-1} is implicitly defined by the set of labeled examples $(x_1, y_1), \ldots, (x_n, y_n)$. So, in the case of

Algorithm 16 Algorithm for computing Littlestone dimension

Require: Hypothesis class H and dimension d.
Ensure: Answer to "Is $\mathrm{Ldim}(H) \leq d$?".

if $d = 0$ **then**
  if $|H| \leq 1$ **then**
    return "Yes".
  else
    return "No".
  end if
end if
for each $x \in \mathcal{X}$ **do**
  Let $H_{x,0} := \{h \in H : h(x) = 0\}$ and $H_{x,1} := \{h \in H : h(x) = 1\}$.
  Recursively run this algorithm on inputs $H_{x,0}$ and $d - 1$.
  Recursively run this algorithm on inputs $H_{x,1}$ and $d - 1$.
end for
if there is some $x \in \mathcal{X}$ such that $\mathrm{Ldim}(H_{x,0}) \not\leq d - 1$ and $\mathrm{Ldim}(H_{x,1}) \not\leq d - 1$ **then**
  return "No".
else
  return "Yes".
end if

linear threshold functions, choosing some $h \in V_{n-1}$ is the same as finding a solution $(w, b) \in \mathbb{R}^d \times \mathbb{R}$ to the following system of linear inequalities:

$$\begin{cases} \langle w, x_i \rangle - b > 0, & \text{for all } i \in [n] \text{ with } y_i = 1, \\ \langle w, x_i \rangle - b \leq 0, & \text{for all } i \in [n] \text{ with } y_i = -1. \end{cases}$$

This can be done using algorithms for linear programming. Note, however, that the mistake bound for Algorithm 13 is trivial here, because $|H| = \infty$. Indeed, without further restrictions, no algorithm can guarantee a finite mistake bound in this case.

We now describe a different algorithm that uses linear threshold functions, the *Online Perceptron* algorithm. To simplify notation, we describe the algorithm for special case of *homogeneous* linear threshold functions, i.e., $H := \{h_w : w \in \mathbb{R}^d\}$, where $h_w = h_{w,0}$. We analyze the algorithm in the following special case of the realizable setting. We assume the target concept is a homogeneous linear threshold function $c = h_{w^*}$ for some weight vector $w^* \in \mathbb{R}^d$, and moreover, for some $r > 0$:

$$\|x_n\|_2 \leq r, \quad |\langle w^*, x_n \rangle| \geq 1, \quad n = 1, 2, \ldots . \tag{2.1}$$

Above, $\|z\|_2 := \langle z, z \rangle^{1/2}$ denotes the Euclidean norm of the vector $z \in \mathbb{R}^d$. The second condition in Equation (2.1) implies that h_{w^*} not only provides the correct answers for every object x_n, but also that its real-valued *margin* $|\langle w^*, x_n \rangle|$ is large (i.e., at least one) for

every object x_n. Also, observe that since $y_n = h_{w^*}(x_n)$ for all n, the second condition in Equation (2.1) implies

$$y_n \langle w^*, x_n \rangle \geq 1, \quad n = 1, 2, \ldots .$$

Algorithm 17 Online Perceptron algorithm

1: Let $w_1 := 0 \in \mathbb{R}^d$.
2: **for** $n = 1, 2, \ldots$ **do**
3: Receive object: $x_n \in \mathbb{R}^d$.
4: Choose prediction: $a_n := h_{w_n}(x_n)$.
5: Receive correct answer (label): $y_n \in \{\pm 1\}$.
6: Update: if $a_n \neq y_n$, then $w_{n+1} := w_n + y_n x_n$; else $w_{n+1} := w_n$.
7: **end for**

Theorem 2.5 (Online Perceptron mistake bound)**.** *In the realizable setting where the target concept is $c = h_{w^*}$ for $w^* \in \mathbb{R}^d$ and the conditions in Equation* (2.1) *hold, a learner using Algorithm 17 (Online Perceptron) makes at most*

$$\|w^*\|_2^2 \, r^2$$

mistakes.

Proof. The analysis is based on monitoring the angle between w^* and the learner's weight vector, which measures how similar the learner's weight vector is to w^*. The cosine of the angle between w^* and w_n is

$$\frac{\langle w^*, w_n \rangle}{\|w^*\|_2 \, \|w_n\|_2} \tag{2.2}$$

as long as neither vector is zero. We want to show that the angle is close to zero (or equivalently, the cosine of the angle is close to one) when the learner has made many mistakes. To do this, we separately monitor the numerator and denominator of Equation (2.2).

Suppose the learner makes a mistake on round n and updates the weight vector. First, we lower-bound the inner product $\langle w^*, w_{n+1} \rangle$. We use the fact that $y_n \langle w^*, x_n \rangle \geq 1$:

$$\langle w^*, w_{n+1} \rangle = \langle w^*, w_n \rangle + y_n \langle w^*, x_n \rangle \geq \langle w^*, w_n \rangle + 1.$$

Therefore, by induction starting with $w_1 = 0$, if the learner makes M mistakes through n rounds,

$$\langle w^*, w_{n+1} \rangle \geq M. \tag{2.3}$$

Now, we upper-bound the (squared) length of w_{n+1}:

$$\|w_{n+1}\|_2^2 = \|w_n\|_2^2 + 2 y_n \langle w_n, x_n \rangle + \|x_n\|_2^2 \leq \|w_n\|_2^2 + r^2,$$

where the inequality uses the first condition in Equation (2.1) and the fact that h_{w_n} makes a mistake on x_n (which implies $y_n \langle w_n, x_n \rangle \leq 0$). By induction, starting with $w_1 = 0$, if the learner makes M mistakes through n rounds,

$$\|w_{n+1}\|_2^2 \leq r^2 M. \tag{2.4}$$

So, after M mistakes, the cosine of the angle between w^* and the learner's weight vector is at least

$$\frac{M}{\|w^*\|_2 \, r\sqrt{M}} = \sqrt{\frac{M}{\|w^*\|_2^2 \, r^2}}.$$

Since the cosine of the angle is at most one, we obtain the inequality

$$M \leq \|w^*\|_2^2 \, r^2,$$

thus proving the claim. □

Exercise 2.2. Design an online classification algorithm such that, in the realizable setting where the target concept is $c = h_{w^*}$ for $w^* \in \mathbb{R}^d$ and the conditions in Equation (2.1) hold, a learner using the algorithm makes at most $\log_2 \left\lceil (1 + \|w^*\|_2 \, r)^d \right\rceil$ mistakes.

D: Other topics: hinge-loss bound, Winnow.

2.1.4 Multi-class classification

We generalize the online classification problem from Section 2.1.2 that of simultaneously learning K disjoint concepts that partition $\mathcal{X}$. This is sometimes called the *K-class*, *K-way*, or simply *multi-class* classification problem. (The $K = 2$ case is usually called *binary classification*.) The protocol is given in Algorithm 18.

Algorithm 18 Protocol for online K-class classification

1: **for** $n = 1, 2, \ldots$ **do**
2: Teacher reveals object: $x_n \in \mathcal{X}$.
3: Learner makes prediction: $a_n \in [K]$.
4: Teacher reveals correct answer (label): $y_n \in [K]$.
5: **end for**

In this problem, we suppose that the learner has in mind a hypothesis class H of functions $h \colon \mathcal{X} \to \{1, \ldots, K\}$. The *realizable setting* here means that if $c_1, \ldots, c_K$ are the K target concepts that partition $\mathcal{X}$, then there exists $h \in H$ such that $h(x) = i$ if and only if $x \in c_i$, for each $i \in [K]$.

In this setting, we can again use the Consistent Hypothesis and Halving algorithms, replacing the "label set" $\{0, 1\}$ with $\{1, \ldots, K\} = [K]$. A learner using these algorithms make at most $|H| - 1$ and $\log_2 |H|$ mistakes, respectively. But, just like in the $K = 2$ case, both may be far from optimal.

Exercise 2.3. Define a generalization of the Littlestone dimension for K-class classification that precisely characterizes the optimal mistake bound achievable by a learning algorithm in the realizable setting.

2.1.5 Multi-class classification with bandit feedback

In Algorithm 18, the teacher provides the correct answer to the learner as feedback. However, it is also natural to consider a setting where the feedback is simply whether or not the learner's prediction was correct. We call this "bandit feedback"—the protocol is given in Algorithm 19—on account of similarities to the multi-armed bandits problem, and refer to the feedback in the setting in Algorithm 18 as "full-information feedback".

Algorithm 19 Protocol for online K-class classification with bandit feedback

for $n = 1, 2, \ldots$ **do**
  Teacher reveals object: $x_n \in \mathcal{X}$.
  Learner makes prediction: $a_n \in [K]$.
  Teacher reveals correctness of prediction: $1_{\{a_n = y_n\}} \in \{0, 1\}$.
end for

We design an algorithm for this problem using as a subroutine an algorithm that expects full-information feedback in each round. Of course, in the present setting, one cannot run such a full-information algorithm directly, because the requisite feedback is not provided. However, we can create an algorithm that provides "fictitious" feedback to multiple executions of the full-information algorithm, ensuring that at least one execution of the algorithm receives valid feedback. Although we do not know which execution receives valid feedback, we can use an "experts"-type algorithm to predict nearly as well as the one that does.

In Algorithm 20, if A is an execution of an algorithm $\mathcal{A}$ for the online K-class classification problem, we denote by $A(x)$ for $x \in \mathcal{X}$ the prediction of A if presented x in the next round (but without actually updating the internal state of A). We only change the internal state of A if we *update* it with a labeled example $(x, y) \in \mathcal{X} \times [K]$.

Theorem 2.6 (Mistake bound for online multi-class classification with bandit feedback)**.** *Suppose that in online K-class classification with full-information feedback, a learner using algorithm $\mathcal{A}$ is guaranteed to make at most M_0 mistakes. Then, in online K-class classification with bandit feedback, a learner using Algorithm 20 with parameters $\mathcal{A}$ and $\eta > \ln(K-1)$ makes at most*

$$\frac{K\eta}{1-(K-1)e^{-\eta}} \cdot M_0$$

mistakes.

Proof. Let Z_n denote the total weight $\sum_{A \in L} w_A$ of all executions of $\mathcal{A}$ at the start of round n, so $Z_1 = 1$. Suppose the learner makes a mistake in round n. Let $\alpha_n \geq 1/K$ denote the

Algorithm 20 Algorithm for online K-class classification with bandit feedback

Require: $\mathcal{A}$, algorithm for online K-class classification with full-information feedback; $\eta > 0$.

Let A be a new execution of $\mathcal{A}$, and create list $L := (A)$.
Let $w_A := 1$.
for $n = 1, 2, \ldots$ **do**
  Receive object: $x_n \in \mathcal{X}$.
  Make prediction: $a_n := \arg\max_{y \in [K]} \sum_{A \in L : A(x_n) = y} w_A$
  Receive correctness of prediction: $1_{\{a_n = y_n\}} \in \{0, 1\}$.
  if $a_n \neq y_n$ **then**
    for each $A \in L$ such that $A(x_n) = a_n$ **do**
      Remove A from L.
      for each $y \in [K] \setminus \{a_n\}$ **do**
        Append a clone A_y of A to L.
        Update A_y with (x_n, y).
        Assign weight $w_{A_y} := e^{-\eta} \cdot w_A$.
      end for
    end for
  end if
end for

fraction of the total weight corresponding to $A \in L$ that predict $A(x_n) = a_n$. Every $A \in L$ accounted for in this fraction is cloned $K - 1$ times and is given weight $e^{-\eta} w_A$. So

$$\begin{aligned} Z_{n+1} &= (1 - \alpha_n) Z_n + (K-1) e^{-\eta} \cdot \alpha_n Z_n \\ &= (1 - \alpha_n (1 - (K-1) e^{-\eta})) Z_n \\ &\leq \left(1 - \frac{1 - (K-1)e^{-\eta}}{K}\right) Z_n. \end{aligned}$$

So if a total of M mistakes have been made through round n, then

$$\begin{aligned} Z_{n+1} &\leq \left(1 - \frac{1 - (K-1)e^{-\eta}}{K}\right)^M \\ &\leq \exp\left(-\frac{1 - (K-1)e^{-\eta}}{K} \cdot M\right). \end{aligned}$$

On the other hand, at least one of the executions of $\mathcal{A}$ in L has received the feedback required in the full-information case, and hence makes at most M_0 mistakes by assumption. This means its weight is at least $e^{-\eta M_0}$, so the total weight satisfies

$$Z_{n+1} \geq e^{-\eta M_0}.$$

Combining the upper- and lower-bounds on Z_{n+1} and rearranging proves the claimed bound on M. □

The blow-up factor in the bound, i.e.,

$$\frac{K\eta}{1-(K-1)e^{-\eta}},$$

is minimized by setting $\eta := -W_{-1}(-1/(e(K-1))) - 1$, where W_{-1} denotes the lower branch of the Lambert-W function[1]. This yields the factor

$$K \cdot \left(-W_{-1}\left(-\frac{1}{e(K-1)}\right)\right).$$

As $K \to \infty$, we have

$$K \cdot \left(-W_{-1}\left(-\frac{1}{e(K-1)}\right)\right) = K \cdot \big(\ln(K-1) + \ln(1+\ln(K-1)) + 1 + o(1)\big).$$

This shows that, in some sense, the setting with bandit feedback is not more than $K \log K$ times as hard as the setting with full-information feedback.

2.2 Statistical framework

2.2.1 The setting

So far, we have been concerned with worst-case sequences of examples that the teacher could provide. An alternative way the teacher could provide examples is to choose them randomly according a probability distribution. A common assumption is that $x_1, x_2, \ldots$ are *independent and identically distributed* (*iid*). In this case, we write $x_1, x_2, \ldots \sim_{\text{iid}} P$, where P is the marginal distribution of x_1 (unknown to the learner). In this iid setting, the learner may be "quizzed" on the classification of an independent draw $x_n \sim P$ in each round n. Alternatively, the learner may be asked to return a hypothesis h such that, for some given $\varepsilon \in (0,1)$, the probability

$$\text{err}_P(h) := \Pr_{x \sim P}(h(x) \neq c(x))$$

is at most ε, where c is the target concept. The probability $\text{err}_P(h)$ is called the *error rate* of h (with respect to P). We are interested in performance guarantees of the learner that hold either in expectation, or with probability at least $1-\delta$ for some given $\delta \in (0,1)$, over the random choice of the sequence $x_1, x_2, \ldots$ and any internal randomness used by the learner.

2.2.2 Mistake bounds to error rate bounds

One may intuit that the worst-case setting from Section 2.1 is more difficult than the iid setting. We can formalize this intuition by showing how to use an algorithm with a mistake

[1] http://en.wikipedia.org/wiki/Lambert_W_function

bound (such as the algorithms from Section 2.1, at least in the realizable setting) to return a hypothesis with low error rate. This scheme is called an *online-to-batch* conversion.

Let $\mathcal{A}$ be an algorithm for online classification with mistake bound M. At the start of round n, the algorithm $\mathcal{A}$ prescribes a hypothesis that the learner uses in the round to classify the object $x_n \in \mathcal{X}$ revealed by the teacher. This hypothesis may come from a particular hypothesis class, or it may simply be regarded as the internal memory state of the learner at the start of the round.[2] We may also assume without loss of generality that if the learner does not make a mistake in a given round, then it uses the same hypothesis in the next round. Such a learner is called a *conservative learner*. We can achieve this by saving the internal state of the learner at the start of each round, and restoring this state if the learner does not make a mistake in the round.

Let $\varepsilon, \delta \in (0, 1)$ be given. We execute the algorithm $\mathcal{A}$ in the iid setting of online classification, which yields a sequence of hypotheses $h^{(0)}, h^{(1)}, \ldots$ where $h^{(i)}$ is the hypothesis produced after the learner commits its i-th mistake. For each i, we check if $h^{(i)}$ predicts correctly on the next

$$n_i := \left\lceil \frac{1}{\varepsilon} \ln \frac{(i+1)(i+2)}{\delta} \right\rceil$$

rounds. We return the first such $h^{(i)}$ that passes this test. If $\mathcal{A}$ is guaranteed to make at most M mistakes, then the algorithm will halt after at most

$$\sum_{i=0}^{M} n_i = O\left(\frac{M}{\varepsilon} \log \frac{M}{\delta}\right)$$

rounds. Let E_i be the event that at least i mistakes are made and the hypothesis $h^{(i)}$ has error rate more than ε. What is the probability that for some $i \in \{0, \ldots, M\}$, E_i holds and $h^{(i)}$ passes the test, i.e.,

$$\exists i \in \{0, \ldots, M\} \,.\, E_i \wedge h^{(i)} \text{ passes the test.}$$

By a union bound, this is at most

$$\sum_{i=0}^{M} \Pr\left(E_i \wedge h^{(i)} \text{ passes the test}\right).$$

The i-th term in the summation can be bounded as

$$\Pr\left(E_i \wedge h^{(i)} \text{ passes the test}\right) = \Pr\left(h^{(i)} \text{ passes the test} \mid E_i\right) \cdot \Pr\left(E_i\right) \leq (1-\varepsilon)^{n_i} \cdot 1,$$

because conditional on the event that at least i mistakes are made and $h^{(i)}$ has error rate more than ε, the chance that $h^{(i)}$ passes the test is at most the probability that a coin with

[2] If the learner uses randomness to make its prediction in a given round, then the hypothesis for that round may be a *randomized hypothesis*. In this case, we extend the definition of error rate to also consider the internal randomness of the randomized hypothesis.

heads bias more than ε comes up tails in n_i independent tosses. Now using the fact that $1 + z \leq e^z$ for any real number z, we have

$$(1-\varepsilon)^{n_i} \leq \exp(-\varepsilon n_i) \leq \exp\left(-\ln\frac{(i+1)(i+2)}{\delta}\right) = \frac{\delta}{(i+1)(i+2)}.$$

So, the probability that for some i, E_i holds and $h^{(i)}$ passes the test is at most

$$\sum_{i=0}^{M}\frac{\delta}{(i+1)(i+2)} = \delta\sum_{i=0}^{M}\frac{1}{i+1}-\frac{1}{i+2} \leq \delta.$$

Theorem 2.7 (Mistake bound to error rate bound)**.** *Suppose $\mathcal{A}$ is an algorithm for online classification such that a learner using $\mathcal{A}$ on any sequence $(x_n)_{n\geq 1}$ makes at most M mistakes. Then there is an algorithm for a learner in the iid setting using $\mathcal{A}$ as a black-box that, given any $\varepsilon, \delta \in (0,1)$, with probability at least $1-\delta$ produces a hypothesis with error rate at most ε after seeing most $O((M/\varepsilon)\log(M/\delta))$ random examples.*

In fact, with a slightly different procedure, we can get away with just $O((M+\log(1/\delta))/\varepsilon)$ random examples.

2.2.3 Consistent Hypothesis algorithm

We now consider the Consistent Hypothesis algorithm (Algorithm 13) in the iid setting. Recall that in the worst-case setting, its mistake bound could be exponentially worse than that of the Halving algorithm ($|H|-1$ versus $\log_2|H|$). However, in the iid setting, the Consistent Hypothesis algorithm may fare much better.

Theorem 2.8 (Consistent Hypothesis error rate bound)**.** *In the iid setting, if a learner uses Algorithm 13 (Consistent Hypothesis), then for any n and $\delta \in (0,1)$, with probability at least $1-\delta$, every hypothesis $h \in V_n$ has error rate*

$$\mathrm{err}_P(h) \leq \frac{\ln(|H|/\delta)}{n}.$$

Proof. Let $B := \{h \in H : \mathrm{err}_P(h) > \ln(|H|/\delta)/n\}$. We'll show that it is unlikely for any hypotheses in B remain in V_n. The probability that a particular $h \in B$ is in V_n is $(1-\mathrm{err}_P(h))^n$. Therefore, by a union bound, the probability that some $h \in B$ remains in V_n is at most

$$\begin{aligned}\sum_{h\in B}(1-\mathrm{err}_P(h))^n &\leq \sum_{h\in B}\exp(-\mathrm{err}_P(h)n)\\ &\leq \sum_{h\in B}\exp(-\ln(|H|/\delta))\\ &\leq \delta,\end{aligned}$$

which proves the claim. □

Therefore, for a given $\varepsilon \in (0,1)$, the learner requires

$$\frac{\ln(|H|/\delta)}{\varepsilon}$$

random examples to guarantee that, with probability at least $1-\delta$, every hypothesis that is correct on these random examples has error rate at most ε.

Exercise 2.4. Determine a bound on the expected number of mistakes made by a learner using the Consistent Hypothesis algorithm (Algorithm 13) after n rounds in the realizable iid setting.

2.2.4 Empirical Risk Minimization

Thus far, we have considered only the realizable setting, where it is assumed that the target concept c is in the hypothesis class H used by the learner. If we do not make this assumption, we instead just want to find a hypothesis $h \in H$ whose error rate is not much worse than that of the *best* hypothesis in H.

In fact, we will consider the following generalization of the iid setting. Let P be an arbitrary probability distribution over $\mathcal{X} \times \mathcal{Y}$ (where $\mathcal{Y} = \{0,1\}$ or $\mathcal{Y} = \{\pm 1\}$ for binary classification, and $\mathcal{Y} = [K]$ for multi-class classification). We let $(x_1, y_1), (x_2, y_2), \ldots \sim_{\text{iid}} P$, and the learner is asked to return a hypothesis h such that, for some given $\varepsilon \in (0,1)$, the difference between the *error rate* of h,

$$\text{err}_P(h) := \Pr_{(x,y)\sim P}(h(x) \neq y),$$

and the smallest error rate among hypotheses in H, is at most ε. This difference

$$\text{Reg}_P(h, H) := \text{err}_P(h) - \min_{h' \in H} \text{err}_P(h')$$

is called the *excess risk* or *regret* of h (analogous to the concept of regret from Section 1.1.4). We call this setting the *agnostic iid setting*.

The Consistent Hypothesis algorithm may not work in the agnostic setting because there may be no hypothesis in H that is correct on every example. However, there is a natural generalization called the *Empirical Risk Minimization* (*ERM*) algorithm (Algorithm 21).[3]

Let P_n denote the *empirical distribution* on $(x_1, y_1), \ldots, (x_n, y_n)$—i.e., P_n is the *random* probability distribution on $\mathcal{X} \times \mathcal{Y}$ that assigns $1/n$ mass to each (x_i, y_i). The *empirical risk* of a hypothesis h on data $(x_1, y_1), \ldots, (x_n, y_n)$ is the error rate of h under P_n:

$$\text{err}_{P_n}(h) = \frac{1}{n} \sum_{i=1}^{n} 1_{\{h(x_i) \neq y_i\}}.$$

[3]ERM usually refers to the non-sequential version of Algorithm 21, which returns a hypothesis $h \in H$ that minimizes the number of mistakes on a given sequence of labeled examples. Here, we shall also use ERM to refer to the sequential procedure in Algorithm 21.

Algorithm 21 Empirical Risk Minimization

1: Pick $h_0 \in H$.
2: **for** $n = 1, 2, \ldots$ **do**
3: Receive object: $x_n \in \mathcal{X}$.
4: Make prediction: $a_n := h_{n-1}(x_n)$.
5: Receive correct answer (label): $y_n \in \{0, 1\}$.
6: Update: pick $h_n \in \arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} 1_{\{h(x_i) \neq y_i\}}$.
7: **end for**

(The error rate $\mathrm{err}_P(h)$ under P is sometimes called the *risk* of h.) Given $(x_1, y_1), \ldots, (x_n, y_n)$, the ERM algorithm picks a minimizer of the empirical risk:

$$h_n \in \arg\min_{h \in H} \mathrm{err}_{P_n}(h).$$

(There could be multiple minimizers.) The hope is that if P_n is close to P in a suitable sense, then the ERM algorithm is almost the same as choosing a hypothesis in $\arg\min_{h \in H} \mathrm{err}_P(h)$.

To motivate how P_n should be close to P, consider that for any fixed $h \in H$, the random variable $t \cdot \mathrm{err}_{P_n}(h)$ has a binomial distribution (with n trials and success probability $\mathrm{err}_P(h)$). Therefore, using properties of the binomial distribution, it is possible to bound $|\mathrm{err}_{P_n}(h) - \mathrm{err}_P(h)|$ (both in expectation and with high probability). This approach, however, does not directly apply to h_n, since h_n itself it chosen based on $(x_1, y_1), \ldots, (x_n, y_n)$.

We instead analyze the maximum deviation $\max_{h \in H} |\mathrm{err}_{P_n}(h) - \mathrm{err}_P(h)|$, which uniformly bounds the deviation of $\mathrm{err}_{P_n}(h)$ from $\mathrm{err}_P(h)$ for *every* $h \in H$. Since the hypothesis h_n chosen by ERM is chosen from H, the maximum deviation provides an upper-bound for $|\mathrm{err}_{P_n}(h_n) - \mathrm{err}_P(h_n)|$.

We shall use the maximum deviation as follows. Fix a particular minimizer $h^* \in \arg\min_{h \in H} \mathrm{err}_P(h)$. Then

$$\begin{aligned}
\mathrm{Reg}(h_n, H) &= \mathrm{err}_P(h_n) - \mathrm{err}_P(h^*) \\
&= (\mathrm{err}_P(h_n) - \mathrm{err}_{P_n}(h_n)) + (\mathrm{err}_{P_n}(h^*) - \mathrm{err}_P(h^*)) + (\mathrm{err}_{P_n}(h_n) - \mathrm{err}_{P_n}(h^*)) \\
&\leq (\mathrm{err}_P(h_n) - \mathrm{err}_{P_n}(h_n)) + (\mathrm{err}_{P_n}(h^*) - \mathrm{err}_P(h^*)) \\
&\leq \max_{h \in H} (\mathrm{err}_P(h) - \mathrm{err}_{P_n}(h)) + (\mathrm{err}_{P_n}(h^*) - \mathrm{err}_P(h^*)).
\end{aligned}$$

The first inequality uses the fact that $\mathrm{err}_{P_n}(h_n) \leq \mathrm{err}_{P_n}(h^*)$ by definition of h_n, and the second uses the fact that $h_n \in H$. In particular, we have

$$\mathbb{E}\,\mathrm{Reg}(h_n, H) \leq \mathbb{E} \max_{h \in H} (\mathrm{err}_P(h) - \mathrm{err}_{P_n}(h)).$$

2.2.5 Maximum deviation bounds for finite classes

In this section, we let P denote a probability distribution over a space Z. Let P_n denote the empirical distribution on $z_1, \ldots, z_n \sim_{\mathrm{iid}} P$, and let F denote a set of $[-1, +1]$-valued

functions on Z. We are interested in the maximum deviation

$$\max_{f \in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f,$$

where

$$\mathbb{E}_P f := \mathbb{E}_{z \sim P} f(z), \qquad \mathbb{E}_{P_n} f := \frac{1}{n} \sum_{i=1}^{n} f(z_i).$$

We can straightforwardly map the setting from Section 2.2.4 onto this one, by considering a class of functions F_H of the form $f_h(x, y) = 2 \cdot 1_{\{h(x) \neq y\}} - 1$ for $h \in H$, in which case

$$\mathrm{err}_P(h) - \mathrm{err}_{P_n}(h) = \frac{1}{2} \left(\mathbb{E}_P f_h - \mathbb{E}_{P_n} f_h \right).$$

The class of functions F_H is called the *(zero-one) loss class* for the hypothesis class H.

For any $f \in F$, the random variable $n \cdot \mathbb{E}_{P_n} f$ can be written as the sum of n independent random variables, each with range $[-1, +1]$. The following lemma provides a bound on the moment generating function of such a random variable.

Lemma 2.1 (Hoeffding's inequality). *Let $X_1, \ldots, X_n$ be independent $[-1, +1]$-valued random variables; let $S := X_1 + \cdots + X_n$. Then*

D: Prove this.

$$\mathbb{E} \exp(\lambda(S - \mathbb{E}(S))) \leq \exp\left(\frac{n\lambda^2}{2}\right), \quad \lambda \in \mathbb{R}.$$

Theorem 2.9. *Let P be any probability distribution, and let P_n be the empirical distribution for an iid sample from P of size n. For any set F of $[-1, +1]$-valued functions,*

$$\mathbb{E}\left(\max_{f \in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f \right) \leq \sqrt{\frac{2 \ln |F|}{n}}.$$

Proof. Fix $\lambda > 0$, whose value will be determined later. We bound the expected maximum deviation as follows. Begin by writing the following identity for the expected maximum deviation:

$$\mathbb{E}\left(\max_{f \in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f \right) = \frac{1}{\lambda} \ln \exp \left[\mathbb{E}\left(\lambda \max_{f \in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f \right) \right].$$

The right-hand side now involves the exponential of an expectation; changing the order of their application can only lead to a larger value, by Jensen's inequality:

$$\exp\left[\mathbb{E}\left(\lambda \max_{f \in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f \right) \right] \leq \mathbb{E}\left[\exp\left(\lambda \max_{f \in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f \right) \right].$$

(The inequality doesn't lose much if λ is small.) Now inside the expectation is the exponential of the maximum deviation. By monotonicity, this is the same as the maximum exponential deviation, and hence at most the sum of the exponential deviations:

$$\exp\left(\lambda \max_{f\in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f\right) = \max_{f\in F} \exp\left(\lambda\, \mathbb{E}_P f - \mathbb{E}_{P_n} f\right) \le \sum_{f\in F} \exp\left(\lambda\, \mathbb{E}_P f - \mathbb{E}_{P_n} f\right).$$

(The inequality doesn't lose much if λ is large.) The expectation of each side allows the expectation on the right-hand side to pass through the summation by linearity, leading to the summation of moment generating functions that can be bounded using Lemma 2.1:

$$\mathbb{E} \exp\left(\lambda \max_{f\in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f\right) \le \sum_{f\in F} \mathbb{E} \exp\left(\lambda\, \mathbb{E}_P f - \mathbb{E}_{P_n} f\right) \le |F| \exp\left(\frac{\lambda^2}{2n}\right).$$

So, overall we have established

$$\mathbb{E}\left(\max_{f\in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f\right) \le \frac{1}{\lambda} \ln\left(|F| \exp\left(\frac{\lambda^2}{2n}\right)\right) = \frac{\ln|F|}{\lambda} + \frac{\lambda}{2n}.$$

Since this holds for arbitrary $\lambda > 0$, we choose the value of λ to minimize the bound, which yields the claimed bound. □

Applying Theorem 2.9 to ERM gives

$$\mathbb{E}\,\mathrm{Reg}_P(h_n, H) \le \mathbb{E}\left(\max_{h\in H} \mathrm{err}_P(h) - \mathrm{err}_{P_n}(h)\right) \le O\left(\sqrt{\frac{\ln|H|}{n}}\right). \tag{2.5}$$

Thus, it suffices for the learner to use

$$O\left(\frac{\ln|H|}{\varepsilon^2}\right)$$

labeled examples to guarantee regret at most ε in expectation.

Exercise 2.5. Prove a bound on $\mathrm{Reg}_P(h_n, H)$ that holds with high probability: for any $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\mathrm{Reg}_P(h_n, H) \le O\left(\sqrt{\frac{\ln(|H|/\delta)}{n}}\right).$$

Exercise 2.6. Determine a bound on the expected number of mistakes made by a learner using the ERM algorithm (Algorithm 21) after n rounds in the agnostic iid setting. Compare the result to that from Exercise 2.4.

2.2.6 Maximum deviation bounds for infinite classes

We now consider the maximum deviation over a function class F with possibly infinite cardinality. For such a function class F, Theorem 2.9 may be very loose or even vacuous, so a more refined analysis capturing the "richness" of F is needed.

We give a bound on the maximum deviation in terms of the *Rademacher complexity* of F, which we now define. Define the inner product $\langle \cdot, \cdot \rangle_n$ over $\mathbb{R}^n$ by

$$\langle a, b \rangle_n := \frac{1}{n} \sum_{i=1}^{n} a_i b_i$$

for vectors $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$ in $\mathbb{R}^n$. Let $\sigma = (\sigma_1, \ldots, \sigma_n)$ be a random vector where $\sigma_1, \ldots, \sigma_n$ are iid *Rademacher random variables*: $\Pr(\sigma_1 = 1) = \Pr(\sigma_1 = -1) = 1/2$. Define the *Rademacher average* of a set of vectors $S \subseteq \mathbb{R}^n$ by

$$\mathrm{Rad}(S) := \mathbb{E}_\sigma \sup_{v \in S} \langle v, \sigma \rangle_n,$$

where the expectation $\mathbb{E}_\sigma$ is taken over σ (conditional on any other random variables). For $z_{1:n} = (z_1, \ldots, z_n) \in Z^n$, define

$$F_{|z_{1:n}} := \{(f(z_1), \ldots, f(z_n)) : f \in F\}$$

to be the set of "behaviors" on $z_1, \ldots, z_n$ realized by functions in F. Finally, define the *Rademacher complexity* of F to be the expectation of the Rademacher average of $F_{|z_{1:n}}$ for an iid sample $z_1, \ldots, z_n \sim_{\text{iid}} P$:

$$\mathrm{Rad}_n(F; P) = \mathrm{Rad}_n(F) := \mathbb{E}\left[\mathrm{Rad}(F_{|z_{1:n}})\right].$$

We drop the notational dependence on P when the probability distribution is clear from context.

The Rademacher complexity $\mathrm{Rad}_n(F)$ is a measure the "richness" of F with respect to data from P.[4] Some example values of $\mathrm{Rad}_n(F)$ are as follows. If F contains just a single $[-1, +1]$-valued function f_0, then

$$\mathrm{Rad}_n(F) = 0.$$

[4]Suppose F is a class of $\{\pm 1\}$-valued functions, and r denotes a uniformly random $\{\pm 1\}$-valued function. Then, $\mathrm{Rad}(F|_{z_{1:n}})$ (i.e., $\mathrm{Rad}_n(F)$ conditional on $z_1, \ldots, z_n$) measures the how close the random function r is to functions in F on average, where we treat functions as members of the function space $L^2(P_n)$:

$$\langle f, g \rangle_{L^2(P_n)} = \frac{1}{n} \sum_{i=1}^{n} f(z_i) g(z_i), \quad \|f - g\|^2_{L^2(P_n)} = \langle f - g, f - g \rangle_{L^2(P_n)} = \frac{1}{n} \sum_{i=1}^{n} (f(z_i) - g(z_i))^2.$$

Indeed, we have

$$\mathrm{Rad}(F|_{z_{1:n}}) = \mathbb{E}_r \sup_{f \in F} \langle f, r \rangle_{L^2(P_n)} = \mathbb{E}_r \sup_{f \in F} \left[1 - \frac{1}{2}\|f - r\|^2_{L^2(P_n)}\right] = 1 - \frac{1}{2}\,\mathbb{E}_r \inf_{f \in F} \|f - r\|^2_{L^2(P_n)},$$

where $\mathbb{E}_r$ is the expectation with respect to the choice of the random function r.

If $F = \{f_0, -f_0\}$, then

$$\mathrm{Rad}_n(F) \le \sqrt{\frac{\mathbb{E}_P f_0^2}{n}}.$$

At the other extreme, if $z_1, \ldots, z_n$ are distinct almost surely and F contains all $\{\pm 1\}$-valued functions, then

$$\mathrm{Rad}_n(F) = 1.$$

An intermediate case is where the cardinality of $F_{|z_{1:n}}$ is always bounded by a polynomial in n, say $O(n^d)$ for some $d \ge 0$. In this case, we have

$$\mathrm{Rad}_n(F) \le O\left(\sqrt{\frac{d \log n}{n}}\right).$$

For example, suppose $F := \{f_b : b \in \mathbb{R}\}$ is the set of $\{\pm 1\}$-valued functions on $Z = \mathbb{R}$ where $f_b(z) = 2 \cdot 1_{\{z>b\}} - 1$. The cardinality of F is infinite, but for any given $z_1, \ldots, z_n \in Z$, the set $F_{|z_{1:n}}$ has cardinality at most $n + 1$. Other examples are considered in Section 2.2.7.

Theorem 2.10. *Let P be any probability distribution, and let P_n be the empirical distribution for an iid sample from P of size n. For any set F of $[-1, +1]$-valued functions,*

$$\mathbb{E}\left(\sup_{f \in F} |\mathbb{E}_P f - \mathbb{E}_{P_n} f|\right) \le 2\,\mathrm{Rad}_n(F).$$

Proof. We use two "symmetrization" tricks to change the stochastic process into one that is easier to analyze. First, we use "symmetrization by a ghost sample". Let $z'_1, \ldots, z'_n \sim_{\text{iid}} P$ (the ghost sample) be independent of $z_1, \ldots, z_n$, and let P'_n be the empirical distribution on $z'_1, \ldots, z'_n$. We have

$$\begin{aligned}
\mathbb{E} \sup_{f \in F} |\mathbb{E}_P f - \mathbb{E}_{P_n} f| &= \mathbb{E} \sup_{f \in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f \\
&= \mathbb{E} \sup_{f \in F} \mathbb{E}[\mathbb{E}_{P'_n} f - \mathbb{E}_{P_n} f \mid z_1, \ldots, z_n] \\
&\le \mathbb{E} \sup_{f \in F} \mathbb{E}_{P'_n} f - \mathbb{E}_{P_n} f,
\end{aligned}$$

where the inequality follows from Jensen's inequality. Now we use "symmetrization by random signs". Observe that

$$\sup_{f \in F} \mathbb{E}_{P'_n} f - \mathbb{E}_{P_n} f = \sup_{f \in F} \frac{1}{n} \sum_{i=1}^{n} (f(z'_i) - f(z_i))$$

has the same distribution as

$$\sup_{f \in F} \frac{1}{n} \sum_{i=1}^{n} s_i (f(z'_i) - f(z_i))$$

for any $s_1, \ldots, s_n \in \{\pm 1\}$. Therefore

$$\begin{aligned}\mathbb{E} \sup_{f \in F} \mathbb{E}_{P'_n} f - \mathbb{E}_{P_n} f &= \mathbb{E}_\sigma \mathbb{E} \sup_{f \in F} \frac{1}{n} \sum_{i=1}^n \sigma_i (f(z'_i) - f(z_i)) \\ &\leq 2 \mathbb{E}_\sigma \mathbb{E} \sup_{f \in F} \frac{1}{n} \sum_{i=1}^n \sigma_i f(z_i) \\ &= 2 \operatorname{Rad}_n(F),\end{aligned}$$

where the inequality follows from the triangle inequality and sub-additivity of supremum. □

Bounds on the expected maximum deviation may be used to obtain bounds that hold with high probability. To do this, we use the following probability concentration inequality.

Theorem 2.11 (McDiarmid's inequality)**.** *Let $g \colon \mathbb{R}^n \to \mathbb{R}$ be a function with the* bounded differences *property: for some $c_1, \ldots, c_n \geq 0$,*

D: Prove this using Doob martingale.

$$\sup_{z_1, \ldots, z_n, z'_i} |g(z_1, \ldots, z_i, \ldots, z_n) - g(z_1, \ldots, z'_i, \ldots, z_n)| \leq c_i, \quad i \in [n].$$

Let $z_1, \ldots, z_n$ be independent random variables. Then for any $\epsilon > 0$,

$$\Pr\left(g(z_1, \ldots, z_n) - \mathbb{E}\, g(z_1, \ldots, z_n) > \epsilon\right) \leq \exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n c_i^2}\right).$$

For a class F of $[-1, +1]$-valued functions, the function given by

$$g(z_1, \ldots, z_n) := \sup_{f \in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f$$

has the bounded differences property with $c_i = 2/n$ for all i. Therefore, by Theorem 2.11, for any $\epsilon > 0$,

$$\Pr\left(\sup_{f \in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f > \mathbb{E}\left(\sup_{f \in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f\right) + \epsilon\right) \leq \exp(-n\epsilon^2/2).$$

Now plugging in the bound from Theorem 2.10, we have for any $\delta \in (0, 1)$ that

$$\sup_{f \in F} \mathbb{E}_P f - \mathbb{E}_{P_n} f \leq 2 \operatorname{Rad}_n(F) + \sqrt{\frac{2 \ln(1/\delta)}{n}} \tag{2.6}$$

holds with probability at least $1 - \delta$.

Theorem 2.12 (ERM error rate bound in terms of Rademacher complexity)**.** *Let H be a hypothesis class. In the agnostic iid setting, for any n and $\delta \in (0,1)$, the following hold with probability at least $1-\delta$:*

$$\mathrm{err}_P(h) - \mathrm{err}_{P_n}(h) \leq \mathrm{Rad}_n(F_H) + \sqrt{\frac{\ln(1/\delta)}{2n}}, \quad h \in H.$$

where F_H is the class of $\{\pm 1\}$-valued functions of the form $f_h(x,y) = 2 \cdot 1_{\{h(x)\neq y\}} - 1$ for $h \in H$, Also, with probability at least $1-\delta$, the hypothesis h_n picked by a learner using the ERM algorithm (Algorithm 21) in round n satisfies

$$\mathrm{Reg}_P(h_n, H) \leq \mathrm{Rad}_n(F_H) + O\left(\sqrt{\frac{\ln(1/\delta)}{n}}\right).$$

It is worth noting that when $\mathcal{Y} = \{\pm 1\}$ (i.e., binary classification), the Rademacher complexity of the class F_H from Theorem 2.12 depends only on the marginal distribution $P_{\mathcal{X}}$ of P over $\mathcal{X}$. To see this, observe that

$$\begin{aligned}
\mathrm{Rad}((F_H)|_{(x_i,y_i)_{i=1}^n}) &= \mathbb{E}_\sigma \sup_{h\in H} \frac{1}{n}\sum_{i=1}^n \sigma_i \left(2 \cdot 1_{\{h(x_i)\neq y_i\}} - 1\right) \\
&= \mathbb{E}_\sigma \sup_{h\in H} \frac{1}{n}\sum_{i=1}^n -\sigma_i y_i h(x_i) \\
&= \mathbb{E}_\sigma \sup_{h\in H} \frac{1}{n}\sum_{i=1}^n \sigma_i h(x_i) \\
&= \mathrm{Rad}(H|_{x_{1:n}}),
\end{aligned}$$

where the penultimate step uses the fact that $(-\sigma_1 y_1, \ldots, -\sigma_n y_n)$ has the same distribution as $(\sigma_1, \ldots, \sigma_n)$.

For any realization of the Rademacher random variables σ, observe that

$$\sup_{h\in H} \frac{1}{n}\sum_{i=1}^n \sigma_i h(x_i) = 1 - 2 \inf_{h\in H} \mathrm{err}_{P_{n,\sigma}}(h),$$

where $P_{n,\sigma}$ is the empirical distribution on $(x_1,\sigma_1),\ldots,(x_n,\sigma_n)$. This is a measure of how well hypotheses in H can fit uniformly random labels, and thus can be computed using ERM. Moreover, the function $g(z_1,\ldots,z_n) := \sup_{h\in H} \frac{1}{n}\sum_{i=1}^n \sigma_i h(x_i)$ (where $z_i = (x_i, y_i)$ for each i) satisfies the bounded differences property from Theorem 2.11 with $c_i = 1/n$ for each i. Therefore, for any $\delta \in (0,1)$, with probability at least $1-\delta$ (over the choice of $x_1,\ldots,x_n \sim_{\text{iid}} P_{\mathcal{X}}$ and the Rademacher random variables σ),

$$\left| \mathrm{Rad}_n(H) - \sup_{h\in H} \frac{1}{n}\sum_{i=1}^n \sigma_i h(x_i) \right| \leq \sqrt{\frac{2\ln(1/\delta)}{n}}. \tag{2.7}$$

Combining this with the bound from Theorem 2.12 (with a union bound) gives an *empirical bound* on the excess risk, meaning that the bound can be computed from the observed data (and the random realization of the Rademacher random variables).

2.2.7 Vapnik-Chervonenkis dimension

We now give a purely combinatorial property of a function class that can be used to bound its Rademacher complexity with respect to any distribution.

We say the set of points $z_1, \ldots, z_n \in Z$ is *shattered* by a class F of $\{\pm 1\}$-valued functions on Z if F realizes all 2^n possible behaviors on $z_{1:n}$, i.e.,

$$|F_{|z_{1:n}}| = |\{(f(z_1), \ldots, f(z_n)) : f \in F\}| = 2^n.$$

For any positive integer n, define $\Pi_F(n) := \max_{z_1,\ldots,z_n \in Z} |F_{|z_{1:n}}|$ to be the maximum number of behaviors of F on n points. The *Vapnik-Chervonenkis (VC) dimension* of F is the size of the largest set shattered by F, i.e., $\sup\{d : \Pi_F(d) = 2^d\}$. (The VC dimension is infinite if there are arbitrarily large shattered sets.)

A first example of the VC dimension is for the hypothesis class of threshold functions $F = \{f_b : b \in \mathbb{R}\}$ on $Z = \mathbb{R}$, where $f_b(z) = \text{sign}(z - b)$. It is possible to shatter any single point with F, so the VC dimension of F is at least one. For any pair of distinct points $(z_1, z_2) \in \mathbb{R}^2$ with $z_1 < z_2$, it is not possible to realize the behavior $(+1, -1)$. So the VC dimension of F is one.

A second example of the VC dimension is for the hypothesis class of interval functions $F = \{f_{a,b} : a, b \in \mathbb{R}\}$ on $Z = \mathbb{R}$, where $f_{a,b}(z) = 1$ if $z \in [a, b]$ and $f(z) = -1$ otherwise. It is easy to see that any pair of distinct points can be shattered by F. However, for any triple of distinct points (z_1, z_2, z_3) with $z_1 < z_2 < z_3$, it is not possible to realize the behavior $(+1, -1, +1)$. So the VC dimension of F is two.

Finally, the class of linear threshold functions in $\mathbb{R}^d$ (defined in Section 2.1.3) has VC dimension $d + 1$. This follows from a result in convex geometry called Radon's theorem.

When the VC dimension of F is finite, then $\Pi_F(n)$ is bounded by a polynomial in n. This is the content of the *Sauer-Shelah lemma*.

Lemma 2.2 (Sauer-Shelah)**.** *If a function class F on domain Z has finite VC dimension $d < \infty$, then*

$$\Pi_F(n) \leq \sum_{i=0}^{d} \binom{n}{i} \leq (n+1)^d.$$

Proof. The proof is by induction on $d + n$. If $d = 0$ and $n = 1$, then a single point is not shattered by F, and hence $\Pi_F(1) = 1 = \binom{1}{0}$. If $d = 1$ and $n = 1$, then there is a single point shattered by F, and hence $\Pi_F(1) = 2 = \binom{1}{0} + \binom{1}{1}$. Now fix $n \geq 2$ and $d \geq 0$, and assume the claim holds for $\Pi_{F'}(n')$ where F' has VC dimension d', as long as $d' + n' \leq d + n - 1$. Consider any F with VC dimension d. Take $z_1, \ldots, z_n \in Z$ such that $|F_{|z_{1:n}}| = \Pi_F(n)$. Define

$$F' := \{f \in F : \exists f' \in F \,.\, f(z_i) = f'(z_i), 1 \leq i \leq n-1, f(z_n) \neq f'(z_n)\}$$

to be the set of functions f in F for which there exists another function f' that agrees with f on the first $n-1$ points but disagrees on the n-th point. Observe that

$$\Pi_F(n) = |F_{|z_{1:n}}| = |F_{|z_{1:n-1}}| + |F'_{|z_{1:n-1}}|.$$

Moreover, the VC dimension of F' (regarded as a function class on domain $z_{1:n}$) is at most $d-1$. Indeed, suppose for sake of contradiction that F' has VC dimension d. Then there is a set of d points, say, $z_1^*, \ldots, z_d^*$ among $z_{1:n}$ shattered by F'; now we can exhibit a set of $d+1$ points $z_1^*, \ldots, z_d^*, z_n$ shattered by F. This contradicts the assumption that F has VC dimension d. So, by the induction hypothesis,

$$|F_{|z_{1:n-1}}| \le \Pi_F(n-1) \le \sum_{i=0}^{d} \binom{n-1}{i} \quad \text{and} \quad |F'_{|z_{1:n-1}}| \le \Pi_{F'}(n-1) \le \sum_{i=0}^{d-1} \binom{n-1}{i},$$

so

$$\Pi_F(n) \le \sum_{i=0}^{d} \binom{n-1}{i} + \sum_{i=0}^{d-1} \binom{n-1}{i} = \binom{n}{0} + \sum_{i=1}^{d} \binom{n-1}{i} + \binom{n-1}{i-1} = \sum_{i=0}^{d} \binom{n}{i},$$

where we use the fact $\binom{n}{i} = \binom{n-1}{i} + \binom{n-1}{i-1}$. □

Lemma 2.2 may be used with the following bound on the Rademacher complexity in terms of $\Pi_F(n)$.

Lemma 2.3. *If F is a class of $\{\pm 1\}$-valued functions on domain Z with VC dimension d, then*

$$\mathrm{Rad}(F|_{z_{1:n}}) \le \sqrt{\frac{2\ln(\Pi_F(n))}{n}}$$

for any $z_1, \ldots, z_n \in Z$.

Lemma 2.3 is a special case of a more general lemma.

Lemma 2.4. *Let $S \subset \mathbb{R}^n$ be a finite set of vectors in $\mathbb{R}^n$. Then*

$$\mathrm{Rad}(S) \le \sqrt{\frac{2\max_{v\in S}\langle v, v\rangle_n \ln|S|}{n}}.$$

Proof. Let $\sigma := (\sigma_1, \ldots, \sigma_n)$ be a random vector where $\sigma_1, \ldots, \sigma_n$ are iid Rademacher random variables. Observe that by Lemma 2.1 and independence, we have

$$\mathbb{E}_\sigma \exp(\langle v, \sigma\rangle_n) \le \exp(\langle v, v\rangle_n/(2n)), \quad v \in \mathbb{R}^n. \tag{2.8}$$

Now, we follow the proof of Theorem 2.9. Fix $\lambda > 0$, and write

$$\mathrm{Rad}(S) = \mathbb{E}_\sigma \max_{v\in S}\langle v, \sigma\rangle_n = \frac{1}{\lambda}\ln\exp\left(\mathbb{E}_\sigma \max_{v\in S}\langle \lambda v, \sigma\rangle_n\right).$$

By Jensen's inequality and monotonicity,

$$\exp\left(\mathbb{E}_\sigma \max_{v\in S}\langle \lambda v, \sigma\rangle_n\right) \le \mathbb{E}_\sigma \exp\left(\max_{v\in S}\langle \lambda v, \sigma\rangle_n\right) \le \mathbb{E}_\sigma \sum_{v\in S}\exp\left(\langle \lambda v, \sigma\rangle_n\right).$$

Using linearity of expectation and Equation (2.8), we have

$$\mathbb{E}_\sigma \sum_{v\in S}\exp\left(\langle \lambda v, \sigma\rangle_n\right) \le |S|\exp(\lambda^2 \max_{v\in S}\langle v, v\rangle_n/(2n)).$$

So, we have established

$$\mathrm{Rad}(S) \le \frac{\ln|S|}{\lambda} + \frac{\lambda \max_{v\in S}\langle v, v\rangle_n}{2n}.$$

for any $\lambda > 0$. Choosing λ to minimize the bound yields the claimed bound. □

Putting Lemmas 2.2 and 2.3 and Theorems 2.10 and 2.11 together gives the following theorem.

Theorem 2.13. *Let P be any probability distribution, and let P_n be the empirical distribution for an iid sample from P of size n. For any set F of $\{\pm 1\}$-valued functions with VC dimension d,*

$$\mathbb{E}\left(\sup_{f\in F}\mathbb{E}_P f - \mathbb{E}_{P_n} f\right) \le \sqrt{\frac{8d\ln(n+1)}{n}}.$$

Also, for any $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\sup_{f\in F}\mathbb{E}_P f - \mathbb{E}_{P_n} f \le \sqrt{\frac{8d\ln(n+1)}{n}} + \sqrt{\frac{2\ln(1/\delta)}{n}}.$$

In Theorem 2.13, the rate as a function of n is $\sqrt{(\log n)/n}$, which is not tight. Using a technique called "chaining", it can be improved to $\sqrt{1/n}$.

2.2.8 Relative deviations

Using similar symmetrization arguments from Section 2.2.6, it is possible to prove relative deviation bounds that hold simultaneously for all $f \in F$.

Theorem 2.14. *Let F be a class of $\{0,1\}$-valued functions with VC dimension d. For any $\delta \in (0,1)$, with probability at least $1-\delta$,*

D: Prove this. Two ways: (i) symmetrization (sharper), (ii) Talagrand concentration (looser?).

$$\mathbb{E}_P f - \mathbb{E}_{P_n} f \le 2\sqrt{\mathbb{E}_P f \cdot \frac{d\ln(2n+1)+\ln(4/\delta)}{n}}, \quad f \in F.$$

Also, with probability at least $1-\delta$,

$$\mathbb{E}_{P_n} f - \mathbb{E}_P f \le 2\sqrt{\mathbb{E}_{P_n} f \cdot \frac{d\ln(2n+1)+\ln(4/\delta)}{n}}, \quad f \in F.$$

In Theorem 2.14, the $\mathbb{E}_P f$ appearing under the square-root should be regarded as an upper-bound on the variance of a Bernoulli random variable (since $p(1-p) \leq p$ for $p \in [0,1]$). The smaller the variance is, the smaller the deviation is.

If non-negative numbers A, B, and C satisfy $A \leq B\sqrt{A} + C$, then it follows that $A \leq B^2 + B\sqrt{C} + C$. Using this fact, we have the following corollary.

Corollary 2.1. *Let F be a class of $\{0,1\}$-valued functions with VC dimension d. For any $\delta \in (0,1)$, define*

$$\varepsilon_{F,n,\delta} := \frac{4\ln(\Pi_F(2n)) + 4\ln(4/\delta)}{n} \leq \frac{4d\ln(2n+1) + 4\ln(4/\delta)}{n}.$$

With probability at least $1-\delta$,

$$\mathbb{E}_P f - \mathbb{E}_{P_n} f \leq \min\left\{\sqrt{\mathbb{E}_P f \cdot \varepsilon_{F,n,\delta}}, \sqrt{\mathbb{E}_{P_n} f \cdot \varepsilon_{F,n,\delta}} + \varepsilon_{F,n,\delta}\right\}, \quad f \in F.$$

Also, with probability at least $1-\delta$,

$$\mathbb{E}_{P_n} f - \mathbb{E}_P f \leq \min\left\{\sqrt{\mathbb{E}_{P_n} f \cdot \varepsilon_{F,n,\delta}}, \sqrt{\mathbb{E}_P f \cdot \varepsilon_{F,n,\delta}} + \varepsilon_{F,n,\delta}\right\}, \quad f \in F.$$

Now consider a hypothesis class H with finite VC dimension. (Here, we only consider binary-valued hypotheses.) We apply Corollary 2.1 to the class $L_H := \{\ell_h : h \in H\}$, where $\ell_h(x,y) = 1_{\{h(x) \neq y\}}$, which has the same VC dimension as H. One implication is the following.

Theorem 2.15 (Consistent Hypothesis error rate bound in terms of VC dimension)**.** *Let H be a hypothesis class with VC dimension d. In the iid setting, if a learner uses Algorithm 13 (Consistent Hypothesis), then for any n and $\delta \in (0,1)$, with probability at least $1-\delta$, every hypothesis $h \in V_n$ has error rate*

$$\mathrm{err}_P(h) \leq \frac{4d\ln(2n+1) + 4\ln(4/\delta)}{n}.$$

This is comparable to Theorem 2.8, which only applies to finite hypothesis classes. The rate, as a function of n, is slightly worse: $(\log n)/n$ rather than $1/n$. In some cases, the theorem is loose, and indeed a $1/n$ rate is correct.

Theorem 2.16 (ERM error rate bound in terms of VC dimension)**.** *Let H be a hypothesis class with VC dimension d. In the agnostic iid setting, for any n and $\delta \in (0,1)$, the following hold with probability at least $1-\delta$:*

$$\mathrm{err}_P(h) - \mathrm{err}_{P_n}(h) \leq \min\left\{\sqrt{\mathrm{err}_P(h) \cdot \varepsilon_{H,n,\delta}}, \sqrt{\mathrm{err}_{P_n}(h) \cdot \varepsilon_{H,n,\delta}} + \varepsilon_{H,n,\delta}\right\}, \quad h \in H,$$

and

$$\mathrm{err}_{P_n}(h) - \mathrm{err}_P(h) \leq \min\left\{\sqrt{\mathrm{err}_{P_n}(h) \cdot \varepsilon_{H,n,\delta}}, \sqrt{\mathrm{err}_P(h) \cdot \varepsilon_{H,n,\delta}} + \varepsilon_{H,n,\delta}\right\}, \quad h \in H,$$

where

$$\varepsilon_{H,n,\delta} := \frac{4\ln(\Pi_H(2n)) + 4\ln(8/\delta)}{n} \leq \frac{4d\ln(2n+1) + 4\ln(8/\delta)}{n}.$$

Also, in the same event, the hypothesis h_n picked by a learner using the ERM algorithm (Algorithm 21) in round n satisfies

$$\mathrm{Reg}_P(h_n, H) \leq C_1 \cdot \sqrt{\inf_{h\in H} \mathrm{err}_P(h) \cdot \varepsilon_{H,n,\delta}} + C_2 \cdot \varepsilon_{H,n,\delta},$$

where C_1 and C_2 are absolute constants (and it suffices to take $C_1 := 1+\sqrt{3/2}$ and $C_2 := 2+\sqrt{3/2}$).

Exercise 2.7. Prove the claim in Theorem 2.16 regarding $\mathrm{Reg}_P(h_n, H)$.

2.3 Selective sampling

2.3.1 The setting

In many practical scenarios where learning from examples takes place, the unlabeled parts (x) are freely available, but the labels (y) are expensive. Motivated by this disparity in the costs of data, we consider a modification of the online classification scenario where the learner must decide in each round whether or not to obtain the label. This modification is called *selective sampling* (or *active learning*).

Algorithm 22 Protocol for online classification under selective sampling

1: **for** $n = 1, 2, \ldots$ **do**
2: Teacher reveals object: $x_n \in \mathcal{X}$.
3: Learner makes prediction: $a_n \in \{0, 1\}$.
4: **if** Learner requests the correct answer (label) **then**
5: Teacher reveals correct answer (label): $y_n \in \{0, 1\}$.
6: **end if**
7: **end for**

In addition to the usual goals in online classification, the learner also seeks to minimize the number of labels requested.

2.3.2 Threshold functions

We first consider, as a toy example, selective sampling for the class of threshold functions $H = \{h_b : b \in \mathbb{R}\}$ on $\mathbb{R}$, where $h_b(x) = 1_{\{x>b\}}$. We assume that the target concept is also a threshold function $h_{b^*} \in H$ for some $b^* \in \mathbb{R}$.

This hypothesis class H has VC dimension one, so in the realizable iid setting, for any $\varepsilon, \delta \in (0, 1)$, a learner can find a hypothesis $h \in H$ with error rate at most ε after

Algorithm 23 Selective sampling algorithm for threshold functions

Let $b_l := -\infty$, $b_u := +\infty$.
for $n = 1, 2, \ldots$ **do**
  Receive object: $x_n \in \mathbb{R}$.
  Make prediction: choose any $b \in [b_l, b_u)$ and set $a_n := h_b(x_n)$.
  if $x_n \in (b_l, b_u]$ **then**
    Request label: $y_n \in \{0, 1\}$.
    if $y_n = 0$ **then** $b_l := x_n$. **else** $b_u := x_n$. **end if**
  end if
end for

$O((1/\varepsilon)\log(1/(\varepsilon\delta)))$ rounds. This, however, assumes that a label is requested in every round. How can a learner achieve the same error rate but using fewer labels?

A simple strategy is shown in Algorithm 23. It maintains lower- and upper-limits for the target threshold. A simple induction argument proves the following.

Theorem 2.17. *In the realizable setting, the quantities b_l and b_u maintained by a learner using Algorithm 23 satisfy the following properties:*

- $b^* \in [b_l, b_u]$;
- *every threshold function h_b with $b \in [b_l, b_u]$ at the start of round n is consistent with all examples $(x_1, h_{b^*}(x_1)), \ldots, (x_{n-1}, h_{b^*}(x_{n-1}))$ from previous rounds, including rounds in which a label is not requested.*

Therefore, after n rounds, the learner may choose any threshold function h_b with $b \in [b_l, b_u]$ and be assured that, with probability at least $1 - \delta$,

$$\mathrm{err}_P(h_b) \leq \frac{4\ln(2n+1) + 4\ln(4/\delta)}{n}$$

as per Theorem 2.15.

What remains is to understand how many labels a learner using Algorithm 23 ends up requesting after several rounds. For simplicity, we assume the marginal of P over $\mathcal{X} = \mathbb{R}$ is a continuous density, so the probability mass assigned to an interval $[a, b] \subset \mathbb{R}$ is the same whether or not we include either endpoint. Let Q_n be the $\{0, 1\}$-valued random variable with $Q_n = 1$ if and only if the learner using Algorithm 23 requests the label in round n. Observe that $Q_n = 1_{\{x_n \in (b_l, b_u]\}}$ for the values of b_l, b_u at the start of round n. In particular, if $x_n \in (b_l, b_u)$, then some hypothesis consistent with all previous examples $(x_1, h_{b^*}(x_1)), \ldots, (x_{n-1}, h_{b^*}(x_{n-1}))$ makes a mistake on x_n. Since H has VC dimension one, Theorem 2.15 implies that in the realizable iid setting, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$ (over the choices of the first $n - 1$ examples), every consistent hypothesis has error rate at most

$$\varepsilon_{n-1,\delta} := \frac{4\ln(2n-1) + 4\ln(4/\delta)}{n-1}.$$

This implies that there is an event E_n of probability at least $1 - \delta$ (over the choices of the first $n - 1$ examples) such that $\Pr(x_n \in (b_l, b_u) \mid E_n) \leq \varepsilon_{n-1,\delta}$. Therefore, in this setting,

$$\Pr(Q_n = 1) \leq \min\{1, \varepsilon_{n-1,\delta} + \delta\}.$$

Since this holds for any $\delta \in (0, 1)$, we choose δ to minimize the bound. So the expected number of labels requested after n rounds can be obtained by summing up the $\Pr(Q_i = 1)$ for $i = 1, \ldots, n$.

Theorem 2.18. *In the realizable iid setting, the expected number of labels requested by a learner using Algorithm 23 after n rounds is*

$$O\left(\log^2 n\right).$$

This is a dramatic improvement over requesting all n labels after n rounds.

2.3.3 Selective Consistent Hypothesis algorithm

We now give a generalization of Algorithm 23 that works with an arbitrary hypothesis class H. This algorithm is called the *Selective Consistent Hypothesis* algorithm (Algorithm 24). For simplicity, we only consider hypothesis classes of $\{\pm 1\}$-valued functions (i.e., binary classification).

Algorithm 24 depends on an algorithm $\mathcal{A}$ that finds a hypothesis in H consistent with a given set of labeled examples (much like the one for linear threshold functions described in Section 2.1.3); it returns $\perp$ if no such hypothesis in H exists.

Algorithm 24 maintains a set of labeled examples S and a current hypothesis h. In each round n, the algorithm $\mathcal{A}$ is used to check if there is a hypothesis h' that agrees with h on all examples in S but disagrees with h on a new point x_n. If no such h' exists, then the label of x_n is not needed.

Theorem 2.19 (Selective Consistent Hypothesis)**.** *In the realizable setting (with $h^* \in H$ being the target concept), the set of labeled examples S maintained by a learner using Algorithm 24 (Selective Consistent Hypothesis) satisfies the following properties:*

1. *h^* is consistent with all labeled examples in S at all times;*
2. *every $h \in H$ consistent with all labeled examples in S at the start of round t is consistent with all examples $(x_1, h^*(x_1)), \ldots, (x_{n-1}, h^*(x_{n-1}))$ from previous rounds, including rounds in which a label is not requested.*

Proof. The first property is true by definition (since we are considering the realizable setting). For the second property, observe that a learner using Algorithm 24 only forgoes a label y_i if every hypothesis in H consistent with S (at the start of round i) agrees on the label of x_i; since this includes h^*, it means that every such hypothesis is consistent with $(x_i, h^*(x_i))$. □

Algorithm 24 Selective Consistent Hypothesis algorithm

Require: $\mathcal{A}$, algorithm for finding hypothesis in H consistent with a given set of labeled examples.

Let $S := \emptyset$, and let $h := \mathcal{A}(S)$.
for $n = 1, 2, \ldots$ **do**
  Receive object: $x_n \in \mathcal{X}$.
  Make prediction: $a_n := h(x_n)$.
  Let $h' := \mathcal{A}(S \cup \{(x_n, -a_n)\})$.
  if $h' \neq \perp$ **then**
    Request label: $y_n \in \mathcal{Y}$.
    Let $S := S \cup \{(x_n, y_n)\}$.
    if $y_n \neq a_n$ **then** $h := h'$. **end if**
  end if
end for

Theorem 2.19 shows that the Selective Consistent Hypothesis algorithm (Algorithm 24) behaves like the Consistent Hypothesis algorithm (Algorithm 13). Thus, in the realizable iid setting, if H has VC dimension d, then for any $\varepsilon, \delta \in (0, 1)$, the learner using Algorithm 24 finds a hypothesis $h \in H$ with error rate at most ε after

$$O\left(\frac{d \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}\right)$$

rounds.

To analyze how many labels are requested by a learner using Algorithm 24, we consider the metric structure over the hypothesis class H arising from the marginal distribution $P_{\mathcal{X}}$ of P over $\mathcal{X}$. Specifically, define the pseudometric ρ over H by

$$\rho(h, h') := P_{\mathcal{X}}(\{x \in \mathcal{X} : h(x) \neq h'(x)\}), \quad h, h' \in H.$$

The pseudometric ρ is non-negative, symmetric, and satisfies the triangle inequality. (It is technically not a metric because it is possible that $\rho(h, h') = 0$ even though $h \neq h'$.) A ρ-ball of radius $r \geq 0$ around some $h \in H$ is denoted by

$$B(h, r) := \{h' \in H : \rho(h, h') \leq r\}.$$

Define the *disagreement region* $D(h, V)$ for $V \subseteq H$ around $h \in H$ by

$$D(h, V) := \{x \in \mathcal{X} : \exists h' \in V \,.\, h(x) \neq h'(x)\}.$$

For shorthand, we write $D(h, r) := D(h, B(h, r))$. Finally, we define *Alexander's capacity function* τ by

$$\tau(h, r) := \frac{P_{\mathcal{X}}(D(h, r))}{r}, \quad h \in H, r > 0.$$

The supremum of $\tau(h^*, r)$ over $r > 0$ is called the *disagreement coefficient*.

Exercise 2.8. Verify the non-negativity, symmetry, and triangle inequality for the pseudo-metric ρ.

Exercise 2.9. Let $P_{\mathcal{X}}$ be the uniform distribution on the interval $[0,1] \subset \mathbb{R}$, and let H be the class of threshold functions (c.f. Section 2.2.7). Show that $\tau(h,r) \leq 2$ for all $h \in H$ and $r > 0$.

Exercise 2.10. Let $P_{\mathcal{X}}$ be the uniform distribution on the interval $[0,1] \subset \mathbb{R}$, and let H be the class of interval functions (c.f. Section 2.2.7). Define $p(h) := P_{\mathcal{X}}(\{x \in [0,1] : h(x) = 1\})$ for $h \in H$. Show that $\tau(h,r) \leq \max\{4, 1/p(h)\}$ for all $h \in H$ and $r > 0$.

Proposition 2.1. *Let $P_{\mathcal{X}}$ be the uniform distribution on the unit sphere $\mathcal{S}^{d-1} := \{x \in \mathbb{R}^d : \|x\|_2 = 1\} \subset \mathbb{R}^d$, and let H be the class of homogeneous linear threshold functions in $\mathbb{R}^d$ (c.f. Section 2.1.3). There is an absolute constant $C > 0$ such that*

$$\tau(h,r) \leq C \cdot \sqrt{d}, \quad h \in H, r > 0.$$

Proof. Every homogeneous linear threshold function may be represented by a weight vector in the unit sphere $\mathcal{S}^{d-1}$. Indeed, we have

$$\rho(w,w') = \frac{\cos^{-1}(\langle w, w' \rangle)}{\pi}, \quad w, w' \in \mathcal{S}^{d-1}.$$

(We abuse notation and identify each $h_w \in H$ by its weight vector $w \in \mathcal{S}^{d-1}$.) This implies

$$\begin{aligned} B(w,r) &= \{w' \in \mathcal{S}^{d-1} : \langle w, w' \rangle \geq \cos(\pi r)\}, \\ D(w,r) &= \{x \in \mathcal{S}^{d-1} : \langle w, x \rangle^2 \leq \sin^2(\pi r)\}, \quad w \in \mathcal{S}^{d-1}, r \geq 0. \end{aligned}$$

Fixed $w \in \mathcal{S}^{d-1}$, and consider a uniformly random unit vector x. It is well-known that $\langle w, x \rangle^2$ has a Beta$(1/2, (d-1)/2)$ distribution. So $\tau(w,r)$ can be computed directly from the Beta$(1/2, (d-1)/2)$ distribution function.

A coarse bound can also be obtained by using the fact that the distribution of $\langle w, x \rangle^2$ is the same as that of

$$\frac{z_1^2}{z_1^2 + \cdots + z_d^2},$$

where $z_1, \ldots, z_d$ are iid standard normal random variables. Therefore, assuming $d > 1$ and $r < 1/(\pi\sqrt{d})$,

$$\Pr\left(\frac{z_1^2}{z_1^2 + \cdots + z_d^2} \leq \sin^2(\pi r)\right) \leq \pi r \sqrt{d} \left(1 + \frac{1 - \pi^2 r^2 d}{d-1}\right)^{\frac{d-1}{2}}$$

by a Chernoff-type bound. D: Prove this. □

Define V_{n-1} to be the subset of hypotheses consistent with S at the start of round n. We know from Theorem 2.15 that for any $\delta \in (0,1)$, with probability at least $1-\delta$, we have

$$V_{n-1} \subseteq B(h^*, \varepsilon_{H,n-1,\delta}),$$

where

$$\varepsilon_{H,n-1,\delta} := \frac{4\ln(\Pi_H(2(n-1))) + 4\ln(8/\delta)}{n-1} \leq \frac{4d\ln(2n-1) + 4\ln(4/\delta)}{n-1}.$$

Conditional on this event (which we call E_n), if a label is requested in round n (i.e., Step 7 is executed in round n), then there exist $h, h' \in V_{n-1} \subseteq B(h^*, \varepsilon_{H,n-1,\delta})$ such that $h(x_n) \neq h'(x_n)$; one of these two hypotheses disagrees with h^*, and thus

$$x_n \in D(h^*, \varepsilon_{H,n-1,\delta}).$$

Therefore, the probability of a label request in round n is at most

$$\Pr(x_n \in D(h^*, \varepsilon_{H,n-1,\delta}) \mid E_n) + (1 - \Pr(E_n)) \leq \tau(h^*, \varepsilon_{H,n-1,\delta}) \cdot \varepsilon_{H,n-1,\delta} + \delta.$$

To obtain a bound on the expected number of label requests, we bound $\tau(h^*, \varepsilon_{H,n-1,\delta})$ by $\sup_{r>0} \tau(h^*, r)$, and sum (the bounds on) these probabilities. This is potentially loose, but at least applicable in certain cases of interest (e.g., Proposition 2.1).

Theorem 2.20 (Selective Consistent Hypothesis label request bound)**.** *In the realizable iid setting, the expected number of labels requested by a learner using Algorithm 24 (Selective Consistent Hypothesis) after n rounds is at most*

$$O\left(\sup_{r>0} \tau(h^*, r) \cdot d\log^2 n\right),$$

where d is the VC dimension of the hypothesis class H, and $h^* \in H$ is the target concept.

2.3.4 Selective Empirical Risk Minimization algorithm

Algorithm 24 only makes sense in the realizable setting (like Algorithm 13), because there may be no hypotheses in H consistent with $(x_1, y_1), \ldots, (x_n, y_n)$. To handle the agnostic setting, we look to ERM. Again, we only consider $\{\pm 1\}$-valued hypotheses.

A preliminary version of a *Selective ERM* algorithm (Algorithm 25) depends an ERM-like algorithm $\mathcal{A}$ with two operating modes[5]:

1. Given labeled examples S, it returns an ERM hypothesis $h \in \arg\min_{h' \in H} \mathrm{err}_S(h')$.
2. Given labeled examples S and a distinguished example (x', y'), it returns a minimizer of the empirical risk on S subject to the constraint $h(x') \neq y'$. In other words, it returns an ERM hypothesis for the constrained hypothesis class $\{h' \in H : h'(x') \neq y'\}$.

[5]In fact, only the second mode of $\mathcal{A}$ is needed, as it can be used to implement the first mode.

For a sequence of labeled examples S and hypothesis h, we use $\mathrm{err}_S(h)$ to denote the empirical risk of h on the examples in S. (If S is empty, then $\mathrm{err}_S(h) = 0$.)

In each round n, Algorithm 25 attempts to determine the label of x_n assigned by a risk minimizing hypothesis $h^* \in \arg\min_{h \in H} \mathrm{err}_P(h)$. If it cannot determine this label, it requests the label y_n. Otherwise, it uses a *predicted label* a_n for x_n.

Algorithm 25 Selective Empirical Risk Minimization algorithm, preliminary version

Require: $\mathcal{A}$, an ERM-like algorithm as described at the beginning of Section 2.3.4; $\delta \in (0, 1)$.

1: Let $S_0 := \emptyset$, $h_0 := \mathcal{A}(S_0)$, $\beta_0 := \infty$.
2: **for** $n = 1, 2, \ldots$ **do**
3: Receive object: $x_n \in \mathcal{X}$.
4: Make prediction: $a_n := h_{n-1}(x_n)$.
5: Let $h'_{n-1} := \mathcal{A}(S_{n-1}; (x_n, a_n))$.
6: **if** $\mathrm{err}_{S_{n-1}}(h'_{n-1}) - \mathrm{err}_{S_{n-1}}(h_{n-1}) \leq 2\beta_{n-1}$ **then**
7: Request label: $y_n \in \mathcal{Y}$.
8: Let $S_n := S_{n-1} \cup \{(x_n, y_n)\}$.
9: **else**
10: Let $S_n := S_{n-1} \cup \{(x_n, a_n)\}$.
11: **end if**
12: Let $h_n := \mathcal{A}(S_n)$, $\beta_n := \mathrm{Rad}_n(H \cup -H) + \sqrt{\frac{\ln(n(n+1)/\delta)}{2n}}$.
13: **end for**

The idea behind the rule in Step 7 of Algorithm 25 is as follows. Suppose h_{n-1} is the ERM hypothesis based on S_{n-1}, and $h'_{n-1} = \mathcal{A}(S_{n-1}, (x_n, h_{n-1}(x_n)))$, so the hypothesis h'_{n-1} is forced to disagree with h_{n-1} on x_n. Note that the empirical risk of h'_{n-1} on S_{n-1} is at least that of h_{n-1} (by definition of h_{n-1}). If, however, the empirical risk of h'_{n-1} is *much higher* than that of h_{n-1}, then one can conclude that h^* cannot be in the set $\{h \in H : h(x_n) \neq h_{n-1}(x_n)\}$—i.e., we can conclude $h^*(x_n) = h_{n-1}(x_n)$.

The quantities β_n in Algorithm 25 are used to determine how large $\mathrm{err}_{S_{n-1}}(h'_{n-1}) - \mathrm{err}_{S_{n-1}}(h_{n-1})$ must be to forgo the label y_n. They are derived from the maximum deviation bounds in Theorem 2.12, with δ replaced by $\delta/(n(n+1))$ so that the bounds hold for all n with probability at least $1 - \delta$.[6]

The maximum deviation bounds from Theorem 2.12 are for iid samples; however, on account of the predicted labels a_i, which are used whenever the y_i are not requested, the labeled examples in S_n are generally not iid. Indeed, even for a fixed hypothesis h, $\mathrm{err}_{S_n}(h)$ is not necessarily an unbiased estimator for $\mathrm{err}_P(h)$ (unlike $\mathrm{err}_{P_n}(h)$, which is unbiased).

However, if every predicted label a_i in S_n is equal to $h^*(x_i)$, then hypotheses that agree with h^* on examples with predicted labels can only appear better. This is formalized in the

[6]The dependence on H and δ is suppressed in β_n. Also, in place $\mathrm{Rad}_n(H \cup -H)$ in β_n, one may instead use the empirical bound based on Equation (2.7) (which hold with high probability).

following lemma.

Lemma 2.5 (Favorable bias)**.** *If $a_i = h^*(x_i)$ for every example in S_n where a_i is used in place of y_i, then, for any hypothesis $h : \mathcal{X} \to \{\pm 1\}$:*

1. $\mathrm{err}_{S_n}(h) - \mathrm{err}_{S_n}(h^*) \geq \mathrm{err}_{P_n}(h) - \mathrm{err}_{P_n}(h^*)$;
2. $\mathrm{err}_{S_n}(h) \leq \mathrm{err}_{P_n}(h) + \mathrm{err}_{P_n}(h^*)$;
3. $\mathrm{err}_{S_n}(h^*) \leq \mathrm{err}_{P_n}(h^*)$.

Proof. Recall that each of the empirical risk differences ($\mathrm{err}_{S_n}(h) - \mathrm{err}_{S_n}(h^*)$ and $\mathrm{err}_{P_n}(h) - \mathrm{err}_{P_n}(h^*)$) is an average of values from $\{-1, 0, +1\}$. Consider any $i \in [n]$ where a_i is used in place of y_i in S_n. Now consider any hypothesis h. If $h(x_i) = h^*(x_i)$, then the i-th example has no contribution to either empirical risk differences. If $h(x_i) \neq h^*(x_i)$, then the i-th example has a positive contribution to $\mathrm{err}_{S_n}(h) - \mathrm{err}_{S_n}(h^*)$, which is at least the contribution to $\mathrm{err}_{P_n}(h) - \mathrm{err}_{P_n}(h^*)$. This proves the first inequality. The other inequalities are proved similarly. □

This means that, when compared relative to h^*, hypotheses that disagree with h^* on examples with predicted labels appear worse under err_{S_n} than they do under err_{P_n}. Thus, even though $\mathrm{err}_{S_n}(h)$ is biased (as an estimator for $\mathrm{err}_P(h)$), the bias is favorable in that it steers ERM towards h^*, which is desirable.

For $n \geq 0$, let E_n be the event that

$$|\mathrm{err}_P(h) - \mathrm{err}_{P_n}(h)| \leq \beta_n, \quad h \in H,$$

where β_n is defined in Algorithm 25. The inequality is trivial for $n = 0$, and by Theorem 2.12 and a union bound, we have $\Pr(\cap_{n \geq 0} E_n) \geq 1 - \delta$.

We now give a theorem that includes the risk bounds for Algorithm 25.

Theorem 2.21 (Selective ERM, preliminary version)**.** *In the agnostic iid setting (with any fixed $h^* \in \arg\min_{h \in H} \mathrm{err}_P(h)$), for any $n \geq 1$, the labeled examples and hypotheses maintained by a learner using Algorithm 25 (Selective ERM, preliminary version) through round n satisfy the following, assuming $\cap_{m=0}^{n-1} E_m$ holds.*

1. *For all $m \in \{0, \ldots, n-1\}$, if $\mathrm{err}_{S_m}(h'_m) - \mathrm{err}_{S_m}(h_m) > 2\beta_m$, then $a_{m+1} = h^*(x_{m+1})$.*
2. *For all $m \in \{0, \ldots, n-1\}$, $\mathrm{Reg}(h_m, H) \leq 2\beta_m$.*

Proof. We prove that the first claim holds for all $n \geq 1$ by induction; the second claim is established later. The base case ($n = 1$) is trivial because $\beta_0 = \infty$. So fix $n \geq 1$, and assume as the inductive hypothesis that the first claim holds, as well as E_n. This implies that $a_m = h^*(x_m)$ for every example in S_n where a_m is used in place of y_m. Thus, by Lemma 2.5 and the assumption of event E_n, we have

$$\mathrm{err}_{S_n}(h^*) - \mathrm{err}_{S_n}(h_n) \leq \mathrm{err}_{P_n}(h^*) - \mathrm{err}_{P_n}(h_n) \leq \mathrm{err}_P(h^*) - \mathrm{err}_P(h_n) + 2\beta_n \leq 2\beta_n. \quad (2.9)$$

Now suppose $a_{n+1} \neq h^*(x_{n+1})$. Then $h'_n(x_{n+1}) = h^*(x_{n+1})$, and by definition of h'_n and Lemma 2.5, and Equation (2.9)

$$\mathrm{err}_{S_n}(h'_n) - \mathrm{err}_{S_n}(h_n) \leq \mathrm{err}_{S_n}(h^*) - \mathrm{err}_{S_n}(h_n) \leq 2\beta_n.$$

This completes the inductive proof.

Now we prove the second claim. The first claim of the present theorem implies the preconditions of Lemma 2.5. Thus, for any $m \in \{0, \ldots, n-1\}$, using Lemma 2.5 and the properties of ERM on the event E_m, we have

$$\begin{aligned}
\mathrm{Reg}(h_m, H) &= \mathrm{err}_P(h_m) - \mathrm{err}_P(h^*) \\
&\leq \mathrm{err}_{P_m}(h_m) - \mathrm{err}_{P_m}(h^*) + 2\beta_m \\
&\leq \mathrm{err}_{S_m}(h_m) - \mathrm{err}_{S_m}(h^*) + 2\beta_m \\
&\leq 2\beta_m.
\end{aligned}$$

□

The second claim in Theorem 2.21 is the essentially same as the ERM guarantee from Theorem 2.12, except for the replacing of δ by $\delta/(m(m+1))$ in β_m so that the guarantee holds for all $m \in \{0, \ldots, n-1\}$.

We now analyze the number of labels requested by a learner using Algorithm 25. The analysis follows that of Algorithm 24.

Lemma 2.6. *Fix any $n \geq 1$, and assume $\cap_{m=0}^{n-1} E_m$ holds. If $\mathrm{err}_{S_{n-1}}(h'_{n-1}) - \mathrm{err}_{S_{n-1}}(h_{n-1}) \leq 2\beta_{n-1}$, then $x_n \in D(h^*, 2\,\mathrm{err}_P(h^*) + 4\beta_{n-1})$.*

Proof. Assume $\cap_{m=0}^{n-1} E_m$ holds and $\mathrm{err}_{S_{n-1}}(h'_{n-1}) - \mathrm{err}_{S_{n-1}}(h_{n-1}) \leq 2\beta_{n-1}$. We first show that there exists a hypothesis $h \in H$ such that $h(x_n) \neq h^*(x_n)$ and $\mathrm{err}_P(h) \leq \mathrm{err}_P(h^*) + 4\beta_{n-1}$. There are two cases to consider: $h_{n-1}(x_n) = h^*(x_n)$ and $h_{n-1}(x_n) \neq h^*(x_n)$. Suppose first that $h_{n-1}(x_n) = h^*(x_n)$. Then we have

$$\begin{aligned}
\mathrm{err}_P(h'_{n-1}) - \mathrm{err}_P(h^*) &\leq \mathrm{err}_{P_{n-1}}(h'_{n-1}) - \mathrm{err}_{P_{n-1}}(h^*) + 2\beta_{n-1} \\
&\leq \mathrm{err}_{S_{n-1}}(h'_{n-1}) - \mathrm{err}_{S_{n-1}}(h^*) + 2\beta_{n-1} \\
&\leq \mathrm{err}_{S_{n-1}}(h'_{n-1}) - \mathrm{err}_{S_{n-1}}(h_{n-1}) + 2\beta_{n-1} \\
&\leq 4\beta_{n-1}.
\end{aligned}$$

Above, the first inequality uses the maximum deviation bounds that hold on E_{n-1}; the second inequality uses Lemma 2.5 (which is applicable on account of the first claim of Theorem 2.21); the third inequality uses the definition of h_{n-1}; the fourth inequality holds by assumption. Now instead suppose that $h_{n-1}(x_n) \neq h^*(x_n)$. Then by a similar argument as before,

$$\begin{aligned}
\mathrm{err}_P(h_{n-1}) - \mathrm{err}_P(h^*) &\leq \mathrm{err}_{P_{n-1}}(h_{n-1}) - \mathrm{err}_{P_{n-1}}(h^*) + 2\beta_{n-1} \\
&\leq \mathrm{err}_{S_{n-1}}(h_{n-1}) - \mathrm{err}_{S_{n-1}}(h^*) + 2\beta_{n-1} \\
&\leq 2\beta_{n-1}.
\end{aligned}$$

So let $h \in H$ be such that $h(x_n) \neq h^*(x_n)$ and $\mathrm{err}_P(h) \leq \mathrm{err}_P(h^*) + 4\beta_{n-1}$. By the triangle inequality,

$$\rho(h^*, h) \leq \mathrm{err}_P(h^*) + \mathrm{err}_P(h) \leq 2\,\mathrm{err}_P(h^*) + 4\beta_{n-1}.$$

Since $h(x_n) \neq h^*(x_n)$, it follows that $x_n \in D(h^*, 2\,\mathrm{err}_P(h^*) + 4\beta_{n-1})$. □

Lemma 2.6 implies that conditional on $\cap_{m=0}^{n-1} E_m$, the probability that a learner using Algorithm 25 requests the label y_n is at most $\tau(h^*, 2\,\mathrm{err}_P(h^*) + 4\beta_{n-1}) \cdot (2\,\mathrm{err}_P(h^*) + 4\beta_{n-1})$. To obtain a bound on the expected number of label requests, we use the fact that $\mathrm{Rad}_m(H \cup -H) = O(\sqrt{(d \log m)/m})$ when H has VC dimension d, and sum (the bounds on) these probabilities.

Theorem 2.22 (Selective ERM, preliminary version, label request bound)**.** *In the agnostic iid setting, the expected number of labels requested by a learner using Algorithm 25 (Selective ERM, preliminary version) after n rounds is at most*

$$\sup_{r>0} \tau(h^*, r) \cdot \left(2\,\mathrm{err}_P(h^*) \cdot n + O\left(\sqrt{dn \log n}\right)\right).$$

where d is the VC dimension of the hypothesis class H, and $h^* \in \arg\min_{h \in H} \mathrm{err}_P(h)$ is a fixed risk minimizer.

The bound on the expected number of label requests grows linearly with n when $\mathrm{err}_P(h^*) > 0$. This turns out to be unavoidable without further assumptions on the distribution P and hypothesis class H.

When $\mathrm{err}_P(h^*) = 0$, the bound from Theorem 2.22 is worse than the bound from Theorem 2.20 ($\sqrt{n \log n}$ vs. $\log^2 n$). This deficiency of Algorithm 25 comes from the use of the maximum deviation bound from Theorem 2.12. This can be rectified by replacing the β_n in Algorithm 25 with quantities based on relative deviation bounds, similar to those from Theorem 2.16. This replacement is non-trivial because the bounds depend on empirical risks, but $\mathrm{err}_{S_n}(h) \neq \mathrm{err}_{P_n}(h)$ in general.

2.3.5 Improved Selective Empirical Risk Minimization algorithm

We now give an improved version of Algorithm 25 as suggested at the end of Section 2.3.4. The improved *Selective ERM* algorithm is shown in Algorithm 26 (again, only for binary classification).

Algorithm 26 is based on the following relative deviation bound, which is slightly different from that in Theorem 2.16.

Theorem 2.23 (Error rate difference bounds)**.** *Let H be a hypothesis class. In the agnostic iid setting, for any n and $\delta \in (0, 1)$, the following hold with probability at least $1 - \delta$:*

$$\left|\left(\mathrm{err}_{P_n}(h) - \mathrm{err}_{P_n}(h^*)\right) - \left(\mathrm{err}_P(h) - \mathrm{err}_P(h^*)\right)\right| \leq 2\sqrt{\rho_\wedge(h, h^*) \cdot \varepsilon_{H,n,\delta}} + \varepsilon_{H,n,\delta}, \quad h \in H,$$

Algorithm 26 Selective Empirical Risk Minimization algorithm, improved version

Require: $\mathcal{A}$, an ERM-like algorithm as described at the beginning of Section 2.3.4; $\delta \in (0,1)$.

Let $S_0 := \emptyset$, $h_0 := \mathcal{A}(S_0)$, $\gamma_0(r) := \infty$ for $r \geq 0$.
for $n = 1, 2, \ldots$ **do**
  Receive object: $x_n \in \mathcal{X}$.
  Make prediction: $a_n := h_{n-1}(x_n)$.
  Let $h'_{n-1} := \mathcal{A}(S_{n-1}; (x_n, a_n))$.
  if $\mathrm{err}_{S_{n-1}}(h'_{n-1}) - \mathrm{err}_{S_{n-1}}(h_{n-1}) \leq \gamma_{n-1}(\mathrm{err}_{S_{n-1}}(h_{n-1}) + \mathrm{err}_{S_{n-1}}(h'_{n-1}))$ **then**
    Request label: $y_n \in \mathcal{Y}$.
    Let $S_n := S_{n-1} \cup \{(x_n, y_n)\}$.
  else
    Let $S_n := S_{n-1} \cup \{(x_n, a_n)\}$.
  end if
  Let $h_n := \mathcal{A}(S_n)$, $\gamma_n(r) := 2\sqrt{r \cdot \varepsilon_{H,n,\delta/(n(n+1))}} + \varepsilon_{H,n,\delta/(n(n+1))}$ for $r \geq 0$.
end for

where

$$\varepsilon_{H,n,\delta} := \frac{4\ln(\Pi_H(2n)) + 4\ln(8/\delta)}{n}, \qquad \rho_\wedge(h,h') := \min\{\rho(h,h'), \rho_n(h,h')\},$$

$$\rho(h,h') := P_{\mathcal{X}}(\{x \in \mathcal{X} : h(x) \neq h'(x)\}), \qquad \rho_n(h,h') := \frac{1}{n}\sum_{i=1}^{n} 1_{\{h(x_i) \neq h'(x_i)\}}.$$

Exercise 2.11. Use Theorem 2.16 to prove Theorem 2.23.

We also use the following bound on $\rho_n(h,h')$ in terms of empirical risks on S_n:

$$\rho_n(h,h') \leq \mathrm{err}_{S_n}(h) + \mathrm{err}_{S_n}(h'), \tag{2.10}$$

$$\rho(h,h') \leq \mathrm{err}_P(h) + \mathrm{err}_P(h'). \tag{2.11}$$

For $n \geq 0$, let E_n be the event that

$$\left(\mathrm{err}_{P_n}(h) - \mathrm{err}_{P_n}(h')\right) - \left(\mathrm{err}_P(h) - \mathrm{err}_P(h')\right) \leq \gamma_n(\rho_\wedge(h,h')), \quad h,h' \in H. \tag{2.12}$$

The inequality is trivial for $n = 0$, and by Theorem 2.23 and a union bound, we have $\Pr(\cap_{n\geq 0} E_n) \geq 1 - \delta$.

The following theorem is an analogue of Theorem 2.21 for Algorithm 26.

Theorem 2.24 (Selective ERM, improved version)**.** *In the agnostic iid setting (with any fixed $h^* \in \arg\min_{h\in H} \mathrm{err}_P(h)$), for any $n \geq 1$, the labeled examples and hypotheses maintained by a learner using Algorithm 25 (Selective ERM, improved version) through round n satisfy the following, assuming $\cap_{m=0}^{n-1} E_m$ holds.*

1. *For all* $m \in \{0, \ldots, n-1\}$, *if* $\mathrm{err}_{S_m}(h'_m) - \mathrm{err}_{S_m}(h_m) > \gamma_m(\mathrm{err}_{S_m}(h_m) + \mathrm{err}_{S_m}(h'_m))$, *then* $a_{m+1} = h^*(x_{m+1})$.

2. *For all* $m \in \{0, \ldots, n-1\}$, $\mathrm{Reg}(h_m, H) \leq \gamma_m(\rho_\wedge(h^*, h_m))$.

Proof. We prove that the first claim holds for all $n \geq 1$ by induction; the second claim is established later. The base case ($n = 1$) is trivial because $\gamma_0 \equiv \infty$. So fix $n \geq 2$, and assume as the inductive hypothesis that the first claim holds, as well as E_n. This implies that $a_m = h^*(x_m)$ for every example in S_n where a_m is used in place of y_m. Thus, by Lemma 2.5 and Equation (2.12), we have

$$\begin{aligned}\mathrm{err}_{S_n}(h^*) - \mathrm{err}_{S_n}(h_n) &\leq \mathrm{err}_{P_n}(h^*) - \mathrm{err}_{P_n}(h_n) \\ &\leq \mathrm{err}_P(h^*) - \mathrm{err}_P(h_n) + \gamma_n(\rho_\wedge(h^*, h_n)) \\ &\leq \gamma_n(\rho_\wedge(h^*, h_n)). \end{aligned} \tag{2.13}$$

Now suppose $a_{n+1} \neq h^*(x_{n+1})$. Then $h'_n(x_{n+1}) = h^*(x_{n+1})$, and hence $\mathrm{err}_{S_n}(h'_n) \leq \mathrm{err}_{S_n}(h^*)$. If $\mathrm{err}_{S_n}(h'_n) \leq \varepsilon_{H,n,\delta/(n(n+1))}$, then $\mathrm{err}_{S_n}(h'_n) - \mathrm{err}_{S_n}(h'_n) \leq \gamma_n(\mathrm{err}_{S_n}(h_n) + \mathrm{err}_{S_n}(h'_n))$. So assume instead that $\mathrm{err}_{S_n}(h'_n) > \varepsilon_{H,n,\delta/(n(n+1))}$. Define

$$F(x) := x - \mathrm{err}_{S_n}(h_n) - \gamma_n(x + \mathrm{err}_{S_n}(h_n)), \quad x \geq 0,$$

and observe that by Equation (2.13) and Equation (2.10), we have $F(\mathrm{err}_{S_n}(h^*)) \leq 0$. It can be checked that F is non-decreasing on $x \geq \varepsilon_{H,n,\delta/(n(n+1))} - \mathrm{err}_{S_n}(h_n)$. Therefore,

$$F(\mathrm{err}_{S_n}(h'_n)) \leq F(\mathrm{err}_{S_n}(h^*)) \leq 0,$$

which implies

$$\mathrm{err}_{S_n}(h'_n) - \mathrm{err}_{S_n}(h_n) \leq \gamma_n(\mathrm{err}_{S_n}(h'_n) + \mathrm{err}_{S_n}(h_n)).$$

This completes the inductive proof.

Now we prove the second claim. The first claim of the present theorem implies the preconditions of Lemma 2.5. Thus, for any $m \in \{0, \ldots, n-1\}$, using Lemma 2.5 and Equation (2.12), we have

$$\begin{aligned}\mathrm{Reg}(h_m, H) &= \mathrm{err}_P(h_m) - \mathrm{err}_P(h^*) \\ &\leq \mathrm{err}_{P_m}(h_m) - \mathrm{err}_{P_m}(h^*) + \gamma_m(\rho_\wedge(h^*, h_m)) \\ &\leq \mathrm{err}_{S_m}(h_m) - \mathrm{err}_{S_m}(h^*) + \gamma_m(\rho_\wedge(h^*, h_m)) \\ &\leq \gamma_m(\rho_\wedge(h^*, h_m)). \end{aligned}$$

□

We now analyze the number of labels requested by a learner using Algorithm 26, following the same basic argument that was carried out for Algorithm 25, with a few adjustments due to the different deviation bound.

Lemma 2.7. *Fix any* $n \geq 1$, *and assume* $\cap_{m=0}^{n-1} E_m$ *holds. If* $\mathrm{err}_{S_{n-1}}(h'_{n-1}) - \mathrm{err}_{S_{n-1}}(h_{n-1}) \leq \gamma_{n-1}(\mathrm{err}_{S_{n-1}}(h'_{n-1}) + \mathrm{err}_{S_{n-1}}(h_{n-1}))$, *then* $x_n \in D(h^*, 2\,\mathrm{err}_P(h^*) + C\sqrt{\mathrm{err}_P(h^*)\varepsilon_{n-1}} + C\varepsilon_{n-1})$ *for some absolute constant* $C > 0$, *where* $\varepsilon_{n-1} := \varepsilon_{H,n-1,\delta/((n-1)n)}$.

Proof. Assume $\cap_{m=0}^{n-1} E_m$ holds and that

$$\mathrm{err}_{S_{n-1}}(h'_{n-1}) - \mathrm{err}_{S_{n-1}}(h_{n-1}) \leq \gamma_{n-1}(\mathrm{err}_{S_{n-1}}(h'_{n-1}) + \mathrm{err}_{S_{n-1}}(h_{n-1})).$$

By the first claim of Theorem 2.24, the preconditions of Lemma 2.5 hold. We first show that there exists a hypothesis $h \in H$ such that $h(x_n) \neq h^*(x_n)$ and $\mathrm{err}_P(h) \leq \mathrm{err}_P(h^*) + C\sqrt{\mathrm{err}_P(h^*)\varepsilon_{n-1}} + C\varepsilon_{n-1}$ for some absolute constant $C > 0$. There are two cases to consider: $h_{n-1}(x_n) = h^*(x_n)$ and $h_{n-1}(x_n) \neq h^*(x_n)$. Suppose first that $h_{n-1}(x_n) = h^*(x_n)$. Then

$$\begin{aligned}
\mathrm{err}_{P_{n-1}}(h'_{n-1}) - \mathrm{err}_{P_{n-1}}(h^*) &\leq \mathrm{err}_{S_{n-1}}(h'_{n-1}) - \mathrm{err}_{S_{n-1}}(h^*) \\
&\leq \mathrm{err}_{S_{n-1}}(h'_{n-1}) - \mathrm{err}_{S_{n-1}}(h_{n-1}) \\
&\leq \gamma_{n-1}(\mathrm{err}_{S_{n-1}}(h'_{n-1}) + \mathrm{err}_{S_{n-1}}(h_{n-1})) \\
&\leq \gamma_{n-1}(\mathrm{err}_{S_{n-1}}(h'_{n-1}) + \mathrm{err}_{S_{n-1}}(h^*)) \\
&\leq \gamma_{n-1}(\mathrm{err}_{P_{n-1}}(h'_{n-1}) + 2\,\mathrm{err}_{P_{n-1}}(h^*)).
\end{aligned}$$

Above, the first and last inequalities use Lemma 2.5; the second and second-to-last inequalities use the definition of h_{n-1}; the third inequality holds by assumption. Solving the inequality for $\mathrm{err}_{P_{n-1}}(h'_{n-1})$ gives

$$\mathrm{err}_{P_{n-1}}(h'_{n-1}) \leq \mathrm{err}_{P_{n-1}}(h^*) + C_1\sqrt{\mathrm{err}_{P_{n-1}}(h^*)\varepsilon_{n-1}} + C_1\varepsilon_{n-1}$$

for some absolute constant $C_1 > 0$. Therefore, by Equation (2.12) and Equation (2.11),

$$\begin{aligned}
\mathrm{err}_P(h'_{n-1}) &\leq \mathrm{err}_P(h^*) + C_1\sqrt{\mathrm{err}_{P_{n-1}}(h^*)\varepsilon_{n-1}} + C_1\varepsilon_{n-1} + \gamma_{n-1}(\rho_\wedge(h'_{n-1}, h^*)) \\
&\leq \mathrm{err}_P(h^*) + C_1\sqrt{\mathrm{err}_{P_{n-1}}(h^*)\varepsilon_{n-1}} + C_1\varepsilon_{n-1} + \gamma_{n-1}(\mathrm{err}_P(h'_{n-1}) + \mathrm{err}_P(h^*)).
\end{aligned}$$

Solving the inequality for $\mathrm{err}_P(h'_{n-1})$ gives

$$\mathrm{err}_P(h'_{n-1}) \leq \mathrm{err}_P(h^*) + C_2\sqrt{\mathrm{err}_P(h^*)\varepsilon_{n-1}} + C_2\varepsilon_{n-1}$$

for some absolute constant $C_2 > 0$.

Now instead suppose that $h_{n-1}(x_n) \neq h^*(x_n)$. Then

$$\begin{aligned}
\mathrm{err}_P(h_{n-1}) - \mathrm{err}_P(h^*) &\leq \mathrm{err}_{P_{n-1}}(h_{n-1}) - \mathrm{err}_{P_{n-1}}(h^*) + \gamma_{n-1}(\rho_\wedge(h_{n-1}, h^*)) \\
&\leq \mathrm{err}_{S_{n-1}}(h_{n-1}) - \mathrm{err}_{S_{n-1}}(h^*) + \gamma_{n-1}(\rho_\wedge(h_{n-1}, h^*)) \\
&\leq \gamma_{n-1}(\rho_\wedge(h_{n-1}, h^*)) \\
&\leq \gamma_{n-1}(\mathrm{err}_P(h_{n-1}) + \mathrm{err}_P(h^*)).
\end{aligned}$$

Above, the first inequality uses Equation (2.12); the second inequality uses Lemma 2.5; the third inequality uses the definition of h_{n-1}; the fourth inequality uses Equation (2.11). Solving the inequality for $\mathrm{err}_P(h_{n-1})$ gives

$$\mathrm{err}_P(h_{n-1}) \leq \mathrm{err}_P(h^*) + C_3\sqrt{\mathrm{err}_P(h^*)\varepsilon_{n-1}} + C_3\varepsilon_{n-1}$$

for some absolute constant $C_3 > 0$.

So let $h \in H$ be such that $h(x_n) \neq h^*(x_n)$ and $\mathrm{err}_P(h) \leq \mathrm{err}_P(h^*) + C\sqrt{\mathrm{err}_P(h^*)\varepsilon_{n-1}} + C\varepsilon_{n-1}$ for some absolute constant $C > 0$. By the triangle inequality,

$$\rho(h^*, h) \leq \mathrm{err}_P(h^*) + \mathrm{err}_P(h) \leq 2\,\mathrm{err}_P(h^*) + C\sqrt{\mathrm{err}_P(h^*)\varepsilon_{n-1}} + C\varepsilon_{n-1}.$$

Since $h(x_n) \neq h^*(x_n)$, it follows that $x_n \in D(h^*, 2\,\mathrm{err}_P(h^*) + C\sqrt{\mathrm{err}_P(h^*)\varepsilon_{n-1}} + C\varepsilon_{n-1})$. □

Theorem 2.25 (Selective ERM, improved version, label request bound)**.** *In the agnostic iid setting, the expected number of labels requested by a learner using Algorithm 26 (Selective ERM, improved version) after n rounds is at most*

$$\sup_{r>0} \tau(h^*, r) \cdot \left(2\,\mathrm{err}_P(h^*) \cdot n + O\left(\sqrt{\mathrm{err}_P(h^*) \cdot dn \log n} + d \log^2 n\right)\right).$$

where d is the VC dimension of the hypothesis class H, and $h^* \in \arg\min_{h \in H} \mathrm{err}_P(h)$ is a fixed risk minimizer.

2.4 Bibliographic references

The Halving mistake bound is due to Barzdin and Freivald (1972). The Standard Optimal Algorithm and the optimal mistake bounds for online classification are due to Littlestone (1988). The Perceptron algorithm is due to Rosenblatt (1958), and its mistake bound was proved by Block (1962) and Novikoff (1962). The algorithm and analysis for multi-class classification with bandit feedback is due to Long (2017).

The online-to-batch conversion is due to Angluin (1988), and the improvement mentioned is due to Littlestone (1989). The general iid setting of statistical learning and empirical risk minimization date back to the work of Vapnik and Chervonenkis (1971), while the realizable iid setting was defined by Valiant (1984) as the *PAC model*. The maximum deviation bounds in terms of Rademacher complexity are due to Bartlett and Mendelson (2002). Vapnik and Chervonenkis (1971) defined the Vapnik-Chervonenkis (VC) dimension and also proved maximum deviation bounds (which are also called *uniform convergence* bounds) and relative deviation bounds given in terms of VC dimension. The Sauer-Shelah lemma was proved by Sauer (1972) and Shelah (1972).

The Selective Consistent Hypothesis algorithm is due to Cohn *et al.* (1994), and the analysis is due to Hsu (2010). Alexander's capacity function was studied by Alexander (1987) and Giné and Koltchinskii (2006); its supremum was called the disagreement coefficient and first used to analyze active learning algorithms by Hanneke (2007). The Selective ERM algorithms and their analyses are simplifications of algorithms and analyses due to Dasgupta *et al.* (2007) and Hsu (2010). In particular, the improved Selective ERM algorithm is based on an analysis due to Hsu (2010) and Zhang (2015).

Bibliography

K. S. Alexander. Rates of growth and sample moduli for weighted empirical processes indexed by sets. *Probability Theory and Related Fields*, 75(3):379–423, 1987.

D. Angluin. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2002.

P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

J. M. Barzdin and R. V. Freivald. On the prediction of general recursive functions. *Soviet Math. Doklady*, 13:1224–1228, 1972.

H. D. Block. The perceptron: A model for brain functioning. *Reviews of Modern Physics*, 34:123–135, 1962.

D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine learning*, 15(2):201–221, 1994.

T. M. Cover. Behavior of sequential predictors of binary sequences. *Transactions on Prague Conference on Information Theory Statistical Decision Functions, Random Processes*, pages 263–272, 1965.

S. Dasgupta, D. Hsu, and C. Monteleoni. A general agnostic active learning algorithm. In *Advances in neural information processing systems*, 2007.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

Y. Freund and R. E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 29:79–103, 1999.

E. Giné and V. Koltchinskii. Concentration inequalities and asymptotic results for ratio type empirical processes. *The Annals of Probability*, 34(3):1143–1216, 2006.

S. Hanneke. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th international conference on Machine learning*, pages 353–360. ACM, 2007.

D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47:663–675, 1952.

D. Hsu. *Algorithms for Active Learning*. PhD thesis, University of California, San Diego, 2010.

N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.

N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

N. Littlestone. From on-line to batch learning. In *Proceedings of the second annual workshop on Computational learning theory*, pages 269–284, 1989.

P. M. Long. New bounds on the price of bandit feedback for mistake-bounded online multiclass learning. In *Proceedings of the 28th International Conference on Algorithmic Learning Theory*, 2017.

A. B. J. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, volume 12, pages 615–622, 1962.

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972.

S. Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972.

L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

V. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264, 1971.

C. Zhang. Personal communication, 2015.