CPSC 532D — STATISTICAL LEARNING THEORY

Danica J. Sutherland

University of British Columbia, Vancouver

This is not written to be read standalone; it's a reference to look at after class.
This version was compiled on November 16, 2022.

CONTENTS

Large parts of this document are heavily inspired by the books of Shalev-Shwartz and Ben-David [SSBD] and Mohri, Rostamizadeh, and Talkwalkar [MRT] as well as the lecture notes of Telgarsky [Tel].

1 SETUP

Lecture 1
September 7, 2022

Our default learning problem is as follows:

- We have a data distribution $\mathcal{D}$ over some domain $\mathcal{Z}$. For *supervised learning*, this is often (but not always) actually a product space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ of (x, y) pairs, where x is an input object (e.g. an image) and y is a label (e.g. whether the image contains a dog).

- We have n independent and identically distributed samples $z_1, \ldots, z_n \sim \mathcal{D}$.

- The sequence $S = (z_1, \ldots, z_n) \sim \mathcal{D}^n$ is our training “set.” (The terminology is very established, but we want to allow repeats, and occasionally we might want to look at the order.)

- We have a *hypothesis class* $\mathcal{H}$; in supervised learning, this is often a set of predictors $h : \mathcal{X} \to \hat{\mathcal{Y}}$, a space of predictions. (We might have $\hat{\mathcal{Y}} = \mathcal{Y}$, but we also might have binary labels $\mathcal{Y} = \{0, 1\}$ but make predictions with a confidence level in $\hat{\mathcal{Y}} = [0, 1]$.)
- We have a *loss function* $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$. In supervised learning, this often takes the form $\ell(h, (x, y)) = \lambda(h(x), y)$ for some $\lambda : \hat{\mathcal{Y}} \times \mathcal{Y} \to \mathbb{R}$. Some common examples:
  - Zero-one loss: $\lambda(\hat{y}, y) = \mathbb{1}(\hat{y} \neq y)$, usually used for $\mathcal{Y} = \hat{\mathcal{Y}}$ a discrete set of labels. This corresponds to one minus the *accuracy* of a predictor (A1 Q2b).
  - Logistic loss: $\lambda(\hat{y}, y) = \log(1 + \exp(-\hat{y}y))$ for $\hat{\mathcal{Y}} = \mathbb{R}$, $\mathcal{Y} = \{-1, 1\}$. This loss $\to 0$ if $\hat{y} \to \infty y$ (very confident in the right direction), is $\log 2$ if $\hat{y} = 0$ (a totally ambiguous prediction), and $\to \infty$ if $\hat{y} \to -\infty y$ (very confident in the wrong direction).
  - Square loss: $\lambda(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$. (Sometimes the $\frac{1}{2}$ isn't included.)
  - See assignment 1 question 2 for some more options.
- $L_{\mathcal{D}}(h) = \mathbb{E}_{z \sim \mathcal{D}} \ell(h, z)$ is called the risk, the population loss, the true loss, etc.
- $L_S(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h, z_i)$ is called the empirical risk, the sample loss, etc.
- A learning algorithm A is a function that takes in a sample S and returns a hypothesis. Ideally, one with low risk.
- The most common learning algorithm we'll think about is *empirical risk minimization*: $\mathrm{ERM}(S) \in \arg\min_{h \in \mathcal{H}} L_S(h)$. (If there are ties, by default we think of the algorithm returning an arbitrary choice.) The returned hypothesis, ERM(S) which we will also often denote $\hat{h}_S$, is called an empirical risk minimizer ("an ERM").

The function $\mathbb{1}$ returns one if its boolean argument is true, and zero if not.

If $\mathcal{H}$ is infinite, there might be not be a minimizer; we usually won't worry about this explicitly, but basically everything we talk about could be generalized to approximate minimizers.

Choosing an appropriate hypothesis class $\mathcal{H}$ is important: if it's too small, you'll never be able to learn the pattern you're looking for, but if it's too large, you'll *overfit* and pick one that seems good by chance. Much of this course will be about knowing when we can expect to overfit.

2 PAC LEARNING AND FINITE HYPOTHESIS CLASSES

2.1 *Realizable finite hypothesis classes*

For a nonnegative loss function, a distribution $\mathcal{D}$ is *realizable* by $\mathcal{H}$ if there is a $h^* \in \mathcal{H}$ with $L_{\mathcal{D}}(h^*) = 0$.

Lecture 2
September 14, 2022

We'll begin by showing that ERM succeeds at learning any *finite* $\mathcal{H}$. Assume that $0 \leq \ell(h, z) \leq 1$, and that $\mathcal{D}$ is realizable by $\mathcal{H}$. Note that this means that

$L_S(h^*) = 0$, although there might be other h with $L_S(h)$ but $L_{\mathcal{D}}(h) > 0$; we'll want to prove that if we get some such h, it can't be too bad. That is,

$$\begin{aligned}
\Pr_{S\sim\mathcal{D}^n}(L_{\mathcal{D}}(\hat{h}_S) > \varepsilon) &\leq \Pr_{S\sim\mathcal{D}^n}\left(S \in \bigcup_{h\in\mathcal{H}\,:\,L_{\mathcal{D}}(h)>\varepsilon} \{S : L_S(h) = 0\}\right) \\
&\leq \sum_{h\in\mathcal{H}\,:\,L_{\mathcal{D}}(h)>\varepsilon} \Pr_{S\sim\mathcal{D}^n}(L_S(h) = 0) \qquad \text{by a \textit{union bound}} \\
&= \sum_{h\in\mathcal{H}\,:\,L_{\mathcal{D}}(h)>\varepsilon} \Pr_{S\sim\mathcal{D}^n}(\forall i \in [n].\ell(h, z_i) = 0) \\
&= \sum_{h\in\mathcal{H}\,:\,L_{\mathcal{D}}(h)>\varepsilon} \prod_{i=1}^{n} \Pr_{z_i\sim\mathcal{D}}(\ell(h, z_i) = 0) \\
&= \sum_{h\in\mathcal{H}\,:\,L_{\mathcal{D}}(h)>\varepsilon} \left(\Pr_{z\sim\mathcal{D}}(\ell(h, z) = 0)\right)^n .
\end{aligned}$$

Now, let $p_0(h) = \Pr_{z\sim\mathcal{D}}(\ell(z, h) = 0)$. We know that

$$L_{\mathcal{D}}(h) = p_0(h) \times 0 + (1 - p_0(h)) \times \underbrace{\mathbb{E}_z[\ell(z, h) \mid \ell(z, h) > 0]}_{\leq 1}.$$

Thus, if $L_{\mathcal{D}}(h) > \varepsilon$, we must have $p_0(h) < 1 - \varepsilon$, and so

$$\begin{aligned}
\Pr_{S\sim\mathcal{D}^n}(L_{\mathcal{D}}(\hat{h}_S) > \varepsilon) &\leq \sum_{h\in\mathcal{H}\,:\,L_{\mathcal{D}}(h)>\varepsilon} (1 - \varepsilon)^n \\
&= |h \in \mathcal{H} \;:\; L_{\mathcal{D}}(h) > \varepsilon|\,(1 - \varepsilon)^n \\
&\leq |\mathcal{H}|\,(1 - \varepsilon)^n \\
&\leq |\mathcal{H}| \exp(-\varepsilon n)
\end{aligned}$$

where the last line used that $1 - t \leq 1 - \exp(-t)$.

Thus, we've shown that:

- The probability of ERM with n samples having error more than ε is at most $|\mathcal{H}| \exp(-\varepsilon n)$.
- With probability at least $1 - \delta$, the error of ERM with n samples is at most $\frac{1}{n}\left[\log|\mathcal{H}| + \log\frac{1}{\delta}\right]$.
- If we have at least $\frac{1}{\varepsilon}\left[\log|\mathcal{H}| + \log\frac{1}{\delta}\right]$ samples, the error of any ERM is at most ε with probability at least $1 - \delta$.

2.2 *PAC learning*

This last form shows that ERM is *probably approximately correct*. That is, there might be some samples S where we just get unlucky and can't really learn well; we allow that to happen a δ fraction of the time. Otherwise, though, we want to be approximately correct, i.e. have a small loss.

2.1 DEFINITION. Let the loss $\ell(h, z)$ be almost surely nonnegative. An algorithm A (realizably) *PAC learns* $\mathcal{H}$ if there exists a function $n : (0, 1)^2 \to \mathbb{N}$ such that, for every $\varepsilon, \delta \in (0, 1)$, for every $\mathcal{D}$ over $\mathcal{Z}$ which is realizable by $\mathcal{H}$, for any $n \geq n(\varepsilon, \delta)$ we have that

$$\Pr_{S \sim \mathcal{D}^n}(L_{\mathcal{D}}(A(S)) > \varepsilon) < \delta.$$

This definition, more or less, was introduced by Valiant [Val84].

If A runs in time polynomial in $1/\varepsilon$, $1/\delta$, n, and some notion of the size of h^*, then we say that A *efficiently PAC learns* $\mathcal{H}$.

We won't worry much about computational complexity in this course, but be aware that some authors use "PAC learning" to mean what we called efficient PAC learning. Note that there are *hypothesis classes which are PAC learnable but not efficiently PAC learnable under standard complexity assumptions; the canonical example is learning three-term boolean clauses in disjunctive normal form. Section 8.2 of [SSBD] overviews this example; Kearns and Vazirani [KV94, Section 1.4] explain it more fully.*

2.2 DEFINITION. A hypothesis class $\mathcal{H}$ is *PAC learnable* if there exists an algorithm A which PAC learns $\mathcal{H}$.

Note that this learning should work for *any distribution* $\mathcal{D}$, with a number of samples totally independent of what distribution nature has chosen for us (as long as it's realizable): a very worst-case kind of notion. We'll talk about distribution-dependent notions of learning later in the course.

Finite classes can do interesting things. In class, we talked about an example of learning Boolean conjunctions on d variables, hypotheses of the form "c and not d and g," and defined a natural ERM algorithm (the most restrictive clause consistent with all the positive samples). There are at most 3^d such conjunctions; so our result above implies that ERM has an error rate at most $\frac{d}{n} \log \frac{3}{\delta}$.

In practice, every hypothesis class we can represent on a computer is also finite. But those classes are *really big*, so bounds of that form tend to be poor, but they can be useful to keep in mind. (They do absolutely come up as a component of proving more advanced types of bounds.)

2.3 *Agnostic PAC learning*

We may not like the realizability assumption; in particular, it can't handle any supervised learning problem where more than y is possible for a given x.

2.3 DEFINITION. An algorithm A *agnostically PAC learns* $\mathcal{H}$ if there exists a function $n : (0, 1)^2 \to \mathbb{N}$ such that, for every $\varepsilon, \delta \in (0, 1)$, for every $\mathcal{D}$ over $\mathcal{Z}$, for any $n \geq n(\varepsilon, \delta)$ we have that

$$\Pr_{S\sim\mathcal{D}^n}\left(L_{\mathcal{D}}(A(S)) > \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \varepsilon\right) < \delta.$$

That is, A(S) can get arbitrarily close to the best possible error in $\mathcal{H}$.

If A runs in time polynomial in $1/\varepsilon$, $1/\delta$, n, and some notion of the size of h^*, then we say that A *efficiently agnostically PAC learns* $\mathcal{H}$.

2.4 DEFINITION. A hypothesis class $\mathcal{H}$ is *agnostically PAC learnable* if there exists an algorithm A which agnostically PAC learns $\mathcal{H}$.

If you don't know what a measurable function is, just think "any function"; we're not going to be overly concerned with the difference in this class.

The term $\inf_{h\in\mathcal{H}} L_{\mathcal{D}}(\mathcal{H})$ depends on your choice of hypothesis class. One thing we can say is that it is at least as big as the *Bayes error* $\inf_{h \text{ measurable}} L_{\mathcal{D}}(h)$, which is a measure of the "inherent noise" in the distribution $\mathcal{D}$. See A1 Q3 for more on the Bayes error.

We won't talk about improper learning much in this course, but it's good to know the term.

We can also do *improper* PAC learning, which allows our learning algorithm to select a hypothesis from $\mathcal{H}'$ that competes with the best hypothesis from $\mathcal{H}$. For example, you might want to show that your neural network learner is able to learn any quadratic target function.

2.4 *Uniform convergence and ERM*

In the argument of Section 2.1, we knew that h^* would achieve the best possible loss of 0; our only worry was that a bad hypothesis might also have zero empirical risk. In the agnostic setting, that's no longer true: there's noise, so the best hypothesis h^* (if there is one...) might not get the minimal error. But, hopefully, $L_S(h)$ will be reasonably close to $L_{\mathcal{D}}(h)$.

Remember we don't assume there exists some best h^* anymore. This kind of bound will often take an arbitrary comparator, which – since it was arbitrary – we can use to argue that, say, $L_{\mathcal{D}}(\hat{h}_S) \leq \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(\mathcal{H}) + 2\varepsilon$.

In particular, assume that $L_S(\hat{h}_S) \geq L_{\mathcal{D}}(\hat{h}_S) - \varepsilon$ ($\hat{h}_S$ doesn't seem way better than it really is), and also that $L_S(h^*) \leq L_{\mathcal{D}}(h^*) + \varepsilon$ for any fixed hypothesis h^*. If so, then we have that

$$L_{\mathcal{D}}(\hat{h}_S) \leq L_S(\hat{h}_S) + \varepsilon \leq L_S(h^*) + \varepsilon \leq L_{\mathcal{D}}(h^*) + 2\varepsilon, \tag{2.1}$$

so $\hat{h}_S$ is not much worse than h^*. The second inequality here used that the ERM $\hat{h}_S$ minimizes the empirical risk, and so by definition $L_S(\hat{h}_S) \leq L_S(h^*)$.

Let's think about the requirement $L_S(h^*) \leq L_{\mathcal{D}}(h^*) + \varepsilon$ first. We can do this with the following lemma, which we'll prove shortly:

2.5 LEMMA. *If the loss $\ell(h, z)$ is almost surely bounded in $[a, b]$, then for any fixed* *hypothesis h:*

$$\Pr_{S\sim\mathcal{D}^n}\left(L_S(h) \le L_{\mathcal{D}}(h) + (b-a)\sqrt{\frac{\log(1/\delta)}{2n}} \right) \ge 1-\delta$$

$$\Pr_{S\sim\mathcal{D}^n}\left(L_S(h) \ge L_{\mathcal{D}}(h) - (b-a)\sqrt{\frac{\log(1/\delta)}{2n}} \right) \ge 1-\delta$$

$$\Pr_{S\sim\mathcal{D}^n}\left(|L_S(h) - L_{\mathcal{D}}(h)| \le (b-a)\sqrt{\frac{\log(2/\delta)}{2n}} \right) \ge 1-\delta.$$

So, that's no real worry.

But, we can't just apply this lemma to $\hat{h}_S$, because which hypothesis is the ERM depends on the whole sample. What we can do, though, is apply it to *every* hypothesis $h \in \mathcal{H}$. Then it'll have to hold for $\hat{h}_S$, since it held for everything. This is called *uniform convergence*, and while it's maybe a little "brute force," it underpins a *lot* of statistical learning theory.

Another way to write this, that's going to be natural for infinite hypothesis classes as well as finite ones, is to say that

$$\sup_{h\in\mathcal{H}} L_{\mathcal{D}}(h) - L_S(h) \le \varepsilon.$$

If this holds with high probability, and we apply Lemma 2.5 for the other direction for h^*, then we can apply the argument of (2.1) to say that $L_{\mathcal{D}}(\hat{h}_S)$ is not too much worse than $L_{\mathcal{D}}(h^*)$ with high probability.

2.5 *ERM agnostically PAC learns finite classes*

Lecture 3
September 21, 2022

So, to show that ERM agnostically PAC learns a finite $\mathcal{H}$, all we have to do is plug in Lemma 2.5 for each $h \in \mathcal{H}$. There are multiple ways we can do this, but one way is to divide our error probability δ equally into $|\mathcal{H}|+1$ parts, since we'll do a lower bound for $|\mathcal{H}|$ hypotheses and an upper bound for one more. Then we have that

$$\begin{aligned} &\forall h \in \mathcal{H}. \quad L_S(h) \ge L_{\mathcal{D}}(h) - (b-a)\sqrt{\frac{1}{2n}\log\frac{|\mathcal{H}|+1}{\delta}} \\ &\text{and} \qquad L_S(h^*) \le L_{\mathcal{D}}(h^*) + (b-a)\sqrt{\frac{1}{2n}\log\frac{|\mathcal{H}|+1}{\delta}}, \end{aligned}$$

which, plugging into (2.1), gives us that with probability at least $1-\delta$,

$$L_{\mathcal{D}}(\hat{h}_S) \le L_{\mathcal{D}}(h^*) + (b-a)\sqrt{\frac{2}{n}\log\frac{|\mathcal{H}|+1}{\delta}}.$$

We can easily convert to the sample complexity form used by PAC learning: ERM agnostically PAC learns $\mathcal{H}$ with $n(\varepsilon, \delta) = \frac{2(b-a)^2}{\varepsilon^2} \log \frac{|\mathcal{H}|+1}{\delta}$.

3 CONCENTRATION INEQUALITIES

We'll now prove Lemma 2.5, and learn a bunch of useful stuff along the way.

3.1 DEFINITION. A random variable X with mean $\mu = \mathbb{E}[X]$ is called *sub-Gaussian with variance parameter* $\sigma \geq 0$, written $X \in \mathcal{SG}(\sigma)$, if $\mathbb{E}[e^{\lambda(X-\mu)}] \leq e^{\frac{1}{2}\lambda^2\sigma^2}$ for all $\lambda \in \mathbb{R}$.

We motivated this definition by noting that a Gaussian $\mathcal{N}(\mu, \sigma^2)$ is $\mathcal{SG}(\sigma)$.

Notice that if $\sigma_1 < \sigma_2$, then anything that's $\mathcal{SG}(\sigma_1)$ is also $\mathcal{SG}(\sigma_2)$.

3.2 PROPOSITION (Hoeffding's lemma). *A real-valued random variable bounded in* $[a, b]$ *is* $\mathcal{SG}\left(\frac{b-a}{2}\right)$.

You'd be able to follow the proof, it's just a little messy and I don't think it's all that insightful or interesting. Wainwright [Wai19] actually has a proof that only shows $\mathcal{SG}(b-a)$ that I do think is pretty neat, though – see his Examples 2.3 and 2.4.

We didn't prove this in class; you can check out a proof in Lemma B.6 of Shalev-Shwartz and Ben-David [SSBD] or Lemma D.1 of Mohri, Rostamizadeh, and Talkwalkar [MRT].

3.3 PROPOSITION. *If* $X_1 \in \mathcal{SG}(\sigma_1)$ *and* $X_2 \in \mathcal{SG}(\sigma_2)$ *are independent,* $X_1 + X_2 \in \mathcal{SG}(\sqrt{\sigma_1^2 + \sigma_2^2})$.

Proof. $\mathbb{E}[e^{\lambda(X_1+X_2-\mathbb{E}[X_1+X_2])}] = \mathbb{E}[e^{\lambda(X_1-\mathbb{E}X_1)}]\,\mathbb{E}[e^{\lambda(X_2-\mathbb{E}X_2)}] \leq e^{\frac{1}{2}\lambda^2\sigma_1^2}\, e^{\frac{1}{2}\lambda^2\sigma_2^2} = e^{\frac{1}{2}\lambda^2\left(\sqrt{\sigma_1^2+\sigma_2^2}\right)^2}$. □

3.4 PROPOSITION. *If* $X \in \mathcal{SG}(\sigma)$, *then* $aX \in \mathcal{SG}(|a|\,\sigma)$ *for any* $a \in \mathbb{R}$.

Proof.
$$\mathbb{E}[e^{\lambda(aX-\mathbb{E}[aX])}] = \mathbb{E}[e^{(a\lambda)(X-\mathbb{E}X)}] \leq e^{\frac{1}{2}(a\lambda)^2\sigma^2} = e^{\frac{1}{2}\lambda^2(|a|\sigma)^2}.$$
□

3.5 PROPOSITION (Markov's inequality). *If* X *is a nonnegative-valued random variable,* $\Pr(X \geq t) \leq \frac{1}{t}\mathbb{E}X$.

Proof. Take expectations of both sides of $t\mathbb{1}(X \geq t) \leq X$. □

3.6 PROPOSITION (Chernoff bound for sub-Gaussians). *If* $X \in \mathcal{SG}(\sigma)$, *then* $\Pr(X \geq \mathbb{E}X + t) \leq \exp\left(-\frac{t^2}{2\sigma^2}\right)$ *for* $t \geq 0$.

Proof. Note that $\Pr(X \geq \mathbb{E}X + t) = \Pr(\exp(\lambda(X - \mathbb{E}X)) \geq \exp(\lambda t))$ for any $\lambda \in \mathbb{R}$. Applying Markov's inequality gives an upper bound of $\exp(-\lambda t)\,\mathbb{E}\exp(\lambda(X - \mathbb{E}X)) \leq \exp(\frac{1}{2}\lambda^2\sigma^2 - \lambda t)$. Plug in $\lambda = t/\sigma^2$. □

This isn't some magical choice of λ; it's just what minimizes the bound, as you can see by setting the derivative $\lambda\sigma^2 - t$ to zero.

Since $-X$ is also $\mathcal{SG}(\sigma)$ by Proposition 3.4, the same bound holds for a lower deviation $\Pr(X \le \mathbb{E} X - t)$. A union bound then immediately gives $\Pr(\left|X - \mu\right| \ge t) \le 2 \exp\left(-\frac{t^2}{2\sigma^2}\right)$.

3.7 PROPOSITION (Hoeffding). *If $X_1, \dots, X_n$ are independent and each $\mathcal{SG}(\sigma_i)$ with mean μ_i, for all $\varepsilon \ge 0$*

$$\Pr\left(\frac{1}{n}\sum_{i=1}^{n} X_i \ge \frac{1}{n}\sum_{i=1}^{n} \mu_i + \varepsilon\right) \le \exp\left(-\frac{n^2\varepsilon^2}{2\sum\limits_{i=1}^{n}\sigma_i^2}\right).$$

Proof. By Propositions 3.3 and 3.4, $\frac{1}{n}\sum\limits_{i=1}^{n} X_i \in \mathcal{SG}\left(\frac{1}{n}\sqrt{\sum\limits_{i=1}^{n}\sigma_i^2}\right)$. Then apply Proposition 3.6. □

If the X_i have the same mean $\mu_i = \mu$ and parameter $\sigma_i = \sigma$, this becomes

$$\Pr\left(\frac{1}{n}\sum_{i=1}^{n} X_i \ge \mu + \varepsilon\right) \le \exp\left(-\frac{n\varepsilon^2}{2\sigma^2}\right), \tag{Hoeffding}$$

which can also be stated as that, with probability at least $1 - \delta$,

$$\frac{1}{n}\sum_{i=1}^{n} X_i \le \mu + \sigma\sqrt{\frac{2}{n}\log\frac{1}{\delta}}. \tag{Hoeffding'}$$

Going back now to what we were trying to prove: Lemma 2.5 follows from combining (Hoeffding') with Proposition 3.2 to the variables $\ell(h, z_i)$.

4 RADEMACHER COMPLEXITY

Lecture 4
September 26, 2022

Although everything we do in practice is finite, analyzing it that way both gives us bad bounds and doesn't really give us much insight about what's going on. So, let's try to study infinite hypothesis classes.

4.1 *Deriving an expectation bound*

Specifically, let's just look at the worst-case generalization gap $\sup_{h\in\mathcal{H}} L_{\mathcal{D}}(h) - L_S(h)$ and try to bound that somehow. Although eventually we'll get a high-probability bound, let's start by analyzing its mean.

To start, note that

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^n} \sup_{h\in\mathcal{H}} L_{\mathcal{D}}(h) - L_S(h) = \mathop{\mathbb{E}}_{S\sim\mathcal{D}^n} \sup_{h\in\mathcal{H}} \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^n} L_{S'}(h) - L_S(h) \le \mathop{\mathbb{E}}_{\substack{S\sim\mathcal{D}^n\\ S'\sim\mathcal{D}^n}} \sup_{h\in\mathcal{H}} L_{S'}(h) - L_S(h)$$

where we first used that L_S is unbiased (A1 Q2a) to introduce a "ghost sample" $S' = (z'_1, \dots, z'_n)$. Then we used the general fact that $\sup_y \mathbb{E}_X f_y(X) \le \mathbb{E}_X \sup_y f_y(X)$: it's true for every realization of X and choice of y that $f_y(x) \le \sup_y f_y(X)$, so take the expectation of both sides and then the sup over y.

This inequality should be intuitive, once you think about it: if the optimization is allowed to see the particular realization of the randomness, it can "overfit" better than if it has to operate on the average over X.

This last form is a natural notion of generalization: it's asking, say, how much it's possible to overfit to a test set, rather than on the distribution as a whole.

Continuing, we're going to think about swapping points between the two sets. Specifically, let $\epsilon_i \in \{-1, 1\}$ for $i \in [n]$, and define $(u_i, u'_i) = \begin{cases} (z_i, z'_i) & \text{if } \varepsilon_i = 1 \\ (z'_i, z_i) & \text{if } \varepsilon_i = -1 \end{cases}$. Then, for any choice of $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_n)$, we have

Watch out that ϵ_i has nothing to do with ε; we'll call the vector $(\epsilon_1, \dots, \epsilon_n)$ by $\boldsymbol{\epsilon}$. Unfortunate, but no option is great here.

$$\ell(h, z'_i) - \ell(h, z_i) = \epsilon_i(\ell(h, u'_i) - \ell(h, u_i)).$$

Because this holds for any arbitrary choice of sign ϵ_i, it also holds if we pick them at random from a Rademacher distribution Unif(±1), a distribution that's 1 half the time and −1 the other half. Then we have that

$$\begin{aligned} \mathop{\mathbb{E}}_{S,S' \sim \mathcal{D}^n} \sup_{h \in \mathcal{H}} L_{S'}(h) - L_S(h) &= \mathop{\mathbb{E}}_{S,S' \sim \mathcal{D}^n} \sup_{h \in \mathcal{H}} \frac{1}{n} \sum_i [\ell(h, z'_i) - \ell(h, z_i)] \\ &= \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} \mathop{\mathbb{E}}_{S,S' \sim \mathcal{D}^n} \mathop{\mathbb{E}}_{U,U'} \left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_i \epsilon_i [\ell(h, u'_i) - \ell(h, u_i)] \,\middle|\, S, S', \boldsymbol{\epsilon} \right]. \end{aligned}$$

Here we're writing $U = (u_1, \dots, u_n)$ and $U' = (u'_1, \dots, u'_n)$ as random variables, but conditional on S, S′, and $\boldsymbol{\epsilon}$, they're actually deterministic. Thus, the term inside the sup is literally identical to what it was before, just written in a more complicated way.

Now, let's rearrange the outer expectations to swap S and U. The *marginal* distribution of U and U′ are just exactly $\mathcal{D}^n$, each a sequence of n iid samples from $\mathcal{D}$, and $\boldsymbol{\epsilon} \mid U, U'$ is still just random signs. Finally, S and S′ are deterministic given $\boldsymbol{\epsilon}$ and U, U′. This gives us

This is guaranteed possible by Fubini's theorem; for a nonnegative loss, it's fine as long as $L_{\mathcal{D}}(h)$ exists. (For a negative loss, it's enough for $\mathbb{E}_z |\ell(h, z)|$ to exist.)

$$\mathop{\mathbb{E}}_{S,S' \sim \mathcal{D}^n} \sup_{h \in \mathcal{H}} L_{S'}(h) - L_S(h) = \mathop{\mathbb{E}}_{U,U' \sim \mathcal{D}^n} \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}} \mathop{\mathbb{E}}_{S,S'} \left[\sup_{h \in \mathcal{H}} \frac{1}{n} \sum_i \epsilon_i [\ell(h, u'_i) - \ell(h, u_i)] \,\middle|\, U, U', \boldsymbol{\epsilon} \right].$$

But… S and S′ no longer appear inside the expectation at all, so we can just drop that expectation, then keep going:

This proof technique of introducing a random sign is called symmetrization.

$$\begin{aligned}
\mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n} \sup_{h\in\mathcal{H}} \mathrm{L}_{S'}(h) - \mathrm{L}_S(h) &= \mathop{\mathbb{E}}_{U,U'\sim\mathcal{D}^n} \mathop{\mathbb{E}}_{\epsilon} \sup_{h\in\mathcal{H}} \frac{1}{n}\sum_i \epsilon_i[\ell(h,u_i') - \ell(h,u_i)] \\
&\le \mathop{\mathbb{E}}_{U,U'\sim\mathcal{D}^n} \mathop{\mathbb{E}}_{\epsilon}\left[\sup_{h\in\mathcal{H}} \frac{1}{n}\sum_i \epsilon_i \ell(h,u_i') + \sup_{h'\in\mathcal{H}} \frac{1}{n}\sum_i (-\epsilon_i)\ell(h,u_i)\right] \\
&= \mathop{\mathbb{E}}_{U,U'\sim\mathcal{D}^n} \mathop{\mathbb{E}}_{\epsilon} \sup_{h\in\mathcal{H}} \frac{1}{n}\sum_i \epsilon_i \ell(h,u_i') + \mathop{\mathbb{E}}_{U,U'\sim\mathcal{D}^n} \mathop{\mathbb{E}}_{\epsilon} \sup_{h'\in\mathcal{H}} \frac{1}{n}\sum_i \epsilon_i \ell(h,u_i) \\
&= 2 \mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n} \mathop{\mathbb{E}}_{\epsilon} \sup_{h\in\mathcal{H}} \frac{1}{n}\sum_i \epsilon_i \ell(h,z_i) \\
&=: 2 \mathop{\mathbb{E}}_{S,S'\sim\mathcal{D}^n} \mathrm{Rad}\left((\ell\circ\mathcal{H})|_S\right)
\end{aligned}$$

using that $\sup_x f(x) + g(x) \le \sup_x f(x) + \sup_{x'} g(x')$, that $-\epsilon_i$ has the same distribution as ϵ_i, and then renaming U to S at the end for comfort.

We've bounded the expected generalization gap by twice the expected *Rademacher complexity* of the set $(\ell\circ\mathcal{H})|_S = \{(\ell(h,z_1),\dots,\ell(h,z_n)) : h\in\mathcal{H}\} \subseteq \mathbb{R}^n$. The notation $\mathcal{F}|_S$ denotes $\{(f(z_1),\dots,f(z_n)) : f\in\mathcal{F}\}$, and $\ell\circ\mathcal{H} = \{z\mapsto \ell(h,z) : h\in\mathcal{H}\}$ is a set of functions from $\mathcal{Z}$ to $\mathbb{R}$.

4.1 DEFINITION. The *Rademacher complexity* of a set $\mathcal{A}\subseteq\mathbb{R}^n$ is given by

$$\mathrm{Rad}(\mathcal{A}) = \mathop{\mathbb{E}}_{\epsilon\sim\mathrm{Unif}(\pm1)^n} \sup_{a\in\mathcal{A}} \frac{1}{n}\sum_{i=1}^n \epsilon_i a_i = \mathop{\mathbb{E}}_{\epsilon\sim\mathrm{Unif}(\pm1)^n} \sup_{a\in\mathcal{A}} \frac{\epsilon\cdot a}{n}.$$

One way to think of it is a measure of how much a set $\mathcal{A}$ extends in the direction of a random binary vector. $\mathrm{Rad}(\mathcal{F}|_S)$ measures how well $\mathcal{F}$ can align with random signs on the particular set S, or equivalently how well it can separate a random subset of S from the rest.

Nothing in this derivation depended on the particular choice of $\ell\circ\mathcal{H}$, and so we've proved:

4.2 THEOREM. *For any function class $\mathcal{F}$ of functions $f:\mathcal{Z}\to\mathbb{R}$, and any distribution $\mathcal{D}$ over $\mathcal{Z}$ with $S=(z_1,\dots,z_n)\sim\mathcal{D}^n$, we have*

$$\mathop{\mathbb{E}}_{S\sim\mathcal{D}^n} \sup_{f\in\mathcal{F}} \left(\mathop{\mathbb{E}}_{z\sim\mathcal{D}}[f(z)] - \frac{1}{n}\sum_{i=1}^n f(z_i) \right) \le 2 \mathop{\mathbb{E}}_{S\sim\mathcal{D}^n} \mathrm{Rad}(\mathcal{F}|_S).$$

4.2 *Consequences for ERM*

Since $\mathbb{E}_S \sup_h \mathrm{L}_{\mathcal{D}}(h) - \mathrm{L}_S(h) \le 2\,\mathbb{E}_S \mathrm{Rad}((\ell\circ\mathcal{H})|_S)$, it holds in particular that

$$\mathop{\mathbb{E}}_S\left[\mathrm{L}_{\mathcal{D}}(\hat{h}_S) - \mathrm{L}_S(\hat{h}_S)\right] \le 2\mathop{\mathbb{E}}_S \mathrm{Rad}((\ell\circ\mathcal{H})|_S).$$

We also have that $\mathrm{L}_\mathrm{S}(\hat{h}_\mathrm{S}) \le \mathrm{L}_\mathrm{S}(h^*)$, so $\mathbb{E}_\mathrm{S} \mathrm{L}_\mathrm{S}(\hat{h}_\mathrm{S}) \le \mathbb{E}_\mathrm{S} \mathrm{L}_\mathrm{S}(h^*) = \mathrm{L}_\mathcal{D}(h^*)$; plugging this in gives

$$\mathop{\mathbb{E}}_\mathrm{S} \mathrm{L}_\mathcal{D}(\hat{h}_\mathrm{S}) \le \mathrm{L}_\mathcal{D}(h^*) + 2 \mathop{\mathbb{E}}_\mathrm{S} \operatorname{Rad}((\ell \circ \mathcal{H})|_\mathrm{S}).$$

If we have for some $\mathcal{D}$, nonnegative loss ℓ, and $\mathcal{H}$ that $\mathbb{E}_{\mathrm{S}\sim\mathcal{D}^n} \operatorname{Rad}((\ell \circ \mathcal{H})|_\mathrm{S}) \to 0$ as $n \to \infty$, this then implies that ERM gets probably approximately the best hypothesis from $\mathcal{H}$ on $\mathcal{D}$; see A2 Q1(b).

4.3 *Basic properties of Rademacher complexity*

First, note that

$$\operatorname{Rad}(\{a\}) = \frac{1}{n} \mathop{\mathbb{E}}_{\boldsymbol\epsilon} \boldsymbol\epsilon \circ a = 0 :$$

no matter the vector, a singleton set has no complexity. (In terms of generalization, this makes sense: any given hypothesis is equally likely to over- or under-estimate the risk.)

On the other extreme, for the vertices of the hypercube

$$\operatorname{Rad}(\{-1,1\}^n) = \frac{1}{n} \mathop{\mathbb{E}}_{\boldsymbol\epsilon} \sup_a \boldsymbol\epsilon \cdot a = \frac{1}{n} \mathop{\mathbb{E}}_{\boldsymbol\epsilon} \boldsymbol\epsilon \cdot \boldsymbol\epsilon = 1.$$

This is also the complexity of the function class of all possible $\{-1,1\}$-valued functions, as long as S has no duplicates. This also makes sense: it's the maximally complex bounded set.

Letting $c\mathcal{A} = \{ca : a \in \mathcal{A}\}$ for any $c \in \mathbb{R}$, we have that

$$\operatorname{Rad}(c\mathcal{A}) = \frac{1}{n} \mathop{\mathbb{E}}_{\boldsymbol\epsilon} \sup_{a\in\mathcal{A}} \boldsymbol\epsilon \cdot (ca) = \frac{1}{n} \mathop{\mathbb{E}}_{\boldsymbol\epsilon} \sup_{a\in\mathcal{A}} |c| \, (\operatorname{sign}(c)\boldsymbol\epsilon) \cdot a = |c| \operatorname{Rad}(\mathcal{A})$$

since $\operatorname{sign}(c)\boldsymbol\epsilon$ has the same distribution as $\boldsymbol\epsilon$.

For $\mathcal{A} + \mathcal{B} = \{a + b : a \in \mathcal{A}, b \in \mathcal{B}\}$ we get

$$\operatorname{Rad}(\mathcal{A}+\mathcal{B}) = \frac{1}{n} \mathop{\mathbb{E}}_{\boldsymbol\epsilon} \sup_{\substack{a\in\mathcal{A}\\ b\in\mathcal{B}}} \boldsymbol\epsilon \cdot (a+b) = \frac{1}{n} \mathop{\mathbb{E}}_{\boldsymbol\epsilon} \sup_{a\in\mathcal{A}} \boldsymbol\epsilon \cdot a + \frac{1}{n} \mathop{\mathbb{E}}_{\boldsymbol\epsilon} \sup_{b\in\mathcal{B}} \boldsymbol\epsilon \cdot b = \operatorname{Rad}(\mathcal{A}) + \operatorname{Rad}(\mathcal{B}).$$

Combined with the fact that $\operatorname{Rad}(\{a\}) = 0$, this means that translating a set by a constant vector does not change its complexity.

4.4 *Talagrand's contraction lemma*

The following lemma is very useful for relating the complexity of $\ell \circ \mathcal{H}$ to that of $\mathcal{H}$, as well as for bounding the complexity of hypothesis classes that are based on compositions (like neural networks).

A 1-Lipschitz function is sometimes called a contraction, *because it doesn't increase the distance between any points, but instead (usually) contracts at least some of those distances.*

4.3 DEFINITION. A function $f : \mathcal{X} \to \mathcal{Y}$ is ρ-Lipschitz with respect to the $\|\cdot\|_\mathcal{X}$ and $\|\cdot\|_\mathcal{Y}$ norms if for all $x, y \in \mathcal{X}$, $\left\|f(x) - f(y)\right\|_\mathcal{X} \le \rho \left\|x - y\right\|_\mathcal{Y}$. The smallest ρ

for which this inequality holds is *the Lipschitz constant*, denoted $\|f\|_{\text{Lip}}$.

The condition for a function from $\mathbb{R}$ to $\mathbb{R}$ is $\left|f(x) - f(y)\right| \le \rho\left|x - y\right|$. If f is differentiable, then $\|f\|_{\text{Lip}} = \sup_{x\in\mathbb{R}} |f'(x)|$. The canonical example of a non-differentiable function that is still Lipschitz is the absolute value function.

4.4 LEMMA (Talagrand). *Let $\phi : \mathbb{R}^n \to \mathbb{R}^n$ be given by $\phi(a) = (\varphi_1(a_1), \dots, \varphi_n(a_n))$, where each ϕ_i is ρ-Lipschitz. Then*

$$\mathrm{Rad}(\phi \circ \mathcal{A}) = \mathrm{Rad}(\{\phi(a) : a \in \mathcal{A}\}) \le \rho\,\mathrm{Rad}(\mathcal{A}).$$

Lecture 5
September 28, 2022

We'll prove this by proving the following special case:

4.5 LEMMA. *Let $\varphi : \mathbb{R} \to \mathbb{R}$ be 1-Lipschitz. Then* $\mathrm{Rad}(\{(\varphi(a_1), a_2, \dots, a_n) : a \in \mathcal{A}\}) \le \mathrm{Rad}(\mathcal{A})$.

To reduce to this case: first, we can change from ρ-Lipschitz to 1-Lipschitz with the function $\phi'(a) = \frac{1}{\rho}\phi(a)$. Then, apply Lemma 4.5 n times in order to apply $\varphi_1, \varphi_2, \dots, \varphi_n$. Because $\rho = 1$, this doesn't blow up the right-hand side.

Proof of Lemma 4.5. We have

$$\begin{aligned}
\mathrm{Rad}(\{(\varphi(a_1), a_2, \dots, a_n) : a \in \mathcal{A}\}) &= \frac{1}{n}\mathop{\mathbb{E}}_{\epsilon}\sup_{a} \epsilon_1\varphi(a_1) + \boldsymbol{\epsilon}_{2:}\cdot a_{2:} \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{2:}}\sup_{a\in\mathcal{A}} \varphi(a_1) + \boldsymbol{\epsilon}_{2:}\cdot a_{2:} + \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{2:}}\sup_{a'\in\mathcal{A}} -\varphi(a_1') + \boldsymbol{\epsilon}_{2:}\cdot a_{2:}' \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{2:}}\sup_{a,a'\in\mathcal{A}} \varphi(a_1) - \varphi(a_1') + \boldsymbol{\epsilon}_{2:}\cdot(a_{2:} + a_{2:}').
\end{aligned}$$

Now, $\varphi(a_1) - \varphi(a_1')$ will always be nonnegative at or near the supremum: if it were negative, we could simply swap a and a' to make that term positive and not affect the rest of the function. Thus we can write

$$\begin{aligned}
\mathrm{Rad}(\{(\varphi(a_1), a_2, \dots, a_n) : a \in \mathcal{A}\}) &= \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{2:}}\sup_{a,a'\in\mathcal{A}} \left|\varphi(a_1) - \varphi(a_1')\right| + \boldsymbol{\epsilon}_{2:}\cdot(a_{2:} + a_{2:}') \\
&\le \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{2:}}\sup_{a,a'\in\mathcal{A}} \left|a_1 - a_1'\right| + \boldsymbol{\epsilon}_{2:}\cdot(a_{2:} + a_{2:}') \\
&= \frac{1}{2}\mathop{\mathbb{E}}_{\boldsymbol{\epsilon}_{2:}}\sup_{a,a'\in\mathcal{A}} a_1 - a_1' + \boldsymbol{\epsilon}_{2:}\cdot(a_{2:} + a_{2:}'),
\end{aligned}$$

using that φ is 1-Lipschitz and then the same argument again to remove the absolute value. Then, split up the supremum again and turn it back into

$$\mathrm{Rad}(\{(\varphi(a_1), a_2, \dots, a_n) : a \in \mathcal{A}\}) \le \mathrm{Rad}(\mathcal{A}). \qquad \square$$

4.5 *From* $\operatorname{Rad}((\ell \circ \mathcal{H}|)_S)$ *to* $\operatorname{Rad}(\mathcal{H}|_S)$

Note: If $S = (z_1, \dots, z_n) = ((x_1, y_1), \dots, (x_n, y_n))$, then I'm using S_x to denote $(x_1, \dots, x_n)$.

Suppose that ℓ is of a form such that $\ell(h, (x, y)) = \lambda_y(h(x))$, where now I'm using a subscript for y instead of two arguments for reasons that'll be clear in a second. We can think of $(\ell \circ \mathcal{H})|_S$ as applying $\phi(a) = (\lambda_{y_1}(a_1), \dots, \lambda_{y_n}(a_n))$ to the vector $(h(x_1), \dots, h(x_n)) = h|_{S_x}$. Thus, if λ_y is ρ-Lipschitz for each y, Talagrand's lemma implies that $\operatorname{Rad}((\ell \circ \mathcal{H})|_S) \le \rho \operatorname{Rad}(\mathcal{H}|_{S_x})$.

Dealing with the complexity of the hypothesis class directly, rather than of the loss function class, is often more intuitive: "how well can my hypothesis class fit random signs"?

Many loss functions for continuous predictions are just "naturally" Lipschitz. For example, absolute value loss (as in mean absolute error regression) $\lambda_y(\hat{y}) = \left|y - \hat{y}\right|$, or logistic loss $\lambda_y(\hat{y}) = \log(1 + \exp(-y\hat{y}))$, are each 1-Lipschitz:

$$\begin{aligned} \left|\frac{\mathrm{d}}{\mathrm{d}\hat{y}} \log(1 + \exp(-y\hat{y}))\right| &= \left|\frac{1}{1 + \exp(-y\hat{y})} \exp(-y\hat{y})(-y)\right| \\ &= \left|\frac{\exp(-y\hat{y})}{1 + \exp(-y\hat{y})} \times \frac{\exp(y\hat{y})}{\exp(y\hat{y})}\right| \left|-y\right| = \left|\frac{1}{1 + \exp(y\hat{y})}\right| \le 1 \end{aligned}$$

This means that, for example, we can get a generalization bound for the loss of logistic regression if we can bound $\mathbb{E}_S \operatorname{Rad}(\mathcal{H}|_{S_x})$. We'll do this next.

4.6 *Complexity of linear classes*

If you do classic, unregularized linear regression, you often overfit, but if you regularize the regression so that $\|w\|$ is not too big, you generally do much better. The usual form of regularization doesn't *quite* fit into our hypothesis class framework – we'll do some more direct regularization-based analyses soon in the course – but it turns out to be essentially equivalent to doing ERM with the logistic loss and a hypothesis class like

$$\mathcal{H}_B = \{x \mapsto \langle w, x \rangle : \|w\| \le B\},$$

the set of linear functions with bounded norm. We can bound that complexity directly as follows:

$$\begin{aligned} n \cdot \operatorname{Rad}(\mathcal{H}|_{S_x}) &= \mathop{\mathbb{E}}_{\epsilon} \sup_{\|w\| \le B} \sum_i \epsilon_i \langle w, x_i \rangle \\ &= \mathop{\mathbb{E}}_{\epsilon} \sup_{\|w\| \le B} \langle w, \sum_i \epsilon_i x_i \rangle \end{aligned}$$

$$
\begin{aligned}
&\le \mathop{\mathbb{E}}_{\epsilon} \sup_{\|w\|\le \mathrm{B}} \|w\| \left\| \sum_i \epsilon_i x_i \right\| && \text{(Cauchy-Shwartz)} \\
&= \mathrm{B} \mathop{\mathbb{E}}_{\epsilon} \left\| \sum_i \epsilon_i x_i \right\| \\
&\le \mathrm{B} \sqrt{\mathop{\mathbb{E}}_{\epsilon} \left\| \sum_i \epsilon_i x_i \right\|^2} && ((\mathbb{E}\,\mathrm{T})^2 \le \mathbb{E}\,\mathrm{T}^2) \\
&= \mathrm{B} \sqrt{\mathop{\mathbb{E}}_{\epsilon} \sum_{ij} \epsilon_i \epsilon_j \langle x_i, x_j \rangle} \\
&= \mathrm{B} \sqrt{\sum_i \mathbb{E}[\underbrace{\epsilon_i^2}_{1}] \|x_i\|^2 + \sum_{i \ne j} \underbrace{\mathop{\mathbb{E}}_{\epsilon}[\epsilon_i \epsilon_j]}_{0} \langle x_i, x_j \rangle}.
\end{aligned}
$$

We can rewrite this final inequality as

$$\mathrm{Rad}(\mathcal{H}|_{\mathrm{S}_x}) \le \frac{\mathrm{B}}{\sqrt{n}} \sqrt{\frac{1}{n} \sum_i \|x_i\|^2},$$

which depends on the particular S that you see. To avoid dealing with that, there are two typical routes.

One is to assume that $\mathcal{D}$ is such that $\|x\| \le \mathrm{M}$ almost surely, something often true in practice. This would imply that $\mathrm{Rad}(\mathcal{H}|_{\mathrm{S}_x}) \le \mathrm{BM}/\sqrt{n}$ almost surely.

The other is to write

$$\mathop{\mathbb{E}}_{\mathrm{S}} \mathrm{Rad}(\mathcal{H}|_{\mathrm{S}_x}) \le \frac{\mathrm{B}}{\sqrt{n}} \mathop{\mathbb{E}}_{\mathrm{S}} \sqrt{\frac{1}{n} \sum_i \|x_i\|^2} \le \frac{\mathrm{B}}{\sqrt{n}} \sqrt{\mathop{\mathbb{E}}_{x} \|x\|^2}. \tag{4.1}$$

This only works for the average Rademacher complexity, which is the only thing we've seen to care about yet, but in some settings you do want a high-probability bound on $\mathrm{Rad}(\mathcal{H}|_{\mathrm{S}_x})$ *rather than an average-case one.*

This allows for broader data distributions, as long as you can bound $\mathbb{E}\,\|x\|^2$: e.g. you can handle Gaussians much more easily.

In either case, this means we've shown an average excess error bound for logistic regression and mean-absolute-error regression with a rate of $\mathcal{O}(1/\sqrt{n})$,

4.7 *Binary classifiers with 0-1 loss*

It's not yet obvious how to handle binary classifiers, though. It turns out you can do this with Talagrand's lemma in the same way: if $h(x) \in \{-1, 1\}$ and $y \in \{-1, 1\}$, then the 0-1 loss is

$$\lambda_y(\hat{y}) = \begin{cases} 0 & \hat{y} = y \\ 1 & \hat{y} \ne y. \end{cases}$$

The trick is: we don't actually care, for computing the loss, what the function λ_y does for *other* values of $\hat{y}$. So, let's just pick a Lipschitz function that agrees at these points, by linear interpolation:

$$\lambda_y(\hat{y}) = \begin{cases} 0 & y\hat{y} \geq 1 \\ \frac{1}{2} - \frac{1}{2} y\hat{y} & 0 \leq y\hat{y} \leq 1 \\ 1 & y\hat{y} \leq 1. \end{cases}$$

This has $\left\|\lambda_y\right\|_{\mathrm{Lip}} = \frac{1}{2}\left|y\right| = \frac{1}{2}$. So, for $\mathcal{H}_{-1,1}$ of binary classifiers mapping to $\{-1, 1\}$,

$$\operatorname{Rad}((\ell_{0-1} \circ \mathcal{H}_{-1,1})|_{S}) \leq \frac{1}{2} \operatorname{Rad}(\mathcal{H}_{-1,1}|_{S_x}). \tag{4.2}$$

Note that, since Rad is "scale-sensitive", this very much depended on the choice to do $\{-1, 1\}$ classifiers. If we have $\{0, 1\}$ classifiers, we can either choose a slightly different function which would be 1-Lipschitz, or note that we can convert from a $\{0, 1\}$ classifier to a $\{-1, 1\}$ classifier by taking $2h - 1$ and use properties from Section 4.3 to see

$$\operatorname{Rad}(\mathcal{H}_{-1,1}|_{S_x}) = \operatorname{Rad}((2\mathcal{H}_{0,1} - 1)|_{S_x}) = 2 \operatorname{Rad}(\mathcal{H}_{0,1}|_{S_x}),$$

so that

$$\operatorname{Rad}((\ell_{0-1} \circ \mathcal{H}_{0,1})|_{S}) \leq \operatorname{Rad}(\mathcal{H}_{0,1}|_{S_x}).$$

4.8 *Finite sets*

Notice that when we're doing a binary classifier in particular, $\mathcal{H}_{S_x} = \{(h(x_1), \ldots, h(x_n)) : h \in \mathcal{H}\}$ can't be too big: even if $\mathcal{H}$ is infinite, there aren't infinitely many possible *behaviours on the set* S_x: there are only 2^n possible bit vectors of length n, and in fact there may be many fewer possible things that $\mathcal{H}$ can do on this particular S_x. So, let's try bounding the Rademacher complexity of an arbitrary finite set based on its size. This is going to turn out to be a very useful thing to do, and will form the basis of the next chunk of the course.

4.6 LEMMA. *Let* $T_1, \ldots, T_m$ *be zero-mean random variables that are each* $\mathcal{SG}(\sigma)$, *which are* ***not*** *necessarily independent. Then*

$$\mathbb{E}\left[\max_{i=1,\ldots,m} T_i\right] \leq \sigma\sqrt{2\log(m)}.$$

Proof. This is exactly A2 Q2(d), with variable names changed. □

4.7 LEMMA. *If* $\mathcal{A}$ *is finite and* $\|a\| \leq B$ *for all* $a \in \mathcal{A}$, *then*

$$\operatorname{Rad}(\mathcal{A}) \leq \frac{B}{n}\sqrt{2\log|\mathcal{A}|}.$$

Proof. We have

$$\mathrm{Rad}(\mathcal{A}) = \frac{1}{n} \mathop{\mathbb{E}}_{\epsilon} \max_{a \in \mathcal{A}} \sum_{i=1}^{n} \epsilon_i a_i.$$

Each of those $\sum_{i=1}^{n} \epsilon_i a_i$ for a different a is some random variable, which all depend on the same $\boldsymbol{\epsilon}$, but that's fine. They each have mean zero, and since ϵ_i is $\mathcal{SG}(\frac{1-(-1)}{2}) = \mathcal{SG}(1)$ by Hoeffding's lemma, $a_i \epsilon_i / n$ is $\mathcal{SG}(|a_i|/n)$. These sub-terms are independent of one another, so applying Proposition 3.3 shows that $\sum_i a_i \epsilon_i \in \mathcal{SG}(\|a\|) \subseteq \mathcal{SG}(B)$. Use Lemma 4.6 and divide by n. □

For binary classifiers, $|h(x)| \le 1$, and so $\left\| h|_{S_x} \right\| \le \sqrt{n}$. Thus we get that

4.8 COROLLARY. *For binary classifiers mapping to* $\{-1, 1\}$,

$$\mathrm{Rad}(\mathcal{H}|_{S_x}) \le \sqrt{\frac{2}{n} \log \left| \mathcal{H}|_{S_x} \right|}.$$

(For binary classifiers mapping to $\{0, 1\}$, it's half that, by the scaling property from Section 4.3.)

Plugging in the 2^n bound would only get us that $\mathrm{Rad}(\mathcal{H}|_{S_x}) \le \sqrt{2 \log 2} \approx 1.18$, which is not very interesting since the generalization gap is trivially at most 1! But, when $\left|\mathcal{H}|_{S_x}\right| = o(2^n)$, this is far more interesting; we'll talk about this case next week.

As an aside, if we just use that $\left|\mathcal{H}|_{S_x}\right| \le |\mathcal{H}|$, we'd get for ρ-Lipschitz losses that

$$\mathbb{E} \sup_{h \in \mathcal{H}} [L_{\mathcal{D}}(\mathcal{H}) - L_S(\mathcal{H})] \le 2\rho \sqrt{\frac{2}{n} \log |\mathcal{H}|},$$

which is interesting to compare to the direct (high-probability) finite-class bound we showed in Section 2.5.

5 NO FREE LUNCH

Lecture 6
October 3, 2022

Nick Harvey did this (and the next) lecture. (Note to self: next time I'll switch the order around a bit!)

So far we've been taking some fixed hypothesis class $\mathcal{H}$, and proving upper bounds on generalization for that class. There's a fundamental tension in choosing your $\mathcal{H}$: too small and $\inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$ will be much bigger than the Bayes error, so even if you generalize you "fail" because your predictor's a lot worse than it could be (the *approximation error* is large). Too large of a $\mathcal{H}$, though, and you can't generalize: your *estimation error* is large. But we haven't yet shown any lower bounds saying that you *can't* learn in certain classes – just bounds where an upper bound doesn't show you can learn. But this doesn't necessarily mean anything: you can always just prove some dumb upper bound that doesn't account for your algorithm actually working. The

Here's one of these dumb upper bounds: using 0-1 loss, $L_{\mathcal{D}}(A(S)) \le 1$ *for any algorithm A. Congrats. But this doesn't mean that no algorithm can learn!*

no free lunch theorem is a *lower bound* establishing that *no algorithm* can learn certain hypothesis classes.

For this theorem, we're going to use the 0-1 loss for binary classifiers, $\mathcal{Y} = \{0, 1\}$. We're going to use slightly different notation than we did before (as Nick did in class, and as [SSBD] does some of the time): for this proof it's a little more convenient to think of having a distribution $\mathcal{D}_x$ over $\mathcal{X}$ and a deterministic labeling function $f : \mathcal{X} \to \mathcal{Y}$. This implies one of our usual joint distributions $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ by just sampling $x \sim \mathcal{D}_x$ and then choosing $(x, f(x))$; I'll abbreviate the loss $L_{\mathcal{D}}(h)$ you'd get by constructing this joint distribution as $L_{\mathcal{D}_x,f}(h)$; note that $L_{\mathcal{D}_x,f}(f) = 0$, so if $f \in \mathcal{H}$ then this problem is realizable.

We'd like to find a hypothesis class that is *not* PAC learnable (recall Definitions 2.1 and 2.2). Specifically, we're going to use the set of *all* functions from $\mathcal{X}$ to $\mathcal{Y}$, and show that no algorithm can PAC-learn this class: no matter what number of samples you see, there is some realizable distribution $\mathcal{D}$ such that $L_{\mathcal{D}}(A(S))$ is still at least $\frac{1}{8}$. Thus the sample complexity function $n(\varepsilon, \delta)$ can't be finite.

5.1 THEOREM (No Free Lunch). *Let* A *be any learning algorithm based on a sample* S *of n training examples, and suppose $n \leq |\mathcal{X}|/2$. Then there exists a distribution $\mathcal{D}_x$ over $\mathcal{X}$ and a labeling function $f : \mathcal{X} \to \mathcal{Y}$ for which the 0-1 loss satisfies*

$$\Pr_{S \sim \mathcal{D}_x^n}\left(L_{\mathcal{D}_x,f}(A(S)) \geq \tfrac{1}{8}\right) \geq \tfrac{1}{7}.$$

Proof. To achieve this, we're first going to pick a finite subset $\tilde{\mathcal{X}} \subseteq \mathcal{X}$ of size $\left|\tilde{\mathcal{X}}\right| = 2n$. We can pick that subset arbitrarily; at least one certainly exists, since $n \leq |\mathcal{X}|/2$. Then we'll pick $\mathcal{D}_x$ to be a discrete uniform distribution on $\tilde{\mathcal{X}}$.

Next, we're at first going to pick a random distribution of possible labeling functions f; we'll settle on a particular one later. We'll let $f : \tilde{\mathcal{X}} \to \mathcal{Y}$ assign its labels uniformly at random: for any $x \in \tilde{\mathcal{X}}$, we flip an independent coin to decide if $f(x) = 0$ or $f(x) = 1$, with equal probability.

(We can let f do whatever we like on $\mathcal{X} \setminus \tilde{\mathcal{X}}$, e.g. always return 0; it doesn't matter to us, since $\mathcal{D}_x$ never samples there. We could let it be uniformly random there too, but if $\mathcal{X}$ is uncountable, that way lies danger, since f won't be measurable.)

Now, for any sample of inputs $S_x = (x_1, \ldots, x_n)$, we can implicitly construct a sample of pairs $S = ((x_1, f(x_1)), \ldots, (x_n, f(x_n)))$; call the result of the algorithm $\hat{h}_S = A(S)$. Its expected loss is

$$\mathop{\mathbb{E}}_{f \sim \mathrm{Unif}(\tilde{\mathcal{X}} \to \mathcal{Y})} \mathop{\mathbb{E}}_{S_x \sim \mathcal{D}_x^n} L_{\mathcal{D}_x,f}(\hat{h}_S) = \mathop{\mathbb{E}}_{f, S_X} \mathop{\mathbb{E}}_{x \sim \mathcal{D}_x} \mathbb{1}(\hat{h}_S(x) \neq f(x)).$$

Using the law of total expectation, let's break this expectation up based on whether the test x is in the training data S or not:

$$\mathop{\mathbb{E}}_{f,\mathrm{S},x} \mathbb{1}(\hat{h}_\mathrm{S}(x) \neq f(x)) = \mathop{\mathbb{E}}_{f,\mathrm{S}_x} \Big[\Pr(x \notin \mathrm{S}_x)\, \mathbb{E}[\mathbb{1}(\hat{h}_\mathrm{S}(x) \neq f(x)) \mid x \notin \mathrm{S}_x] \\ + \Pr(x \in \mathrm{S}_x)\, \mathbb{E}[\mathbb{1}(\hat{h}_\mathrm{S}(x) \neq f(x)) \mid x \in \mathrm{S}_x] \Big].$$

For the second term, we're not going to worry about what the algorithm does on the data it's actually seen: we'll just bound this as being at least zero.

For the first term, we know since $\mathcal{D}_x$ is uniform and $|\mathrm{S}_x| \leq n$ that

$$\Pr(x \notin \mathrm{S}_x) = \frac{\left|\tilde{\mathcal{X}} \setminus \mathrm{S}_x\right|}{\left|\tilde{\mathcal{X}}\right|} \geq \frac{n}{2n} = \frac{1}{2}.$$

Also, since our labels $f(x)$ are uniformly random and totally independent of one another, and $\hat{h}_\mathrm{S}$ has no information about it, it's just a random guess: $\mathbb{E}[\mathbb{1}(\hat{h}_\mathrm{S}(x) \neq f(x)) \mid x \notin \mathrm{S}_x] = \frac{1}{2}$.

Combining, we know that

$$\mathop{\mathbb{E}}_{f \sim \mathrm{Unif}(\mathcal{X} \to \mathcal{Y})} \mathop{\mathbb{E}}_{\mathrm{S}_x \sim \mathcal{D}_x^n} \mathrm{L}_{\mathcal{D}_x, f}(\hat{h}_\mathrm{S}) \geq \tfrac{1}{4}.$$

But, if the *average* over functions f of the expected loss $\mathbb{E}_{\mathrm{S}_x \sim \mathcal{D}_x^n} \mathrm{L}_{\mathcal{D}_x,f}(\hat{h}_\mathrm{S})$ is at least $\frac{1}{4}$, then there must be at least one *particular* function f such that the expected loss is at least $\frac{1}{4}$! Pick one such f; this will be the labeling function claimed by the theorem.

This proof technique is known as the probabilistic method, *and often attributed to Paul Erdős.*

We've shown the average loss is large, but we want to show that the loss has high probability of being large. Now, $\mathrm{L}_{\mathcal{D}_x,f}(\hat{h}_\mathrm{S})$ is a random variable bounded in $[0, 1]$, and we already know one way to bound those variables in terms of their means: Markov's inequality (Proposition 3.5). But, unfortunately, Markov's inequality bounds the probability of things being *big*, and we want to bound the probability of this being *small*. So we'll need to switch it around:

5.2 PROPOSITION (Reverse Markov). *If* $\Pr(\mathrm{X} \leq b) = 1$, *then* $\Pr(\mathrm{X} \leq t) \leq \frac{\mathbb{E}[b - \mathrm{X}]}{b - t}$.

Proof. Apply Markov's inequality to the random variable $b - \mathrm{X}$. □

Since our expected loss is bounded in $[0, 1]$, reverse Markov gives us

$$\Pr(\mathrm{L}_{\mathcal{D}_x,f}(\hat{h}_\mathrm{S}) \leq \tfrac{1}{8}) \leq \frac{1 - \mathbb{E}\,\mathrm{L}_{\mathcal{D}_x,f}(\hat{h}_\mathrm{S})}{1 - \frac{1}{8}} \leq \frac{1 - \frac{1}{4}}{\frac{7}{8}} = \frac{3}{4} \cdot \frac{8}{7} = \frac{6}{7}.$$

Thus, for the f and $\mathcal{D}_x$ we picked above,

$$\Pr_{\mathrm{S} \sim \mathcal{D}_x^n}\left(\mathrm{L}_{\mathcal{D}_x,f}(\hat{h}_\mathrm{S}) > \tfrac{1}{8}\right) \geq \frac{1}{7}.$$ □

5.3 COROLLARY. *If $|\mathcal{X}| = \infty$, the set of all functions from $\mathcal{X}$ to $\{0, 1\}$ is not PAC learnable with 0-1 loss.*

I used 0.1 here as an arbitrary number smaller than $\frac{1}{8}$ and $\frac{1}{7}$, just to avoid worrying about $\leq$ versus $<$. . . .

Proof. Suppose it were PAC learnable. Then there would be a sample complexity function $n(\varepsilon, \delta)$ such that if $n \geq n(0.1, 0.1)$, for any realizable $\mathcal{D}$, we would have $\Pr_{S\sim\mathcal{D}^n}(L_\mathcal{D}(A(S)) > 0.1) < 0.1$. But this contradicts Theorem 5.1. □

6 VC DIMENSION

The space of *all* functions from $\mathcal{X}$ to $\{0, 1\}$ that we were just talking about is pretty big ($2^{|\mathcal{X}|}$, and we assumed $|\mathcal{X}|$ is infinite. . .). But, as we saw in the proof of Theorem 5.1, we don't actually care about the behaviour of f *everywhere* in $\mathcal{X}$; we just need a subset of size $2n$. So, we can refine this a bit.

Nick made up the name lunchable (sort of); shattered is the usual name, I think because $\mathcal{H}$ can separate anything from anything else, i.e. it shatters it into many pieces.

6.1 DEFINITION. For any hypothesis class $\mathcal{H}$, we say a set $C \subseteq \mathcal{X}$ is ~~lunchable~~ *shattered* by $\mathcal{H}$ if functions from $\mathcal{H}$ can achieve any labeling of C.

6.2 COROLLARY (to Theorem 5.1). *Let $\mathcal{H}$ be a hypothesis class shattering a set of size $|C|$, and suppose that $n \leq |C|/2$. Then there exists a distribution $\mathcal{D}_x$ over $\mathcal{X}$ and a labeling function $f : \mathcal{X} \to \mathcal{Y}$ for which the 0-1 loss satisfies*

$$\Pr_{S\sim\mathcal{D}_x^n}(L_{\mathcal{D}_x,f}(A(S)) \geq \tfrac{1}{8}) \geq \tfrac{1}{7}.$$

Slightly generalizing the notation $\mathcal{F}|_{S_x}$ used in Section 4, we can write $\mathcal{H}|_C$ to represent the set of possible labelings of a finite set $C \subseteq \mathcal{X}$:

$$\mathcal{H}|_C = \{(h(c_1), \ldots, h(c_{|C|})) : h \in \mathcal{H}\}.$$

Here we've assumed some implicit order on $C = \{c_1, \ldots, c_{|C|}\}$. A set C being shattered by $\mathcal{H}$ means exactly that $\left|\mathcal{H}|_C\right| = 2^{|C|}$, since there are $2^{|C|}$ possible binary labelings of C.

The letters VC are after Vladimir Vapnik and Alexey Chervonenkis, who developed this theory starting in the 60s in the Soviet Union (well before the definition of PAC learning); the English translation of the first key paper is [VC71].

6.3 DEFINITION. The *VC dimension* of a hypothesis class $\mathcal{H}$ is the size of the largest set that can be shattered by $\mathcal{H}$. If $\mathcal{H}$ can shatter sets of arbitrary size, we say it has infinite VC dimension.

6.4 COROLLARY. *Learning a hypothesis class such that it achieves 0-1 error below 1/8 on every possible $\mathcal{D}$ requires $\Omega(\mathrm{VCdim}(\mathcal{H}))$ samples.*

Lecture 7
October 5, 2022

6.1 *Examples of computing VC dimension*

It will be useful for all of our examples below to note that if you can't shatter any set of size n, you also can't shatter any set of size $n' > n$: if you could, then by definition you could shatter any size-n subset of the larger set.

6.1.1 Threshold functions

Let $h_a : \mathbb{R} \to \{0,1\}$ denote a threshold function $h_a(x) = \mathbb{1}(x \geq a)$, and let $\mathcal{H} = \{h_a : a \in \mathbb{R}\}$.

To start: we can shatter, say, $\mathrm{C} = \{0\}$, because $h_{-1}(0) = 1$ and $h_1(0) = 0$. Thus $\mathrm{VCdim}(\mathcal{H}) \geq |\mathrm{C}| = 1$.

We can shatter any set of size 1, but for VC dimension we only have to show that we can shatter one *particular set of that size.*

But we can't shatter any set C of size $|\mathrm{C}| \geq 2$. Let $a, b \in \mathrm{C}$ with $a < b$. We can't get $h(a) = 1$ and $h(b) = 0$ with the same $h \in \mathcal{H}$, since all $h \in \mathcal{H}$ are nondecreasing. Thus C cannot be shattered, and so $\mathrm{VCdim}(\mathcal{H}) < 2$.

Thus $\mathrm{VCdim}(\mathcal{H}) = 1$.

6.1.2 Circles

For $\mathcal{X} = \mathbb{R}^2$, consider $\mathcal{H} = \{h_{r,c} : r > 0, c \in \mathbb{R}^2\}$ with $h_{r,c}(x) = \mathbb{1}(\|x - c\| \leq r\}$, the set of indicator functions of circles.

This is like the problem from A1 Q1, but not necessarily centred at the origin.

We can shatter any set of size two, since we can draw a circle that includes both points, one that includes either point, or one that includes neither point.

We can also shatter *some* sets of size three, since if we put them in an equilateral triangle we can pick out none, or any one, two, or all three points. (If we put the three points in a line, we can't pick out the two edges but not the middle – but that's okay, VC dimension is about *the largest* set you can shatter.)

A bunch of these examples are easier to see if you draw them out! But I'm not taking the time to draw the diagrams with TikZ this time – sorry. Try drawing them yourself, or watch the recordings.

Claim: we cannot shatter any set of size four, and so $\mathrm{VCdim}(\mathcal{H}) = 3$. If we think of the points as lying roughly in a rectangle, then we can't pick out opposite corners without including at least one of the other points, but we didn't formalize this argument in class.

6.1.3 Homogeneous linear threshold functions in $\mathbb{R}^2$

Let $\mathcal{X} = \mathbb{R}^2$ and consider $\mathcal{H} = \{x \mapsto \mathrm{sgn}(w^\mathsf{T} x) : w \in \mathbb{R}^2\}$: hyperplanes passing through the origin. We're now using $\mathcal{Y} = \{-1, 1\}$, because it's more natural for linear classifiers, and we're going to define a function sgn which is like the sign except that $\mathrm{sgn}(0) = 1$ – yeah, yeah, that's gross but that's what we're doing in this context. If you want to stick to $\mathcal{Y} = \{0,1\}$, then instead use $\mathbb{1}(w^\mathsf{T} x \geq 0)$; that's much nicer to write down, but more annoying to work with.

We can shatter at least some sets of size 2: e.g. $\{(-1,1),(1,1)\}$, we can put the hyperplane along the x-axis to get both the same sign, or put it in along the y-axis to get them with opposite signs.

We can't shatter any sets of size 3. If the convex hull of the points contains the origin, then we can't get them all with the same sign; if the hull doesn't contain the origin, then we can't label them like $(1, 0, 1)$.

Reminder: a convex hull *of a set is the smallest convex set containing the original set:* $\mathrm{conv}(\mathcal{A}) = \{\alpha a + (1-\alpha)a' : a, a' \in \mathcal{A}, \alpha \in [0,1]\}$. *If you have some points in* $\mathbb{R}^2$, *you draw straight lines connecting the "outside" points to include all the points.*

So homogenous 2-d linear threshold functions have VC dimension 2.

6.1.4 Homogeneous linear threshold functions in $\mathbb{R}^d$

6.5 PROPOSITION. *Let $\mathcal{H} = \{x \mapsto \operatorname{sgn}(w^\mathsf{T} x) : w \in \mathbb{R}^d\}$. Then* $\operatorname{VCdim}(\mathcal{H}) = d$.

Proof. We can shatter a set of size d: take the set $\{e_1, \dots, e_d\}$ for e_i the ith standard basis vector, i.e. the "one-hot" vector with a 1 in the ith position and 0 everywhere else. Then we can achieve an arbitrary labeling $(y_1, \dots, y_d) \in \{0, 1\}^d$ by setting $w_i = y_i$: we get $w^\mathsf{T} e_i = y_i$.

Now, let $x_1, \dots, x_{d+1}$ be a set of $d + 1$ points in $\mathbb{R}^d$. Then they can't be linearly independent: there must be some $\beta_1, \dots, \beta_{d+1}$ such that $\sum_{i=1}^{d+1} \alpha_i x_i = 0$, with not all the α_i zero. Let $\mathcal{I}_+ = \{i \in [d + 1] : \beta_i > 0\}$, $\mathcal{I}_0 = \{i \in [d + 1] : \beta_i = 0\}$, and $\mathcal{I}_- = \{i \in [d + 1] : \beta_j < 0\}$.

Now, if $\mathcal{H}$ can shatter $\{x_1, \dots, x_{d+1}\}$, we can ask it to assign 1 to the x_i with $i \in \mathcal{I}_+ \cup \mathcal{I}_0$, and -1 to the x_i with $i \in \mathcal{I}_-$. Then we'd have

$$0 = w^\mathsf{T} 0 = w^\mathsf{T} \sum_{i=1}^{d+1} (\beta_i x_i) = \sum_{i \in \mathcal{I}_+} \underbrace{\beta_i}_{>0} \underbrace{w^\mathsf{T} x_i}_{\geq 0} + \sum_{i \in \mathcal{I}_-} \underbrace{\beta_i}_{<0} \underbrace{w^\mathsf{T} x_i}_{<0} .$$

This is the case I was complaining about to Nick that [SSBD] doesn't touch; he didn't either! :(

I claim that the sum on the right-hand side is strictly positive, meaning we've shown $0 < 0$, a contradiction; thus $\mathcal{H}$ cannot shatter $\{x_1, \dots, x_{d+1}\}$. This is easiest to see if $\mathcal{I}_-$ is nonempty: those terms will all be strictly positive. It will also be positive if there are any points in $\mathcal{I}_+$ with $w^\mathsf{T} x_i > 0$. Otherwise, the only case left is if $w^\mathsf{T} x_i = 0$ for all $i \in \mathcal{I}_+$ and $\mathcal{I}_- = \{\}$, meaning that this w labels all of the data points as positive. Recall that, if $\mathcal{I}_- = \{\}$, we must have $\sum_{i \in \mathcal{I}_+} \beta_i x_i = 0$. Now, suppose that $\tilde{w}$ is some weight vector that labels all these points as negative, $\tilde{w}^\mathsf{T} x_i < 0$ for all $i \in \mathcal{I}_+$; this must be possible if the set is shattered. Then we'd have

$$0 = \tilde{w}^\mathsf{T} 0 = \tilde{w}^\mathsf{T} \left(\sum_{i \in \mathcal{I}_+} \beta_i x_i \right) = \sum_{i \in \mathcal{I}_+} \underbrace{\beta_i}_{>0} \underbrace{\tilde{w}^\mathsf{T} x_i}_{<0} < 0,$$

a contradiction. Thus $\mathcal{H}$ cannot shatter $\{x_1, \dots, x_{d+1}\}$. □

6.1.5 Inhomogeneous linear threshold functions in $\mathbb{R}^d$

Nick didn't do this in class; I'll come back to it soon, but it makes more sense here in the notes.

What about if we don't enforce that the hyperplane passes through the origin, $\mathcal{H} = \{x \mapsto \operatorname{sgn}(w^\mathsf{T} x + b) : w \in \mathbb{R}^d, b \in \mathbb{R}\}$?

We could analyze this directly; Mohri, Rostamizadeh, and Talkwalkar [MRT, Example 3.12] do this if you want to see it, proving something called Radon's Theorem that's similar to what we showed above but a little cleaner and a standard theorem.

But we can also reduce to the set of homogeneous linear classifiers: if we have

d-dimensional data, we can model that as homogeneous linear classifiers on $(d+1)$-dimensional data with an extra "dummy feature" that's always 1. The weight w_0 corresponding to that feature will just be the offset b.

Using this reduction, we can see:

6.6 PROPOSITION. *For $x \in \mathbb{R}^d$, $\operatorname{VCdim}\left(\left\{x \mapsto w^\mathsf{T} x + b : w \in \mathbb{R}^d, b \in \mathbb{R}\right\}\right) = d + 1$.*

Proof. First, we can shatter the set $\{0, e_1, \dots, e_d\}$, which has size $d+1$, like before. We set $w_0 = y_0/2$ and $w_i = y_i$; the $y_0/2$ only affects the sign if all the other weights are "off", i.e. only on the 0 vector.

Also, we can't shatter any set of size $d+2$. If we could, then there would be $d+2$ vectors in $\mathbb{R}^{d+1}$ shattered by the class of homogeneous thresholds; but that class has VC dimension $d+1$ by Proposition 6.5, so that's not possible. □

6.2 *Growth function bounds based on VC dimension (Sauer-Shelah)*

The VC dimension only talks about exact shattering, i.e. $\left|\mathcal{H}|_S\right| = 2^{|S|}$. We could imagine that there are sets that aren't shattered, but are *nearly* shattered, say e.g. $\left|\mathcal{H}|_S\right| = 2^{|S|} - 1$; then the argument of a very slightly weaker no free lunch theorem would apply, and we'd still get big sample complexity. Let's give a name to the worst-case number of labelings:

I'm using S instead of S_x now just for laziness; it should be unambiguous below anyway.

6.7 DEFINITION. The *growth function* of a hypothesis class $\mathcal{H}$ is $\Pi_\mathcal{H}(n) = \max_{S \subseteq \mathcal{X} : |S| = n} \left|\mathcal{H}|_S\right|$.

It turns out that the "almost 2^n" growth we were worrying about doesn't happen: if $\operatorname{VCdim}(\mathcal{H}) = d$, then $\Pi_\mathcal{H}(n) = \mathcal{O}(n^d)$.

This is weird: the growth function is exponential for a while, being exactly 2^n up to $n = d$, but then it drops off to just polynomial growth.

6.8 LEMMA (Sauer-Shelah). *Let $\operatorname{VCdim}(\mathcal{H}) \le d < \infty$. Then $\Pi_\mathcal{H}(n) \le \sum_{i=0}^{d} \binom{n}{i}$.*

6.9 COROLLARY. *If $n \ge d = \operatorname{VCdim}(\mathcal{H})$, then $\Pi_\mathcal{H}(n) \le \left(\frac{en}{d}\right)^d$.*

This e is $\exp(1) \approx 2.718$.

We're going to prove Lemma 6.8 as a corollary to Lemma 6.10 below, and then finally come back to prove Corollary 6.9 by bounding binomial coefficients.

6.10 LEMMA (Pajor). *For all finite $S \subseteq \mathcal{X}$, $\left|\mathcal{H}|_S\right| \le \left|\{T \subseteq S : T \text{ is shattered by } \mathcal{H}\}\right|$.*

If S is shattered, both sides of the inequality are $2^{|S|}$, but otherwise it's not obvious that these things should be related.

Proof of Lemma 6.10. We'll proceed by induction on $\left|\mathcal{H}|_S\right|$.

Base case: $\left|\mathcal{H}|_S\right| = 1$. For the right-hand side, the empty set is trivially shattered by any $\mathcal{H}$, so the RHS is always at least 1 as well, and the inequality holds.

Inductive case: $\left|\mathcal{H}|_S\right| \geq 2$ and the inequality holds for any T with $\left|\mathcal{H}|_T\right| < \left|\mathcal{H}|_S\right|$. Then, since there two distinct labelings, there must be at least one point $x \in S$ that achieves both $h(x) = 1$ and $h'(x) = 0$ for some $h, h' \in \mathcal{H}$. Partition $\mathcal{H}$ into $\mathcal{H}_+ = \{h \in \mathcal{H} : h(x) = 1\}$ and $\mathcal{H}_- = \{h \in \mathcal{H} : h(x) = 0\}$. Now,

$$\left|\mathcal{H}|_S\right| = \left|\mathcal{H}_+|_S\right| + \left|\mathcal{H}_-|_S\right|,$$

since the two produce disjoint labelings on S (they always disagree on x). They also produce fewer labelings than $\mathcal{H}|_S$ itself (there's at least one labeling in each), so we can apply the inductive hypothesis to each.

Defining $\mathrm{Shat}_{\mathcal{H}}(S) = \{T \subseteq S : T \text{ is shattered by } \mathcal{H}\}$, we've shown that

$$\left|\mathcal{H}|_S\right| \leq \left|\mathrm{Shat}_{\mathcal{H}_+}(S)\right| + \left|\mathrm{Shat}_{\mathcal{H}_-}(S)\right|.$$

Note the right-hand side is exactly, counting up the "double-counted" sets,

$$\left|\mathrm{Shat}_{\mathcal{H}_+}(S) \cup \mathrm{Shat}_{\mathcal{H}_-}(S)\right| + \left|\mathrm{Shat}_{\mathcal{H}_+}(S) \cap \mathrm{Shat}_{\mathcal{H}_-}(S)\right|;$$

it remains to argue that this is at most $|\mathrm{Shat}_{\mathcal{H}}(S)|$. To see this, first note that $\mathrm{Shat}_{\mathcal{H}_+}(S) \cup \mathrm{Shat}_{\mathcal{H}_-}(S) \subseteq \mathrm{Shat}_{\mathcal{H}}(S)$.

Now, consider a set $T \in \mathrm{Shat}_{\mathcal{H}_+}(S) \cap \mathrm{Shat}_{\mathcal{H}_-}(S)$, so that it's been double-counted. Then note that $T' = T \cup \{x\}$ is not in either $\mathrm{Shat}_{\mathcal{H}_+}(S)$ or $\mathrm{Shat}_{\mathcal{H}_-}(S)$, since these classes cannot shatter $\{x\}$ and so can't shatter a superset of $\{x\}$ either. But $\mathcal{H}$ can shatter T': there's a hypothesis in $\mathcal{H}_-$ to achieve any desired labeling with $h(x) = 0$ (since $T \in \mathrm{Shat}_{\mathcal{H}_-}(S)$), and likewise there's a hypothesis in $\mathcal{H}_+$ for any labeling with $h(x) = 1$. So $T' \in \mathrm{Shat}_{\mathcal{H}}(S)$. Also, each such double-counted T corresponds to a different T', since we're adding the same x to each. Thus

$$\left|\mathrm{Shat}_{\mathcal{H}_+}(S) \cap \mathrm{Shat}_{\mathcal{H}_-}(S)\right| \leq \left|\mathrm{Shat}_{\mathcal{H}}(S) \setminus \left(\mathrm{Shat}_{\mathcal{H}_+}(S) \cup \mathrm{Shat}_{\mathcal{H}_-}(S)\right)\right|,$$

and so $\left|\mathcal{H}|_S\right| \leq |\mathrm{Shat}_{\mathcal{H}}(S)|$ as desired. □

Lecture 8
October 12, 2022

Proof of Sauer-Shelah, Lemma 6.8. To bound the number of shattered subsets of S in Lemma 6.10, recall there can't possibly be any with size larger than $d = \mathrm{VCdim}(\mathcal{H})$; the number of sets it can shatter is thus upper-bounded by the number of subsets of S of size at most d, which is just $\sum_{i=0}^{d} \binom{n}{i}$ for $n = |S|$. □

Proof of Corollary 6.9. We need to show that $\sum_{i=0}^{d} \binom{n}{i} \leq \left(\frac{en}{d}\right)^d$ for $n \geq d$. We can do this by

$$
\begin{aligned}
\sum_{i=0}^{d} \binom{n}{i} &\leq \sum_{i=0}^{d} \binom{n}{i} \left(\frac{n}{d}\right)^{d-i} && \text{multiply each term by} \geq 1 \\
&\leq \sum_{i=0}^{n} \binom{n}{i} \left(\frac{n}{d}\right)^{d-i} && \text{add nonnegative terms} \\
&= \left(\frac{n}{d}\right)^{d} \sum_{i=0}^{n} \binom{n}{i} \left(\frac{d}{n}\right)^{i} \\
&= \left(\frac{n}{d}\right)^{d} \left(1 + \frac{d}{n}\right)^{n} && \text{binomial theorem} \\
&\leq \left(\frac{n}{d}\right)^{d} e^{d} && 1 + x \leq \exp(x). \qquad \square
\end{aligned}
$$

6.3 *VC dimension and generalization*

Remember way back to (4.8), where we showed that for ±1 binary classifiers,

$$\operatorname{Rad}(\mathcal{H}|_{S_x}) \leq \sqrt{\frac{2}{n} \log \left|\mathcal{H}|_{S_x}\right|}.$$

Thus using (4.2) for the Rademacher complexity of binary classifiers and the symmetrization result from Section 4.1,

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^n} \sup_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - L_S(h) \leq \mathbb{E} \sqrt{\frac{2}{n} \left|\mathcal{H}|_{S_x}\right|}.$$

By definition, we can bound this with the growth function: $\left|\mathcal{H}|_{S_x}\right| \leq \Pi_{\mathcal{H}}(n)$, and then bound that based on the VC dimension with Corollary 6.9: if $\operatorname{VCdim}(\mathcal{H}) = d$,

$$\mathop{\mathbb{E}}_{S \sim \mathcal{D}^n} \sup_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - L_S(h) \leq \sqrt{\frac{2d}{n} [1 + \log n - \log d]}. \tag{6.1}$$

Now we've dropped the distribution dependence completely on the right-hand side. To show that ERM PAC-learns $\mathcal{H}$ with finite VC dimension, the only thing left to do is to turn this expectation bound into a high-probability bound.

6.4 *High-probability bounds for generalization*

It turns out that, for bounded losses, *any* expectation bound implies a high-probability bound. Let $\Phi(S) = \sup_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - L_S(h)$ be the worst-case generalization gap.

If we could argue that $\Pr_S(\Phi(S) \geq 0) = 1$, then we could use Markov's inequality (Proposition 3.5) to say that $\Phi(S) \leq \frac{1}{\delta} \mathbb{E}\, \Phi(S)$ with probability at least $1 - \delta$. That condition will be true in many cases (e.g. for 0-1 loss binary classification if $\mathcal{H}$ is symmetric), but the $\frac{1}{\delta}$ multiplicative rate is really bad anyway. We can instead show a sub-Gaussian type rate, as we'll do now, based on a new tool: McDiarmid's inequality.

6.11 DEFINITION. We say a function $f : \mathcal{X}^n \to \mathbb{R}$ has *bounded differences* with parameters $(c_1, \dots, c_n)$ if for all $i \in [n]$, we have

$$\sup_{x \in \mathcal{X}^n} \sup_{x^{(k)} \in \mathcal{X}^n : \forall j \neq i, x_j = x_j^{(i)}} \left| f(x) - f(x^{(i)}) \right| \leq c_i.$$

That is, changing the ith x to something totally different can't change the output of f by more than c_i.

6.12 PROPOSITION (McDiarmid). *Let* $X = (X_1, \dots, X_n)$ *have independent components, and let* f *have bounded differences with parameters* $(c_1, \dots, c_n)$*. Then*

$$\Pr\left(f(X) \geq \mathbb{E}\, f(X) + \varepsilon\right) \leq \exp\left(\frac{-2\varepsilon^2}{\sum_{i=1}^n c_i^2}\right) \qquad \Pr\left(f(X) \leq \mathbb{E}\, f(X) - \varepsilon\right) \leq \exp\left(\frac{-2\varepsilon^2}{\sum_{i=1}^n c_i^2}\right).$$

For a proof, see Nick's randomized algorithms class, or [MRT, Sections D.6-D.7], or [Wai19, Section 2.2].

Solving for ε, we get that with probability at least $1 - \delta$ the deviation is at most $\sqrt{\frac{1}{2} \sum_{i=1}^n c_i^2 \log \frac{1}{\delta}}$. In the common case where $c_i = c/n$, this becomes $c\sqrt{\frac{1}{2n} \log \frac{1}{\delta}}$.

Note that if $f(x) = \frac{1}{n} \sum_i x_i$ for $x_i \in [a, b]$, we have bounded differences with $c_i = (b - a)/n$, in which case the bound becomes identical to the classical version of Hoeffding's inequality (Proposition 3.7 with Proposition 3.2).

For the generalization gap $\Phi(S)$,

$$\begin{aligned} \left|\Phi(S) - \Phi(S^{(i)})\right| &= \left|\sup_{h \in \mathcal{H}} [L_{\mathcal{D}}(h) - L_S(h)] - \sup_{h' \in \mathcal{H}} [L_{\mathcal{D}}(h') - L_{S^{(i)}}(h')]\right| \\ &\leq \sup_{h \in \mathcal{H}} |L_{S^{(i)}}(h) - L_S(h)| \\ &= \frac{1}{n} \sup_{h \in \mathcal{H}} \left|\ell(h, z_i) - \ell(h, z_i')\right| \\ &\leq \frac{b - a}{n} \qquad\qquad \text{if } a \leq \ell(h, z) \leq b. \end{aligned}$$

Thus

6.13 THEOREM. *If $\ell(h,z) \in [a,b]$, then with probability at least $1-\delta$ over $S \sim \mathcal{D}^n$,*

$$\sup_{h\in\mathcal{H}} L_{\mathcal{D}}(h) - L_S(h) \le \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^n} \sup_{h\in\mathcal{H}} [L_{\mathcal{D}}(h) - L_{S'}(h)] + (b-a)\sqrt{\tfrac{1}{2n}\log\tfrac{1}{\delta}}. \tag{6.2}$$

Thus any ERM $\hat{h}_S$ has with probability at least $1-\delta$ that

$$L_{\mathcal{D}}(\hat{h}_S) - \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) \le \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^n} \sup_{h\in\mathcal{H}} [L_{\mathcal{D}}(h) - L_{S'}(h)] + (b-a)\sqrt{\tfrac{2}{n}\log\tfrac{2}{\delta}}.$$

Proof. We just proved the first part above. The ERM part follows from

$$\begin{aligned} L_{\mathcal{D}}(\hat{h}_S) &\le L_S(\hat{h}_S) + \mathop{\mathbb{E}}_{S'} \sup_{h\in\mathcal{H}} [L_{\mathcal{D}}(h) - L_{S'}(h)] + (b-a)\sqrt{\tfrac{1}{2n}\log\tfrac{2}{\delta}} \\ &\le L_S(h^*) + \mathop{\mathbb{E}}_{S'} \sup_{h\in\mathcal{H}} [L_{\mathcal{D}}(h) - L_{S'}(h)] + (b-a)\sqrt{\tfrac{1}{2n}\log\tfrac{2}{\delta}} \\ &\le L_{\mathcal{D}}(h^*) + (b-a)\sqrt{\tfrac{1}{2n}\log\tfrac{2}{\delta}} + \mathop{\mathbb{E}}_{S'} \sup_{h\in\mathcal{H}} [L_{\mathcal{D}}(h) - L_{S'}(h)] + (b-a)\sqrt{\tfrac{1}{2n}\log\tfrac{2}{\delta}}, \end{aligned}$$

using uniform convergence, the definition of ERM, and Lemma 2.5. □

Plugging in Theorem 4.2 to bound the expected worst-case gap:

6.14 COROLLARY. *If $\ell(h,z) \in [a,b]$, then with probability at least $1-\delta$ over $S \sim \mathcal{D}^n$,*

$$\sup_{h\in\mathcal{H}} L_{\mathcal{D}}(h) - L_S(h) \le 2 \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^n} \operatorname{Rad}\left((\ell\circ\mathcal{H})|_{S'_x}\right) + (b-a)\sqrt{\tfrac{1}{2n}\log\tfrac{1}{\delta}},$$

and any ERM $\hat{h}_S$ has with probability at least $1-\delta$ that

$$L_{\mathcal{D}}(\hat{h}_S) - \inf_{h\in\mathcal{H}} L_{\mathcal{D}}(h) \le 2 \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^n} \operatorname{Rad}\left((\ell\circ\mathcal{H})|_{S'_x}\right) + (b-a)\sqrt{\tfrac{2}{n}\log\tfrac{2}{\delta}}.$$

Plugging this together with Equations (4.2) and (6.1) gives:

6.15 COROLLARY. *For a class $\mathcal{H}$ of binary classifiers mapping to $\{-1,1\}$ with $\operatorname{VCdim}(\mathcal{H}) = d$ and ℓ the 0-1 loss, with probability at least $1-\delta$ over the choice of $S \sim \mathcal{D}^n$,*

$$\begin{aligned} \sup_{h\in\mathcal{H}} L_{\mathcal{D}}(h) - L_S(h) &\le \mathop{\mathbb{E}}_{S'\sim\mathcal{D}^n} \operatorname{Rad}\left(\mathcal{H}|_{S'_x}\right) + \sqrt{\tfrac{1}{2n}\log\tfrac{1}{\delta}} \\ &\le \sqrt{\frac{2d}{n}[\log n + 1 - \log d]} + \sqrt{\tfrac{1}{2n}\log\tfrac{1}{\delta}}, \end{aligned}$$

and hence any ERM $\hat{h}_S$ has with probability at least $1 - \delta$ that

$$
\begin{aligned}
L_{\mathcal{D}}(\hat{h}_S) - \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h) &\le \mathop{\mathbb{E}}_{S' \sim \mathcal{D}^n} \operatorname{Rad}\left(\mathcal{H}|_{S'_x}\right) + \sqrt{\tfrac{2}{n} \log \tfrac{2}{\delta}} \\
&\le \sqrt{\frac{2d}{n}[\log n + 1 - \log d]} + \sqrt{\tfrac{2}{n} \log \tfrac{2}{\delta}}.
\end{aligned}
$$

6.5 *The fundamental theorem of statistical learning*

For any loss function bounded in $[a, b]$ and hypothesis class $\mathcal{H}$ with $\operatorname{VCdim}(\mathcal{H}) = d$, plugging (6.1) into (6.2) gives that with probability at least $1 - \delta$,

Note that $1 - \log d \le 0$ for $d \ge 3$, so we can replace that term in brackets with just $\log n$ when $d \ge 3$.

$$
\sup_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - L_S(h) \le \sqrt{\frac{2d}{n}[\log n + 1 - \log d]} + (b - a)\sqrt{\tfrac{1}{2n} \log \tfrac{1}{\delta}}. \tag{6.3}
$$

As long as $\frac{n}{\log n + 1 - \log d} > \frac{d}{2\varepsilon^2}$ and $n > \frac{(b-a)^2 \log(2/\delta)}{4\varepsilon^2}$, we can see that the generalization gap is bounded as $\sup_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - L_S(h) \le \varepsilon/2$ with probability at least $\delta/2$. This n will also be large enough to get $\Pr(L_S(h^*) \le L_{\mathcal{D}}(h^*) + \varepsilon/2) \ge 1 - \delta/2$ with standard Hoeffding, giving like in Section 2.4 that an ERM $\hat{h}_S$ satisfies

$$
L_{\mathcal{D}}(\hat{h}_S) \le L_S(\hat{h}_S) + \tfrac{\varepsilon}{2} \le L_S(h^*) + \tfrac{\varepsilon}{2} \le L_{\mathcal{D}}(h^*) + \varepsilon
$$

with probability at least $1 - \delta$: any ERM algorithm agnostically PAC-learns $\mathcal{H}$.

This $\log n$ is kind of annoying; you can find a big enough n using an argument like Lemma A.1 of [SSBD]. Better yet, you can drop the $\log n$ using a more advanced kind of argument called chaining*; we might cover this later in the course.*

6.16 THEOREM (Fundamental Theorem of Statistical Learning). *For $\mathcal{H}$ a class of functions $h : \mathcal{X} \to \{0, 1\}$ and with the 0-1 loss, the following are equivalent:*

This name is only, as far as I know, used by [SSBD].

1. *Uniform convergence: for all $\varepsilon, \delta \in (0, 1)$, we have that $\sup_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - L_S(h) < \varepsilon$ with probability at least $1 - \delta$ as long as $n \ge n^{\mathrm{UC}}(\varepsilon, \delta) < \infty$.*
2. *Any ERM rule agnostically PAC-learns $\mathcal{H}$.*
3. *$\mathcal{H}$ is agnostically PAC learnable.*
4. *Any ERM rule PAC-learns $\mathcal{H}$.*
5. *$\mathcal{H}$ is PAC learnable.*
6. $\operatorname{VCdim}(\mathcal{H}) < \infty$.

[SSBD] use two-sided uniform convergence: in the setting of the theorem here, one-sided bounds imply two-sided ones, but (a) one-sided is what we really use, and (b) in more general settings the distinction can matter.

Proof. We just showed in (6.3) that 6 implies 1.

1 implying 2 is by Section 2.4, with the argument just repeated above.

2 implying 3, and 4 implying 5, are immediate.

2 implying 4, and 3 implying 5, is A2 Q1a (which should be straightforward).

Finally, Corollary 5.3 shows that 5 implies 6. □

It's worth emphasizing that this theorem is only for 0-1 loss on binary classification, but various parts of it still hold more broadly.

7 SRM AND NONUNIFORM LEARNABILITY

Lecture 9
October 17, 2022

We now understand the (agnostic) PAC learnability of a fixed hypothesis class $\mathcal{H}$ pretty well, at least for 0-1 loss binary classification, which says that e.g. ERM will do not too much worse than the best thing in $\mathcal{H}$ with enough samples. This lets us control the *estimation error* in the following decomposition, which we need to trade off against the hard-to-understand *approximation error* of how close the class can get to the best-possible irreducible Bayes error L^*:

$$\underbrace{\mathrm{L}_{\mathcal{D}}(\hat{h}) - \mathrm{L}^*_{\mathcal{D}}}_{\text{excess error}} = \underbrace{\left(\mathrm{L}_{\mathcal{D}}(\hat{h}) - \inf_{h\in\mathcal{H}} \mathrm{L}_{\mathcal{D}}(h)\right)}_{\text{estimation error}} - \underbrace{\left(\inf_{h\in\mathcal{H}} \mathrm{L}_{\mathcal{D}}(h) - \mathrm{L}^*_{\mathcal{D}}\right)}_{\text{approximation error}}.$$

Although we can analyze this approximation error gap in some cases if we assume things about the form of $\mathcal{D}$, it's generally hard to know for any specific problem, and there's not usually a clear way to estimate it (or just $\inf_{h\in\mathcal{H}} \mathrm{L}_{\mathcal{D}}(h)$) from data, either.

The practical solution is generally to just try a bunch of different $\mathcal{H}$ and/or a bunch of different learning algorithms, then pick the best based on a validation set V. This is a good idea in practice, and we can make some theoretical guarantees on its generalization based on L_{V} being close to $\mathrm{L}_{\mathcal{D}}$; more in the homework. But it's still hard to use that approach to handle the approximation error.

7.1 *Structural Risk Minimization*

SRM says: let's use a *huge* $\mathcal{H}$, one where the approximation error is going to be small or maybe even zero. This will probably mean we have infinite VC dimension, bad Rademacher complexity, etc. in $\mathcal{H}$. But let's decompose

$$\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2 \cup \cdots = \bigcup_{k\in\mathbb{N}} \mathcal{H}_k.$$

For instance, we might have $\mathcal{H}_k$ the set of decision trees of depth k, the set of degree-k polynomials, or the set of linear classifiers with $\|w\| \le 2^k$. We're going to assume that *each* $\mathcal{H}_k$ has uniform convergence:

$$\forall k \in \mathbb{N}. \quad \Pr_{\mathrm{S}\sim\mathcal{D}^n}\left(\sup_{h\in\mathcal{H}_k} \mathrm{L}_{\mathcal{D}}(h) - \mathrm{L}_{\mathrm{S}}(h) \le \varepsilon_k(n, \delta)\right) \ge 1 - \delta \tag{7.1}$$

for functions ε_k satisfying that for all k and all $\delta \in (0, 1)$, $\lim_{n\to\infty} \varepsilon_k(n, \delta) = 0$.

We'll also need a set of weights $w_k \ge 0$ such that $\sum_{k=1}^{\infty} w_k \le 1$; a typical choice is $6/(\pi^2 k^2) \approx 0.61/k^2$, since $\sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$.

This is the problem that made Euler famous.

7.1 PROPOSITION. *Let $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2 \cup \ldots$ satisfy (7.1), and let $w_k \geq 0$ have $\sum_{k=1}^{\infty} w_k \leq 1$. Then for any $\mathcal{D}$, with probability at least $1 - \delta$ over the choice of $S \sim \mathcal{D}^n$, we have*

$$\forall h \in \mathcal{H}. \quad L_{\mathcal{D}}(h) \leq L_S(h) + \min_{k : h \in \mathcal{H}_k} \varepsilon_k(n, \delta w_k).$$

Proof. Similarly to A2 Q1c, we just allocate a failure probability of δw_k to each class, giving total failure probability of at most $\delta \sum_k w_k \leq \delta$. □

SRM is then the algorithm that minimizes this upper bound $L_{\mathcal{D}}(h)$:

7.2 DEFINITION. Given bounds on a decomposition of $\mathcal{H}$ as in (7.1), and weights $w_k \geq 0$ with $\sum w_k \leq 1$ and $\bigcup_{k : w_k > 0} \mathcal{H}_k = \mathcal{H}$, *structural risk minimization* is given by

$$\mathrm{SRM}_{\mathcal{H}}(S) \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \left[L_S(h) + \varepsilon_{k_h}(n, \delta w_{k_h}) \right] \qquad \text{where } k_h \in \operatorname*{arg\,min}_{k : h \in \mathcal{H}_k} \varepsilon_k(n, w_k \delta).$$

Typically, $k_h = \min\{k : h \in \mathcal{H}_k\}$.

We can implement this potentially-infinite minimization by a finite number of calls to an "ERM oracle", as long as our loss is lower-bounded by $a \leq \ell(h, z)$ (typically $a = 0$):

```
function $\mathrm{SRM}_{\mathcal{H}}(S)$
    best $\leftarrow \infty$
    for $k = 1, 2, \ldots$ do
        $h_k \leftarrow \mathrm{ERM}_{\mathcal{H}_k}(S)$
        cand $\leftarrow L_S(h_k) + \varepsilon_k(n, w_k \delta)$
        if cand $<$ best then
            $\hat{h} \leftarrow h_k$
            best $\leftarrow$ cand
        if $\min_{k' > k} a + \varepsilon_{k'}(n, w_{k'} \delta) >$ best then
            break
    return $\hat{h}$
```

Note that if we "decompose" as $\mathcal{H}_1 = \mathcal{H}$, then SRM becomes just $\mathrm{ERM}_{\mathcal{H}}$.

7.3 THEOREM. *Let $h^* \in \mathcal{H}$ be any fixed hypothesis in the setup of Definition 7.2, and let $a \leq \ell(h, z) \leq b$ for all $h \in \mathcal{H}$, $z \in \mathcal{Z}$. Then, with probability at least $1 - \delta$, SRM based on n samples satisfies*

$$L_{\mathcal{D}}(\mathrm{SRM}_{\mathcal{H}}(S)) \leq L_{\mathcal{D}}(h^*) + \varepsilon_{k_{h^*}}\left(n, \tfrac{1}{2} w_{k_{h^*}} \delta\right) + (b - a)\sqrt{\tfrac{1}{2n} \log \tfrac{2}{\delta}}.$$

Proof. Let $\hat{h}_S = \mathrm{SRM}_{\mathcal{H}}(S)$. We have that

$$\begin{aligned} L_{\mathcal{D}}(\hat{h}_S) &\le L_S(\hat{h}) + \varepsilon_{k_{\hat{h}_S}}(n, w_{k_{\hat{h}_S}}\delta/2) && \text{by Proposition 7.1, prob} \ge \frac{\delta}{2} \\ &\le L_S(h^*) + \varepsilon_{k_{h^*}}(n, w_{k_{h^*}}\delta/2) && \text{by def of SRM;} \end{aligned}$$

the conclusion follows by applying Lemma 2.5 with probability $\delta/2$ to upper-bound $L_S(h^*)$. □

Note that the number of samples n required to achieve a particular error ε depends on the choice of h^*, unlike in PAC learning!

7.2 *Nonuniform learnability*

7.4 DEFINITION. An algorithm $A(S)$ (ε, δ)-*competes with* a hypothesis h if it satisfies $\Pr_{S\sim\mathcal{D}^n}(L_{\mathcal{D}}(A(S)) \le L_{\mathcal{D}}(h) + \varepsilon) \ge 1 - \delta$.

7.5 DEFINITION. An algorithm A *nonuniformly learns* $\mathcal{H}$ there is a finite sample complexity function $n(\varepsilon, \delta, h)$ such that for all $\varepsilon, \delta \in (0, 1)$ and $h \in \mathcal{H}$ and any $\mathcal{D}$, $A(S)$ (ε, δ)-competes with h.

7.6 DEFINITION. A hypothesis class $\mathcal{H}$ is *nonuniformly learnable* if there exists an algorithm A which nonuniformly learns $\mathcal{H}$.

Theorem 7.3 establishes that SRM nonuniformly learns any $\mathcal{H}$ which we can decompose into a countable union of $\mathcal{H}_k$ which each allow for uniform convergence.

In fact, for binary classifiers with 0-1 loss, SRM nonuniformly learns any $\mathcal{H}$ which is nonuniformly learnable:

7.7 PROPOSITION. *If $\mathcal{H}$ of binary classifiers is nonuniformly learnable under the 0-1 loss, it can be written as a countable union of $\mathcal{H}_k$ with finite VC dimension.*

Proof. Define

$$\mathcal{H}_k = \left\{h \in \mathcal{H} : n\left(\tfrac{1}{8}, \tfrac{1}{7}, h\right) \le k\right\},$$

where $n(\varepsilon, \delta, h)$ is the sample complexity function of an algorithm A that nonuniformly learns $\mathcal{H}$. Then $\mathcal{H} = \bigcup_{k\ge1} \mathcal{H}_k$.

For any k, consider a $\mathcal{D}$ which is realizable under $\mathcal{H}_k$. If $\mathcal{H}_k$ is nonempty, there must be some such distribution (and otherwise $\mathrm{VCdim}(\mathcal{H}_k) = 0$). Then there exists an $h^* \in \mathcal{H}_k$ with zero loss, and since $A(S)$ competes with that h^*, $\Pr_{S\sim\mathcal{D}^n}(L_{\mathcal{D}}(A(S)) \le \frac{1}{8}) \ge \frac{6}{7}$. But by the No Free Lunch theorem (in particular Corollary 6.2), the fact that we can do this for *every* realizable $\mathcal{D}$ implies that $\mathcal{H}_k$ has finite VC dimenison. □

Note that the set of all measurable $\mathcal{H}$ is *not* a countable union of finite-VC classes.

7.3 *SRM based on Rademacher complexity*

Plugging Corollary 6.15 into (7.1), we have that, for 0-1 loss on $\mathcal{H}$ mapping to $\{-1, 1\}$,

$$L_{\mathcal{D}}(h) \le L_S(h) + \mathop{\mathbb{E}}_{S'} \operatorname{Rad}\left(\mathcal{H}_{k_h}|_{S'_x}\right) + \sqrt{\frac{1}{2n} \log \frac{1}{w_{k_h}\delta}}.$$

Let's choose the concrete set of weights $w_k = 6/(\pi^2 k^2)$. We can make things look a little nicer by noticing that

$$\log \frac{1}{w_{k_h}\delta} = \log \frac{1}{2w_{k_h}} + \log \frac{2}{\delta} = \log \frac{\pi^2 k_h^2}{12} + \log \frac{2}{\delta} \le 2 \log k + \log \frac{2}{\delta}$$

$$\sqrt{\frac{1}{2n} \log \frac{1}{w_{k_h}\delta}} \le \sqrt{\frac{1}{n} \log k_h + \frac{1}{2n} \log \frac{2}{\delta}} \le \sqrt{\frac{1}{n} \log k_h} + \sqrt{\frac{1}{2n} \log \frac{2}{\delta}}.$$

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

$$\forall h \in \mathcal{H}. \quad L_{\mathcal{D}}(h) \le L_S(h) + \mathop{\mathbb{E}}_{S'} \operatorname{Rad}\left(\mathcal{H}_{k_h}|_{S'_x}\right) + \sqrt{\frac{1}{n} \log k_h} + \sqrt{\frac{1}{2n} \log \frac{2}{\delta}}.$$

Using this upper bound in SRM gives a nice version that doesn't depend on δ:

$$\hat{h}_S \in \mathop{\arg\min}_{h \in \mathcal{H}} L_S(h) + \mathop{\mathbb{E}}_{S'} \operatorname{Rad}\left(\mathcal{H}_{k_h}|_{S'_x}\right) + \sqrt{\frac{1}{n} \log k_h}. \tag{7.2}$$

The proof of Theorem 7.3 then establishes that

$$L_{\mathcal{D}}(\hat{h}_S) \le L_{\mathcal{D}}(h^*) + \mathop{\mathbb{E}}_{S'} \operatorname{Rad}\left(\mathcal{H}_{k_h}|_{S'_x}\right) + \sqrt{\frac{1}{n} \log k_h} + \sqrt{\frac{2}{n} \log \frac{2}{\delta}}.$$

Comparing to the bound Corollary 6.15 for running ERM on the "right" $\mathcal{H}_{k_h}$ in the first place, it's only worse by a factor of $\sqrt{\frac{1}{n} \log k_h}$ – usually not a big deal, if we've picked our $\mathcal{H}_k$ appropriately!

7.4 *Singleton Classes*

Lecture 10
October 19, 2022

Suppose we have a countable $\mathcal{H} = \{h_1, h_2, \dots\}$. Then we could partition it into *singleton* sub-classes, $\mathcal{H}_k = \{h_k\}$. Denoting the weight for the class consisting of the hypothesis h by w_h, each of these $\mathcal{H}_k$ have "uniform convergence" via a simple Hoeffding bound with

$$\varepsilon_k(n, w_h\delta) \le (b-a)\sqrt{\frac{2}{n} \log \frac{1}{w_h\delta}} \le (b-a)\sqrt{\frac{2}{n} \log \frac{1}{w_h}} + (b-a)\sqrt{\frac{2}{n} \log \frac{1}{\delta}},$$

splitting out the dependence on δ for simplicity as in the previous section. SRM then becomes

$$\mathrm{SRM}_{\mathcal{H}}(\mathrm{S}) \in \operatorname*{arg\,min}_{h\in\mathcal{H}} \mathrm{L}_{\mathrm{S}}(h) + \sqrt{\frac{2}{n}\log\frac{1}{w_h}},$$

and this has the guarantee by Theorem 7.3 that

$$\mathrm{L}_{\mathcal{D}}(\mathrm{SRM}_{\mathcal{H}}(\mathrm{S})) \le \mathrm{L}_{\mathcal{D}}(h^*) + (b-a)\sqrt{\frac{2}{n}\log\frac{1}{w_{h^*}}} + 2(b-a)\sqrt{\frac{2}{n}\log\frac{2}{\delta}}.$$

This is interesting to compare to A2 Q1(c).

But... how should we set w_h?

7.5 *Minimum Description Length*

One popular way to decide on weights is based on choosing some *prefix-free binary language* to determine the hypotheses: for example, the binary representation of a `gzip`ed Python program implementing that hypothesis. Then we can choose a weight according to the following result:

7.8 PROPOSITION (Kraft's inequality). *If $\mathcal{S} \subseteq \{0,1\}^*$ is prefix-free (there are no $\sigma \ne \sigma' \in \mathcal{S}$ such that σ is a prefix of σ'), then*

$$\sum_{\sigma\in\mathcal{S}} 2^{-|\sigma|} \le 1.$$

Proof. Define the following random process: starting with the empty string, add either a 0 or a 1 with equal probability. If the current string is in $\mathcal{S}$, terminate; if no element of $\mathcal{S}$ begins with the current string, also terminate; otherwise, repeat. Since $\mathcal{S}$ is prefix-free, this process hits any string $\sigma \in \mathcal{S}$ with probability $2^{-|\sigma|}$; these probabilities must sum to at most one. □

Thus, we can choose a representation for $\mathcal{H}$ and assign $w_h = 2^{-|h|}$. This gives

$$\mathrm{MDL}_{\mathcal{H}}(\mathrm{S}) \in \operatorname*{arg\,min}_{h\in\mathcal{H}} \mathrm{L}_{\mathrm{S}}(h) + \sqrt{\frac{2\log 2}{n}|h|}$$

$$\mathrm{L}_{\mathcal{D}}(\mathrm{MDL}_{\mathcal{H}}(\mathrm{S})) \le \mathrm{L}_{\mathcal{D}}(h^*) + (b-a)\sqrt{\frac{2\log 2}{n}|h^*|} + 2(b-a)\sqrt{\frac{2}{n}\log\frac{2}{\delta}}.$$

This is one formalization of Occam's razor: if there are multiple explanations of the data ($\mathrm{L}_{\mathrm{S}}(h_1) = 0 = \mathrm{L}_{\mathrm{S}}(h_2)$), prefer the simplest one (the one with shortest explanation).

But we need to *pre-commit* to a notion of description length before seeing the data. A nice analogy: `codegolf.stackexchange.com`, a site where people compete to find the shortest implementation of a program doing some task, prohibits by default any language written after the contest was started.

8 CONSISTENCY

So far we've studied:

- (Realizable) PAC learning.
  A competes with any h^* on any realizable $\mathcal{D}$ with $n(\varepsilon, \delta)$ samples.
  Binary 0-1 loss: ERM works iff $\operatorname{VCdim}(\mathcal{H}) < \infty$.
- Agnostic PAC learning.
  A competes with any h^* on any $\mathcal{D}$ with $n(\varepsilon, \delta)$ samples.
  Binary 0-1 loss: ERM works iff $\operatorname{VCdim}(\mathcal{H}) < \infty$.
- Nonuniform learning.
  A competes with h^* on any $\mathcal{D}$ with $n(\varepsilon, \delta, h^*)$ samples.
  Binary 0-1 loss: SRM can work iff $\mathcal{H}$ is countable union of finite-VC $\mathcal{H}_k$.

Now we're going to add a new one: *consistency*, where A competes with h^* with $n(\varepsilon, \delta, h^*, \mathcal{D})$ samples.

Sorry, I'm going to come back and fill this in soon!

9 LINEAR CLASSIFIERS AND MARGINS

Remember that a linear classifier is given by $h(x) = \operatorname{sgn}(w^\mathsf{T} x + b \geq 0)$; a *homogeneous* linear classifier is $h(x) = \operatorname{sgn}(w^\mathsf{T} x)$. You can reduce from a general linear classifier to a homogeneous one by changing the data: use $\tilde{x} = \begin{bmatrix} 1 & x \end{bmatrix} \in \mathbb{R}^{d+1}$ and $\tilde{w} = \begin{bmatrix} b & w \end{bmatrix}$. So, for now, we're only going to worry about homogeneous classifiers. (Sometimes adding an intercept back in ends up being nontrivial, though – pay attention to that step!)

Lecture 11
October 24, 2022
Lecture 12
October 26, 2022
(These two lectures were pretty intermixed, because I was pretty disorganized the first time!)

Letting $\mathcal{H} = \{x \mapsto \operatorname{sgn}(w^\mathsf{T} x) : w \in \mathbb{R}^d\}$, we know from Proposition 6.5 that $\operatorname{VCdim}(\mathcal{H}) = d$, and hence for the 0-1 loss, Corollary 6.15 gives each of the following with probability at least $1 - \delta$:

$$\sup_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - L_S(h) \leq \sqrt{\frac{2d}{n}[\log n + 1 - \log d]} + \sqrt{\frac{1}{2n} \log \frac{1}{\delta}}$$

$$L_{\mathcal{D}}(\hat{h}_S) - \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h) \leq \sqrt{\frac{2d}{n}[\log n + 1 - \log d]} + \sqrt{\frac{2}{n} \log \frac{2}{\delta}}$$

where $\hat{h}_S$ is an ERM.

So, for any fixed d, this means that ERM will work once n is big enough. But sometimes we have a really big d, and this only gives us very slow convergence in n. Sometimes we even have an *infinite* d, and then this doesn't tell us anything at all; this is often the case with kernel methods, as we'll see later.

Often, though, when d is big we end up with a hypothesis h that has *small norm*. This might be because we explicitly *try* to find a small-norm solution, and/or because our optimization algorithm implicitly prefers small-norm solutions; more on both situations later in the course.

To analyze that, let's define $\mathcal{H}_B = \{x \mapsto w^\mathsf{T} x : \|w\| \le B\}$ – note this is a class that outputs continuous real numbers, not "hard" classifications, but we can get a class of binary classifiers out with $\operatorname{sgn} \circ \mathcal{H}_B$.

But note that $\operatorname{VCdim}(\operatorname{sgn} \circ \mathcal{H}_B) = d$ for any B: since VC dimension is worst-case over all possible input distributions, we can take any set that the full $\mathcal{H}$ can shatter and just scale it up so that we can still shatter it with a small-norm predictor. So we'll need a distribution-dependent notion of complexity to do better than this; something like Rademacher complexity.

Now, recall from (4.1) that $\mathbb{E}_S \operatorname{Rad}\left(\mathcal{H}_B\big|_{S_x}\right) \le \frac{B}{\sqrt{n}}\sqrt{\mathbb{E}\|x\|^2}$. To use this in a generalization bound for the 0-1 loss, though, we'd need to bound $\mathbb{E}_S \operatorname{Rad}\Big((\ell_{0-1} \circ \operatorname{sgn} \circ \mathcal{H}_B)\big|_{S_x}\Big)$. The only way we really know how to deal with "peeling" off functions like that is Lipschitz functions, with Lemma 4.4. But $\ell_{0-1} \circ \operatorname{sgn}$ isn't Lipschitz. (In Section 4.7 we pretended ℓ_{0-1} was Lipschitz, but we could only do that because our $\mathcal{H}$ mapped to $\{-1, 1\}$; we can't play any similar trick with sgn for $\mathcal{H}_B$ mapping to $\mathbb{R}$.)

Another problem is that computing the ERM with respect to 0-1 loss, in the case where $L_\mathcal{D}(h^*) > 0$, is actually NP-hard [BS00]! (You can reduce a SAT variant to it.)

9.1 *Surrogate losses*

We can work around both problems with *surrogate losses*.

One version we've already talked about is by using the logistic loss, which is 1-Lipschitz, so we can apply Lemma 4.4. But then we'd be bounding things only in terms of the logistic loss, which is hard to relate directly to accuracy; if we care more about accuracy, it's difficult to say anything.

The logistic loss isn't "naturally" bounded, but if we assume a hard bound on $\|x\|$ then we can upper-bound the possible logistic loss for anything in $\mathcal{H}_B$.

Instead, suppose that we have some loss ℓ_{surr} such that $\ell_{surr}(h, z) \ge \ell_{0-1}(h, z)$ for all h, z. Then $L_\mathcal{D}^{surr}(h) = \mathbb{E}_z \ell_{surr}(h, z) \ge \mathbb{E}_z \ell_{0-1}(h, z) = L_\mathcal{D}^{0-1}(h)$. Thus, if we pick such a surrogate loss that's also ρ-Lipschitz and bounded in $[a, b]$, we get by combining Theorems 4.2 and 6.13 and Lemma 4.4 that

$$L_\mathcal{D}^{0-1}(h) \le L_\mathcal{D}^{surr}(h) \le L_S^{surr}(h) + 2\rho \mathop{\mathbb{E}}_S \operatorname{Rad}(\mathcal{H}|_{S_x}) + (b-a)\sqrt{\frac{1}{2n}\log\frac{1}{\delta}}.$$

Ideally, we'd have a surrogate loss that also makes ERM easy to solve with respect to that loss; if $L_S^{surr}(h)$ is small, this would give small 0-1 loss as well. We'll hold off on that problem for a bit, though, and just worry about uniform convergence for now.

9.2 *Analysis with ramp loss*

One natural way to get a bounded, 1-Lipschitz upper bound on the 0-1 loss is with the *ramp loss*

$$\ell_{ramp}(h,(x,y)) = \lambda_y^{ramp}(h(x)) = \begin{cases} 1 & yh(x) \le 0 \\ 1 - yh(x) & 0 \le yh(x) \le 1 \\ 0 & 1 \le yh(x) \end{cases}.$$

That is, if we make an incorrect prediction $\operatorname{sgn}(h(x)) \neq y$, we get 1 loss. If we make a correct prediction and are confident enough in it, $|h(x)| \ge 1$, we get 0 loss. But in between, we incur some partial loss even if we're right if we're not confident enough. This is indeed an upper bound on the 0-1 loss, λ_y^{ramp} is 1-Lipschitz, and it's bounded in $[0,1]$, so we have with probability at least $1-\delta$ for all h in a real-valued $\mathcal{H}$ that

$$L_{\mathcal{D}}^{0\text{-}1}(h) \le L_{\mathcal{D}}^{ramp}(h) \le L_{S}^{ramp}(h) + 2\mathop{\mathbb{E}}_{S} \operatorname{Rad}(\mathcal{H}|_{S_x}) + \sqrt{\frac{1}{2n}\log\frac{1}{\delta}}. \tag{9.1}$$

Now let's look at linear classifiers, and assume $\mathbb{E}\|x\|^2 \le R^2$. For predictors from $\mathcal{H}_B = \{x \mapsto w^\mathsf{T} x : \|w\| \le B\}$, we have

$$L_{\mathcal{D}}^{0\text{-}1}(h) \le L_{S}^{ramp}(h) + \frac{2RB}{\sqrt{n}} + \sqrt{\frac{1}{2n}\log\frac{1}{\delta}}. \tag{9.2}$$

What about that ramp loss term?

One nice special case when the distribution is *separable* with margin 1, meaning that there's a w^* such that $\Pr_{(x,y)\sim\mathcal{D}}\left(yx^\mathsf{T} w^* \ge 1\right) = 1$. Then we know that $\inf_{h\in\mathcal{H}_{\|w^*\|}} L_{\mathcal{D}}^{ramp}(h) = 0$. This tells us that any predictor $\hat{h} = h_{\hat{w}} = \operatorname{sgn}(\hat{w}^\mathsf{T} x)$ with $L_S^{ramp}(\hat{h}) = 0$ and $\|\hat{w}\| \le \|w^*\|$ has

$$L_{\mathcal{D}}^{0\text{-}1}(\hat{h}) \le \frac{2R\|w^*\|}{\sqrt{n}} + \sqrt{\frac{1}{2n}\log\frac{1}{\delta}}.$$

Since $L_S^{ramp}(h^*) = 0$ in the separable case, the *minimum-norm interpolator* $\hat{w} = \arg\min_{w : L_S^{ramp}(h_w)=0} \|w\|$ will have $\|\hat{w}\| \le \|w^*\|$ and so satisfies this bound for separable data, meaning it's a decent learning algorithm on separable data. We'll think about this algorithm more in a moment.

The actual value of the bound, though, depends on $\|w^*\|$, which we don't know. We can use an argument like we used for SRM to get a bound that only depends on $\|\hat{w}\|$:

9.1 PROPOSITION. *Let $\mathbb{E}_{(x,y)\sim\mathcal{D}}\|x\|^2 \le R^2$, and $h_w(x) = \operatorname{sgn}(\hat{w}^\mathsf{T} x)$. Then for any $\delta \in (0,1)$ and $r > 0$ fixed before seeing the data, we have with probability at least*

You can think of either $r = 1$ or r small; it's an annoying technicality. (The best choice is $r = \|\hat{w}\|$, but we can't choose it based on data.) Theorem 26.14 of [SSBD] doesn't have it, but that's because that theorem is wrong.

$1 - \delta$ *over the choice of sample* $S \sim \mathcal{D}^n$ *that for all* $w \in \mathbb{R}^d$*,*

$$L_{\mathcal{D}}^{0-1}(h_w) \le L_S^{ramp}(h_w) + \frac{1}{\sqrt{n}}\left[4R\max\{r, \|w\|\} + \max\left\{0, \sqrt{\log\log_2 \frac{2\,\|w\|}{r}}\right\} + \sqrt{\frac{1}{2}\log\frac{2}{\delta}}\right].$$

Proof. Define $B_i = r2^i$ and $\delta_i = \frac{6\delta}{\pi^2 i^2}$ for all $i \ge 1$, noting $\sum_{i=1}^{\infty} \delta_i = \delta$. For each i, it holds with probability at least $1 - \delta_i$ that

$$\forall h \in \mathcal{H}_{B_i}. \quad L_{\mathcal{D}}^{0-1}(h) \le L_S^{ramp}(h) + \frac{2B_i R}{\sqrt{n}} + \sqrt{\frac{1}{2n}\log\frac{1}{\delta_i}}.$$

For any $h = h_w$, let $i_w = \max\left\{1, \left\lceil \log_2 \frac{\|w\|}{r} \right\rceil\right\}$; then $h \in \mathcal{H}_{B_{i_w}}$. We can then see

$$B_{i_w} = r2^{i_w} = r\max\left\{2, 2^{\left\lceil \log_2 \frac{\|w\|}{r}\right\rceil}\right\} \le r\max\{2, 2\frac{\|w\|}{r}\} = 2\max\{r, \|w\|\}$$

and

$$\frac{1}{\delta_{i_w}} = \frac{\pi^2 i_w^2}{6\delta} = \frac{\pi^2/6}{\delta}\max\left\{1, \left\lceil \log_2 \frac{\|w\|}{r}\right\rceil\right\}^2.$$

Using that $\pi^2/6 < 2$ and $\lceil \log_2 a \rceil < \log_2(a) + 1 = \log_2(2a)$,

$$\log\frac{1}{\delta_{i_w}} \le \log\frac{2}{\delta} + 2\max\left\{0, \log\log_2 \frac{2\,\|w\|}{r}\right\}.$$

Taking a union bound over all $i \ge 1$, specializing to $\hat{h}$ which has $L_s^{ramp}(\hat{h}) = 0$, and slightly simplifying, we get the desired result. □

If we pick an r that's much smaller than any reasonable $\|\hat{w}\|$ but not so small that $\log\log_2 \frac{1}{r}$ is significant, we get for $\hat{w}$ that separate the sample S with margin 1 that, roughly, $L_{\mathcal{D}}^{0-1}(\hat{w}) = \mathcal{O}(\|\hat{w}\|/\sqrt{n})$ with high probability. This reinforces that the minimum-norm interpolator seems like a good idea.

9.3 *Hard SVMs and margin maximization*

We've argued that it seems to make sense to compute the minimum-norm interpolator

$$\hat{w} = \arg\min_w \|w\| \quad \text{s.t. } L_S^{ramp}(h_w) = 0.$$

Expanding out the definition of L_S^{ramp}, this is equivalent to

$$\hat{h} = h_{\hat{w}}; \qquad \hat{w} = \arg\min_w \|w\|^2 \quad \text{s.t. } \forall i \in [n],\ y_i w^{\mathsf{T}} x_i \ge 1. \qquad \text{(HardSVM)}$$

This form is a *convex quadratic program,* a well-studied class of optimization problems. This is known as a *(hard) support vector machine* (SVM).

The usual motivation for SVMs is in terms of margin maximization. We can see this by noting that (HardSVM) is equivalent to

$$\begin{aligned}\hat{w} &= \operatorname*{arg\,max}_w \frac{1}{\|w\|} && \text{s.t. } \forall i \in [n],\quad y_i w^\mathsf{T} x_i \ge 1 \\ &= \operatorname*{arg\,max}_w \frac{1}{\|w\|} \min_{i\in[n]} w^\mathsf{T} x_i && \text{s.t. } \forall i \in [n],\quad y_i w^\mathsf{T} x_i \ge 1 \\ &\supseteq \operatorname*{arg\,max}_w \min_{i\in[n]} \frac{w^\mathsf{T} x_i}{\|w\|} && \text{s.t. } \forall i \in [n],\quad y_i w^\mathsf{T} x_i > 0.\end{aligned}$$

In the second line, $\min_{i\in[n]} w^\mathsf{T} x_i$ will equal 1 for any optimal w: it must be at least 1 for the constraint to hold, and if it were bigger we could just scale down w to also scale down all the predictions, which would improve the objective while keeping the constraints valid.

In the third line, the objective is invariant to scaling w by a constant, so any multiple of a w that minimized the second line will minimize the third line.

Also, if we scale any minimizer for the third line by $\min_{i\in[n]} y_i w^\mathsf{T} x_i$, we'll get a minimizer for the second line. Note that scaling by a positive constant doesn't change the hard classifier, $\operatorname{sgn}(w^\mathsf{T} x) = \operatorname{sgn}(c w^\mathsf{T} x)$ for $c > 0$; it just changes our confidence score.

The quantity $w^\mathsf{T} x_i / \|w\|$ is the *geometric margin* of the point x_i: it's the distance of x_i from the hyperplane $\{z : w^\mathsf{T} z = 0\}$. (For a formal proof of this fact, see Claim 15.1 of [SSBD].)

So, (HardSVM) maximizes the worst-case geometric margin on the training set, and anything maximizing the geometric margin will be a multiple of a solution to (HardSVM).

I should add similar illustrations here eventually.

For a graphical illustration of these concepts, see Figures 5.1 to 5.3 of [MRT].

Note that, as a convex QP, we can solve (HardSVM) in polynomial time – e.g. with a generic interior point algorithm [YT89], although there are many specialized solvers and other possibilities. Thus, if $n \ge \frac{1}{\varepsilon^2}\left[2\mathrm{R}\,\|w^*\| + \sqrt{2\log\frac{2}{\delta}}\right]^2$ then we efficiently achieve 0-1 loss less than ε with probability at least $1 - \delta$. This doesn't violate NP-hardness for 0-1 loss ERM, since it's only for separable distributions. It also doesn't contradict our VC dimension lower bounds, since we have two assumptions on $\mathcal{D}$ here: separability with a margin and the bound R on the norm of the data. (It doesn't even establish nonuniform learning, because of the dependence on R, only consistency.)

9.4 *Hinge loss and Soft SVM*

When the data isn't separable, (HardSVM) will just fail: the constraints aren't achievable, so it's minimizing over an empty set.

A natural idea is to try to trade off between having a small $\mathrm{L}_\mathrm{S}^{ramp}(h)$ and a small

$\|w\|$. For example, like in SRM, we could try to minimize the upper bound in Proposition 9.1. We could try to literally do that, but there's some cruft in the bound based on r and so on that we might prefer to avoid worrying about. So, let's be a little fuzzy, and pretend we pick an r small enough that $\max\{r, \|w\|\} = \|w\|$ for any "reasonable" w but not so small that $\log\log_2 \frac{1}{r}$ is relevant to anything. The $\sqrt{\log\log_2 \|w\|}$ term is also not going to be at all relevant compared to the $\|w\|$ term. So, it seems reasonable to try to pick

$$\arg\min_w \mathrm{L}_\mathrm{S}^{ramp}(h_w) + \frac{4\mathrm{R}}{\sqrt{n}} \|w\| .$$

Unfortunately, solving this problem is NP-hard [MI15, Theorem 2.3].

To avoid this, we can *again* take a surrogate loss, $\ell_{hinge} \geq \ell_{ramp} \geq \ell_{0-1}$ given by

$$\ell_{hinge}(h, (x, y)) = \lambda_y^{hinge}(h(x)) = \begin{cases} 1 - yh(x) & \text{if } yh(x) \leq 1 \\ 0 & \text{if } yh(x) \geq 1. \end{cases}$$

This is like the ramp loss, except once it starts going, it never stops: you get more loss for a more-confident wrong answer. This loss is still 1-Lipschitz, but it's not bounded. More importantly, though, it's *convex*, which makes it easy to optimize. (We'll talk more about convexity shortly.)

9.4.1 Hinge loss ERM

It then makes sense to try to have both small $\mathrm{L}_\mathrm{S}^{hinge}$ and small $\|w\|$. We can see from (9.2) that, for example, if

$$\hat{h}_\mathrm{B} = \arg\min_{h \in \mathcal{H}_\mathrm{B}} \mathrm{L}_\mathrm{S}^{hinge}(h_w), \tag{9.3}$$

then, using $\mathrm{L}_\mathrm{S}^{hinge}(h_{\hat{w}_\mathrm{B}}) \leq \mathrm{L}_\mathrm{S}^{hinge}(h^*)$, with probability at least $1 - \delta$

$$\mathrm{L}_\mathcal{D}^{0-1}(\hat{h}_\mathrm{B}) \leq \mathrm{L}_\mathrm{S}^{hinge}(\hat{h}_\mathrm{B}) + \frac{2\mathrm{RB}}{\sqrt{n}} + \sqrt{\frac{1}{2n} \log \frac{1}{\delta}}.$$

While ℓ_{hinge} is unbounded, we know that $\sup_{h,x} |h(x)| \leq 1 + \sup_{h,x} \|w\| \|x\|$. Thus if we strengthen our assumption on $\mathcal{D}$ to $\Pr(\|x\| \leq \mathrm{R}) = 1$, we get

$$\mathrm{L}_\mathcal{D}^{0-1}(h_{\hat{w}_\mathrm{B}}) \leq \inf_{h \in \mathcal{H}_\mathrm{B}} \mathrm{L}_\mathrm{S}^{hinge}(h) + \frac{2\mathrm{RB}}{\sqrt{n}} + (2 + \mathrm{RB})\sqrt{\frac{1}{2n} \log \frac{2}{\delta}}.$$

9.4.2 Bound minimization

Rather than picking a hard constraint B, maybe difficult to choose a priori, we could do something SRM-like with Proposition 9.1 and let $\hat{h} = h_{\hat{w}}$ minimize

$$L_S^{hinge}(h_w) + \frac{1}{\sqrt{n}}\left[4R\max\{r, \|w\|\} + \max\left\{0, \sqrt{\log\log_2 \frac{2\,\|w\|}{r}}\right\}\right]. \tag{9.4}$$

Then, like in SRM, we know that quantity is minimized for $\hat{w}$, and so can say that, for any arbitrary $h^* = h_{w^*}$, assigning $\frac{2}{3}\delta$ failure probability for the bound of Proposition 9.1 and $\frac{1}{3}\delta$ probability for a Hoeffding bound on $L_S^{hinge}(h^*)$, and again assuming that $\|x\| \le R$ a.s.,

$$\begin{aligned}
L_{\mathcal{D}}^{0-1}(\hat{h}) &\le L_S^{hinge}(\hat{h}) + \frac{1}{\sqrt{n}}\left[4R\max\{r, \|\hat{w}\|\} + \max\left\{0, \sqrt{\log\log_2 \frac{2\,\|\hat{w}\|}{r}}\right\} + \sqrt{\frac{1}{2}\log\frac{3}{\delta}}\right] \\
&\le L_S^{hinge}(h^*) + \frac{1}{\sqrt{n}}\left[4R\max\{r, \|w^*\|\} + \max\left\{0, \sqrt{\log\log_2 \frac{2\,\|w^*\|}{r}}\right\} + \sqrt{\frac{1}{2}\log\frac{3}{\delta}}\right] \\
&\le L_{\mathcal{D}}^{hinge}(h^*) + \frac{1}{\sqrt{n}}\left[4R\max\{r, \|w^*\|\} + \max\left\{0, \sqrt{\log\log_2 \frac{2\,\|w^*\|}{r}}\right\} + (2 + R\,\|w^*\|)\sqrt{\frac{1}{2}\log\frac{3}{\delta}}\right].
\end{aligned}$$

This is like a nonuniform learning bound, but only for $\mathcal{D}$ satisfying $\|x\| \le R$.

9.4.3 Soft SVM

Unfortunately, the optimization problem (9.4) is kind of a huge pain. It's not even convex, both because of the $\max\{r, \|w\|\}$ thing and the $\sqrt{\log\log_2 \|w\|}$ term. Again, we can reason that we can probably ignore r and the $\sqrt{\log\log_2 \|w\|}$ term, and argue for minimizing

$$L_S^{hinge}(h_w) + \frac{4R}{\sqrt{n}}\|w\|.$$

It's not obvious that these bounds are especially tight, though, so maybe $\frac{4R}{\sqrt{n}}$ isn't the right constant to trade off between the loss and $\|w\|$. Also, it turns out to be more convenient to minimize with $\|w\|^2$ rather than $\|w\|$. *Soft SVMs* use the squared norm of w and replace $4R/\sqrt{n}$ with a hyperparameter λ:

$$\hat{h}_\lambda = h_{\hat{w}_\lambda}; \qquad \hat{w}_\lambda = \operatorname*{arg\,min}_w L_S^{hinge}(h_w) + \lambda\,\|w\|^2. \tag{SoftSVM}$$

(In the version with an intercept b, we typically *don't* add λb^2 to the loss; this is one difference from the homogeneous reduction.)

(9.3) and (SoftSVM) are in fact dual to each other, in the sense that for any B there is some λ such that $\hat{h}_B$'s weight vector agrees with $\hat{w}_\lambda$, and vice

If you're familiar with convex optimization: set up the Lagrangian of either problem, and use Slater's condition to show that strong duality holds.

versa. (We can't just write down a given B for a given λ or vice-versa, though, unfortunately.)

Soft SVMs also have a nice motivation in terms of margin maximization. If h_w classifies a point x correctly with margin at least 1, then it doesn't contribute to the objective at all. If it's "inside" the margin or even misclassified, though, we get loss equal to the distance by which we're on the wrong side of the margin. One way to consider this is as a hard SVM on a modified problem, where we drag points around to be on the margin, and penalize how much dragging around we need to do.

The classic framing is $C\,L_S^{hinge}(h_w) + \|w\|^2$*; there the penalty for moving points around is* C*. You can think of* $C = \frac{1}{\lambda}$.

In the limit as $\lambda \to 0$, on separable data, (SoftSVM) becomes (HardSVM). Soft SVMs with a nonzero λ might give different results from hard SVMs, though, even on separable data: they might allow a few points to violate a bigger "theoretical" margin.

To analyze $\hat{w}_\lambda$ directly, we can still use Proposition 9.1 and (if we like) bound the ramp loss by the hinge loss: the result holds for all linear predictors. This gives us an upper bound on $L_{\mathcal{D}}^{0\text{-}1}(\hat{h}_\lambda)$ in terms of $L_S^{hinge}(\hat{h}_\lambda)$ and $\|\hat{w}_\lambda\|$. It's more difficult to relate this to the loss of a comparison hypothesis h^*, though we can maybe take some solace in (SoftSVM) being similar to (9.4).

Or, instead, we can use stability bounds (discussed soon).

9.5 *Hard SVM duality*

The following stuff is historically very important, serves as a nice segue into our next topic, explains the name "support vector machine," and introduces an area of math that's profoundly important to optimization / often useful in theory / beautiful in its own right. It's not, however, as practically important as it once was.

Starting from (HardSVM), we can rewrite these hard constraints by introducing *dual variables* α_i for $i \in [n]$:

$$\min_w \frac{1}{2}\|w\|^2 \text{ s.t. } \forall i.\ y_i w^\mathsf{T} x_i \geq 1 = \min_w \max_{\alpha_i \geq 0} \frac{1}{2}\|w\|^2 + \sum_{i=1}^n \alpha_i (1 - y_i w^\mathsf{T} x_i).$$

If any of the $y_i w^\mathsf{T} x_i$ are less than one, then the inner maximizer can drive $\alpha_i \to \infty$ and make the objective arbitrarily big; the outer minimizer, then, can't allow that to happen. If they're all at least one, then the inner maximizer is best by just picking $\alpha_i = 0$ (or unconstrained for any that are exactly one).

Now, it's always the case that $\min_x \max_y f(x,y) \geq \max_y \min_x f(x,y)$; take $\max_y f(x,y) \geq f(x,y')$ for any (x,y'), minimize both sides in x, then maximize in y'. In this setting, this is called *weak (Lagrangian) duality*. In this case, though, we actually have *strong* duality via something called *Slater's condition*:

swapping the min and the max doesn't change the value.

$$\min_w \frac{1}{2}\|w\|^2 \text{ s.t. } \forall i.\ y_i w^\mathsf{T} x_i \geq 1 = \max_{\alpha_i \geq 0} \min_w \frac{1}{2}\|w\|^2 + \sum_{i=1}^n \alpha_i (1 - y_i w^\mathsf{T} x_i).$$

The inner minimization in w is differentiable and unconstrained, so we can find its value by setting the gradient to zero:

$$w + \sum_{i=1}^n (-\alpha_i y_i x_i) = 0 \quad \Rightarrow \quad w = \sum_{i=1}^n \alpha_i y_i x_i,$$

and hence

$$\|w\|^2 = \sum_{i=1}^n \sum_{j=1}^n \alpha_i y_i x_i^\mathsf{T} x_j y_j \alpha_j = \alpha^\mathsf{T} \operatorname{diag}(y) \mathbf{X}\mathbf{X}^\mathsf{T} \operatorname{diag}(y) \alpha,$$

where $\alpha \in \mathbb{R}^n$ is the vector of α_is, $\operatorname{diag}(y) \in \mathbb{R}^{n \times n}$ is a matrix with y_i as its (i,i)th entry and zero off-diagonal, and $\mathbf{X} \in \mathbb{R}^{n \times d}$ has ith row x_i. Thus we've shown that (HardSVM) is equivalent to

$$\max_{\alpha_i \geq 0} \mathbf{1}^\mathsf{T} \alpha - \frac{1}{2} \alpha^\mathsf{T} \operatorname{diag}(y) \mathbf{X}\mathbf{X}^\mathsf{T} \operatorname{diag}(y) \alpha, \tag{HardSVM'}$$

where once we find α we can recover w as $\mathbf{X}^\mathsf{T} \operatorname{diag}(y)\alpha$, meaning that $h_w(x) = \alpha^\mathsf{T} \operatorname{diag}(y) \mathbf{X} x = \sum_{i=1}^n \alpha_i y_i x_i^\mathsf{T} x$.

This is called the *dual form* of (HardSVM). We've transformed the *primal* form, a constrained optimization over $w \in \mathbb{R}^b$, to an unconstrained optimization over $\alpha \in \mathbb{R}^n_{\geq 0}$. We can solve this with any of several algorithms: it's also a convex quadratic program, and there are many specialized algorithms for (HardSVM') in particular, but since the constraints are simple we can also think about easy things like projected gradient descent.

SUPPORT VECTORS (HardSVM') also motivates the name *support vector machine*. As we mentioned when we first introduced the dual variables, if $y_i w^\mathsf{T} x_i > 1$ for some i, then we necessarily have $\alpha_i = 0$ at optimum. We can only have $\alpha_i \neq 0$ if $y_i w^\mathsf{T} x_i = 0$, i.e. the point (x_i, y_i) is exactly on the margin of the hard SVM. These points are called support vectors, because they "support" the position of the margin. This sparsity in the solution has some other nice consequences as well.

This is called complementary slackness *in the KKT conditions.*

9.6 *Soft SVM duality*

Start by introducing auxiliary variables ξ_i accounting for the hinge loss in (SoftSVM), then go through the same kind of argument, where now we'll additionally have dual variables β for the nonnegativity constraints on ξ. We're also going to use our dual variables for the margin constraints as $2\lambda\alpha_i$

We didn't do this out in class, just mentioned the result.

instead of just α_i, because it just makes stuff work out nicer in the end.

$$\begin{aligned}
&\min_{w,\xi} \lambda \|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \xi_i \quad \text{s.t. } \forall i,\ y_i w^\mathsf{T} x_i \geq 1 - \xi_i \ \text{ and } \ \xi_i \geq 0 \\
&\quad = \min_{w,\xi} \max_{\alpha\geq 0, \beta\geq 0} \lambda \|w\|^2 + \frac{1}{n}\sum_{i=1}^{n} \xi_i + \sum_{i=1}^{n} 2\lambda\alpha_i (1 - y_i w^\mathsf{T} x_i - \xi_i) - \sum_{i=1}^{n} \beta_i \xi_i \\
&\quad = \max_{\tilde\alpha\geq 0, \beta\geq 0} \min_{w,\xi} \lambda \|w\|^2 + \frac{1}{n}\mathbf{1}^\mathsf{T}\xi + 2\lambda\alpha^\mathsf{T}\left[\mathbf{1} - \operatorname{diag}(y)\mathbf{X}w - \xi\right] - \beta^\mathsf{T}\xi.
\end{aligned}$$

Setting the w gradient to zero, $2\lambda w - 2\lambda \mathbf{X}^\mathsf{T} \operatorname{diag}(y)\alpha = 0$ and so $w = \mathbf{X}^\mathsf{T} \operatorname{diag}(y)\alpha$ as before. For ξ, $\frac{1}{n}\mathbf{1} - 2\lambda\alpha - \beta = 0$ means that $\beta = \frac{1}{n}\mathbf{1} - 2\lambda\alpha$. We can easily achieve this as long as $\alpha_i < \frac{1}{2\lambda n}$, getting the dual form

$$(2\lambda) \max_{0 \leq \alpha_i \leq \frac{1}{2\lambda n}} \mathbf{1}^\mathsf{T}\alpha - \frac{1}{2}\alpha^\mathsf{T} \operatorname{diag}(y)\mathbf{X}\mathbf{X}^\mathsf{T} \operatorname{diag}(y)\alpha. \qquad \text{(SoftSVM')}$$

Remarkably, this is *exactly* (HardSVM') with an extra upper bound on α.

Using the same kind of argument as we made for support vectors earlier, we can see that indeed $\xi_i = 0$ unless $y_i w^\mathsf{T} x_i < 1$: we only "move the input points" if we need to. For these points, $\beta_i = 0$, meaning that $\alpha_i = \frac{1}{2\lambda n}$, and we can immediately tell which points are misclassified or classified correctly with too small a margin. Any points with $0 < \alpha_i < \frac{1}{2\lambda n}$ have $\xi_i = 0$ but $y_i w^\mathsf{T} x_i = 1$, and so lie exactly on the margin as before.

9.6.1 Including an intercept

So far, we've been assuming that intercept terms, $\operatorname{sgn}(w^\mathsf{T}x + b)$ rather than $\operatorname{sgn}(w^\mathsf{T}x)$, are handled via $\tilde w = [b, w]$, $\tilde x = [1, x]$. But then note that $\|\tilde w\|^2 = b^2 + \|w\|^2$: we're regularizing the intercept as well, which isn't motivated in terms of the geometric margin and is also counter to usual statistical practice. So, it's maybe worth figuring out what happens if we explicitly include b and don't regularize it.

Compared to the derivation of (SoftSVM'), the constraint is $y_i(w^\mathsf{T}x_i + b) \geq 1 - \xi_i$, which only adds a term $2\lambda\alpha_i y_i b$. This doesn't affect the optimization for w or ξ, and b will have zero derivative iff $\alpha^\mathsf{T} y = 0$. This gives the dual

$$\max_{0 \leq \alpha_i \leq \frac{1}{2\lambda n} \text{ and } \alpha^\mathsf{T} y = 0} \mathbf{1}^\mathsf{T}\alpha - \frac{1}{2}\alpha^\mathsf{T} \operatorname{diag}(y)\mathbf{X}\mathbf{X}^\mathsf{T} \operatorname{diag}(y)\alpha.$$

Our final predictor is $w^\mathsf{T}x + b = \alpha^\mathsf{T} \operatorname{diag}(y)\mathbf{X}x + b$, so we still need to figure out the value of b. But note that, for points with $0 < \alpha_i < \frac{1}{2\lambda n}$, we know that $y_i(w^\mathsf{T}x_i + b) = 1$: so, once we've found α, we can just pick any such i and set $b = y_i - w^\mathsf{T}x_i = y_i - \alpha^\mathsf{T} \operatorname{diag}(y)\mathbf{X}x_i$.

9.7 *Aside (not in class): margin analysis*

The following is a slightly different way to frame ramp loss analysis that can sometimes be easier to think about. It's also more natural to look at for general hypothesis classes. It's based on the ρ-margin loss, which gives us full credit if our confidence is at least ρ:

$$\ell_{\rho\text{−}margin}(h,(x,y)) = \lambda_{\rho\text{−}margin,y}(h(x)) = \begin{cases} 1 & \text{if } yh(x) \le 0 \\ 1 - \frac{yh(x)}{\rho} & \text{if } 0 \le yh(x) \le \rho \\ 0 & \text{if } yh(x) \ge \rho. \end{cases}$$

This upper-bound to 0-1 loss ramps at ρ instead of 1, and is $\frac{1}{\rho}$-Lipschitz. So, the analogue of (9.1) is that for any fixed ρ, with probability at least $1-\delta$, we have for any $\mathcal{H}$ of real-valued hypotheses that

$$\forall h \in \mathcal{H}, \quad L_{\mathcal{D}}^{0\text{−}1}(h) \le L_S^{\rho\text{−}margin}(h) + \frac{2}{\rho} \mathop{\mathbb{E}}_{S' \sim \mathcal{D}^n} \operatorname{Rad}(\mathcal{H}|_{S'_x}) + \sqrt{\frac{1}{2n} \log \frac{1}{\delta}}. \qquad (9.5)$$

We can avoid committing to a particular margin, similarly to in Proposition 9.1. Another slight improvement is that we don't have to assume $\mathcal{H}_B$, but allow general real-valued $\mathcal{H}$.

9.2 PROPOSITION. *Let $\mathcal{H}$ contain functions mapping to $\mathbb{R}$, and fix some margin upper bound $r > 0$. Then for any $\delta \in (0,1)$, we have with probability at least $1-\delta$ over the choice of $S \sim \mathcal{D}^n$ that for all $h \in \mathcal{H}$ and $\rho \in (0, r]$,*

$$L_{\mathcal{D}}^{0\text{−}1}(\hat{h}) \le L_S^{\rho\text{−}margin}(h) + \frac{4}{\rho} \mathop{\mathbb{E}}_{S' \sim \mathcal{D}^n} \operatorname{Rad}(\mathcal{H}|_{S'_x}) + \sqrt{\frac{1}{n} \log \log_2 \frac{2r}{\rho}} + \sqrt{\frac{1}{2n} \log \frac{2}{\delta}}.$$

Proof. Let $\rho_i = r2^{-i}$ for all $i \ge 0$, and $\delta_i = \frac{6\delta}{\pi^2 i^2}$ for $i \ge 1$; note that $\sum_{i=1}^{\infty} \delta_i = \delta$. By (9.5), it holds with probability at least $1 - \delta_i$ for each ρ_i that

$$\forall h \in \mathcal{H}, \quad L_{\mathcal{D}}^{0\text{−}1}(h) \le L_S^{\rho_i\text{−}margin}(h) + \frac{2}{\rho_i} \mathop{\mathbb{E}}_{S' \sim \mathcal{D}^n} \operatorname{Rad}(\mathcal{H}|_{S'_x}) + \sqrt{\frac{1}{2n} \log \frac{1}{\delta_i}}.$$

For any $\rho \in (0, r]$, the smallest i such that $\rho_i \le \rho$ is given by $i = \left\lceil \log_2 \frac{r}{\rho} \right\rceil$.

We have $\ell_{\rho'\text{−}margin} \le \ell_{\rho\text{−}margin}$ for any $\rho' \le \rho$, so $L_S^{\rho_i\text{−}margin}(h) \le L_S^{\rho\text{−}margin}(h)$.

We also know that $\rho \le \rho_{i-1} = 2\rho_i$, so $\frac{1}{\rho_i} \le \frac{2}{\rho}$.

Finally, from $\log \frac{1}{\delta_i} = \log \frac{\pi^2}{6\delta} + 2 \log \log_2 \left\lceil \log_2 \frac{r}{\rho} \right\rceil$ we use that $\pi^2/6 < 2$ and $\lceil \log_2 a \rceil < \log_2(a) + 1 = \log_2(2a)$. □

We do have to commit to some predefined upper bound on the margin r, but the resulting bound only depends on it through $\log\log_2 r$ so we can pick something big. (This r corresponds to $\frac{1}{r}$ from Proposition 9.1.)

In this bound, we think of having a fixed $\mathcal{H}$ but then trading off in our analysis between trying to get a large margin (to decrease the $\frac{1}{\rho}$ terms) and having a small margin loss.

REFERENCES

[BS00] Shai Ben-David and Hans Ulrich Simon. "Efficient learning of linear perceptrons." *Advances in Neural Information Processing Systems*. 2000.

[KV94] Michael J. Kearns and Umesh Vazirani. *An Introduction to Computational Learning Theory*. The MIT Press, 1994.

[MI15] Søren Frejstrup Maibing and Christian Igel. "Computational Complexity of Linear Large Margin Classification With Ramp Loss." *AISTATS*. 2015.

[MRT] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talkwalkar. *Foundations of Machine Learning*. 2nd ed. MIT Press, 2018.

[SSBD] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[Tel] Matus Telgarsky. *Deep learning theory lecture notes*. Version: 2021-10-27 v0.0-e7150f2d (alpha). 2021.

[Val84] Leslie G. Valiant. "A Theory of the Learnable." *Commun. ACM* 27.11 (1984), pp. 1134–1142.

[VC71] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. "On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities." *Theory of Probability & Its Applications* 16.2 (1971), pp. 264–280.

[Wai19] Martin Wainwright. *High-dimensional statistics: a non-asymptotic viewpoint*. Cambridge University Press, 2019.

[YT89] Yinyu Ye and Edison Tse. "An extension of Karmarkar's projective algorithm for convex quadratic programming." *Mathematical Programming* 44 (1989), pp. 157–159.