Lecture 21. (Sketch.)

- Project proposal meetings today!

Rademacher recap (same slide as last time!).

Concentration controlled one function at a time. To control many functions, our main tool is (unnormalized) Rademacher complexity:

\[ \text{URad}(V) := \mathbb{E} \sup_{u \in V} \langle \epsilon, u \rangle, \quad \text{Rad}(V) := \frac{1}{n} \text{URad}(V). \]

Given data \( S := (Z_1, \ldots, Z_n) \) and functions \( \mathcal{F} \), define vectors

\[ \mathcal{F}_S := \{ (f(Z_1), \ldots, f(Z_n)) : f \in \mathcal{F} \}. \]

Our main generalization tool involves \( \text{URad}(\mathcal{F}_S) \), and is a consequence of our two symmetrization lemmas and McDiarmid's inequality.

**Theorem.** Let \( \mathcal{F} \) be given with \( f(z) \in [a, b] \) a.s.. With probability \( \geq 1 - \delta \),

\[ \sup_{f \in \mathcal{F}} \mathbb{E} f - \hat{\mathbb{E}}_n f \leq 2 \frac{n}{n} \text{URad}(\mathcal{F}_S) + 3(b - a) \sqrt{\frac{\ln(2/\delta)}{2n}}. \]

2. VC Theory.

First, some definitions. First, the zero-one/classification risk/error:

\[ \mathcal{R}_z(f) = \mathbb{P}[\text{sgn}(f(X)) \neq Y], \quad \hat{\mathcal{R}}_z(f) = \frac{1}{n} \sum_{i=1}^{n} I[\text{sgn}(f(x_i)) \neq y_i] \]

The earlier Rademacher bound will now have

\[ \text{URad} \left( \{ (x, y) \mapsto I[\text{sgn}(f(x)) \neq y] : f \in \mathcal{F}_S \} \right). \]

This is at most \( 2^n \); we’ll reduce it to a combinatorial quantity:

\[ \text{sgn}(U) := \{ (\text{sgn}(u_1), \ldots, \text{sgn}(u_n)) : u \in V \}, \]

\[ \text{Sh}(\mathcal{F}_S) := \text{sgn}(\mathcal{F}_S), \]

\[ \text{Sh}(\mathcal{F}; n) := \sup_{S \in \mathbb{P}_n} \text{sgn}(\mathcal{F}_S), \]

\[ \text{VC}(\mathcal{F}) := \sup \{ i \in \mathbb{Z}_{\geq 0} : \text{Sh}(\mathcal{F}; i) = 2^i \}. \]
Remarks.

- Sh is “shatter coefficient”, VC is “VC dimension”.
- Both quantities are criticized as being too tied to their worst case; bounds here depend on (empirical quantity!) $URad(sgn(F|S))$, which can be better, but throws out the labels.

First step of proof: pull out the zero-one loss.

Lemma.

$URad(\{(x, y) \mapsto 1[sgn(f(x)) \neq y] : f \in F\}|S) \leq URad(sgn(F|S))$.

Proof. For each $i$, define

$$\ell_i(z) := \max \left\{ 0, \min \left\{ 1, \frac{1 - y_i(2z - 1)}{2} \right\} \right\},$$

which is 1-Lipschitz, and satisfies

$$\ell_i(sgn(f(x_i))) = 1[sgn(f(x_i)) \neq y_i].$$

(Indeed, it is the linear interpolation.) Then

$$URad(\{(x, y) \mapsto 1[sgn(f(x)) \neq y] : f \in F\}|S) = URad((\ell_1(sgn(f(x_1))), \ldots, \ell_n(sgn(f(x_n)))) : f \in F\}|S) = URad(\ell \circ sgn(F)|S) \leq URad(sgn(F)|S).$$

Theorem (“VC Theorem”). With probability at least $1 - \delta$, every $f \in F$ satisfies

$$R_z(sgn(f)) \leq \hat{R}_z(sgn(f)) + \frac{2}{n} URad(sgn(F|S)) + 3\sqrt{\frac{\ln(2/\delta)}{2n}},$$

and

$$URad(sgn(F|S)) \leq \sqrt{2n \ln Sh(F|S)},$$

$$\ln Sh(F|S) \leq \ln Sh(F ; n) \leq VC(F) \ln(n + 1).$$

Remarks.

- Need $Sh(F|S) = o(n)$ “in order to learn”.
- $VC(F) < \infty$ suffices; many considered this a conceptual breakthrough, namely “learning is possible”!
- The quantities (VC, Sh) appeared in prior work (not by V-C). Symmetrization apparently too, though I haven’t dug this up.

Our next step is a general Rademacher bound for finite sets.

Theorem (Massart finite lemma).

$$URad(V) \leq \sup_{u \in V} \|u\|_2 \sqrt{2 \ln |V|}.$$
We’ll prove this via a few lemmas.

**Lemma.** If \((X_1, \ldots, X_n)\) are \(c^2\)-subgaussian, then 
\[
\mathbb{E} \max_i X_i \leq c\sqrt{2\ln(n)}.
\]

**Proof.** Similar to homework 2.

**Lemma.** If \((X_1, \ldots, X_n)\) are \(c^2\)-subgaussian and independent, 
\[
\sum_i X_i \text{ is } \|\vec{c}\|_2^2\text{-subgaussian}.
\]

**Proof.** We did this in the concentration lecture, but here it is again:
\[
\mathbb{E}\exp(t\sum_i X_i) = \prod_i \mathbb{E}\exp(tX_i) \leq \prod_i \exp(t^2c_i^2/2) = \exp(t^2\|\vec{c}\|_2^2/2).
\]

Plugging this into our bound gives
\[
\text{URad}(\text{sgn}(F|S)) \leq \sqrt{2n\text{Sh}(F|S)}.
\]

One last lemma remains for the proof.


Let \(F\) be given, and define \(V := \text{VC}(F)\). Then
\[
\text{Sh}(F; n) \leq \begin{cases} 2^n & \text{when } n \leq V, \\ \left(\frac{en}{V}\right)^V & \text{otherwise}. \end{cases}
\]

Moreover, \(\text{Sh}(F; n) \leq n^V + 1\).

**Proof.** Omitted. Exists in many standard texts.)

**Proof (of Massart finite lemma).**

Let \(\tilde{e}\) be iid Rademacher and fix \(u \in V\). Define \(X_{u,i} := \epsilon_i u_i\) and \(X_u := \sum_i X_{u,i}\).

By Hoeffding lemma, \(X_{u,i}\) is \((u_i - u_i)^2/4 = u_i^2\)-subgaussian, thus (by Lemma) \(X_u\) is \(\|u\|_2^2\)-subgaussian. Thus
\[
\text{URad}(V) = \mathbb{E}_{u \in V} \max \langle e, u \rangle = \mathbb{E}_{u \in V} X_u \leq \max_{u \in V} \|u\|_2\sqrt{2\ln |V|}.
\]

**Remarks.** (on the VC theorem.)

▶ Minimizing \(\hat{R}_2\) is NP-hard in many trivial cases, but those require noise and neural networks can often get \(\hat{R}_2(\text{sgn}(f)) = 0\).

▶ Recent work prefers real-valued / scale-sensitive complexity measures, where it is easier (?) to depend on things like weight matrix norms in neural networks.
3. Margin bounds.

- Rather than looking at just \( \text{sgn}(f(x)) \), let’s evaluate the magnitude of \( f \).
- These bounds will be sensitive to real-valued (rather than combinatorial) properties of \( \mathcal{F} \), and also to the labels (encoded via a “margin assumption” implicit in assuming the training margin risk \( \mathcal{R}_\gamma \) is small).

Remark.

- Started with Bartlett ’96 “For valid generalization, the size of the weights is more important than the size of the network”. (Worst-case VC can’t handle scale: \( \text{sgn}(f) = \text{sgn}(cf) \) for \( c > 0 \). Margin bounds can handle scale.)
- Intuition: can wiggle (rotate up to \( \gamma \)) predictor without changing output labels.
- To invoke theorem, we need to show that algorithms actually give a small \( \mathcal{R}_\gamma \) (which is stronger than requiring small \( \mathcal{R}_z \)). We’ll see in homework that we often have something like this for convex losses.
- Often these bounds are used with \( l_1 \) balls of predictors, which is the same as \( \text{conv}(\mathcal{F} \cup -\mathcal{F}) \). (Next page gives some tools for this.)

Following properties can help apply margin bounds.

**Lemma.**

1. \( \text{URad}(V) \geq 0 \).
2. \( \text{URad}(cV + \{u\}) \leq |c| \text{URad}(V) \).
3. \( \text{URad}(\text{conv}(V)) \leq \text{URad}(V) \).
4. Let \( (V_i)_{i \geq 0} \) be given with \( \sup_{u \in V_i} \langle u, \epsilon \rangle \geq 0 \forall \epsilon \in \{-1, +1\}^n \). (E.g., \( V_i = -V_i \) or \( 0 \in V_i \).) Then \( \text{URad}(\cup_i V_i) \leq \sum_i \text{URad}(V_i) \).
5. \( \text{URad}(V) = \text{URad}(-V) \).

**Remarks.**

- \( (3) \) is a mixed blessing: “Rademacher is insensitive to convex hulls”,
- \( (4) \) is true for \( \text{URad}_{l_1} \) directly: define \( W_i := V_i \cup -V_i \), which satisfies the conditions, and note \( (\cup_i V_i) \cup -(\cup_i V_i) = \cup_i W_i \). Since \( \text{URad}_{l_1}(V_i) = \text{URad}(W_i) \), then \( \text{URad}_{l_1}(\cup_i V_i) = \text{URad}(\cup_i W_i) \leq \sum_{i \geq 1} \text{URad}(W_i) = \sum_{i \geq 1} \text{URad}_{l_1}(V_i) \).

Define \( \ell_\gamma(z) := \max\{0, \min\{1, 1 + z/\gamma\}\} \), \( \mathcal{R}_\gamma(f) := \mathcal{R}_{\ell_\gamma}(f) = \mathbb{E}\ell_\gamma(-Yf(X)) \).

**Theorem.** With probability \( \geq 1 - \delta \), \( \forall f \in \mathcal{F} \),

\[
\mathcal{R}_z(f) \leq \mathcal{R}_\gamma(f) \leq \hat{\mathcal{R}}_\gamma(f) + \frac{2}{n\gamma} \text{URad}(\mathcal{F}) + 3\sqrt{\frac{\ln(2/\delta)}{2n}}.
\]

**Proof.** Since

\[
\mathbb{I}[\text{sgn}(f(x)) \neq y] \leq \mathbb{I}[f(x)y \geq 0] \leq \ell_\gamma(-f(x)y),
\]

then \( \mathcal{R}_z(f) \leq \mathcal{R}_\gamma(f) \). The bound between \( \mathcal{R}_\gamma \) and \( \hat{\mathcal{R}}_\gamma \) follows from the fundamental Rademacher bound, and by peeling the \( 1/\gamma \)-Lipschitz function \( \ell_\gamma \).
Proof.

(1.) Fix any \( u_0 \in V \); then \( \mathbb{E}_\epsilon \sup_{u \in V} \langle \epsilon, v \rangle \geq \mathbb{E}_\epsilon \langle \epsilon, u_0 \rangle = 0 \).

(2.) Either check directly, or use the \(|c|\)-Lipschitz functions \( \ell_i(r) := c \cdot r + u_i \).

(4.) Using the condition,
\[
\mathbb{E}_\epsilon \sup_{u \in \bigcup V_i} \langle \epsilon, u \rangle = \mathbb{E}_\epsilon \sup_{i} \sup_{u \in V_i} \langle \epsilon, u \rangle \leq \mathbb{E}_\epsilon \sum_{i \geq 1} \text{URad}(V_i).
\]

(5.) Since integrating over \( \epsilon \) is the same as integrating over \(-\epsilon\) (the two are equivalent distributions),
\[
\text{URad}(-V) = \mathbb{E}_\epsilon \sup_{u \in V} \langle \epsilon, -u \rangle = \mathbb{E}_\epsilon \sup_{u \in V} \langle -\epsilon, -u \rangle = \text{URad}(V).
\]

Proof (continued).

(3.) This follows since optimization over a polytope is achieved at a corner. In detail,
\[
\text{URad} (\text{conv}(V)) = \mathbb{E}_\epsilon \sup_{k \geq 1} \sup_{\alpha \in \Delta_k} \left( \epsilon, \sum_{j} \alpha_j u_j \right)
\]
\[
= \mathbb{E}_\epsilon \sup_{k \geq 1} \sum_{\alpha \in \Delta_k} \alpha_j \sup_{u_j \in V} \langle \epsilon, u_j \rangle
\]
\[
= \mathbb{E}_\epsilon \left( \sup_{k \geq 1} \sum_{\alpha \in \Delta_k} \alpha_j \right) \sup_{u \in V} \langle \epsilon, u \rangle
\]
\[
= \text{URad}(V).
\]