CS 294-220

Computational Learning Theory

Lecture Notes

Druv Pai

Contents

1 Language Identification In The Limit 3

1.1 Definitions . . . 3

1.2 Simple Examples . . . 5

1.3 Negative Result: Gold's Theorem, Locking Sequence . . . 6

1.4 Identifiable Sets of Languages . . . 7

1.5 Discussion . . . 10

2 Probably Approximately Correct Learning 11

2.1 The PAC Definition . . . 11

2.2 A Better PAC Definition . . . 17

2.3 Discussion . . . 19

3 Concentration of Measure 20

3.1 Introduction . . . 20

3.2 Moment Generating Functions . . . 21

3.3 Sub-Gaussian Distributions . . . 22

3.4 Hoeffding's Inequality . . . 23

4 Agnostic PAC, Uniform Convergence, and VC Dimension 26

4.1 Introduction . . . 26

4.2 The Agnostic PAC Definition . . . 26

4.3 Error Decomposition: The Bias-Complexity Tradeoff . . . 29

4.4 Agnostic PAC Learning via Uniform Convergence . . . 30

4.5 The VC Dimension: Uniform Convergence for Infinite Classes . . . 32

4.6 Discussion . . . 34

5 The Fundamental Theorem of PAC Learning 35

5.1 Introduction . . . 35

5.2 The Growth Function and Sauer's Lemma . . . 35

5.3 Uniform Convergence for Classes of Finite VC Dimension . . . 37

5.4 The Fundamental Theorem of PAC Learning . . . 41

5.5 Discussion . . . 43

6 Rademacher Complexity 44

6.1 Introduction . . . 44

6.2 Concentration of Measure for Uniform Convergence . . . 44

6.3 Definition of Rademacher Complexity . . . 46

6.4 Bounding the Rademacher Complexity . . . 49

6.5 Discussion . . . 52

7 Covering Numbers and Chaining 53

7.1 Definitions . . . 53

7.2 Intuition for COvering and Packing Numbers . . . 54

7.3 Packing Numbers for VC Claasses . . . 55

7.4 Uniform Convergence via Covering Numbers . . . 57

7.5 Chaining . . . 59
7.6 Learning Bounds via Chaining . . . 62

8 Boosting 64
8.1 Weak Learnability . . . 64
8.2 AdaBoost . . . 65
8.3 Linear Combinations of Base Hypotheses . . . 67

9 Non-Uniform Learning 68
9.1 Nonuniform Learnability . . . 68
9.2 Structural Risk Minimization . . . 68
9.3 Minimum Description Length and Occam's Razor . . . 71
9.4 PAC-Bayes Bounds . . . 73

10 Regularization and Stability 75
10.1 Regularized Loss Minimization . . . 75
10.2 Stable Rules Do Not Overfit . . . 76
10.3 Controlling the Fitting-Stability Tradeoff . . . 77

1 Language Identification In The Limit

When most people discuss machine learning today, it's mostly focused on applications. There is a rich algorithmic and mathematical background to the study of machine learning, and we want to pursue it in this course.

A fundamental question we ask is, "what is learning"? We study an older model of learning first, before we consider the modern variants.

1.1 Definitions

1.1.1 Preliminaries

Notation 1.1
We use $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$, which will help when talking about string length.

Notation 1.2
For s a (possibly finite) sequence, we use s^n to denote $(s_i)_{i \in [n]}$, the length-n *prefix*.

Notation 1.3
Sometimes we write $x \in X$ for a sequence x to denote that the elements of x are contained in X. Similar notation applies for subset relations.

Definition 1.4 (Alphabet, Language)
A *alphabet* Σ is a finite set. If Σ is an alphabet then each member $s \in \Sigma$ is a *symbol*. A *string* of length n over Σ is an ordered tuple $s = s^n$ such that $s_i \in \Sigma$. We often write $s = s_1 \cdots s_n$. We write $|s| = n$ and $x \in s$ to indicate $x = s_i$ for some $i \in [n]$.

The set of all strings of length n over Σ is denoted Σ^n. The unique string of length 0 is called the *empty string* and is denoted by ε.

Let $\Sigma^* = \bigcup_{n \in \mathbb{N}_0} \Sigma^n$ be the set of strings over Σ (this is the *Kleene star notation*). For two strings $s, t \in \Sigma^*$, we denote their *concatenation* $s \circ t$ by $s_1 \cdots s_{|s|} t_1 \cdots t_{|t|}$.

A *language* L over Σ is a set of strings $L \subseteq \Sigma^*$. We say L is *empty* if $L = \varnothing$, *finite* if $|L| < |\mathbb{N}|$, and *infinite* if $|L| \geq |\mathbb{N}|$.

Definition 1.5 (Encoding)
For an object x in an underlying set X, we write $\langle x \rangle$ to denote a string encoding of x. The map $x \mapsto \langle x \rangle$ is a reasonable computable bijection $X \to \Sigma^*$. The details don't matter too much, other than existence of this bijection.

1.1.2 Language Identification in the Limit

Definition 1.6 (Positive Presentation)
Let $L \subseteq \Sigma^*$ be a language. An infinite sequence $s = (s_n)_{n \in \mathbb{N}} \in \Sigma^*$ is called a *positive presentation of* L if $L = \{s_n\}_{n \in \mathbb{N}}$.

Remark 1.7. The empty language does not have any positive presentations. Every non-empty language has a positive presentation, and any language with more than one string has infinitely many positive presentations.

Definition 1.8 (Computable Indexing)
Let $\mathcal{L}$ be a collection of languages over Σ. An infinite sequencing of languages $(L_n)_{n \in \mathbb{N}} \subseteq \Sigma^*$ is a *computable indexing of* $\mathcal{L}$ if

(i) $\mathcal{L} = \{L_n\}_{n \in \mathbb{N}}$.

(ii) There exists a Turing Machine (TM) D such that for all $n \in \mathbb{N}$ and all $s \in \Sigma^*$,

$$D(\langle n, s \rangle) = \mathbb{1}(s \in L_n)\,.$$

Namely, if D is executed with the encoding of (n, s) as input then D halts, and furthermore D outputs 1 if $s \in L_n$ and it outputs 0 otherwise.

Remark 1.9. If a set of languages $\mathcal{L}$ has a computable indexing, then

- Each language in $\mathcal{L}$ is computable,
- $\mathcal{L}$ is finite or countable, and
- $\mathcal{L}$ is recursively enumerable in the sense that $(D(\langle n, \cdot \rangle))_{n \in \mathbb{N}}$ is an enumeration of deciders of the languages in $\mathcal{L}$.

Definition 1.10 (Sequence of Outputs)
Let L be a language and let s be a positive presentation of L. Let M be a TM. The *sequence of outputs of* M *on* s is $(m_n)_{n \in \mathbb{N}}$ such that

$$m_n = M\left(\langle s^n \rangle\right).$$

for all $n \in \mathbb{N}$. Namely, m_n is the output of M when executed on the input $\langle s^n \rangle = \langle s_1, \ldots, s_n \rangle$ which consists of the first n strings in the positive presentation s. If M does not halt on this input, we write $m_n = \perp$.

Definition 1.11 (Convergence of Turing Machines)
A Turing Machine M *converges* to $k \in \mathbb{N}$ on s and write

$$M(s) \to K$$

if

- $m_n \neq \perp$ for all $n \in \mathbb{N}$, and
- there exists $N \in \mathbb{N}$ such that $m_n = k$ for all $n > N$.

If M does not converge to any number k on s, then M *diverges* on s.

Definition 1.12 (Identification in the Limit)
Let $\mathscr{L}$ be a set of languages over an alphabet Σ, let $L \in \mathscr{L}$, let $I = (L_n)_{n\in\mathbb{N}}$ be a computable indexing of $\mathscr{L}$, and let M be a TM. We say that M *identifies L in the limit w.r.t indexing I* if for any positive presentation s of L, $M(s) \to n$ for $n \in \mathbb{N}$ such that $L = L_n$. We say that M *identifies $\mathscr{L}$ in the limit w.r.t. indexing I* if, for every $L \in \mathscr{L}$, M identifies L in the limit w.r.t. indexing I.

We say that a set of languages $\mathscr{L}$ (or a language L) is identifiable in the limit if there exists a TM M and a computable indexing I of $\mathscr{L}$ such that M identifies $\mathscr{L}$ (or L) in the limit w.r.t. indexing I.

Remark 1.13. Because the empty language does not have any positive presentations, every TM identifies it in the limit in a vacuous sense.

1.2 Simple Examples

Example 1.14 (Finite Languages are Identifiable)
The set $\mathscr{L}$ of all finite languages over an alphabet Σ is identifiable in the limit.

Proof Sketch. We define the learner M to be a TM as follows. Let L be the target language and let s be a positive presentation of L. For any finite prefix s^n of s, the learner outputs $M(\langle s^n \rangle) = k$ such that $L_k = \{s_i\}_{i\in[n]}$. Observe that because s is a positive presentation, for every $\sigma \in L$ there exists a minimal index $m = m(\sigma) \in \mathbb{N}$ such that $s_m = \sigma$. Because L is finite, the value $N = \max\{m(\sigma) \colon \sigma \in L\}$ is finite. For all $n \geq N$, the language $L_k = \{s_i\}_{i\in[n]}$ as above will contain all the strings in L, and only strings in L. Thus, $M(s) \to k$ where $L_k = L$ as desired.

It is required, but not difficult to show, that the behavior of M is computable. □

Example 1.15 (Co-Singleton Sets are Identifiable)
The set $\mathscr{L}$ of co-singleton sets, namely $\mathscr{L} = \{\mathbb{N} \setminus \{n\} : n \in \mathbb{N}\}$, is identifiable in the limit.

Proof Sketch. Using the same notation as in Example 1.14, the learner M will compute

$$c = \min\left\{m \in \mathbb{N} \colon m \notin \{s_i\}_{i\in[n]}\right\}$$

and then output an index k such that $L_k = \mathbb{N} \setminus \{c\}$. Assume that the target language is $L = \mathbb{N} \setminus \{t\}$ for some fixed $t \in \mathbb{N}$. Let

$$N = \max\{m(s) \colon s < t, s \in \mathbb{N}\},$$

and note that N is a finite integer. For all $n \geq N$ it holds that $M(\langle s^n \rangle) = t$ as desired, because all the natural numbers smaller than t appear in the prefix s^n (by the choice of N), and t doesn't appear in s^n because $t \notin L$.

To complete the proof we note that there exists a computable indexing of $\mathscr{L}$ such that the mapping $c \mapsto k$ such that $L_k = \mathbb{N} \setminus \{c\}$ is computable. Hence M is a well-defined TM. □

Remark 1.16. Language identification in the limit explicitly models only learning from positive examples ("$x \in L$" is a "yes" instance) and the learner does not see negative examples ("$x \notin L$" is a "no" instance). Nevertheless, we can use the same formalism more generally to model learning functions and sequences. If X and Y are finite or countable sets and $f \colon X \to Y$ is a function by considering t he identification in the limit of the language $L = \{(x, y) \colon y = f(x), x \in X\}$. In particular, if $Y = \{0, 1\}$, we can think of instances of the form $(x, 1)$ as "yes" instances and examples of the form $(x, 0)$ as "no" instances. Additionally, a sequence is a function where the domain is $\mathbb{N}$, so we can use this identification in the limit to model a form of sequence learning. However, the presentation of the sequence to the learner might not be in order.

1.3 Negative Result: Gold's Theorem, Locking Sequence

Definition 1.17 (Locking Sequence for a Language)
Let $L \subseteq \Sigma^*$ be a language, let M be a TM, and let $k \in \mathbb{N}$. We say that a finite sequence of strings $s^n \in \Sigma^*$ is a *locking sequence for* L *that locks* M *onto* k if the following conditions hold:

(i) $\{s_i\}_{i \in [n]} \subseteq L$, and

(ii) For any (possibly empty) finite sequence of strings $t^m \in \Sigma^*$, if $\{t_j\}_{j \in [m]} \subseteq L$, then

$$M\left(\langle s^n \circ t^m \rangle\right) = k.$$

Lemma 1.18 (Locking Sequence Lemma)
Let $\mathscr{L}$ be a set of languages with a computable indexing $(L_n)_{n \in \mathbb{N}}$, let $L \in \mathscr{L}$, and let M be a TM. If M identifies L in the limit with respect to $(L_n)_{n \in \mathbb{N}}$ then there exists a locking sequence for L that locks M onto k such that $L = L_k$.

Proof. Assume for contradiction that there does not exist a locking sequence for L that locks M onto any number $k \in \mathbb{N}$ such that $L_k = L$. We make two observations.

Claim 1. If there exists a locking sequence for L that locks M onto some number k, then $L_k = L$.

Proof. Assume for contradiction that there is a locking sequence s for L that locks M onto k such that $L_k \neq L$. Let p be a positive presentation of L. Then the concatenation $s \circ p$ is also a positive presentation of L. From the assumption that s is a locking sequence, $M(s \circ p) \to k$. Seeing as $s \circ p$ is a positive presentation for L and $L_k \neq L$, this is a contradiction to the assumption that M identifies L in the limit. ■

Claim 2. For any finite sequence s of strings from L, there exists a finite sequence $\text{unlock}(s)$ of strings from L such that $M(\langle s \rangle) \neq M(\langle s \circ \text{unlock}(s) \rangle)$.

Proof. If there exists a sequence s of strings from L that does not have a suitable sequence $\text{unlock}(s)$, then s would be a locking sequence for L. From Claim 1 it must be that s locks M onto k such that $L = L_k$, in contradiction to the assumption that no such locking sequence exists. ■

Now let p be a positive presentation of L. We will construct a positive presentation q for L on which M diverges, which is a contradiction to the assumption that M identifies L in the limit:

Input: A positive presentation p of L.
Output: A positive presentation q of L on which M diverges.

```
$q \leftarrow$ empty sequence
for $n \in \mathbb{N}$ do
    $q \leftarrow q \circ (p_n)$
    $q \leftarrow q \circ \text{unlock}(q)$
```

The construction alternates between extending q with the next string from p, and extending q with its own unlocking sequence. Note that:

Claim 3. q is a positive presentation for L.

Proof. q is a well-defined infinite sequence of strings. All the strings in q are elements of L by the definitions of p and $\text{unlock}(\cdot)$. Every string in L appears in q at some point, because p is a positive presentation of L. ■

Claim 4. M diverges on q.

Proof. For each step n in the construction of q,

$$M(\langle q\rangle) \neq M(\langle q \circ \text{unlock}(q)\rangle)$$

so the output of M changes infinitely many times when executing on the input q. ■

Taken together, Claims 3 and 4 are a contradiction to the assumption that M identifies L in the limit. □

Definition 1.19 (Transfinite Language)
A set of languages is called *transfinite* if it contains all finite languages over some alphabet Σ and at least one infinite language over Σ.

Theorem 1.20 (Gold's Theorem)
Every transfinite set of languages is not identifiable in the limit.

Proof. Let $\mathcal{L}$ be a transfinite set of languages, and assume for contradiction that there exists a TM M that identifies $\mathcal{L}$ in the limit with respect to some computable indexing $(L_n)_{n\in\mathbb{N}}$. Let $L_\infty \in \mathcal{L}$ be an infinite language. From Lemma 1.18 there exists a finite locking sequence s for L_∞ that locks M onto k such that $L_k = L_\infty$. Let L_s be the finite set of strings in s. Let s_1 be the first string in s, and consider the infinite sequence $p = s \circ (s_1)_{n\in\mathbb{N}}$. Then p is a positive presentation for L_s. Because $\mathcal{L}$ is transfinite, $L_s \in \mathcal{L}$, and therefore M identifies L_s in the limit. In particular, $M(p) \to t$ such that $L_t = L_s$. Because $L_s \neq L_\infty$, $t \neq k$. This is a contradiction to s being a locking sequence that locks M onto k. □

1.4 Identifiable Sets of Languages

Remark 1.21. The proof of Gold's theorem highlights the main problem that makes language identification in the limit difficult to achieve: at some point, the learner might make up its mind that the target language is some language L (e.g., if the learner has seen a locking sequence for L), and from that point onward, it will not change its mind unless it sees a string that does not belong to L. If the correct target language is actually a subset $L' \subsetneq L$, then the learner will fail to identify it. We call this phenomena over-generalization, because the learner fixates on the solution L that is too general (too big a set), instead of identifying the correct solution L' that is more particular (is a smaller set).

The following theorem basically says that over-generalization is the only problem that can make a set of languages not identifiable. A set of languages is identifiable if and only if over-generalization can be avoided.

Definition 1.22 (Telltale Set)
Let $\mathcal{L}$ be a set of languages, and let $L \in \mathcal{L}$. A finite set of strings $T \subseteq L$ is a *telltale set for* L with respect to $\mathcal{L}$ if for any $L' \in \mathcal{L}$,

$$T \subseteq L' \implies L' \not\subset L,$$

namely every language $L' \in \mathcal{L}$ that is a strict subset of L does not contain T.

Theorem 1.23 (Angluin's Theorem)
A set of languages $\mathcal{L}$ over an alphabet Σ is identifiable in the limit if and only if the following two conditions are satisfied:

(i) Every language $L \in \mathcal{L}$ has a telltale set with respect to $\mathcal{L}$, and furthermore

(ii) The telltale sets are recursively enumerable in the following sense. There exists a computable indexing $(L_n)_{n \in \mathbb{N}}$ of $\mathcal{L}$ and a TM T such that for every $n \in \mathbb{N}$, if T is executed with input $\langle n \rangle$ then T enumerates a set T_n that is a telltale set for L_n.

Proof. "If" direction.

We assume that $\mathcal{L}$ satisfies conditions (i) and (ii) and prove that $\mathcal{L}$ is identifiable in the limit. Let $(L_n)_{n \in \mathbb{N}}$ be the computable indexing of $\mathcal{L}$ from condition (ii). Write $T_i^{(n)}$ to denote the subset of the telltale set T_i that is printed during the first n steps of executing T on input $\langle i \rangle$. Note that the mapping $(i, n) \mapsto T_i^{(n)}$ is computable. The TM M that identifies $\mathcal{L}$ operates as follows.

```
Input: A finite sequence of strings $(s_i)_{i \in [n]}$.
  for $k \in [n]$ do
    if $T_k^{(n)} \subseteq \{s_i\}_{i \in [n]} \subseteq L_k$ then        ▷ $T_k^{(n)}$ is a subset of the telltale set for $L_k$
      return $k$ and halt
  return 0 and halt
```

Let $L \in \mathcal{L}$, and let s be a positive presentation of L. Let $j \in \mathbb{N}$ be the minimal number such that $L_j = L$. We show that $M(s) \to j$. Let T_j be the telltale set enumerated for L_j by T. Because $T_j \subseteq L_j = L$ and s is a positive presentation of L, it holds that $T_j \subseteq \{s_i\}_{i \in [n]}$ for all n large enough, and therefore $T_j^{(n)} \subseteq \{s_i\}_{i \in [n]}$ for all n large enough. Therefore for $k = j$ the condition

$$T_k^{(n)} \subseteq \{s_i\}_{i \in [n]} \subseteq L_k. \tag{$*$}$$

is satisfied for all n large enough.

To prove that $M(s) \to j$, it suffices to show that for all $k < j$, condition $(*)$ is satisfied at most finitely many times. Fix $k < j$ and let N be large enough such that $T_k^{(n)} = T_k$ for all $n \geq N$. In fact N exists because T_k is finite. Assume that this value of k satisfies $(*)$ for some fixed $n \geq N$. Then $(*)$ is satisfied and $T_k^{(n)} = T_k$, so $T_k \subseteq \{s_i\}_{i \in [n]}$. And because s is a positive presentation of L, $\{s_i\}_{i \in [n]} \subseteq L$. Hence $T_k \subseteq L$. Because T_k is a telltale set for L_k, this implies that $L \not\subset L_k$. Furthermore, from the minimality of j, $L \neq L_k$. Hence $L \setminus L_k$ is nonempty. Let $x \in L \setminus L_k$. Because s is a positive presentation of L, there exists an index m such that $s_m = x$. Therefore this value of k does not satisfy $(*)$ whenever $n \geq m$. Thus, j is the minimal number that satisfies $(*)$ for all n large enough, and therefore $M(s) \to j$. Thus $\mathcal{L}$ is identifiable in the limit.

"Only if" direction.

We assume that $\mathcal{L}$ is identifiable in the limit and prove that $\mathcal{L}$ satisfies condition (ii), and this implies that condition (i) is also satisfied. Let M be a TM that identifies $\mathcal{L}$ with respect to some computable indexing $(L_n)_{n \in \mathbb{N}}$ of $\mathcal{L}$. The proof proceeds in 5 steps.

Claim 1. There is a TM E (for enumerate) that on input $\langle n \rangle$ enumerates all the finite sequences of strings in L_n.

Proof. The proof is by construction. It uses D the Turing machine for the computable indexing $(L_n)_{n \in \mathbb{N}}$ such that $s \in L_n$ if and only if $D(\langle n \rangle\, s) = 1$.

```
procedure $E(n)$                                   ▷ enumerates finite sequences of strings from $L_n$
  for $i \in \mathbb{N}_0$ do
    $S \leftarrow \varnothing$
    for $s$ in first $n$ strings of $\Sigma^*$ do
      if $D(\langle n, s \rangle) = 1$ then
        $S \leftarrow S \cup \{s\}$
    for $q \in H(S, i)$ do
      print $q$
```

```
procedure $H(S, n)$                ▷ enumerates all sequences of length $\leq n$ from finite set $S$
    $Q \leftarrow \varnothing$
    if $n = 0$ then
        $Q \leftarrow Q \cup \{()\}$                ▷ () denotes the empty sequence.
    else
        for $q \in H(S, n-1)$ do
            $Q \leftarrow Q \cup \{q\}$
            for $s \in S$ do
                $Q \leftarrow Q \cup \{q \circ (s)\}$
```

■

Claim 2. There is a TM T that on input $\langle n \rangle$ enumerates the telltale sets of L_n.

Proof. Again the proof is by construction. We use the fact that for every $n \in \mathbb{N}$ it is possible to compute the members p_i of a positive presentation p for L_n (by enumerating Σ^* and using D to filter out strings that do not belong to L_n).

```
procedure $T(n)$
    $q \leftarrow ()$
    $m_0 \leftarrow M(\langle q \rangle)$
    for $j \in \mathbb{N}$ do
        for $f \in E(n)$ do
            $m_j \leftarrow M(q \circ f)$
            if $m_j \neq m_{j-1}$ then
                for $s \in f \circ (p_j)$ do
                    print $s$
                $q \leftarrow q \circ f \circ (p_j)$
                break
```

■

Claim 3. For any $n \in \mathbb{N}$, T prints only a finite number of strings when executed on input $\langle n \rangle$.

Proof. Assume for contradiction that T prints an infinite number of strings for some $n \in \mathbb{N}$. Then the sequences q and $m = (m_k)_{k \in \mathbb{N}}$ (that are constructed during the computation) are both infinite. However, observe that q contains only members of L_n, and it contains all members of L_n (because in each step j we append p_j to q). Hence q is a positive presentation of L_n. But, from the construction of m_j, it holds that $m_j \neq m_{j-1}$ for all $j \in \mathbb{N}$. Hence $M(q)$ diverges, in contradiction to M identifying L_n in the limit. Thus the set of strings T_n that T prints on input $\langle n \rangle$ is finite. ■

Claim 4. Let t be the set of strings printed out when executing T on $\langle n \rangle$. Then t is a locking sequence for L_n that locks M onto n.

Proof. Observe that there exists no nonempty finite sequence f of strings from L_n such that $M(\langle t \circ f \rangle) \neq M(\langle t \rangle)$, because if such a sequence f existed then T would have eventually enumerated it and then printed out the strings in f after it printed t, which is a contradiction to our choice of t. This implies that t is a locking sequence for L that locks M onto some index $k = M(\langle t \rangle)$. To see that $k = n$, note that $t \circ p$ is a positive presentation of L_n, and so $M(t \circ p) \to n$. In particular there exists some finite prefix f of p such that $M(\langle t \circ f \rangle) = n$, and so $k = M(\langle t \rangle) = M((t \circ f)) = n$. ■

Claim 5. The set T_n of strings in t is indeed a telltale set for L_n.

> *Proof.* Clearly $T_n \subseteq L_n$, and from the third point T_n is finite. Assume for contradiction that there exists $m \in \mathbb{N}$, $m \neq n$, such that $T_n \subseteq L_m \subset L_n$. Let w be a positive presentation of L_m. Then $t \circ w$ is also a positive presentation of L_m. Hence $M(t \circ w) \to m$, in contradiction to t being a locking sequence for L_n that locks M onto n. ■

Note that in the edge case $L_n = \varnothing$, it holds that $\varnothing$ is a telltale set for L_n, and indeed T does not print any strings on input $\langle n \rangle$. □

1.5 Discussion

Golds's theorem shows that in some cases, identification in the limit is not possible because of over-generalization, and Angluin's theorem tells us that over-generalization is essentially the only type of problem that can make identification in the limit not possible. However, there are good reasons to believe that when humans learn in the real world, over-generalization is a difficulty that is readily and routinely overcome. The motto is:

Absence of evidence can be evidence of absence.

It would be valuable to consider a model of learning in which absence of evidence can count as evidence of absence, and that such a model could capture instances of learning that are common in the real world. As we will see, this can be done using *probability*.

Beyond that, another issue ignored (by design) in the model of identification in the limit is the issue of *resources*. Because real-world learners have limitations on their computational space and runtime complexities, as well as the amount of training data that they receive, it makes sense to ask what can be learned efficiently? (namely, using a reasonable amount of time, space, and data).

2 Probably Approximately Correct Learning

2.1 The PAC Definition

In this section we present the Probably Approximately Correct (PAC) definition of learning. We will see a few other definitions throughout the course, but this definition (and a variant of it called *agnostic* PAC) will be our central definition.

PAC is a *probabilistic* definition of learning. Two main motivations for taking a probabilistic approach are that this allows us to:

- Formally make inferences in which "absence of evidence is evidence of absence," as we discussed in the previous lecture.

- Model learning in the presence of uncertainty and noise.

Additionally, the PAC definition allows us to model learning with finite amounts of data and computational resources (instead of infinite positive presentations, and infinite computations that only converge in the limit).

2.1.1 Components of a Learning Problem

In a *learning problem*, a learner is attempting to predict the specific *labels* $y \in \mathcal{Y}$ that correspond to specific instances $x \in \mathcal{X}$. More fully, the learner is attempting to learn an unknown function $f : \mathcal{X} \to \mathcal{Y}$. To do so, it is given example input-output pairs $(x, f(x))$, where x is chosen randomly from $\mathcal{X}$. The setting includes the following components:

- $\mathcal{X}$ – a set called the *domain* or the *instance space*. We define a measurable space $(\mathcal{X}, \mathcal{F})$.

- $\mathcal{Y}$ – a set called the *co-domain* or the *label space*.

- The unknown system:
  - $f : \mathcal{X} \to \mathcal{Y}$ – a *target function*.
  - $\mathcal{D}$ – a distribution (probability measure) over $\mathcal{X}$.

- S – a random variable called the *sample* or the *training set*. This is the data presented to the learner. There are a number of ways in which the data might be generated, but we will mostly consider the following setting: $S = \{(x_i, y_i)\}_{i \in [m]}$, such that for every $i \in [m]$, $x_i \in \mathcal{X}$ is sampled independently according to $\mathcal{D}$ and $y_i = f(x_i) \in \mathcal{Y}$. The assumption that the x_i's are sampled independently from $\mathcal{D}$ is called the *i.i.d. assumption*.

- Learner – a (possibly randomized) algorithm that takes S as input and produces a *hypothesis function* $h : \mathcal{X} \to \mathcal{Y}$ as output. Also called a *learning algorithm*.

- $L_{\mathcal{D},f}(h)$ – a *loss function* that measures how well the hypothesis h is able to predict the labels assigned by f with respect to the distribution $\mathcal{D}$. We will generally assume that

  $$L_{\mathcal{D},f}(h) = \mathop{\mathsf{E}}_{x \sim \mathcal{D}}[\ell(f(x), h(x))]$$

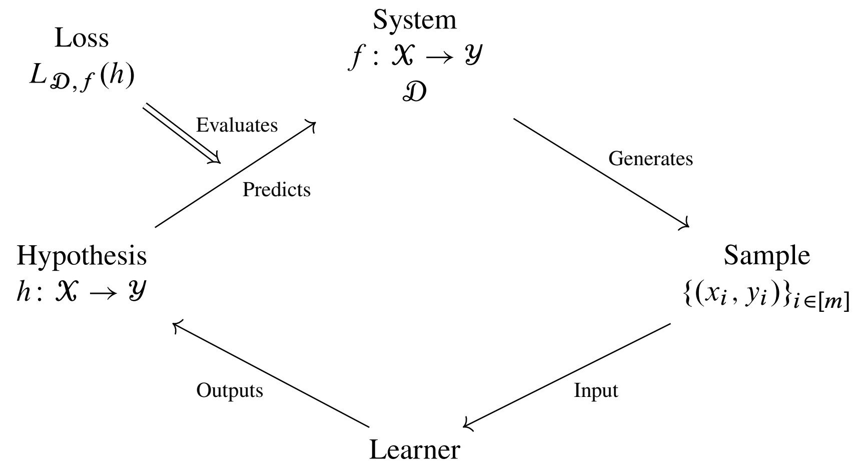
  for some function ℓ. In particular, we will mostly focus on the "0-1 loss", where $\ell(y, y') = \mathbb{1}(y \neq y')$, which yields

  $$L_{\mathcal{D},f}(h) = \mathop{\mathsf{P}}_{x \sim \mathcal{D}}[h(x) \neq f(x)] \, .$$

However, there are many other reasonable choices for loss functions. For example, if $\mathcal{Y} = \mathbb{R}$ we can choose $\ell(y, y') = (f(x) - h(x))^2$, which yields the expected square error loss,

$$L_{\mathcal{D},f}(h) = \mathop{\mathsf{E}}_{x\sim\mathcal{D}}\left[(h(x) - f(x))^2\right].$$

Here is a diagram of the system:



All details of the setting are known to the learner except for the target function f and the distribution $\mathcal{D}$. The learner's objective is to output a hypothesis h such that $L_{\mathcal{D},f}(h)$ is as small as possible.

Remark 2.1. The unknown distribution $\mathcal{D}$ plays a double role: it determines how the sample S is generated, and it also defines the loss function $L_{\mathcal{D},f}$ used to evaluate the performance of the learner.

Remark 2.2. The current setting does not meet our objective of dealing with uncertainty (e.g., noisy labels). We have made a fairly strong assumption about the unknown system – that the label y is a deterministic function of x. We will see how this assumption can be relaxed.

2.1.2 PAC Definition – First Attempt

We want the learner to find a hypothesis h that has loss as small as possible. It might seem desirable to require that the learner find h such that $L_{\mathcal{D},f}(h)$, namely $h = f$ on all points in the support of $\mathcal{D}$. However, there are two issues with this:

- The learner only gets a limited amount of information about the target function f, because it gets a sample that only contains a finite number m of input-output pairs. Therefore we generally cannot hope for the learner to reconstruct f precisely. Instead, we can go for the next best thing, which is to require that $L_{\mathcal{D},f}(h) \leq \varepsilon$ for some small positive ε. That is, we require that h be *approximately correct*. Moreover, we can make $\varepsilon > 0$ arbitrarily small. The more data the learner gets, the better its predictions can be. That is, we require that for every $\varepsilon > 0$ there exists $m \in \mathbb{N}$ such that if the algorithm receives a sample of size m then it will output h such that $L_{\mathcal{D},f}(h) \leq \varepsilon$. (Basically, $L_{\mathcal{D},f}(h) \to 0$ as $m \to \infty$, with upper bounds on $L_{\mathcal{D},f}(h)$ also for finite sample sizes m and not only in the limit.)

- Seeing as the sample is generated randomly according to $\mathcal{D}$, there is always a possibility that the learner will be unlucky and get a "bad" sample. For instance, there is a nonzero probability that the sample $S = \{(x_i, y_i)\}_{i\in[m]}$ will have $x_1 = \cdots = x_m$, and so the learner only receives information on a single input-output pair, even though the sample is of size $m > 0$. Therefore, we cannot hope for the learner to always achieve low loss. However, because the probability of getting a bad sample is low, we can require that it will *probably* succeed, i.e., the learner achieves low loss with high probability.

Combining these two considerations, we get the *probably approximately correct* (PAC) definition of learning. Following is a first (problematic) attempt at formalizing this notion.

Definition 2.3 (Naive PAC Learning)
Let $\mathcal{X}$ and $\mathcal{Y}$ be sets. We say that a (possibly randomized) algorithm A is a *naive probably approximately correct learner for functions* $\mathcal{X} \to \mathcal{Y}$ if for any precision parameter $\varepsilon \in (0,1)$ and confidence parameter $\delta \in (0,1)$ there exists a sample size $m \in \mathbb{N}$ such that for every target function $f \colon \mathcal{X} \to \mathcal{Y}$ and every distribution $\mathcal{D}$ over $\mathcal{X}$, if A receives as input the parameters ε and δ and a sample S such that $S = \{(x_i, f(x_i))\}_{i \in [m]}$ where $\{x_i\}_{i \in [m]} \overset{\text{i.i.d.}}{\sim} \mathcal{D}$, then A halts and outputs a hypothesis $h \colon \mathcal{X} \to \mathcal{Y}$ such that, with probability at least $1 - \delta$ has loss $L_{\mathcal{D},f}(h) \leq \varepsilon$.

Notation 2.4
Let $\mathcal{X}$ and $\mathcal{Y}$ be sets, $\mathcal{D}$ a distribution on $\mathcal{X}$ and $f \colon \mathcal{X} \to \mathcal{Y}$ be a function. We write $z \sim (\mathcal{D}, f)$ to denote that z is a random variable such that $z = (x, y)$ where $x \in \mathcal{X}$ is sampled according to $\mathcal{D}$ and $y = f(x)$.

The requirement of naive PAC learning can be summarized precisely as

$$\mathop{\mathsf{P}}_{S}\big[L_{\mathcal{D},f}(h) \leq \varepsilon\big] \geq 1 - \delta \tag{2.1}$$

where $h = A(\varepsilon, \delta, S)$ and the probability is over the randomness of the sample S and any randomness used by A.

The number m in the definition is a number of examples that depend on ε and δ and is sufficient to guarantee that the algorithm satisfies Equation 2.1.

Definition 2.5
Let A be an algorithm that is a naive PAC learner for functions $\mathcal{X} \to \mathcal{Y}$, for some sets $\mathcal{X}, \mathcal{Y}$. The *sample complexity of* A is a function $m \colon (0,1)^2 \to \mathbb{N}$ such that for every $\varepsilon, \delta \in (0,1)$, the number $m(\varepsilon, \delta)$ is the minimal sample size for which A satisfies Equation 2.1.

Proposition 2.6
Let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty sets and assume that $\mathcal{X}$ is finite. There exists an algorithm A that naive PAC learns the functions $\mathcal{X} \to \mathcal{Y}$ and has sample complexity

$$m(\varepsilon, \delta) \leq \left\lceil \frac{|\mathcal{X}| + \log(1/\delta)}{\varepsilon} \right\rceil.$$

The learning algorithm simply "memorizes" the sample. Given a sample $S = \{(x_i, y_i)\}_{i \in [m]}$, it outputs the following hypothesis h. For every $x \in \mathcal{X}$, if there exists $i \in [m]$ such that $x = x_i$, then $h(x) = y_{i^*}$ where $i^* = \min\{i \in [m] \colon x_i = x\}$. Otherwise, $h(x) = y'$ for some arbitrary fixed $y' \in \mathcal{Y}$.

The idea of the proof is simple: if we take a number of samples that is linear in $|\mathcal{X}|$, then we will see the correct labels for nearly all of $\mathcal{X}$, except possibly some small fraction of weight at most ε, and so by memorizing these labels we achieve loss less than ε.

Proof of Proposition 2.6. Fix $\varepsilon > 0$, $\delta > 0$, a distribution $\mathcal{D}$, and a target function f. We show that if the memorization algorithm described in the previous paragraph receives a sample S of size $m = \left\lceil \frac{|\mathcal{X}| + \log(1/\delta)}{\varepsilon} \right\rceil$, then it outputs h such that $\mathsf{P}_S\big[L_{\mathcal{D},f}(h) > \varepsilon\big] < \delta$.

We say that a set $X \subseteq \mathcal{X}$ is *bad* if $\mathcal{D}(X) > \varepsilon$ and for all $x \in X$, $h(x) \neq f(x)$. Note that

$$L_{\mathcal{D},f}(h) = \mathop{\mathsf{P}}_{x \sim \mathcal{D}}[h(x) \neq f(x)] = \mathcal{D}(\{x \in \mathcal{X} : h(x) \neq f(x)\})$$

so

$$L_{\mathcal{D},f}(h) > \varepsilon \iff \exists X \subseteq X : X \text{ is bad.}$$

Hence

$$\begin{aligned}\mathop{\mathsf{P}}_{S}\big[L_{\mathcal{D},f}(h) > \varepsilon\big] &= \mathop{\mathsf{P}}_{S}[\exists X \subseteq \mathcal{X} : X \text{ is bad}] \\ &= \mathop{\mathsf{P}}_{S}\left[\bigcup_{X \subseteq \mathcal{X}} \{X \text{ is bad}\}\right] \\ &\leq \sum_{X \subseteq \mathcal{X}} \mathop{\mathsf{P}}_{S}[X \text{ is bad}] .\end{aligned}$$

Let $S_x = \{x_i : i \in [m]\} \subseteq \mathcal{X}$ and observe that if $X \cap S_x \neq \varnothing$ then X is not bad, because if $x \in S_x$ then h will memorize the correct label for x and then $h(x) = f(x)$.

Next, we show that for any $X \subseteq \mathcal{X}$, it holds that $\mathop{\mathsf{P}}_{S}[X \text{ is bad}] < \frac{\delta}{2^{|\mathcal{X}|}}$. This suffices to complete the proof, because it implies that $\sum_{X \subseteq \mathcal{X}} \mathop{\mathsf{P}}_{S}[X \text{ is bad}] < \delta$.

Notice that for any $X \subseteq \mathcal{X}$, if $\mathcal{D}(X) \leq \varepsilon$ then X is never bad, and if $\mathcal{D}(X) > \varepsilon$ then

$$\begin{aligned}\mathop{\mathsf{P}}_{S}[X \text{ is bad}] &\leq \mathop{\mathsf{P}}_{S}[X \cap S_x = \varnothing] \\ &= \mathop{\mathsf{P}}_{S}\left[\bigcap_{i \in [m]} \{x_i \notin X\}\right] = \prod_{i \in [m]} \mathop{\mathsf{P}}_{S}[x_i \notin X] \\ &\leq \prod_{i \in [m]} (1 - \varepsilon) = (1 - \varepsilon)^m \leq \mathrm{e}^{-\varepsilon m}.\end{aligned}$$

But

$$\mathrm{e}^{\varepsilon m} < \frac{\delta}{2^{|\mathcal{X}|}} \iff m > -\frac{\log\left(\frac{\delta}{2^{|\mathcal{X}|}}\right)}{\varepsilon} = \frac{|\mathcal{X}| \log(2) + \log\left(\frac{1}{\delta}\right)}{\varepsilon}$$

and so taking

$$m = \left\lceil \frac{|\mathcal{X}| + \log\left(\frac{1}{\delta}\right)}{\varepsilon} \right\rceil \geq \frac{|\mathcal{X}| \log(2) + \log\left(\frac{1}{\delta}\right)}{\varepsilon}$$

is sufficient to ensure that $\mathrm{e}^{-\varepsilon m} < \frac{\delta}{2^{|\mathcal{X}|}}$, as desired. □

This result is not very surprising. Clearly, if we see the labels for nearly all $x \in \mathcal{X}$ then we can perform well on these x's. But this only captures memorization, which is a rudimentary (and quite boring) type of learning. Memorization performs poorly in more realistic situations where the training set contains only a small part of $\mathcal{X}$. Broadly speaking, in learning theory we are more interested in learners that can *generalize* – can use the instances received in the training sample in order to make good predictions about new, as yet unseen instances not present in the training sample.

Definition 2.7

Let $S = \{(x_i, y_i)\}_{i \in [m]}$ be a sample and $h : \mathcal{X} \to \mathcal{Y}$ be a hypothesis. The *empirical loss of h with respect to S*

is $L_S(h) = \dfrac{1}{m} \sum\limits_{i \in [m]} \ell(y_i, h(x_i))$ which in the case of the 0-1 loss is

$$L_S(h) = \frac{1}{m} \sum_{i \in [m]} \mathbb{1}(y_i \neq h(x_i)) = \frac{|\{i \in [m] \colon h(x_i) \neq y_i\}|}{m}.$$

The *out-of-sample loss of h with respect to a distribution $\mathcal{D}$ and target function f* is given by

$$\mathop{\mathsf{E}}_{x \sim \mathcal{D}}[\ell(f(x), h(x)) \mid x \notin S_x],$$

which for the 0-1 loss is

$$\mathop{\mathsf{P}}_{x \sim \mathcal{D}}[f(x) \neq h(x) \mid x \in S_x]$$

where $S_x = \{x_i\}_{i \in [m]}$.

Remark 2.8. To distinguish $L_{\mathcal{D},f}$ from the empirical loss L_S and from the out-of-sample loss, we will sometimes refer to $L_{\mathcal{D},f}$ as the *true loss* or the *population loss*.

The memorization algorithm achieves a perfect empirical loss of $L_S(h) = 0$, but its out-of-sample loss is basically the same as what we would expect if we simply assigned labels arbitrarily or by chance – about as poor as possible. Our goal is to achieve generalization, which can be formalized either as saying that $|L_S(h) - L_{\mathcal{D},f}(h)|$ is small, or as saying that the out-of-sample loss is small. Unfortunately, we will see in the next section that this is not possible in the general case.

2.1.3 No Free Lunch

In Definition 2.3 we did not make any assumptions on the target function f – it could be any function whatsoever. Therefore, the values $f(x_i)$ that the learner receives for x_i's in the sample convey no information at all about the value $f(x)$ for any x not in the sample. Therefore, neither memorization or any other algorithm can perform better than chance on out-of-sample instances. Formally:

Theorem 2.9 (No Free Lunch)
Let $\mathcal{X}$ be a nonempty finite set, $\mathcal{Y} = \{0, 1\}$. Let $\mathcal{U}$ be the uniform distribution over $\mathcal{X}$. Then for any learning algorithm A for functions $\mathcal{X} \to \mathcal{Y}$ and any $m \in \mathbb{N}$,

$$\frac{1}{2} - \frac{m}{2|\mathcal{X}|} \leq \mathop{\mathsf{E}}_{f \sim \mathrm{Uni}(\mathcal{Y}^{\mathcal{X}})}\left[\mathop{\mathsf{E}}_{S \sim (\mathcal{U}, f)^m}\left[L_{\mathcal{U},f}(h)\right]\right] \leq \frac{1}{2} + \frac{m}{2|\mathcal{X}|},$$

where $h = A(S)$.

Proof. Define a sample $S = \{(x_i, y_i)\}_{i \in [m]}$ and a function f to be *consistent* if $y_i = f(x_i)$ for all $i \in [m]$, and likewise S is *self-consistent* if there exists some function f such that S and f are consistent. Observe that in the theorem we have two random variables f and S in a joint finite probability space $\left(\mathcal{Y}^{\mathcal{X}} \times \mathcal{X}^n, \sigma\left(\mathrm{pow}\left(\mathcal{Y}^{\mathcal{X}} \times \mathcal{F}^n\right)\right)\right)$, and we are interested in the expectation of the function $g(f, S) = L_{\mathcal{U},f}(A(S))$. The marginal distribution of f is uniform over $\mathcal{Y}^{\mathcal{X}}$, the marginal of S is uniform over all self-consistent samples, and the joint distribution is such that f and S are consistent $\mathbb{P}$-a.e..

Fix $f \in \mathcal{Y}^{\mathcal{X}}$ and let S_x denote the subset of $\mathcal{X}$ that appears in S. Then

$$\mathop{\mathsf{E}}_{f}\left[\mathop{\mathsf{E}}_{S}\left[L_{\mathcal{U},f}(h)\right]\right] = \mathop{\mathsf{E}}_{f}\left[\mathop{\mathsf{E}}_{S}\left[\frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \mathbb{1}(f(x) \neq h(x))\right]\right]$$

$$
\begin{aligned}
&= \mathop{\mathsf{E}}_{f}\left[\mathop{\mathsf{E}}_{S}\left[\frac{1}{|\mathcal{X}|}\sum_{x \in \mathcal{X} \setminus S_x} \mathbb{1}(f(x) \neq h(x))\right]\right] \\
&\quad + \mathop{\mathsf{E}}_{f}\left[\mathop{\mathsf{E}}_{S}\left[\frac{1}{|\mathcal{X}|}\sum_{x \in S_x} \mathbb{1}(f(x) \neq h(x))\right]\right].
\end{aligned}
$$

For the first term, using the tower rule,

$$
\mathop{\mathsf{E}}_{f}\left[\mathop{\mathsf{E}}_{S}\left[\frac{1}{|\mathcal{X}|}\sum_{x \in \mathcal{X} \setminus S_x} \mathbb{1}(f(x) \neq h(x))\right]\right] = \frac{1}{|\mathcal{X}|}\mathop{\mathsf{E}}_{S}\left[\sum_{x \in \mathcal{X} \setminus S_x} \mathop{\mathsf{E}}_{f}\left[\mathbb{1}(f(x) \neq h(x)) \,\middle|\, S\right]\right].
$$

The random variable $f \mid S$ is uniform over $f \in \mathcal{Y}^{\mathcal{X}}$ that are consistent with S; in particular, for any fixed $x \in \mathcal{X} \setminus \mathcal{S}_x$, the value $f(x)$ is uniform over $\mathcal{Y}$, and hence

$$
\mathop{\mathsf{E}}_{f}\left[\mathbb{1}(f(x) \neq h(x)) \,\middle|\, S\right] = \frac{1}{2}.
$$

Hence

$$
\frac{1}{|\mathcal{X}|}\mathop{\mathsf{E}}_{S}\left[\sum_{x \in \mathcal{X} \setminus S_x} \mathop{\mathsf{E}}_{f}\left[\mathbb{1}(f(x) \neq h(x)) \,\middle|\, S\right]\right] = \frac{|\mathcal{X}| - |S_x|}{2\,|\mathcal{X}|} = \frac{1}{2} - \frac{|S_x|}{2\,|\mathcal{X}|}.
$$

For the second term, note that

$$
0 \leq \mathop{\mathsf{E}}_{f}\left[\sum_{x \in \mathcal{X} \setminus S_x} \mathop{\mathsf{E}}_{S}\left[\frac{1}{|\mathcal{X}|}\sum_{x \in S_x} \mathbb{1}(f(x) \neq h(x))\right]\right] \leq \frac{|S_x|}{|\mathcal{X}|}.
$$

Finally, plugging these expressions back in gives

$$
\frac{1}{2} - \frac{|S_x|}{2\,|\mathcal{X}|} \leq \mathop{\mathsf{E}}_{S}\left[L_{\mathcal{U},f}(h)\right] \leq \frac{1}{2} + \frac{|S_x|}{2\,|\mathcal{X}|}.
$$

Noting that $|S_x| \leq m$ completes the proof. □

Hence, without any assumption on the target function f, it doesn't matter which learning algorithm we choose, if the sample size is small relative to $\mathcal{X}$ the the expected loss would approximately equal $\frac{1}{2}$, which is the same as it would be if we guessed the labels at random. In particular, as the following corollary shows, naive PAC learning is not possible unless the sample size is linear in $\mathcal{X}$.

Corollary 2.10

Let $\mathcal{X}$ be a nonempty finite set and $\mathcal{Y} = \{0, 1\}$. The following holds with respect to the uniform distribution $\mathcal{U}$ over $\mathcal{X}$. For any learning algorithm A for functions $\mathcal{X} \to \mathcal{Y}$, there exists a function $f \colon \mathcal{X} \to \mathcal{Y}$ such that if A receives a sample S of size $m \leq \frac{|\mathcal{X}|}{2}$ from $\mathcal{U}$ labeled according to f then A is a poor naive PAC learner in the sense that

$$
\mathop{\mathsf{P}}_{S}\left[L_{\mathcal{U},f}(h) \geq \frac{1}{8}\right] \geq \frac{1}{7}.
$$

Proof. Fix a learning algorithm A. From Theorem 2.9 there exists a function f such that

$$
\mathop{\mathsf{E}}_{S}\left[L_{\mathcal{U},f}(h)\right] \geq \frac{1}{2} - \frac{m}{2\,|\mathcal{X}|} \geq \frac{1}{4},
$$

for $h = A(S)$. Let q denote the accuracy of h with respect to f, i.e., $q = 1 - L_{\mathcal{U},f}(h)$. From Markov's inequality,

$$\mathop{\mathsf{P}}_{S}\left[q \geq \frac{7}{8}\right] \leq \frac{8\,\mathsf{E}_S[q]}{7} = \frac{8}{7}\left(1 - \mathop{\mathsf{E}}_{S}\left[L_{\mathcal{U},f}(h)\right]\right) \leq \frac{8}{7}\left(1 - \frac{1}{4}\right) = \frac{6}{7}.$$

in other words,

$$\frac{1}{7} \leq 1 - \mathop{\mathsf{P}}_{S}\left[q \geq \frac{7}{8}\right] = \mathop{\mathsf{P}}_{S}\left[q < \frac{7}{8}\right] = \mathop{\mathsf{P}}_{S}\left[L_{\mathcal{U},f}(h) > \frac{1}{8}\right].$$

Hence, if the domain is infinite, naive PAC learning is impossible. □

Corollary 2.11

Let $\mathcal{X}$ be an infinite set and $\mathcal{Y} = \{0, 1\}$. Then there does not exist an algorithm that naive PAC learns the functions $\mathcal{X} \to \mathcal{Y}$.

Proof. Assume for sake of contradiction that A is a learning algorithm that naive PAC learns the functions $\mathcal{X} \to \mathcal{Y}$. Then there exists $m \in \mathbb{N}$ such that for all distributions $\mathcal{D}$ and target functions f, if $S \sim (\mathcal{D}, f)^m$ and $h = A(S)$ satisfies

$$\mathop{\mathsf{P}}_{S}\left[L_{\mathcal{D},f}(h) < \frac{1}{10}\right] > \frac{9}{10}.$$

Fix $X \subseteq \mathcal{X}$ such that $|X| = 2m$, and let $\mathcal{U}$ be the uniform distribution on X. By Corollary 2.10, there exists a target function f such that, for $S \sim (\mathcal{U}, f)^m$,

$$\mathop{\mathsf{P}}_{S}\left[L_{\mathcal{U},f}(h) \geq \frac{1}{8}\right] \geq \frac{1}{7}$$

a contradiction. □

The last corollary suggests that the definition of naive PAC learning might be too strong (be too hard to achieve), because there are many examples in the real world where it appears to be possible to generalize well when learning over an infinite domain. For example, spam filters are fairly good at classifying whether an email is spam or ham, even though the suitable domain $\mathcal{X}$ (the set of all possible emails) is infinite.

2.2 A Better PAC Definition

2.2.1 Incorporating Inductive Bias

In the previous section we presented one possible formalization of Hume's observation, showing that if we do not assume anything about the unknown system then we cannot generalize from instances that appear in the sample to unseen instances. Therefore, it appear that successful generalization requires making more assumptions about the unknown system.

These assumptions are called inductive bias, because the learner has an a-priori preference for (or bias in favor of) some hypotheses over other hypotheses.

One example is the preference in science for simple hypotheses. All else being equal, scientists will usually prefer a simple hypothesis over a more complex one, even if they both fit the evidence equally well. This bias is known as Occam's razor, and we will revisit it later on in the course.

Notice that in Corollary 2.10 we say that even if we know exactly what the unknown distribution $\mathcal{D}$ is (specifically, even if we know that it is the uniform distribution over $\mathcal{X}$), we still cannot generalize well. Therefore, a reasonable approach to inductive bias is to make assumption on the target function f. In particular, a natural assumption which we will explore in depth, is that f belongs to some subset, or *hypothesis class* $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. We will see that if $\mathcal{H}$ is simple in some sense, or has some structure, then generalization is indeed possible.

Formally, adopting the assumption that $f \in \mathcal{H}$ for some class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ yields the following definition of learning.

Definition 2.12 (PAC Learning)
Let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty sets and let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a class of functions. We say that a (possibly randomized) algorithm A is a *probably approximately correct (PAC) learner for* $\mathcal{H}$ if there exists a sample complexity function $m\colon (0,1)^2 \to \mathbb{N}$ such that for every precision parameter $\varepsilon \in (0,1)$, every confidence parameter $\delta \in (0,1)$, every target function $f \in \mathcal{H}$, and every distribution $\mathcal{D}$ over $\mathcal{X}$, if A receives as input the parameters ε and δ and a sample S of size $m = m(\varepsilon, \delta)$ such that $S = \{(x_i, f(x_i))\}_{i\in[m]}$ where $\{x_i\}_{i\in[m]} \overset{\text{i.i.d.}}{\sim} \mathcal{D}$, then A halts and outputs a hypothesis $h \in \mathcal{H}$ that with probability at least $1 - \delta$ (over the sample S and the randomness of A) has loss $L_{\mathcal{D},f}(h) \le \varepsilon$.

We say that a class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is *PAC learnable* if there exists an algorithm that is a PAC learner for $\mathcal{H}$.

This definition is similar to Definition 2.3, the difference being that we assume that $f \in \mathcal{H}$. This is called the realizability assumption.

2.2.2 Finite Hypothesis Classes are PAC Learnable

Perhaps the simplest assumption to make about the hypothesis class $\mathcal{H}$ is that it is finite. We now show that this suffices to ensure PAC learnability, even if the domain is infinite.

Lemma 2.13
Let $\mathcal{X}$ and $\mathcal{Y}$ be (finite or infinite) nonempty sets, and let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be finite. Then $\mathcal{H}$ is PAC learnable with sample complexity

$$m(\varepsilon, \delta) = \left\lceil \frac{\log(|\mathcal{H}|/\delta)}{\varepsilon} \right\rceil.$$

The idea is similar to that of Proposition 2.6, which showed that naive PAC learning is possible for finite domains using the memorization algorithm. However, observe that the memorization algorithm does not satisfy Definition 2.12. Intuitively, in order to avoid the no-free-lunch theorems, we need an algorithm which utilizes inductive bias, namely, which uses the assumption that $f \in \mathcal{H}$ to its advantage - but memorization doesn't do this. More concretely, Corollary 2.10 states that for any algorithm A (including a memorization algorithm), there exists a target function $f_{\text{fail-}A}$ for which the algorithm A fails, and so in particular A is a not a PAC learner for the finite class $\mathcal{H} = \{f_{\text{fail-}A}\}$. To avoid this issue, we need to design our algorithm A in a manner that depends on the class $\mathcal{H}$ so that $f_{\text{fail-}A} \notin \mathcal{H}$. Thus, under the assumption that the target function $f \in \mathcal{H}$, we know that A will not be executed with the target function $f = f_{\text{fail}-A}$, and so it is possible for A to succeed.

We design a learner A hat uses the assumption that $f \in \mathcal{H}$, and in particular ensures that $f_{\text{fail-}A} \notin \mathcal{H}$. We use an approach called *empirical risk minimization (ERM)*.

```
procedure ERM_$\mathcal{H}$($S$)
    return any $h \in \mathcal{H}$ such that $L_S(h) = 0$.
```

From the realizability assumption, there exists at least one hypothesis $h \in \mathcal{H}$ such that $L_S(h) = 0$. If there are multiple such hypotheses, the algorithm chooses one arbitrarily.

We can think of $\mathrm{ERM}_{\mathcal{H}}$ as a constrained version of the memorization algorithm. It outputs a hypothesis h such that $h(x_i) = y_i$ for all x_i in the sample, but in addition it satisfies the constraint that $h \in \mathcal{H}$.

Proof of Lemma 2.13. Fix a finite class $\mathcal{H}$, a target function $f \in \mathcal{H}$, parameters $\varepsilon > 0$ and $\delta > 0$, and a distribution $\mathcal{D}$. Let $S \sim (\mathcal{D}, f)^m$. We show that

$$\mathop{\mathsf{P}}_{S}\big[L_{\mathcal{D},f}(\mathrm{ERM}_{\mathcal{H}}(S)) > \varepsilon\big] < \delta$$

for $m = \left\lceil \frac{\log(|\mathcal{H}|/\delta)}{\varepsilon} \right\rceil$.

Let $h \in \mathcal{H}$. We say that the sample S is *bad for h* if $L_S(h) = 0$ and $L_{\mathcal{D},f}(h) > \varepsilon$. Notice that if $L_{\mathcal{D}}(\mathrm{ERM}_{\mathcal{H}}(S)) > \varepsilon$, then S is bad for the hypothesis $h = \mathrm{ERM}_{\mathcal{H}}(S)$. Hence

$$\begin{aligned}
\underset{S}{\mathsf{P}}\big[L_{\mathcal{D},f}(\mathrm{ERM}_{\mathcal{H}}(S)) > \varepsilon\big] &\leq \underset{S}{\mathsf{P}}[\exists h \in \mathcal{H} \colon S \text{ is bad for } h] \\
&= \underset{S}{\mathsf{P}}\left[\bigcup_{h \in \mathcal{H}} \{S \text{ is bad for } h\}\right] \\
&\leq \sum_{h \in \mathcal{H}} \underset{S}{\mathsf{P}}[S \text{ is bad for } h]
\end{aligned}$$

We know show that for every $h \in \mathcal{H}$, $\mathsf{P}_S[S \text{ is bad for } h] < \frac{\delta}{|\mathcal{H}|}$. Fix $h \in \mathcal{H}$. If $L_{\mathcal{D},f}(h) \leq \varepsilon$ then S is never bad for h. Otherwise,

$$\begin{aligned}
\underset{S}{\mathsf{P}}[S \text{ is bad for } h] &= \underset{S}{\mathsf{P}}[L_S(h) = 0] \\
&= \underset{S}{\mathsf{P}}\left[\bigcap_{i \in [m]} \{f(x_i) = h(x_i)\}\right] \\
&= \prod_{i \in [m]} \underset{S}{\mathsf{P}}[f(x_i) = h(x_i)] \\
&< (1 - \varepsilon)^m \leq \mathrm{e}^{-\varepsilon m} \leq \frac{\delta}{|\mathcal{H}|}
\end{aligned}$$

where the last inequality holds whenever $m \geq \frac{\log(|\mathcal{H}|/\delta)}{\varepsilon}$. By combining these inequalities, we see that taking a sample of size $m(\varepsilon, \delta)$ as in the statement ensures that

$$\underset{S}{\mathsf{P}}\big[L_{\mathcal{D},f}(\mathrm{ERM}_{\mathcal{H}}(S)) > \varepsilon\big] < \sum_{h \in \mathcal{H}} \frac{\delta}{|\mathcal{H}|} = \delta$$

as desired. □

2.3 Discussion

So far we have seen two extreme cases: if the hypothesis class $\mathcal{H}$ is finite, then it is PAC learnable, while if $\mathcal{H}$ is the set of all functions over an infinite domain, then it is not PAC learnable. But there are many useful classes that lie in between these two extremes. A major goal for us will be to chart this territory. Ideally, we would like to find a simple characterization that applies to all classes, and tells us precisely which classes are PAC learnable, and with what sample complexity.

Additionally, our definition of PAC learning required a relatively strong assumption, namely that the labels of the unknown system perfectly match some function f. But in many real-world scenarios this is not the case. Next lecture we will see how we can modify our definition of learnability to model these scenarios as well.

3 Concentration of Measure

3.1 Introduction

When learning in a statistical setting, the more evidence we see, the more confident we become. For example, if we tossed a (possibly biased) coin once and it came out heads, that does not necessarily mean that heads is a more likely outcome than tails. But if we tossed the coin 100 times and it always came out heads, then we can be fairly certain that the coin is strongly biased in favor of heads.

Formally, this phenomena is captured by *concentration of measure*. In a Gaussian distribution $\mathcal{N}\left(\mu, \sigma^2\right)$, more than 95% of the probability mass is concentrated within two standard deviations of the mean. Namely,

$$\underset{X \sim \mathcal{N}(\mu,\sigma^2)}{\mathsf{P}}[|X - \mu| \leq 2\sigma] \geq \frac{19}{20}.$$

Notation 3.1
If we don't specify the codomain of a random variable, we will assume the codomain is (a subset of) $\mathbb{R}$.

Notation 3.2
Suppose $(X_n)_{n \in \mathbb{N}}$ is a sequence of random variables. Then we define

$$\overline{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i .$$

Theorem 3.3 (Central Limit Theorem)
Suppose $\{X_n\}_{n \in \mathbb{N}}$ is an i.i.d. sequence of random variables for which $\mathsf{E}\left[X_1^2\right] < \infty$, then

$$\overset{d}{\lim_{n \to \infty}} \sqrt{n}\left(\overline{X}_n - \mathsf{E}[X_1]\right) = \mathcal{N}\left(0, \mathsf{Var}(X_1)\right).$$

Theorem 3.4 (Weak Law of Large Numbers)
Suppose $\{X_n\}_{n \in \mathbb{N}}$ is an i.i.d. sequence of random variables for which $\mathsf{E}[|X_1|] < \infty$, then

$$\overset{p}{\lim_{n \to \infty}} \overline{X}_n = \mathsf{E}[X_1].$$

Theorem 3.5 (Strong Law of Large Numbers)
Suppose $\{X_n\}_{n \in \mathbb{N}}$ is an i.i.d. sequence of random variables for which $\mathsf{E}[|X_1|] < \infty$, then

$$\overset{\text{a.s.}}{\lim_{n \to \infty}} \frac{1}{n} \sum_{i=1}^{n} X_i = \mathsf{E}[X_1].$$

The problem is that the central limit theorem and the law of large numbers are asymptotic results that only tell us what happens in the limit. In order to use concentration of measure for learning with a finite sample complexity, we need a quantitative version of these results that can tell us how good our estimates are after seeing a finite number n of examples.

In this unit we will prove a quantitative theorem of this form called Hoeffding's inequality, which roughly says that

$$\mathsf{P}\left[\left|\overline{X}_n - \mu\right| > \varepsilon\right] \leq \mathrm{e}^{-\Omega(n\varepsilon^2)}.$$

This theorem will be very useful for us when we continue to investigate the PAC model of learning.

3.2 Moment Generating Functions

Definition 3.6 (Moment)
Let X be a random variable, such that $\mathsf{E}\left[|X|^p\right] < \infty$. Then the *$p^{th}$ moment of X* is $\mathsf{E}[X^p]$.

The first moment of a random variable is its mean, the second moment is its variance (if $\mathsf{E}[X] = 0$), and the higher moments convey further types of "global" information about the random variable.

Definition 3.7 (Moment Generating Function)
Let X be an $\mathbb{R}$-valued random variable such that $\mathsf{E}\left[\mathrm{e}^{tX}\right] < \infty$ for some t. Then the *moment generating function (MGF) of X* is the function $M_X : \mathbb{R} \to \mathbb{R}$ given by

$$M_X(t) = \mathsf{E}\left[\mathrm{e}^{tX}\right]$$

for all t where the expectation exists.

Proposition 3.8
Let X be a random variable and assume M_X exists and is finite in some neighborhood of 0. For any positive integer p,

$$\frac{\mathrm{d}^p M_X}{\mathrm{d}t^p}(0) = \mathsf{E}\left[X^p\right].$$

Proof. Taylor expansion and Fubini's theorem gives

$$M_X(t) = \mathsf{E}\left[\mathrm{e}^{tX}\right] = \mathsf{E}\left[\sum_{n=0}^{\infty} \frac{(tX)^n}{n!}\right] = \sum_{n=0}^{\infty} \mathsf{E}\left[\frac{(tX)^n}{n!}\right] = \sum_{n=0}^{\infty} \frac{t^n}{n!} \mathsf{E}\left[X^n\right].$$

Then

$$\frac{\mathrm{d}^p M_X}{\mathrm{d}t^p}(0) = \frac{\partial^p}{\partial t^p} \sum_{n=0}^{\infty} \frac{t^n}{n!} \mathsf{E}\left[X^n\right] = \sum_{n=0}^{\infty} \mathsf{E}\left[X^n\right] \frac{\partial^p}{\partial t^p} \frac{t^n}{n!} = \sum_{n=0}^{\infty} \mathsf{E}\left[X^n\right] \mathbb{1}(n = p) = \mathsf{E}\left[X^p\right].$$

□

Theorem 3.9 (Uniqueness Theorem)
If there exists $\delta > 0$ such that

$$M_X(t) = M_Y(t) < \infty \quad \forall t \in (-\delta, \delta)$$

then $X \overset{d}{=} Y$.

Proposition 3.10

Let X and Y be independent random variables and $M_X(t)$ and $M_Y(t)$ exist for some t. Then

$$M_{X+Y}(t) = M_X(t) M_Y(t).$$

Proof. Simply,

$$M_{X+Y}(t) = \mathsf{E}\left[e^{t(X+Y)}\right] = \mathsf{E}\left[e^{tX}e^{tY}\right] = \mathsf{E}\left[e^{tX}\right]\mathsf{E}\left[e^{tY}\right] = M_X(t)M_Y(t).$$

□

3.3 Sub-Gaussian Distributions

As mentioned in the introduction, the Gaussian distribution is concentrated near its mean.

Definition 3.11

Let $\mu, \sigma \in \mathbb{R}$. The *Gaussian distribution with mean μ and variance σ^2* is denoted $\mathcal{N}\left(\mu, \sigma^2\right)$ and is defined by the following probability density function:

$$p_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

Proposition 3.12

Let $X \sim \mathcal{N}\left(\mu, \sigma^2\right)$. Then for any $t > 0$,

$$\mathsf{P}[X - \mu > t] \leq \frac{1}{\sqrt{2\pi}} \frac{e^{-\frac{t^2}{2\sigma^2}}}{t}$$
$$\mathsf{P}[X - \mu < -t] \leq \frac{1}{\sqrt{2\pi}} \frac{e^{-\frac{t^2}{2\sigma^2}}}{t}$$

and therefore

$$\mathsf{P}[|X - \mu| > t] \leq \sqrt{\frac{2}{\pi}} \frac{e^{-\frac{t^2}{2\sigma^2}}}{t}.$$

In other words, a Gaussian random variable satisfies

$$\mathsf{P}[|X - \mu| > t] \leq \exp\left(-\frac{t^2}{2\sigma^2}\right).$$

The following definition captures this concentration property. It will be convenient to express this using the logarithm of the MGF,

$$\psi_X(s) = \log(M_X(s)) = \log\left(\mathsf{E}\left[e^{sX}\right]\right).$$

Definition 3.13
Let X be a random variable such that $\mathsf{E}\left[e^{tX}\right] < \infty$ for all t. We say that X is *sub-Gaussian with variance factor* v if $\psi_X(s)$ exists for all $s \in \mathbb{R}$, and

$$\psi_X(s) \le \frac{s^2 v}{2}.$$

The bound on the tails of a sub-Gaussian distribution can be derived using the Cramér-Chernoff method as follows.

Proposition 3.14
Let X be a sub-Gaussian random variable with variance factor σ^2 and $\mathsf{E}[|X|] < \infty$. For any $t > 0$,

$$\mathsf{P}[X > t] \le \exp\left(-\frac{t^2}{2\sigma^2}\right) \quad \text{and} \quad \mathsf{P}[X < -t] \le \exp\left(-\frac{t^2}{2\sigma^2}\right).$$

Proof. For the first inequality, let $s > 0$ be some scalar to be chosen later. Then by Markov's inequality,

$$\mathsf{P}[X > t] = \mathsf{P}[sX > st] = \mathsf{P}\left[e^{sX} > e^{sT}\right] \le \frac{\mathsf{E}\left[e^{sX}\right]}{e^{\mathrm{st}}} = e^{\psi_X(s) - st} \le e^{\frac{s^2\sigma^2}{2} - st}.$$

We finish by choosing the value of s that minimizes the expression. Indeed,

$$\frac{\partial}{\partial s}\left(\frac{s^2\sigma^2}{2} - st\right) = s\sigma^2 - t = 0 \implies s = \frac{t}{\sigma^2}.$$

Plugging back into the bound yields

$$\mathsf{P}[X > t] \le \exp\left(-\frac{t^2}{2\sigma^2}\right).$$

The proof for the second inequality is similar. □

3.4 Hoeffding's Inequality

The following theorem is our main concentration of measure result for sums of independent random variables.

Theorem 3.15 (Hoeffding's Inequality)
Let $\{Z_i\}_{i \in [m]}$ be i.i.d. random variables such that $\mathsf{E}[|Z_i|] < \infty$. Assume that there exist $a, b, \mu \in \mathbb{R}$ such that for all $i \in [m]$, $\mathsf{E}[Z_i] = \mu$ and $\mathsf{P}[a \le Z_i \le b] = 1$. Then for any $\varepsilon > 0$,

$$\mathsf{P}\left[\overline{Z}_m - \mu > \varepsilon\right] \le \exp\left(-2m\left(\frac{\varepsilon}{b-a}\right)^2\right)$$
$$\mathsf{P}\left[\overline{Z}_m - \mu < -\varepsilon\right] \le \exp\left(-2m\left(\frac{\varepsilon}{b-a}\right)^2\right).$$

Therefore

$$\mathsf{P}\left[\left|\overline{Z}_m - \mu\right| > \varepsilon\right] \le 2\exp\left(-2m\left(\frac{\varepsilon}{b-a}\right)^2\right).$$

The proof of Theorem 3.15 relies on the following two lemmas.

Lemma 3.16 (Popoviciu's Inequality)
Let $a \le b$ be real numbers, let X be a random variable such that $\mathsf{E}\left[X^2\right] < \infty$, and assume that $\mathsf{P}[a \le X \le b] =$

1. Then

$$\mathsf{Var}(X) \leq \frac{(b-a)^2}{4}.$$

Proof. Consider the function $g(t) = \mathsf{E}\big[(X-t)^2\big]$. This is minimized at $t = \mathsf{E}[X]$. Let $c = \frac{1}{2}(a+b)$. Then

$$\begin{aligned} \mathsf{Var}(X) = g(\mathsf{E}[X]) \leq g(c) = \mathsf{E}\left[\left(X - \frac{a+b}{2}\right)^2\right] &= \frac{1}{4}\mathsf{E}\big[((X-a)+(X-b))^2\big] \\ &\leq \frac{1}{4}\mathsf{E}\big[((X-a)-(X-b))^2\big] = \frac{1}{4}\mathsf{E}\big[(b-a)^2\big] \\ &= \frac{(b-a)^2}{4}. \end{aligned}$$

□

Lemma 3.17 (Hoeffding's Lemma)

Let $a, b \in \mathbb{R}$, $a \leq b$, and let X be a random variable such that $\mathsf{P}[a \leq X \leq b] = 1$ and $\mathsf{E}[X] = 0$. Then X is sub-Gaussian with variance factor $\frac{(b-a)^2}{4}$. Namely,

$$\mathsf{E}\Big[\mathrm{e}^{tX}\Big] \leq \mathrm{e}^{\frac{t^2(b-a)^2}{8}}.$$

Proof. For fixed $t \in \mathbb{R}$, there exists a random variable U such that $\mathsf{E}[|U|] < \infty$, and for any $f : \Omega \to \mathbb{R}$ such that $\mathsf{E}[f(U)] < \infty$,

$$\mathsf{E}[f(U)] = \frac{\mathsf{E}\big[f(X)\mathrm{e}^{tX}\big]}{\mathsf{E}\big[\mathrm{e}^{tX}\big]}.$$

We saw this in Stat. 210A as an exponential tilting. We define U via its measure, and we define its measure via the Radon-Nikodym derivative:

$$\frac{\mathrm{d}(\mathbb{P}\circ U^{-1})}{\mathrm{d}(\mathbb{P}\circ X^{-1})}(x) = \frac{\mathrm{e}^{tx}}{\mathsf{E}\big[\mathrm{e}^{tX}\big]}$$

Namely, for $A \in \mathcal{B}_{\mathbb{R}}$,

$$\mathsf{P}[U \in A] = \mathsf{E}\Big[\mathbb{1}(U \in A)\Big] = \frac{\mathsf{E}\big[\mathrm{e}^{tX}; X \in A\big]}{\mathsf{E}\big[\mathrm{e}^{tX}\big]}.$$

Thus, for such f,

$$\mathsf{E}[f(U)] = \frac{\mathsf{E}\big[f(X)\mathrm{e}^{tX}\big]}{\mathsf{E}\big[\mathrm{e}^{tX}\big]}$$

as desired.

Second, we note that

- With $f(U) = U$ or $f(U) = U^2$,

$$\mathsf{E}[U] = \frac{\mathsf{E}\big[X\mathrm{e}^{tX}\big]}{\mathsf{E}\big[\mathrm{e}^{tX}\big]} \quad \text{and} \quad \mathsf{E}\big[U^2\big] = \frac{\mathsf{E}\big[X^2\mathrm{e}^{tX}\big]}{\mathsf{E}\big[\mathrm{e}^{tX}\big]}.$$

- With $f(U) = \mathbb{1}(a \leq U \leq b)$,

$$\mathsf{P}[a \leq U \leq b] = \mathsf{E}\Big[\mathbb{1}(a \leq U \leq b)\Big] = \frac{\mathsf{E}\big[\mathbb{1}(a \leq X \leq b)\,\mathrm{e}^{tX}\big]}{\mathsf{E}\big[\mathrm{e}^{tX}\big]} = 1.$$

Third, consider the following function ψ_X. We use Taylor's theorem to get the requisite bound. To do that we need the derivatives of ψ_X.

$$\psi'(t) = \frac{\partial}{\partial t}\log\left(\mathsf{E}\left[\mathrm{e}^{tX}\right]\right) = \frac{\frac{\partial}{\partial t}\mathsf{E}\left[\mathrm{e}^{tX}\right]}{\mathsf{E}\left[\mathrm{e}^{tX}\right]} = \frac{\mathsf{E}\left[\frac{\partial}{\partial t}\mathrm{e}^{tX}\right]}{\mathsf{E}\left[\mathrm{e}^{tX}\right]} = \frac{\mathsf{E}\left[X\mathrm{e}^{tX}\right]}{\mathsf{E}\left[\mathrm{e}^{tX}\right]}.$$

The passing in the derivative is taken care of by DCT. Similarly,

$$\begin{aligned}\psi''(t) &= \frac{\mathsf{E}\left[X^2\mathrm{e}^{tX}\right]}{\mathsf{E}\left[\mathrm{e}^{tX}\right]} - \left(\frac{\mathsf{E}\left[X\mathrm{e}^{tX}\right]}{\mathsf{E}\left[\mathrm{e}^{tX}\right]}\right)^2 \\ &= \mathsf{E}\left[U^2\right] - \mathsf{E}[U]^2 \\ &= \mathsf{Var}(U) \leq \frac{(b-a)^2}{4}.\end{aligned}$$

Lastly, note that

$$\psi(0) = \log(1) = 0 \quad \text{and} \quad \psi'(0) = \mathsf{E}[X] = 0.$$

By Taylor's theorem, there exists $\theta \in [0, t]$ such that

$$\psi(t) = \psi(0) + t\psi'(0) + \frac{t^2}{2}\psi''(\theta) \leq \frac{t^2(b-a)^2}{8}.$$

□

Proof of Theorem 3.15. It suffices to prove it for the case $\mu = 0$, because in the general case we can prove it for the shifted random variables $Z_i' = Z_i - \mu$, and this implies the result for Z_i. Let $S_m = \sum_{i\in[m]} Z_i$. For any $t > 0$,

$$\mathsf{P}\left[\overline{Z}_m > \varepsilon\right] = \mathsf{P}[S_m > \varepsilon m] = \mathsf{P}[tS_m > \varepsilon mt] = \mathsf{P}\left[\mathrm{e}^{tS_m} > \mathrm{e}^{\varepsilon mt}\right] \leq \frac{\mathsf{E}\left[\mathrm{e}^{tS_m}\right]}{\mathrm{e}^{\varepsilon mt}}$$

where the last inequality follows from Markov's inequality. Then

$$\begin{aligned}\mathsf{E}\left[\mathrm{e}^{tS_m}\right] &= \prod_{i=1}^{m}\mathsf{E}\left[\mathrm{e}^{tZ_i}\right] \leq \prod_{i=1}^{m}\mathrm{e}^{t^2(b-a)^2/8} \\ &= \mathrm{e}^{mt^2(b-a)^2/8}.\end{aligned}$$

Plugging this back, we obtain

$$\begin{aligned}\mathsf{P}\left[\overline{Z}_m > \varepsilon\right] &\leq \frac{\mathrm{e}^{mt^2(b-a)^2/8}}{\mathrm{e}^{\varepsilon mt}} \\ &= \exp\left(-\varepsilon mt + \frac{mt^2(b-a)^2}{8}\right) \\ &= \exp\left(-\frac{4\varepsilon m^2}{(b-a)^2} + \frac{2\varepsilon^2 m}{(b-a)^2}\right) \\ &= \exp\left(-\frac{2\varepsilon^2 m}{(b-a)^2}\right)\end{aligned}$$

upon using the substitution $t = \frac{4\varepsilon}{(b-a)^2}$. The proof of the other equation is similar and the last equation follows by union bound. □

4 Agnostic PAC, Uniform Convergence, and VC Dimension

4.1 Introduction

In the previous lecture we presented the PAC definition of learning. We had three goals in mind: (i) learning with finite computational and sample complexity; (ii) modeling uncertainty or noise; and (iii) using absence of evidence as evidence of absence. We saw that at least for finite hypothesis classes, the PAC definition achieves the first goal. But the PAC definition does not achieve the second goal, because it makes the strong assumption that there exists a target function f such that the labels are a deterministic function of the instance, $y = f(x)$.

Furthermore, we saw that it is necessary also to assume that $f \in \mathcal{H}$ for some restricted class of function $\mathcal{H}$ (for instance, if $\mathcal{H}$ is finite then PAC learning is possible). This is problematic because even if there exists such a class $\mathcal{H}$, the nature of unknown systems is that we often won't know what $\mathcal{H}$ is.

Our goals for this lecture and the next are the following:

- Show how to relax the assumption that the target function f satisfies $f \in \mathcal{H}$ (while still guaranteeing that learning is possible with finite sample complexities).
- Show how to relax the assumption that a target function exists, i.e., that the label y is a deterministic function of x (while still maintaining finite sample complexity).
- Continue characterizing which hypothesis classes $\mathcal{H}$ are learnable and with what complexity.

4.2 The Agnostic PAC Definition

Agnostic PAC leaning is a relaxation of PAC learning in which we are *agnostic*, in the sense that we attempt to make as few assumptions as possible about the unknown system. The main idea is that, to enable us to weaken our assumptions, we will also weaken the requirements that the output hypothesis h chosen by the learner is required to meet. Instead of aiming for h to be good in an *absolute* sense, we will only aim only for h to be good in a *competitive* (relative) sense. We present this idea in two steps.

Step 1. Removing the assumption $f \in \mathcal{H}$. Recall that a successful PAC learner should output a hypothesis h such that with high probability, h is good in the absolute sense $L_{\mathcal{D},f}(h) \leq \varepsilon$. In contrast, consider the following competitive objective: with high probability, for all $h' \in \mathcal{H}$, $L_{\mathcal{D},f}(h) \leq L_{\mathcal{D},f}(h') + \varepsilon$. Using the notation $L_{\mathcal{D},f}(\mathcal{H}) = \inf_{h' \in \mathcal{H}} L_{\mathcal{D},f}(h')$, this can also be rephrased as saying that with high probability,

$$L_{\mathcal{D},f}(h) \leq L_{\mathcal{D},f}(\mathcal{H}) + \varepsilon. \tag{4.1}$$

This means that h is almost as good as any hypothesis in $\mathcal{H}$. This is a conditional guarantee. We do not *assume* that $f \in \mathcal{H}$. Instead, we say that *if* $f \in \mathcal{H}$, then $L_{\mathcal{D},f}(h) \leq \varepsilon$; otherwise, *if* f is α-close to $\mathcal{H}$ in the sense that for all $h' \in \mathcal{H}$, $\mathsf{P}_{x\sim\mathcal{D}}[h'(x) \neq f(x)] \leq \alpha$, then $L_{\mathcal{D},f}(h) \leq \alpha + \varepsilon$; however, if f is far from $\mathcal{H}$, then nothing meaningful is guaranteed.

Step 2. Removing the assumption that labels are deterministic. Using the competitive objective, we can go further and eliminate the assumption that a target function f exists at all. Instead of saying that y is a deterministic function of x, namely $y = f(x)$ for some target function f, we can instead say that y is a random variable that depends on x. Namely, there is a conditional probability function $\mathsf{P}[y \mid x]$. Another way to say this is that there exists a probability distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ where

$$\mathcal{D}(A \times B) = \mathsf{P}[(x, y) \in A \times B] = \mathsf{P}[y \in B \in \mid x \in A]\,\mathsf{P}[x \in A]\,.$$

Notice that in the regular PAC section, y was constant depending on x. In this way, although in that section $\mathcal{D}$ was only a distribution over $\mathcal{X}$, it could have been viewed as a distribution over $\mathcal{X} \times \mathcal{Y}$.

Because we are no longer assuming that there exists a target function, we will redefine the loss function as follows.

Definition 4.1

Let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty sets, and let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$.

(a) Let $h \colon \mathcal{X} \to \mathcal{Y}$ be a hypothesis. For any function $\ell \colon \mathcal{Y}^2 \to \mathbb{R}$, the *loss function corresponding to* ℓ is

$$L_{\mathcal{D}}(h) = \mathop{\mathsf{E}}_{(x,y)\sim\mathcal{D}}[\ell(h(x), y)]\,.$$

In particular, the 0-1 *loss of h with respect to* $\mathcal{D}$ is $\mathsf{P}_{(x,y)\sim\mathcal{D}}[h(x) \neq y]$.

(b) Let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$. The *loss of* $\mathcal{H}$ *with respect to* $\mathcal{D}$ *is*

$$L_{\mathcal{D}}(\mathcal{H}) = \inf\{L_{\mathcal{D}}(h) \colon h \in \mathcal{H}\}\,.$$

This quantity is also called the *approximation error of* $\mathcal{H}$ *with respect to* $\mathcal{D}$.

Remark 4.2. Formally, we require ℓ to be integrable; however, outside of Chapter 3 and any heavy probability theory, we will generally make the assumption that our random variables are in L^p for any p we would like.

Using this definition we can define the competitive objective as $L_{\mathcal{D}}(h) \leq L_D(\mathcal{H}) + \varepsilon$, and this makes sense even if we do not assume that a target function exists (note that this differs from Equation 4.1). We only require that *if* there is a hypothesis $h' \in \mathcal{H}$ that has loss α, *then* the learner's output will satisfy $L_{\mathcal{D}}(h) \leq \alpha + \varepsilon$.

These steps yield the notion of *Agnostic PAC Learning*, which is the main definition of learning in this course.

Definition 4.3 (Agnostic PAC Learning)

Let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty sets, let $\mathcal{F}$ be the set of all functions $\mathcal{X} \to \mathcal{Y}$, and let $\mathcal{H} \subseteq \mathcal{F}$ be a class of functions. We say that a (possibly randomized) algorithm A is an *agnostic PAC learner for* $\mathcal{H}$ if there exists a sample complexity function $m \colon (0, 1)^2 \to \mathbb{N}$ such that for every precision parameter $\varepsilon \in (0, 1)$, every confidence parameter $\delta \in (0, 1)$, and every distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, if A receives as input the parameters ε and δ and a sample S of size $s = m(\varepsilon, \delta)$ such that $S = \{(x_i, y_i)\}_{i\in[m]} \overset{\text{i.i.d.}}{\sim} \mathcal{D}$, then A halts and outputs a hypothesis $h \in \mathcal{F}$ such that

$$\mathop{\mathsf{P}}_{S}[L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon] \geq 1 - \delta.$$

We say that a class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ is *agnostic PAC learnable* if there exists an algorithm that is an agnostic PAC learner for $\mathcal{H}$.

Remark 4.4.

- Every agnostic PAC learner for $\mathcal{H}$ is in particular also a PAC learner for $\mathcal{H}$. Hence, the notion of agnostic PAC is stronger (harder to satisfy) than regular PAC. This is despite the fact that the competitive objective in agnostic PAC is weaker (easier to satisfy) than the absolute objective in PAC (namely, it is possible that h does not satisfy $L_{\mathcal{D}}(h) \leq \varepsilon$ but does satisfy $L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon$).

- The “agnostic” setting is actually not completely agnostic, because it still makes two non-trivial assumptions: that there exists a distribution that generates the sample (the sample is not arbitrary), and that the loss to be minimized is also measured according to the same distribution that generates the sample. These assumptions are very minimal and natural. Nonetheless, later on in the course we will see the setting of *online learning*, which employs the same idea of a competitive objective to discard even these minimal assumptions.

4.2.1 Learning Scenarios

PAC learning and agnostic PAC will serve as our main definitions of learning for most of the course. Now that we have fully introduced them, it makes sense to reflect what real-world scenarios are captured by these definitions. Following are three such types of real-world scenarios.

1. Binary classification. The label space $\mathcal{Y}$ has cardinality 2, for instance, $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{1, -1\}$, etc. Examples include:
   - Classifying an email as spam or ham (not spam).
   - Deciding whether a photograph depicts a cat.
   - Given input to a fingerprint reader, decide whether to unlock a device or not (whether the fingerprint belongs to the owner of the device or not).
   - Given data from a medical examination (e.g., an MRI scan), diagnose whether a patient has a specific medical condition or not (e.g., cancer).
   - In a vehicle, decide whether or not to activate the emergency brake assist (EBA) system.
   - Given a text message, decide whether it is intended humorously or literally.

   The 0-1 loss is a natural loss function for most cases of binary classification.

2. Multi-class classification. The label space $\mathcal{Y}$ is discrete and has more than two elements. Examples include:
   - Given a text, identify what language it is written in.
   - Given a photograph of an object, identify the object (“chair”, “cat”, etc.).
   - Given the current state in a game of chess, output a good next move.
   - Structured learning tasks, in which the required label has a nontrivial internal structure. For example, given a sentence in English, produce a syntactical parse tree of the sentence.
   - Ranking: given a set of objects, order the objects according to some criteria. For instance, a search engine is given a query and a set of documents, and is required to order the documents according to their relevance to the query. The label space is the set of possible permutations of the documents.

   The 0-1 loss can be a natural loss function for the first two examples above, but the remaining examples might be better served by a richer loss function that distinguishes between different levels of correctness.

3. Regression. The label space $\mathcal{Y}$ is $\mathbb{R}$, or some other large metric space, and the objective is to output a label that is close to the correct label with respect to the metric. Examples include:
   - Given sensor input from a self-driving car, try to identify the distance to an object ahead (e.g., a pedestrian).
   - Given current weather conditions, predict the temperature tomorrow.
   - Given recent sport statistics, predict the probability that a specific sports team will win an upcoming match.

There are a variety of natural loss functions for regression, including (for real-valued labels) square error loss $\ell(y, y') = (y - y')^2$, absolute error loss $\ell(y, y') = |y - y'|$ hinge loss, cross entropy loss, etc.

All of the above are instances of *supervised learning*, meaning that the learner first receives a training set with labeled instances, and then is required to predict the labels of previously unseen instances. There are also many important learning scenarios that are not instances of supervised learning and are not covered by the PAC and agnostic PAC models. These include reinforcement learning, online learning, transductive learning, semi-supervised learning, active learning, and learning with queries, as well as unsupervised learning scenarios such as clustering and dimensionality reduction. We will touch on some of these settings in later stages of the course.

4.3 Error Decomposition: The Bias-Complexity Tradeoff

The Bayes optimal error is the best (minimal) loss that can be achieved when predicting labels for a specific distribution, and it is determined by how noisy the labels are. Formally:

Definition 4.5
Let $\mathcal{X}$ and $\mathcal{Y}$ be sets, assume $|\mathcal{Y}| < \infty$, let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$, and let $L_{\mathcal{D}}$ denote the 0-1 loss.

1. The *Bayes optimal error for $\mathcal{D}$* is
$$L_{\mathcal{D}}^* = \inf\left\{L_{\mathcal{D}}(f) \colon f \in \mathcal{Y}^{\mathcal{X}}\right\}.$$
2. The *Bayes optimal classifier for $\mathcal{D}$* is
$$f^*(x') = \operatorname*{argmax}_{y' \in \mathcal{Y}} \mathop{\mathsf{P}}_{(x,y)\sim\mathcal{D}}\left[y = y' \,\middle|\, x = x'\right].$$
3. The *noise of $\mathcal{D}$ at instance $x' \in \mathcal{X}$* is
$$\operatorname{noise}(x') = 1 - \max_{y' \in \mathcal{Y}} \mathop{\mathsf{P}}_{(x,y)\sim\mathcal{D}}\left[y = y' \,\middle|\, x = x'\right].$$
4. The *noise of $\mathcal{D}$* is
$$\operatorname{noise}(\mathcal{D}) = \mathop{\mathsf{E}}_{(x,y)\sim\mathcal{D}}[\operatorname{noise}(x)].$$

Fact 4.6
$\operatorname{noise}(\mathcal{D}) = L_{\mathcal{D}}^* = L_{\mathcal{D}}(f^*)$.

Assume an agnostic PAC learning algorithm outputs hypotheses from a class $\mathcal{H}$ of functions $\mathcal{X} \to \mathcal{Y}$. Let $h \in \mathcal{H}$, and let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$. Then $L_{\mathcal{D}}(h) - L_{\mathcal{D}}^*$ is a measure of of how well h compares to the best possible function f^* (which will typically not be a member of $\mathcal{H}$). We can always decompose this loss as follows.

$$L_{\mathcal{D}}(h) - L_{\mathcal{D}}^* = \underbrace{L_{\mathcal{D}}(h) - L_{\mathcal{D}}(\mathcal{H})}_{\text{estimation error}} + \underbrace{L_{\mathcal{D}}(\mathcal{H}) - L_{\mathcal{D}}^*}_{\text{approximation error}}$$

There is a tradeoff between these two terms. Assume we make the class $\mathcal{H}$ larger and more complex, so that the algorithm has more output functions to choose from. Then

- The approximation error will typically become smaller, because $\mathcal{H}$ will gain functions that are closer to f^*. However,

- The estimation error will become larger, because finding a good hypothesis in $\mathcal{H}$ is harder when $\mathcal{H}$ is larger.

Hence, when designing a learning algorithm, we need to carefully consider which class of functions the algorithm should use. This decision will typically also be informed by any prior knowledge available about the learning problem.

4.4 Agnostic PAC Learning via Uniform Convergence

We develop the notion of *uniform convergence*, using the following lemma as an example. It generalizes the result for finite classes to the agnostic setting.

Lemma 4.7
Let $\mathcal{X}$ and $\mathcal{Y}$ be (finite or infinite) nonempty sets, and let $\mathcal{H}$ be a finite subset of $\mathcal{Y}^{\mathcal{X}}$. Then $\mathcal{H}$ is agnostic PAC learnable with sample complexity $m(\varepsilon, \delta) = \left\lceil \frac{2\log(2\left|\mathcal{H}\right|/\delta)}{\varepsilon^2} \right\rceil$

Remark 4.8. Note that the dependence on $\frac{1}{\varepsilon}$ is quadratic, unlike in the PAC case.

We previously saw a proof that finite classes are PAC learnable. That outline of the proof was:

1. For any single hypothesis $h \in \mathcal{H}$, we can efficiently estimate whether h is ε-good or not using a few samples. (Taking $\frac{\log(1/\delta)}{\varepsilon}$ samples suffices to check whether $L_{\mathcal{D}}(h) \leq \varepsilon$ with confidence $1 - \delta$.)
2. Because $\mathcal{H}$ is finite, we can use a union bound in order to estimate this for all $h \in \mathcal{H}$ simultaneously. (Taking $\frac{\log(\left|\mathcal{H}\right|/\delta)}{\varepsilon}$ samples suffices to check this for all $h \in \mathcal{H}$).

To prove Lemma 4.7, we will follow the same outline, using a notion of *uniform convergence*. Recall that in calculus, their is a notion of uniform convergence for sequences of functions, which contrasts with pointwise convergence:

Definition 4.9
Let Ω be a set, let $\{f_n\}_{n\in\mathbb{N}}$ be an infinite sequence of functions $\Omega \to \mathbb{R}$, and let $f^* \colon \Omega \to \mathbb{R}$ be a function.

1. We say that $\{f_n\}_{n\in\mathbb{N}}$ *converges pointwise to* f^* if for every $\varepsilon > 0$, for every $x \in \Omega$, there exists $N \in \mathbb{N}$, such that for every $n \geq N$,
$$\left|f_n(x) - f^*(x)\right| \leq \varepsilon.$$
2. We say that $\{f_n\}_{n\in\mathbb{N}}$ *converges uniformly to* f^* if for every $\varepsilon > 0$, there exists $N \in \mathbb{N}$, such that for every $x \in \Omega$, for every $n \geq N$,
$$\left|f_n(x) - f^*(x)\right| \leq \varepsilon.$$

Similarly, we introduce the following definition which concerns the convergence of the empirical loss functions L_{S_m} to the population loss $L_{\mathcal{D}}$, where L_{S_m} is the empirical loss for a sample of size m.

Definition 4.10
Let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$. We say that $\mathcal{H}$ *satisfies the uniform convergence property* if for every $\varepsilon, \delta \in (0, 1)$, there exists $M \in \mathbb{N}$, such that for every distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, for every $h \in \mathcal{H}$, and for every $m \geq M$, if $S \sim \mathcal{D}^m$,
$$\mathop{\mathsf{P}}_{S}[\left|L_S(h) - L_{\mathcal{D}}(h)\right| > \varepsilon] \leq \delta.$$

Let $m_{\mathscr{H}}^{\mathrm{UC}} \colon (0,1)^2 \to \mathbb{N}$ be a function. We say that $\mathscr{H}$ *satisfies uniform convergence with sample complexity* $m_{\mathscr{H}}^{UC}$ if taking $M = m_{\mathscr{H}}^{\mathrm{UC}}(\varepsilon, \delta)$ satisfies the uniform convergence property.

The following lemma states that uniform convergence is sufficient for agnostic learnability.

Lemma 4.11

Let $\mathscr{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$. If $\mathscr{H}$ satisfies the uniform convergence property with sample complexity $m_{\mathscr{H}}^{\mathrm{UC}}$, then $\mathrm{ERM}_{\mathscr{H}}$ is an agnostic PAC learner for $\mathscr{H}$ with sample complexity $m(\varepsilon, \delta) = m_{\mathscr{H}}^{\mathrm{UC}}\left(\frac{\varepsilon}{2}, \delta\right)$.

Proof. Fix a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, fix $\varepsilon, \delta \in (0,1)$, and let $h = \mathrm{ERM}_{cH}(S)$ where $S \sim \mathcal{D}^m$ and $m = m_{\mathscr{H}}^{\mathrm{UC}}(\frac{\varepsilon}{2}, \delta)$. From uniform convergence, with probability at least $1 - \delta$,

$$\left|L_S(h') - L_{\mathcal{D}}(h')\right| \leq \frac{\varepsilon}{2} \quad \text{for all } h' \in \mathscr{H}. \tag{4.2}$$

In this case, for every $h' \in \mathscr{H}$,

$$\begin{aligned} L_{\mathcal{D}}(h) &\leq L_S(h) + \frac{\varepsilon}{2} && \text{(from uniform convergence Equation 4.2)} \\ &\leq L_S(h^*) + \frac{\varepsilon}{2} && (h = \mathrm{ERM}_{cH}(S)) \\ &\leq L_{\mathcal{D}}(h^*) + \frac{\varepsilon}{2} + \frac{\varepsilon}{2} && \text{(from uniform convergence Equation 4.2)} \\ &= L_{\mathcal{D}}(h^*) + \varepsilon. \end{aligned}$$

□

To prove Lemma 4.7, we will show that finite classes satisfy uniform convergence.

Proposition 4.12

Let $\mathscr{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$. If $\mathscr{H}$ is finite, then $\mathscr{H}$ satisfies uniform convergence with sample complexity $m_{\mathscr{H}}^{\mathrm{UC}}(\varepsilon, \delta) = \left\lceil \frac{\log(2\left|\mathscr{H}\right|/\delta)}{2\varepsilon^2} \right\rceil$.

Proof. Let $h \in \mathscr{H}$. Fix a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ and parameters $\varepsilon, \delta \in (0,1)$. Let $S \sim \mathcal{D}^m$ for $m = m_{\mathscr{H}}^{\mathrm{UC}}(\varepsilon, \delta)$ as in the statement. Fix $h \in \mathscr{H}$. To complete the proof it suffices to show that $\mathsf{P}_S[|L_S(h) - L_{\mathcal{D}}(h)| > \varepsilon] \leq \frac{\delta}{|\mathscr{H}|}$, and then the result follows from a union bound over all $h \in \mathscr{H}$.

Let $S = \{(x_i, y_i)\}_{i \in [m]}$. Note that $\{\mathbb{1}(h(x_i) \neq y_i)\}_{i \in [m]}$ are i.i.d. random variables with support in $\{0,1\}$ and expectation $L_{\mathcal{D}}(h)$. Hence, Hoeffding's inequality yields

$$\begin{aligned} \mathop{\mathsf{P}}_S[|L_S(h) - L_{\mathcal{D}}(h)| > \varepsilon] &= \mathop{\mathsf{P}}_S\left[\left|\frac{1}{m}\sum_{i \in [m]} \mathbb{1}(h(x_i) \neq y_i) - L_{\mathcal{D}}(h)\right| > \varepsilon\right] \\ &\leq 2\exp\bigl(2m\varepsilon^2\bigr) \\ &\leq \frac{\delta}{|\mathscr{H}|}. \end{aligned}$$

□

As a corollary, we obtain Lemma 4.7.

Proof of Lemma 4.7. From Proposition 4.12, $\mathcal{H}$ satisfies uniform convergence with sample complexity $m^{\mathrm{UC}}_{\mathcal{H}}(\varepsilon, \delta) = \left\lceil \frac{\log(2|\mathcal{H}|/\delta)}{2\varepsilon^2} \right\rceil$. From Lemma 4.11, ERM_{cH} is an agnostic PAC learner for $\mathcal{H}$ with sample complexity $m_{\mathcal{H}}(\varepsilon, \delta) = \left\lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\varepsilon^2} \right\rceil$. □

4.5 The VC Dimension: Uniform Convergence for Infinite Classes

So far, we have seen that the set of all functions over an infinite domain is not PAC learnable (and therefore not agnostic PAC learnable), while finite classes of functions are agnostic PAC learnable (and therefore also PAC learnable). But what about classes in between these two extremes?

First, we observe that some infinite classes over infinite domains are agnostic PAC learnable, and therefore being finite is a sufficient condition for learnability, but is not a necessary condition.

Lemma 4.13

Let

$$\mathcal{T} = \left\{ f_t \in \{0, 1\}^{\mathbb{R}} : f_t(x) = \mathbb{1}(x \geq t) \right\}$$

be the set of threshold functions (boolean monotone increasing functions over the domain $\mathbb{R}$). Then $\mathrm{ERM}_{\mathcal{T}}$ is a PAC learner for $\mathcal{T}$ with sample complexity $m(\varepsilon, \delta) = \left\lceil \frac{\log(2/\delta)}{\varepsilon} \right\rceil$.

Proof. Fix $\varepsilon, \delta \in (0, 1)$, a distribution $\mathcal{D}$ over $\mathbb{R}$, and a target function $f_{t^*} \in \mathcal{T}$. Let $S \sim (\mathcal{D}, f_{t^*})^m$ and $h = \mathrm{ERM}_{\mathcal{T}}(S)$. We need to show that $\mathsf{P}_S[L_{\mathcal{D}}(h) > \varepsilon] < \delta$.

Let $t_h \in \mathbb{R}$ be such that $h = f_{t_h}$. Let B be the set of points that are misclassified by h. Namely,

$$B = \begin{cases} (\min\{t_h, t^*\}, \max\{t_h, t^*\}] & t_h \neq t^* \\ \varnothing & t_h = t^* \end{cases}.$$

Note that $L_{\mathcal{D}}(h) = \mathcal{D}(B)$, so

$$\begin{aligned} \mathop{\mathsf{P}}_{S}[L_{\mathcal{D}}(h) > \varepsilon] &= \mathop{\mathsf{P}}_{S}[\mathcal{D}(B) > \varepsilon] \\ &= \mathop{\mathsf{P}}_{S}\big[t_h < t^*, D(B) > \varepsilon\big] + \mathop{\mathsf{P}}_{S}\big[t_h > t^*, D(B) > \varepsilon\big] \end{aligned} \tag{4.3}$$

Hence, it suffices to show that each of the two summands in Equation 4.3 is bounded by $\frac{\delta}{2}$.

We prove that $\mathsf{P}_S[t_h < t^*, \mathcal{D}(B) > \varepsilon] \leq \frac{\delta}{2}$. Note that if $F_{\mathcal{D}}(t^*) = \mathcal{D}((-\infty, t^*]) \leq \varepsilon$, then we are done, because $\mathcal{D}(B) \leq \mathcal{D}((-\infty, t^*]) = F_{\mathcal{D}}(t^*) < \varepsilon$. Otherwise, $F_{\mathcal{D}}(t^*) > \varepsilon$. Since $\lim_{t \to -\infty} F_{\mathcal{D}}(t) = 0$ and $F_{\mathcal{D}}(t)$ is right-continuous, by the intermediate value theorem there exists $t_\varepsilon \in (-\infty, t^*)$ such that $F_{\mathcal{D}}(t_\varepsilon) = F_{\mathcal{D}}(t^*) - \varepsilon$. Namely, the interval $I_\varepsilon = (t_\varepsilon, t^*)$ satisfies $\mathcal{D}(I_\varepsilon) = F_{\mathcal{D}}(t^*) - F_{\mathcal{D}}(t_\varepsilon) = \varepsilon$.

Let $S = \{(x_i, y_i)\}_{i \in [m]}$, and observe that if $t_h < t^*$ and there exists $i \in [m]$ such that $x_i \in I_\varepsilon$, then $t_\varepsilon < x_i < t_h < t^*$, and this implies $\mathcal{D}(B) = \mathcal{D}((t_h, t^*]) \leq \mathcal{D}(I_\varepsilon) = \varepsilon$. Hence,

$$\begin{aligned} \mathop{\mathsf{P}}_{S}\big[t_h < t^*, \mathcal{D}(B) > \varepsilon\big] &\leq \mathop{\mathsf{P}}_{S}\left[\bigcap_{i \in [m]} \{x_i \notin I_\varepsilon\}\right] \\ &= \prod_{i \in [m]} \mathop{\mathsf{P}}_{x_i \sim \mathcal{D}}[x_i \notin I_\varepsilon] && (x_i\text{'s are independent}) \\ &= (1 - \varepsilon)^m \leq \mathrm{e}^{-\varepsilon m} \leq \frac{\delta}{2}. \end{aligned}$$

This proves the bound for the left summand in Equation 4.3. The proof for the right summand is analogous. □

Towards developing a condition that precisely characterizes which classes are PAC or agnostic PAC learnable, we start with a slightly more general version of Corollary 2.11, which stated that the class of all functions from an infinite domain is not learnable.

This followed from Corollary 2.10, which implied that PAC learning the class of all functions from a set $\mathcal{X}$ of cardinality n to $\{0, 1\}$ with parameters $\varepsilon, \delta < \frac{1}{8}$ is not possible with less than $\frac{n}{2}$ samples. This implied Corollary 2.11 because for any $n \in \mathbb{N}$, the set of all functions from an infinite domain contains within it a set of all functions from a subset of the domain of cardinality n, and so PAC learning requires $\frac{n}{2}$ samples for all $n \in \mathbb{N}$, namely a non-finite number of samples. We can therefore reformulate Corollary 2.10 and Corollary 2.11 in the following, more general way.

Definition 4.14

Let $\mathcal{X}$ be a nonempty set, $\mathcal{Y} = \{0, 1\}$, $\mathcal{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$, $f \in \mathcal{H}$, and let $X \subseteq \mathcal{X}$.

1. The *restriction of the function* f *to the subset* X is the function $f|_X : X \to \mathcal{Y}$ such that for all $x \in X$, $f|_X(x) = f(x)$.
2. The *restriction of the class* $\mathcal{H}$ *to the subset* X, denoted $\mathcal{H}|_X$, is the set of functions $\{f|_X : f \in \mathcal{H}\}$.
3. We say that $\mathcal{H}$ *shatters* X if the restriction of $\mathcal{H}$ to X is the set of all functions $X \to \mathcal{Y}$, namely, if $|\mathcal{H}|_X| = 2^{|X|}$.

Proposition 4.15

Let $\mathcal{X}$ be a set, let $\mathcal{Y} = \{0, 1\}$, and let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$.

1. Assume $\mathcal{H}$ shatters some subset $X \subseteq \mathcal{X}$ such that $|X| = n$. Then any algorithm that PAC learns $\mathcal{H}$ with parameters $\varepsilon, \delta < \frac{1}{8}$ must use a sample of size at least $\frac{n}{2}$.
2. Assume that for any $n \in \mathbb{N}$, there exists $X \subseteq \mathcal{X}$ such that $|X| = n$ and $\mathcal{H}$ shatters X. Then $\mathcal{H}$ is not PAC learnable.

Proof. For (1), assume that A is an algorithm that PAC learns $\mathcal{H}$ with parameters $\varepsilon, \delta < \frac{1}{8}$. Then in particular A PAC learns $\mathcal{H}$ with parameters $\varepsilon, \delta < \frac{1}{8}$ for any unknown distribution $\mathcal{D}$ such that $\mathcal{D}(X) = 1$. This is equivalent to A PAC learning the class $\mathcal{H}|_X$ of functions $X \to \mathcal{Y}$ with parameters $\varepsilon, \delta < \frac{1}{8}$. Because $\mathcal{H}$ shatters X, $\mathcal{H}|_X$ is the set of all functions $X \to \mathcal{Y}$, and so from Corollary 2.10, A must use a sample of size at least $\frac{n}{2}$.

For (2), it follows from (1) as follows. Assume for contradiction that there exists an algorithm A that PAC learns $\mathcal{H}$ with parameters $\varepsilon, \delta < \frac{1}{8}$ and uses m samples. From the assumption, there exists a set $X \subseteq \mathcal{X}$ such that $\mathcal{H}$ shatters X and $|X| > 2m$. Then from (1), A uses strictly more than m samples, a contradiction. □

This motivates the following definition.

Definition 4.16

Let $\mathcal{X}$ be a nonempty set, $\mathcal{Y} = \{0, 1\}$, and let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$. The *Vapnik-Charvonenkis (VC) dimension of* $\mathcal{H}$ is

$$\mathsf{VC}(\mathcal{H}) = \sup\{|X| : X \subseteq \mathcal{X}, \mathcal{H} \text{ shatters } X\}.$$

Namely, $\mathsf{VC}(\mathcal{H})$ is the largest number $n \in \mathbb{N}$ such that $\mathcal{H}$ shatters a set of cardinality n, and if $\mathcal{H}$ shatters sets of unbounded size then $\mathsf{VC}(\mathcal{H}) = \infty$.

Example 4.17

Let $\mathcal{T} = \left\{f_t \in \{0,1\}^{\mathbb{R}} : f_t(x) = \mathbb{1}(x \geq t)\right\}$ be the class of thresholds (as in Lemma 4.13). Then $\mathsf{VC}(\mathcal{T}) = 1$.

To see this, we need to show two things:

- There exists a set $A \subseteq \mathbb{R}$ such that $|A| = 1$ and $\mathcal{T}$ shatters A. Indeed, let $A = \{0\}$, and note that $|\mathcal{T}|_A| = 2 = 2^{|A|}$ because $f_{-1}(0) = 1$ and $f_1(0) = 0$. Hence $\mathcal{T}$ shatters A.

- For any set $A \subseteq \mathbb{R}$ such that $|A| > 1$, $\mathcal{T}$ does not shatter A. It suffices to show that no set of size 2 is shattered, because if a set of size $n > 2$ is shattered then in particular every subset of size 2 of that set is also shattered. Fix a set $A = \{x, y\}$ with $x < y$. There does not exist $f \in \mathcal{T}$ such that $f(x) = 1$ and also $f(y) = 0$. Hence, A is not shattered.

Example 4.18

An axis-aligned rectangle in $\mathbb{R}^2$ is a function $f_{a,b} \colon \mathbb{R}^2 \to \{0,1\}$ with $a = (a_1, a_2) \in \mathbb{R}^2$ and $b = (b_1, b_2) \in \mathbb{R}^2$ such that

$$f_{a,b}(x, y) = \mathbb{1}(a_1 \leq x \leq b_1, a_2 \leq y \leq b_2)\,.$$

Let $\mathcal{R}$ be the class of all axis-aligned rectangles, $\mathcal{R} = \left\{f_{a,b} \colon a, b \in \mathbb{R}^2\right\}$. Then $\mathsf{VC}(\mathcal{R}) = 4$.

To see this, we need to show two things:

- There exists a set $A \subseteq \mathbb{R}$ such that $|A| = 4$ and $\mathcal{R}$ shatters A. Indeed, the set $A = \{(1,0), (-1,0), (0,1), (0,-1)\}$ is shattered.

- For any set $A \subseteq \mathbb{R}$ such that $|A| > 4$, $\mathcal{R}$ does not shatter A. Fix a set A with $|A| \geq 5$. We claim that A is not shattered. Let ℓ, r, t, b denote the leftmost, rightmost, topmost, and bottommost members of A (if more than one members of A are equally extreme in some direction, choose one of them arbitrarily). Let $c \in A \setminus \{\ell, r, t, b\}$ (c exists because $|A| \geq 5$). To see that A is not shattered it suffices to observe that for any rectangle $f_{z,z'}$, if we assume that $f_{z,z'}(\ell) = f_{z,z'}(r) = f_{z,z'}(t) = f_{z,z'}(b) = 1$, then $f_{z,z'}(c) = 1$. This observation is true because the assumption implies $z_1 \leq \ell_1 \leq c_1 \leq r_1 \leq z'_1$, and similarly $z_2 \leq c_2 \leq z'_2$, and therefore $f_{z,z'}(c) = 1$.

Example 4.19

Let $\mathcal{H}$ be a class of functions with $|\mathcal{H}| < \infty$. Then $\mathsf{VC}(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$. Indeed, let A be a shattered subset of the domain. Then

$$2^{|A|} = |\mathcal{H}|_{\mathcal{A}}| \leq |\mathcal{H}| \implies |A| \leq \log_2(|\mathcal{H}|)\,.$$

So $\mathcal{H}$ can shatter sets of cardinality at most $\log_2(|\mathcal{H}|)$.

The above inequality is tight when $\mathcal{H}$ is the set of all boolean functions over a finite domain. Notice that $\mathsf{VC}(\mathcal{H})$ can be much smaller than $\log_2(|\mathcal{H}|)$. For instance, let $\mathcal{H} \subseteq \mathcal{T}$ where $\mathcal{T}$ is as in Example 4.17 and $|\mathcal{H}| < \infty$. Then $\mathsf{VC}(\mathcal{H}) = 1$.

4.6 Discussion

In this unit, we have introduced the agnostic PAC models, and the concepts of uniform convergence and VC dimension. In the next unit we will see how to combine these concepts to prove the fundamental theorem of PAC learning, which states that the VC dimension completely characterizes the learnability of classes in the PAC and agnostic PAC models.

5 The Fundamental Theorem of PAC Learning

5.1 Introduction

In Proposition 4.15 we state a lower bound of $\Omega(\mathsf{VC}(\mathcal{H}))$ on the sample complexity of PAC learning $\mathcal{H}$, and Lemma 4.11 states an upper bound on the sample complexity of agnostic PAC learning that follows from uniform convergence. Our main result on characterizing learnability and sample complexity will tie together the notions of uniform convergence and VC dimension, showing that these two bounds are tight: every class of finite VC dimension satisfies uniform convergence with sample complexity $O(\mathsf{VC}(\mathcal{H}))$. Hence, a class is PAC and agnostic PAC learnable if and only if it has finite VC dimension, and if the VC dimension is finite then the sample complexity of learning is $\Theta(\mathsf{VC}(\mathcal{H}))$.

5.2 The Growth Function and Sauer's Lemma

The growth function measures how rich a class $\mathcal{H}$ is on finite subsets of the domain.

Definition 5.1
Let $\mathcal{X}$ and $\mathcal{Y}$ be sets, and let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$. The growth function of $\mathcal{H}$ is a function $\tau_{\mathcal{H}} : \mathbb{N} \to \mathbb{N}$ given by

$$\tau_{\mathcal{H}}(m) = \sup \{ |\mathcal{H}|_A| : A \subseteq \mathcal{X}, |A| = m \} .$$

Notice that the growth function and the VC dimension are related by

$$\mathsf{VC}(\mathcal{H}) = \sup \{ m \in \mathbb{N} \colon \tau_{\mathcal{H}}(m) = 2^m \} .$$

The following combinatorial lemma describes how the growth function behaves more generally, and its relation to the VC dimension. Conceptually, it has two phases: below the VC dimension, $\tau_{\mathcal{H}}$ grows exponentially; above the VC dimension, $\tau_{\mathcal{H}}$ grows at most polynomially.

Lemma 5.2 (Sauer)
Let $\mathcal{X}$ be a set and let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \{0,1\}$ such that $\mathsf{VC}(\mathcal{H}) = d < \infty$. Then for natural numbers m,

$$\begin{aligned} \tau_{\mathcal{H}}(m) &= 2^m, \quad \text{for all } m \leq d, \\ \tau_{\mathcal{H}}(m) &\leq \sum_{i=0}^{d} \binom{m}{i} \leq \left(\frac{em}{d}\right)^d, \quad \text{for all } m > d \end{aligned} \tag{5.1}$$

The lemma is a consequence of the following claim.

Proposition 5.3
Let $\mathcal{X}$ be a set and let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \{0,1\}$ such that $\mathsf{VC}(\mathcal{H}) = d < \infty$. Then for any $A \subseteq \mathcal{X}$,

$$\left| \left\{ B \subseteq A \,\middle|\, \exists h \in H \colon B = A \cap h^{-1}(\{1\}) \right\} \right| \leq |\{ B \subseteq A \mid \mathcal{H} \text{ shatters } B \}| .$$

Proof of Lemma 5.2. If $m \leq d$ then $\tau_{\mathcal{H}}(m) = 2^m$ by the definition of $\tau_{\mathcal{H}}$ and the VC dimension.

For the case $m > d$, fix $m \in \mathbb{N}$ such that $d < m \leq |\mathcal{H}|$. For any $A \subseteq \mathcal{X}$ such that $|A| = m$,

$$\begin{aligned}
|\mathcal{H}|_A| &= \left|\left\{B \subseteq A \,\middle|\, \exists h \in \mathcal{H} \colon B = A \cap h^{-1}(1)\right\}\right| \\
&\leq |\{B \subseteq A \mid \mathcal{H} \text{ shatters } B\}| && \text{(Proposition 5.3)} \\
&= |\{B \subseteq A \mid \mathcal{H} \text{ shatters } B, |B| \leq d\}| \\
&\leq |\{B \subseteq A \mid |B| \leq d\}| \\
&= \sum_{i=0}^{d} \binom{m}{i}.
\end{aligned}$$

Hence,

$$\tau_{\mathcal{H}}(m) = \max_{A \subseteq \mathcal{X}, |A| = m} |\mathcal{H}|_A| \leq \sum_{i=0}^{d} \binom{m}{i}.$$

This completes the proof of the first inequality in Equation 5.1.

For the second inequality in Equation 5.1,

$$\begin{aligned}
\sum_{i=0}^{d} \binom{m}{i} &\leq \sum_{i=0}^{m} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i} \\
&= \left(\frac{m}{d}\right)^{d} \sum_{i=0}^{m} \binom{m}{i} \left(\frac{d}{m}\right)^{i} \\
&= \left(\frac{m}{d}\right)^{d} \left(1 + \frac{d}{m}\right)^{m} \\
&\leq \left(\frac{m}{d}\right)^{d} \mathrm{e}^{d} = \left(\frac{\mathrm{e}m}{d}\right)^{d}.
\end{aligned}$$

□

Proof of Proposition 5.3. We proceed by induction on $n = |A|$. For the base case $n = 0$,

$$\left\{B \subseteq A \,\middle|\, \exists h \in \mathcal{H} \colon B = A \cap h^{-1}(\{1\})\right\} = \{\varnothing\} = \{B \subseteq A \mid \mathcal{H} \text{ shatters } B\}.$$

For the induction step, we assume the claim holds for all sets $A \subseteq \mathcal{X}$ of cardinality n, and prove that it holds for all sets of cardinality $n+1$. Fix a set $A \subseteq \mathcal{X}$ with $|A| = n+1$, fix $a \in A$, and let $A' = A \setminus \{a\}$. Define

$$\begin{aligned}
H_1' &= \left\{g \colon A' \to \{0,1\} \,\middle|\, \exists h \in \mathcal{H} \colon g = h|_{A'}\right\} = \mathcal{H}|_{A'} \\
H_2' &= \left\{g \colon A' \to \{0,1\} \,\middle|\, \exists h_1, h_2 \in \mathcal{H} \colon g = h_1|_{A'} = h_2|_{A'}, h_1(a) \neq h_2(a)\right\} \subseteq \mathcal{H}|_{A'}, \\
H_2 &= \{g \colon A \to \{0,1\} \mid \exists h_1, h_2 \in \mathcal{H} \colon g|_{A'} = h_1|_{A'} = h_2|_{A'}, h_1(a) \neq h_2(a)\} \subseteq \mathcal{H}|_A.
\end{aligned}$$

Namely, $H_2' = H_2|_{A'}$.

Observe that

$$\begin{aligned}
|\mathcal{H}|_A| &= |\{g \colon A \to \{0,1\} \mid \exists h \in \mathcal{H} \colon g = h|_A\}| \\
&= |\{g \colon A \to \{0,1\} \mid \exists h \in \mathcal{H} \colon g = h|_{A'}, h(a) = 0\}| \\
&\quad + |\{g \colon A \to \{0,1\} \mid \exists h \in \mathcal{H} \colon g = h|_{A'}, h(a) = 1\}| \\
&= \left|\left\{g \colon A' \to \{0,1\} \,\middle|\, \exists h \in \mathcal{H} \colon g = h|_{A'}\right\}\right| + \\
&\quad + \left|\left\{g \colon A' \to \{0,1\} \,\middle|\, \exists h_1, h_2 \in \mathcal{H} \colon g = h_1|_{A'} = h_2|_{A'}, h_1(a) = 0, h_2(a) = 1\right\}\right| && (5.2) \\
&= |H_1'| + |H_2'|. && (5.3)
\end{aligned}$$

Equation 5.2 holds because we can count the restrictions of $\mathscr{H}$ to A by counting the number of restrictions of $\mathscr{H}$ to A', where we count twice any restriction that has both possible extensions $h(a) = 0$ and $h(a) = 1$.

Consider each term separately.

$$\begin{aligned}
\left|H_1'\right| &= \left|\left\{g\colon A' \to \{0,1\} \,\middle|\, \exists h \in \mathscr{H}\colon g = h|_{A'}\right\}\right| \\
&= \left|\left\{B \subseteq A' \,\middle|\, \exists h \in \mathscr{H}\colon B = A \cap h^{-1}(\{1\})\right\}\right| \\
&\leq \left|\left\{B \subseteq A' \,\middle|\, \mathscr{H} \text{ shatters } B\right\}\right| \\
&= \left|\left\{B \subseteq A' \,\middle|\, \mathscr{H} \text{ shatters } B, a \notin B\right\}\right|. \qquad (5.4) \\
\left|H_2'\right| &= \left|H_2|_{A'}\right| \\
&= \left|\left\{B \subseteq A' \,\middle|\, \exists h \in H_2\colon B = A \cap h^{-1}(\{1\})\right\}\right| \\
&\leq \left|\left\{B \subseteq A' \,\middle|\, H_2 \text{ shatters } B\right\}\right| \qquad (5.5) \\
&= \left|\{B \subseteq A \mid H_2 \text{ shatters } B, a \in B\}\right| \qquad (5.6) \\
&\leq \left|\{B \subseteq A \mid \mathscr{H} \text{ shatters } B, a \in B\}\right|. \qquad (5.7)
\end{aligned}$$

To understand the equality in Equation 5.6, notice by the definition of H_2 that H_2 shatters $B \subseteq A'$ if and only if H_2 shatters $B \cup \{a\}$. Hence $B \mapsto B \cup \{a\}$ is a bijection from the set in Equation 5.5 to Equation 5.6.

Combining Equation 5.3, Equation 5.4, and Equation 5.7 yields

$$\begin{aligned}
|\mathscr{H}|_A| &\leq |\{B \subseteq A \mid \mathscr{H} \text{ shatters } B, a \notin B\}| + |\{B \subseteq A \mid \mathscr{H} \text{ shatters } B, a \in B\}| \\
&= |\{B \subseteq A \mid \mathscr{H} \text{ shatters } B\}|
\end{aligned}$$

as desired. □

5.3 Uniform Convergence for Classes of Finite VC Dimension

Definition 5.4

A *set system* is a pair $(\Omega, \mathscr{R})$ such that Ω is a nonempty set and $\mathscr{R}$ is a set of subsets of Ω.

Definition 5.5

Let $(\Omega, \mathscr{R})$ be a set system, let $\mathscr{D}$ be a distribution over Ω, and let $S \in \Omega^m$ for some $m \in \mathbb{N}$. Let $\varepsilon > 0$.

(i) We say S *is an ε-net for* $(\Omega, \mathscr{R})$ *with respect to distribution* $\mathscr{D}$ *if*

$$\mathscr{D}(R) > \varepsilon \implies S \cap R \neq 0 \quad \text{for all } R \in \mathscr{R}.$$

(ii) We say S *is an ε-representative sample for* $(\Omega, \mathscr{R})$ *with respect to distribution* $\mathscr{D}$ *if*

$$\left|\frac{|S \cap R|}{m} - \mathscr{D}(R)\right| \leq \varepsilon. \quad \text{for all } R \in \mathscr{R}.$$

Observe that every ε-representative sample for $(\Omega, \mathscr{R})$ is also an ε-net for for $(\Omega, \mathscr{R})$, but not vice versa. Similarly, we can define ε-nets and ε-representative samples for classes of binary functions.

Definition 5.6

Let $\mathcal{X}$ be a nonempty set, let $\mathscr{H}$ be a set of functions $\mathcal{X} \to \{0,1\}$, let $\Omega = \mathcal{X} \times \{0,1\}$, let $S = ((x_i, y_i))_{i \in [m]} \in \Omega^m$, and let $\mathscr{D}$ be a distribution over Ω. For each $h \in \mathscr{H}$, let

$$R_h = \{(x, y) \in \Omega \colon y \neq h(x)\},$$

and let $\mathcal{R} = \{R_h : h \in \mathcal{H}\}$. We say that S *is an ε-net (ε-representative sample) for class $\mathcal{H}$ with respect to distribution $\mathcal{D}$* if it is an ε-net (ε-representative sample) for $(\Omega, \mathcal{R})$ with respect to $\mathcal{D}$.

In other words, the 0-1 loss satisfies that:

- S is an ε-net for $\mathcal{H}$ with respect to $\mathcal{D}$ if
$$L_{\mathcal{D}}(h) > \varepsilon \implies L_S(h) > 0 \quad \text{for all } h \in \mathcal{H}.$$
- S is an ε-representative sample for $\mathcal{H}$ with respect to $\mathcal{D}$ if
$$|L_S(h) - L_{\mathcal{D}}(h)| \leq \varepsilon \quad \text{for all } h \in \mathcal{H}.$$

Remark 5.7. $\mathcal{H}$ has the uniform convergence property if and only if for any $\varepsilon, \delta \in (0,1)$ there exists $m \in \mathbb{N}$ such that for any distribution $\mathcal{D}$,
$$\mathsf{P}_{S \sim \mathcal{D}^m}[S \text{ is an } \varepsilon\text{-representative sample}] \geq 1 - \delta.$$

Theorem 5.8

There exists a constant $c > 0$ as follows. Let $\varepsilon, \delta \in (0,1)$, let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty sets, let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$, let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$, and assume $\mathsf{VC}(\mathcal{H}) = d < \infty$. Let $S \sim \mathcal{D}^m$, where
$$m \geq c \cdot \frac{d \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}.$$
Then
$$\mathsf{P}_{S \sim \mathcal{D}^m}[S \text{ is an } \varepsilon\text{-net for } \mathcal{H} \text{ with respect to } \mathcal{D}] \geq 1 - \delta.$$

Theorem 5.9

There exists a constant $c > 0$ as follows. Let $\varepsilon, \delta \in (0,1)$, let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty sets, let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$, let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$ with $\mathsf{VC}(\mathcal{H}) = d < \infty$, let $\ell \colon \mathcal{H} \times (\mathcal{X} \times \mathcal{Y}) \to [0,1]$ be a loss function, and let $S \sim \mathcal{D}^m$ where
$$m \geq c \cdot \frac{d \log(d/\varepsilon) + \log(1/\delta)}{\varepsilon^2}.$$
Then

(a) $\mathsf{P}_S[\exists h \in \mathcal{H} \colon |L_S(h) - L_{\mathcal{D}}(h)| > \varepsilon] \leq 4\tau_{\mathcal{H}}(2m) \exp\left(-\frac{\varepsilon^2 m}{8}\right)$.

(b) $\mathsf{P}_S[S$ is an ε-representative sample for $\mathcal{H}$ with respect to $\mathcal{D}$ and $\ell] \geq 1 - \delta$.

Remark 5.10. The proofs of Theorem 5.9 and Remark 5.10 are very similar and contain the same ideas. We present the proof of Theorem 5.9, which is slightly more involved, and leave proving Theorem 5.8 as an exercise.

Proof of Theorem 5.9. Let $S = \{z_i\}_{i \in [m]} \sim \mathcal{D}^m$ and let $S' = \{z_i'\}_{i \in [m]} \sim \mathcal{D}^m$ independently of S. For any $h \in \mathcal{H}$, let $\Delta_S(h) = |L_S(h) - L_{\mathcal{D}}(h)|$. Consider the following two events:
$$E_1 = \{\exists h \in \mathcal{H} \colon \Delta_S(h) > \varepsilon\},$$
$$E_2 = \left\{\exists h \in \mathcal{H} \colon \Delta_S(h) > \varepsilon, \Delta_{S'}(h) \leq \frac{\varepsilon}{2}\right\}.$$
We need to show that $\mathsf{P}_{S,S'}[E_1] \leq \delta$. The proof is partitioned into three claims.

Claim 1. $\mathsf{P}_{S,S'}[E_1] \leq 2\,\mathsf{P}_{S,S' \overset{\text{i.i.d.}}{\sim} \mathcal{D}^m}[E_2]$.

Proof. Intuitively, the idea is that for a fixed $h \in \mathcal{H}$, each of $L_S(h)$ and $L_{S'}(h)$ is a good estimate of $L_{\mathcal{D}}(h)$, and they are independent. Hence, even if we fix a particular h such that $L_S(h)$ is a bad estimate, we can still expect that $L_{S'}(h)$ will be a good estimate.

Note that $\mathsf{P}_{S,S'}[E_2] \geq \mathsf{P}_{S,S'}[E_1 \cap E_2] = \mathsf{P}_{S,S'}[E_2 \mid E_1]\,\mathsf{P}_{S,S'}[E_1]$. To prove Claim 1 it suffices to show that $\mathsf{P}_{S,S'}[E_2 \mid E_1] \geq \frac{1}{2}$. Seeing as E_2 is a subset of E_1,

$$\begin{aligned} \mathop{\mathsf{P}}_{S,S'}[E_2 \mid E_1] &= \mathop{\mathsf{P}}_{S,S'}\left[\exists h \in \mathcal{H} \colon \Delta_S(h) > \varepsilon, \Delta_{S'}(h) \leq \frac{\varepsilon}{2} \,\middle|\, \exists g \in \mathcal{H} \colon \Delta_S(g) > \varepsilon\right] \\ &\geq \mathop{\mathsf{P}}_{S,S'}\left[\Delta_{S'}(g) \leq \frac{\varepsilon}{2} \,\middle|\, \exists g \in \mathcal{H} \colon \Delta_S(g) > \varepsilon\right]. \end{aligned} \tag{5.8}$$

Notice that for any $g \in \mathcal{H}$,

$$L_{S'}(g) - L_{\mathcal{D}}(g) = \frac{1}{m}\sum_{i \in [m]} \left(\ell(g, z_i') - L_{\mathcal{D}}(g)\right)$$

and furthermore $Z_i = \ell(g < z_i') - L_{\mathcal{D}}(g)$ for $i \in [m]$ are i.i.d. random variables with expectation 0 and support in $[-1, 1]$. So for any fixed $g \in \mathcal{H}$, Hoeffding's inequality implies

$$\mathop{\mathsf{P}}_{S,S'}\left[\Delta_{S'}(g) > \frac{\varepsilon}{2}\right] = \mathop{\mathsf{P}}_{S,S'}\left[\left|\frac{1}{m}\sum_{i \in [m]} Z_i\right| > \frac{\varepsilon}{2}\right] \leq 2\exp\left(-\frac{\varepsilon^2 m}{8}\right) \leq \frac{1}{2}, \tag{5.9}$$

where the last inequality holds for m as in the statement of the theorem. This holds also when conditioning on the event $\Delta_S(g) > \varepsilon$, because $S' \perp\!\!\!\perp S$. Combining Equation 5.8 and Equation 5.9 implies $\mathsf{P}_{S,S'}[E_2 \mid E_1] \geq \frac{1}{2}$, concluding the proof. ■

Claim 2. $\mathsf{P}_{S,S'}[E_2] \leq \mathcal{T}_{\mathcal{H}}(2m) \cdot 2\exp\left(-\frac{\varepsilon^2 m}{8}\right)$.

Proof. Intuitively, seeing as $L_S(h)$ and $L_{S'}(h)$ are both good estimates of the same value $L_{\mathcal{D}}(h)$, the probability that they be markedly different for a particular h is small. Let $S_x, S_x' \subseteq \mathcal{X}$ be the set of domain elements that appear in S and S' respectively. A key idea in the proof is that even though $\mathcal{H}$ is an infinite class, the event in which $L_S(h)$ and $L_{S'}(h)$ are very different for a particular h is an event that concerns only how h behaves on the set $X = S_x \cup S_x'$, which is a finite subset of the domain (and this event does not depend on how h and $\mathcal{D}$ behave outside of X). Hence, we can restrict our attention to the projections $h|_X \in \mathcal{H}|_X$ instead of considering functions $h \in \mathcal{H}$. For any particular restricted function $h|_X$, the probability that $L_S(h)$ and $L_{S'}(h)$ are very different vanishes exponentially. Seeing as $\mathcal{H}|_X$ is a finite set of functions ($|\mathcal{H}|_X| \leq \tau_{\mathcal{H}}(2m)$ because $|X| \leq 2m$), we can apply the union bound, and this yields the inequality.

To make this argument formal, there are two issues we need to consider. The first issue is that in order to apply a union bound using the fact that $\mathcal{H}|_X$ is finite, we need the set X to be fixed (if the class $\mathcal{H}|_X$ is itself a random variable, we cannot apply a union bound). We solve this by generating our samples via a two-step process: (1) a vector Z of $2m$ i.i.d. samples is chosen from $\mathcal{D}$; (2) Z is partitioned into two vectors S and S' of length m. Thus, for each fixed value $X = Z_x = S_x \cup S_x'$, the set $\mathcal{H}|_X$ is finite and fixed, and we can apply a union bound separately for each value of Z (using the law of total probability).

The second issue is that for each $h \in \mathcal{H}|_X$, we want to use Hoeffding's inequality to bound the probability that $|L_S(h) - L_{S'}(h)|$ is large. To do so, we need to present this quantity as an average of independent random variables $Q_i = \ell(h, z_i) - \ell(h, z_i')$. To ensure that $Q_1, \ldots, Q_m$ are independent ,we specify that in the two-step process above, Z is partitioned into S and S' in a specific manner as follows. Denote

$Z = (a_1, \ldots, a_m, b_1, \ldots, b_m)$. For each $i \in [m]$, with probability $\frac{1}{2}$, we set $z_i = a_i$ and $z_i' = b_i$, and with probability $\frac{1}{2}$ we make the opposite assignment, namely $z_i = b_i$ and $z_i' = a_i$. Thus, for each fixed value Z, the variables Q_i are independent, and furthermore $\mathsf{E}[Q_i] = 0$ for all $i \in [m]$. Observe that sampling (S, S') using this two step process produces the same joint distribution as sampling $S \sim \mathcal{D}^m$ and $S' \sim \mathcal{D}^m$ independently. This technique of "mixing" or "swapping" the samples between S and S' is known as *symmetrization*.

Putting this all together,

$$\begin{aligned}\underset{S,S'}{\mathsf{P}}[E_2] &= \underset{S,S'}{\mathsf{P}}\left[\exists h \in \mathcal{H} \colon \Delta_S(h) > \varepsilon, \Delta_{S'}(h) \leq \frac{\varepsilon}{2}\right] \\ &\leq \underset{S,S'}{\mathsf{P}}\left[\exists h \in \mathcal{H} \colon |L_S(h) - L_{S'}(h)| \geq \frac{\varepsilon}{2}\right] \\ &= \underset{S,S'}{\mathsf{P}}\left[\exists h \in \mathcal{H}|_X \colon |L_S(h) - L_{S'}(h)| \geq \frac{\varepsilon}{2}\right] \\ &= \underset{Z}{\mathsf{E}}\left[\underset{S,S'}{\mathsf{P}}\left[\exists h \in \mathcal{H}|_X \colon |L_S(h) - L_{S'}(h)| \geq \frac{\varepsilon}{2} \,\middle|\, Z\right]\right].\end{aligned}$$

For any fixed value of Z,

$$\begin{aligned}\underset{S,S'}{\mathsf{P}}\left[\exists h \in \mathcal{H}|_X \colon |L_S(h) - L_{S'}(h)| \geq \frac{\varepsilon}{2} \,\middle|\, Z\right] &\leq \sum_{h \in \mathcal{H}|_X} \mathsf{P}\left[|L_S(h) - L_{S'}(h)| \geq \frac{\varepsilon}{2} \,\middle|\, Z\right] \\ &= \sum_{h \in \mathcal{H}|_X} \mathsf{P}\left[\left|\frac{1}{m}\sum_{i \in [m]} (\ell(h, z_i) - \ell(h, z_i'))\right| \geq \frac{\varepsilon}{2} \,\middle|\, Z\right] \\ &= \sum_{h \in \mathcal{H}|_X} \mathsf{P}\left[\left|\frac{1}{m}\sum_{i \in [m]} Q_i\right| \geq \frac{\varepsilon}{2} \,\middle|\, Z\right] \\ &\leq \sum_{h \in \mathcal{H}|_X} 2\exp\left(-\frac{\varepsilon^2 m}{8}\right) \\ &\leq \tau_{\mathcal{H}}(2m) \cdot 2\exp\left(-\frac{\varepsilon^2 m}{8}\right).\end{aligned}$$

This establishes Claim 2. ■

Combining the two claims yields

$$\underset{S,S'}{\mathsf{P}}[E_1] \leq 4\tau_{\mathcal{H}}(2m)\exp\left(-\frac{\varepsilon^2 m}{8}\right)$$

completing the proof of (a).

To prove (b), the main idea is that by Sauer's lemma, $\tau_{\mathcal{H}}(2m) \leq (2m)^d$, so the number of possible projections $h|_X$ only grows polynomially in m, while Hoeffding's inequality above showed that the probability of the bad event for a particular $h|_X$ vanishes exponentially in m. Namely,

$$\underset{S}{\mathsf{P}}[E_1] \leq 4(2m)^d \exp\left(-\frac{\varepsilon^2 m}{8}\right). \tag{5.10}$$

Seeing as the exponential factor dominates the polynomial factor, $\lim_{m \to \infty} \mathsf{P}_S[E_1] = 0$. Numerically, we need to show that taking m as in the problem statement suffices to show that $\mathsf{P}_S[E_1] \leq \delta$.

Claim 3. For m as in the statement, $\mathsf{P}_S[E_1] \leq \delta$.

Proof. We will show that $\log(\mathsf{P}_S[E_1]) \leq \log(\delta)$, which implies the theorem because the logarithm is monotone increasing. Specifically, fix $c = 24 \cdot 36$, and denote

$$w(m) = \frac{1}{3} \cdot \frac{\varepsilon^2 m}{8} = 36\,(d \log(d/\varepsilon) + \log(1/\delta))\,.$$

Using Claim 2, we show the following:

$$\begin{aligned}\log\left(\underset{S}{\mathsf{P}}[E_1]\right) &\leq \log(4) + d \log(2m) - \frac{\varepsilon^2 m}{8} \\ &= \underbrace{(\log(4) - w(m))}_{\leq 0} + \underbrace{(d \log(2m) - w(m))}_{\leq 0} + \underbrace{(-w(m))}_{\leq \log(\delta)} \leq \log(\delta)\,.\end{aligned}$$

To see this, consider each summand separately. Clearly,

$$\log(4) - w(m) \leq \log(4) - 36 \leq 0,$$

and

$$-w(m) = -36\left(d \log\left(\frac{d}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right)\right) \leq \log(\delta)\,.$$

To establish that the remaining summand satisfies $d \log(2m) - w(m) \leq 0$, we will prove the inequality for the specific value

$$m = m_0 = \frac{24 \cdot 6d}{\varepsilon^2} \log\left(\frac{6d}{\varepsilon}\right) \leq \frac{cd}{\varepsilon^2} \log\left(\frac{d}{\varepsilon}\right),$$

■

and that implies the inequality for all the larger values of m that appear in the statement, because the negative term $w(m)$ is linear in m, and therefore dominates the positive logarithmic term $d \log(2m)$.

$$\begin{aligned} d \log(2m_0) - w(m_0) &= d \log(2m_0) - \frac{1}{3} \cdot \frac{\varepsilon^2 m_0}{8} \\ &= d \log\left(\frac{2 \cdot 24 \cdot 6d}{\varepsilon^2} \log\left(\frac{6d}{\varepsilon}\right)\right) - \frac{6d}{\varepsilon^2} \log\left(\frac{6d}{\varepsilon}\right) \\ &= d \log\left(\frac{48 \cdot 6d}{\varepsilon^2} \log\left(\frac{6d}{\varepsilon}\right)\right) - \frac{6d}{\varepsilon^2} \log\left(\frac{6d}{\varepsilon}\right) \\ &\leq d\left(\log\left(\frac{48 \cdot 6d}{\varepsilon^2} \log\left(\frac{6d}{\varepsilon}\right)\right) - 6 \log\left(\frac{6d}{\varepsilon}\right)\right). \end{aligned}$$

Finally,

$$\log\left(\frac{48 \cdot 6d}{\varepsilon^2} \log\left(\frac{6d}{\varepsilon}\right)\right) = \log\left(\frac{48 \cdot 6d}{\varepsilon^2} \log\left(\left(\frac{d}{\varepsilon}\right)^6\right)\right) < 0.$$

□

Remark 5.11. The constant c chosen in this proof is far from tight.

5.4 The Fundamental Theorem of PAC Learning

We have shown that classes with finite VC dimension have the uniform convergence property, and so they are agnostic PAC learnable (and therefore PAC learnable). Together with the lower bound of Proposition 4.15, this yields the following important characterization of learnability.

Theorem 5.12 (Fundamental Theorem of PAC Learning – Qualitative Version)
Let $\mathcal{X}$ be a nonempty set, and let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \{0, 1\}$. The following conditions are equivalent:

1. $\mathsf{VC}(\mathcal{H}) < \infty$.
2. $\mathcal{H}$ has the uniform convergence property.
3. Every $\mathrm{ERM}_{\mathcal{H}}$ algorithm is an agnostic PAC learner for $\mathcal{H}$.
4. $\mathcal{H}$ is agnostic PAC learnable.
5. Every $\mathrm{ERM}_{\mathcal{H}}$ algorithm is a PAC learner for $\mathcal{H}$.
6. $\mathcal{H}$ is PAC learnable.

Proof. We show the following implications.

- $1 \implies 2$. This follows from Theorem 5.9 and Remark 5.7.
- $2 \implies 3$. This follows from Lemma 4.11.
- $3 \implies 4 \implies 6$ and $3 \implies 5 \implies 6$. These implications are immediate from the definitions of PAC and agnostic PAC learning.
- $6 \implies 1$. This is the contrapositive of Proposition 4.15.

□

Furthermore, it is possible to give quantitative bounds on the sample complexity for classes with finite VC dimension.

Theorem 5.13 (Fundamental Theorem of PAC Learning – Quantitative Version)
Let $\mathcal{X}$ be a nonempty set, and let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \{0, 1\}$, and let $d = \mathsf{VC}(\mathcal{H})$. Assume $d = \mathsf{VC}(\mathcal{H}) < \infty$. Then there exist constants $c_0, c_1 > 0$ such that:

1. $\mathcal{H}$ has the uniform convergence property with sample complexity
$$c_0 \cdot \frac{d + \log\left(\frac{1}{\delta}\right)}{\varepsilon^2} \leq m_{\mathcal{H}}^{\mathrm{UC}}(\varepsilon, \delta) \leq c_1 \cdot \frac{d + \log\left(\frac{1}{\delta}\right)}{\varepsilon^2}.$$
2. $\mathcal{H}$ is agnostic PAC learnable with sample complexity
$$c_0 \cdot \frac{d + \log\left(\frac{1}{\delta}\right)}{\varepsilon^2} \leq m_{\mathcal{H}}(\varepsilon, \delta) \leq c_1 \cdot \frac{d + \log\left(\frac{1}{\delta}\right)}{\varepsilon^2}.$$
3. $\mathcal{H}$ is PAC learnable with sample complexity
$$c_0 \cdot \frac{d + \log\left(\frac{1}{\delta}\right)}{\varepsilon^2} \leq m_{\mathcal{H}}(\varepsilon, \delta) \leq c_1 \cdot \frac{d \log\left(\frac{1}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right)}{\varepsilon^2}.$$

The upper bounds in Theorem 5.13 for the realizable and agnostic cases are related to Theorem 5.8 and Theorem 5.9 respectively. More specifically, item 1 is similar to Theorem 5.9, and item 2 follows from item 1 because uniform convergence implies agnostic PAC learnability (by Lemma 4.11). Similarly, for the realizable case, if the sample is an ε-net, then with probability $1 - \delta$ any $\mathrm{ERM}_{\mathcal{H}}$ algorithm is a PAC learner, and therefore item 3 follows from Theorem 5.8.

However, employing Theorem 5.9 to prove items 1 and 2 in the manner just discussed yields upper bounds of

$$c \cdot \frac{d\log\left(\frac{d}{\varepsilon}\right) + \log\left(\frac{1}{\delta}\right)}{\varepsilon^2},$$

which is not tight. This expression is a factor of $\log(d/\varepsilon)$ larger than in the statement of Theorem 5.13. Usually this would not make a big difference in practice, but nonetheless, in upcoming units we will present an analysis using Rademacher complexity and covering numbers, which provide a different perspective on this theorem and yield the tighter bounds.

For the lower bounds in Theorem 5.13, we have already seen that if ε and δ are constants that are less than $\frac{1}{8}$, then Proposition 4.15 implies a lower bound of $\Omega(d)$ on the sample complexity.

5.5 Discussion

This unit tied together many of the notions we have seen so far in the course, creating one unified theory of PAC learning. In a sentence, this can be summarized as follows: The VC dimension determines the general outline of the growth function, which in turn determines whether a class satisfies uniform convergence, which is equivalent to agnostic PAC learning, which implies PAC learning, which (by the no free lunch theorems) implies a finite VC dimension.

In the next two units we will see a different perspective on these issues, employing Rademacher complexity, covering numbers, and chaining.

6 Rademacher Complexity

6.1 Introduction

In the previous unit we saw how the VC dimension controls the growth function, which in turn determines the sample complexity of uniform convergence. In this unit we provide somewhat different perspective on uniform convergence, via Rademacher complexity. The Rademacher complexity depends both on the hypothesis class $\mathcal{H}$ and on the unknown distribution $\mathcal{D}$, in contrast to the VC dimension that depends solely on the hypothesis class. Thus, the Rademacher complexity can be understood as an average-case analysis, in contrast to the worst-case analysis offered by the VC dimension. As we will see in the next Unit, the more nuanced Rademacher complexity analysis also lends itself to a technique called chaining, which will allow us to eliminate the unnecessary logarithmic factor the that appeared in our derivation of the fundamental theorem in the previous unit.

6.2 Concentration of Measure for Uniform Convergence

The following notation will be handy for our analysis of uniform convergence.

Notation 6.1
We write

$$\begin{aligned}\Delta_S(\mathcal{H}) &= \sup_{h\in\mathcal{H}} |L_S(h) - L_{\mathcal{D}}(h)| \\ \Delta_S^+(\mathcal{H}) &= \sup_{h\in\mathcal{H}} (L_S(h) - L_{\mathcal{D}}(h)) \\ \Delta_S^-(\mathcal{H}) &= \sup_{h\in\mathcal{H}} (L_{\mathcal{D}}(h) - L_S(h))\end{aligned}$$

where $\mathcal{H}$ is a class of functions, S is a sample, $\mathcal{D}$ is a distribution, and the loss function should be understood from the context.

Uniform convergence means that if S is a large i.i.d. sample then with high probability $\Delta_S(\mathcal{H})$ is small. The following claim says that $\Delta_S(\mathcal{H})$ is close to its expectation. This implies that, in order to show that $\Delta_S(\mathcal{H})$ is small, it will suffice to bound its expectation.

Proposition 6.2
Let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty sets, let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$, let $\ell\colon \mathcal{Y}^2 \to [0,1]$ be a loss function, and let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$. Then for any $m \in \mathbb{N}$ and $\varepsilon > 0$, if $S' \sim \mathcal{D}^m$ is i.i.d. with S,

$$\mathop{\mathsf{P}}_{S}\left[\Delta_S(\mathcal{H}) \geq \mathop{\mathsf{E}}_{S'}[\Delta_{S'}(\mathcal{H})] + \varepsilon\right] \leq \exp\left(-2m\varepsilon^2\right),$$

and similarly,

$$\begin{aligned}\mathop{\mathsf{P}}_{S}\left[\Delta_S^+(\mathcal{H}) \geq \mathop{\mathsf{E}}_{S'}\left[\Delta_{S'}^-(\mathcal{H})\right] + \varepsilon\right] &\leq \exp\left(-2m\varepsilon^2\right), \\ \mathop{\mathsf{P}}_{S}\left[\Delta_S^-(\mathcal{H}) \geq \mathop{\mathsf{E}}_{S'}\left[\Delta_{S'}^-(\mathcal{H})\right] + \varepsilon\right] &\leq \exp\left(-2m\varepsilon^2\right).\end{aligned}$$

This claim follows from the following concentration-of-measure theorem.

Theorem 6.3 (McDiarmid's Inequality)
Let Ω be a set and let $f : \Omega^m \to \mathbb{R}$ be a function. Assume there exist $c_1, \ldots, c_m \in \mathbb{R}$ such that f satisfies the following bounded differences property:

$$\forall z_1, \ldots, z_m z_1', \ldots, z_m' \in \Omega\ \forall i \in [m]\colon \left|f(z_1, \ldots, z_i, \ldots, z_m) - f(z_1, \ldots, z_i', \ldots, z_m)\right| \leq c_i.$$

Let $Z_1, \ldots, Z_m$ be independent random variables taking values in Ω. Then for any $\varepsilon > 0$,

$$\mathsf{P}\Big[f\,(Z_1, \ldots, Z_m) - \mathsf{E}[f(Z_1, \ldots, Z_m)] \geq \varepsilon\Big] \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^n c_i^2}\right)$$

and

$$\mathsf{P}\Big[\mathsf{E}[f(Z_1, \ldots, Z_m)] - f\,(Z_1, \ldots, Z_m) \geq \varepsilon\Big] \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^n c_i^2}\right).$$

Remark 6.4.

- In the special case where $c_i = \frac{1}{m}$ for all $i \in [m]$, the bound specifies $\exp(-2m\varepsilon^2)$.
- McDiarmid's inequality is a generalization of Hoeffding's inequality. Hoeffding's is a concentration of measure result for the average of independent random variables, whereas McDiarmid's is a concentration of measure result for any function of independent random variables that satisfies the bounded differences property (including the average).
- McDiarmid's inequality is very powerful, because the function f can be arbitrarily complex so long as it satisfies the bounded differences property. Below we will apply McDiarmid's inequality to the function $f(S) = \Delta_S(\mathcal{H})$, which is a non-trivial function that involves a supremum over a possibly infinite class $\mathcal{H}$.

Proof of Proposition 6.2. We prove the first inequality, the proof for theo ther two inequalities is similar.

Fix $m \in \mathbb{N}$. First, we claim that $\Delta_S(\mathcal{H})$ satisfies the following bounded difference property. For any $S, S' \in (\mathcal{X} \times \mathcal{Y})^m$ with $S = \{(x_i, y_i)\}_{i \in [m]}$, $S' = \left\{(x_i', y_i')\right\}_{i \in [m]}$, if there exists $j \in [m]$ such that $(x_i, y_i) = (x_i', y_i')$ for all $i \neq j$, then

$$|\Delta_S(\mathcal{H}) - \Delta_{S'}(\mathcal{H})| \leq \frac{1}{m}. \tag{6.1}$$

To see this, notice that

$$\begin{aligned}
\Delta_{S'}(\mathcal{H}) &= \sup_{h \in \mathcal{H}} |L_{S'}(h) - L_{\mathfrak{D}}(h)| = \sup_{h \in H} |L_{S'}(h) - L_S(h) + L_S(h) - L_{\mathfrak{D}}(h)| \\
&\leq \sup_{h \in \mathcal{H}} \left(|L_{S'}(h) - L_S(h)| + |L_S(h) - L_{\mathfrak{D}}(h)|\right) \\
&= \sup_{h \in \mathcal{H}} \left(\left|\frac{\ell(h(x_j'), y_j') - \ell(h(x_j), y_j)}{m}\right| + |L_S(h) - L_{\mathfrak{D}}(h)|\right) \\
&\leq \frac{1}{m} + \sup_{h \in \mathcal{H}} |L_S(h) - L_{\mathfrak{D}}(h)| \\
&= \frac{1}{m} + \Delta_S(\mathcal{H}).
\end{aligned}$$

Applying the same argument with roles of S and S' reversed implies that $|\Delta_S(\mathcal{H}) - \Delta_{S'}(\mathcal{H})| \leq \frac{1}{m}$. Hence, by Theorem 6.3,

$$\mathsf{P}_S\left[\Delta_S(\mathcal{H}) \geq \mathsf{E}_{S'}[\Delta_{S'}(\mathcal{H})] + \varepsilon\right] \leq \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^m (1/m)^2}\right) = \exp\left(-2m\varepsilon^2\right).$$

□

6.3 Definition of Rademacher Complexity

Definition 6.5
Let $A \subseteq \mathbb{R}^m$ be a bounded set of vectors. The *Rademacher Average of A* is

$$\mathsf{Rad}(A) = \mathop{\mathsf{E}}_{\sigma}\left[\sup_{a \in A} \frac{\langle \sigma, a \rangle}{m}\right] = \mathop{\mathsf{E}}_{\sigma}\left[\sup_{a \in A} \frac{1}{m} \sum_{i=1}^{m} \sigma_i a_i\right],$$

where $\sigma = (\sigma_1, \ldots, \sigma_m) \sim \mathrm{Uni}\big(\{\pm 1\}^m\big)$ is a vector of *Rademacher variables*.

The Rademacher average quantifies how well A correlates with a random vector $\sigma \in \{\pm 1\}^M$. Similarly, the following definition quantifies the *richness* or *complexity* of a class of functions by measuring how well the functions can correlate with random labels.

Definition 6.6
Fix $m \in \mathbb{N}$. Let $\mathcal{X}$ be a nonempty set, and let $\mathcal{H}$ be a class of functions $\mathcal{X} \to [-1, 1]$. For any set $S = \{x_i\}_{i \in [m]} \in \mathcal{X}^m$, let

$$\mathcal{H}(S) = \big\{(f(x_i))_{i \in [m]} : f \in \mathcal{H}\big\} \subseteq \mathbb{R}^m.$$

(i) Fix $S \sim \mathcal{X}^m$. The *empirical Rademacher complexity of $\mathcal{H}$ with respect to S* is

$$\mathsf{Rad}_S(\mathcal{H}) = \mathsf{Rad}(\mathcal{H}(S)) = \mathop{\mathsf{E}}_{\sigma}\left[\sup_{f \in \mathcal{H}} \frac{1}{m} \sum_{i \in [m]} \sigma_i f(x_i)\right].$$

(ii) Let $\mathcal{D}$ be a distribution over $\mathcal{X}$. The *Rademacher complexity of size m of $\mathcal{H}$ with respect to $\mathcal{D}$* is, if $S \sim \mathcal{D}^m$,

$$\mathsf{Rad}_{\mathcal{D},m}(\mathcal{H}) = \mathop{\mathsf{E}}_{S}[\mathsf{Rad}(\mathcal{H}(S))] = \mathop{\mathsf{E}}_{S,\sigma}\left[\sup_{f \in \mathcal{H}} \frac{1}{m} \sum_{i \in [m]} \sigma_i f(x_i)\right].$$

Definition 6.7
Let $\mathcal{X}$ and $\mathcal{Y}$ be sets, let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$, and let $\ell \colon \mathcal{Y}^2 \to [0, 1]$ be a loss function. The *loss class of $\mathcal{H}$ with respect to ℓ* is

$$\mathcal{L} = \{(x, y) \mapsto \ell(h(x), y) \colon h \in \mathcal{H}\}.$$

(Namely, the loss class $\mathcal{L}$ is a set of functions $\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$.)

Proposition 6.8
Let $\mathcal{X}$ be a set, let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \{\pm 1\}$, and let

$$\mathcal{L} = \Big\{(x, y) \mapsto \mathbb{1}(h(x) \neq y) : h \in \mathcal{H}\Big\}$$

be the loss class of $\mathcal{H}$ with respect to the 0–1 loss. Let $S = \{(x_i, y_i)\}_{i \in [m]} \in (\mathcal{X} \times \{\pm 1\})^m$ be a sample, and let $S_x = \{x_i\}_{i \in [m]}$. Then

$$\mathsf{Rad}_S(\mathcal{L}) = \frac{1}{2}\mathsf{Rad}_{S_x}(\mathcal{H})\,.$$

Proof.

$$\begin{aligned}
\mathsf{Rad}_S(\mathcal{L}) &= \mathop{\mathsf{E}}_{\sigma}\left[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i\in[m]} \sigma_i\, \mathbb{1}(h(x_i) \neq y_i)\right] \\
&= \mathop{\mathsf{E}}_{\sigma}\left[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i\in[m]} \sigma_i \frac{1 - y_i h(x_i)}{2}\right] \\
&= \frac{1}{2}\mathop{\mathsf{E}}_{\sigma}\left[\sup_{h\in\mathcal{H}} \left(\frac{1}{m}\sum_{i\in[m]} \sigma_i + \frac{1}{m}\sum_{i\in[m]} \sigma_i(-y_i)h(x_i)\right)\right] \\
&= \frac{1}{2}\mathop{\mathsf{E}}_{\sigma}\left[\frac{1}{m}\sum_{i\in[m]} \sigma_i + \sup_{h\in\mathcal{H}} -\frac{1}{m}\sum_{i\in[m]} \sigma_i y_i h(x_i)\right] \\
&= \frac{1}{2}\mathop{\mathsf{E}}_{\sigma}\left[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i\in[m]} \sigma_i(-y_i)h(x_i)\right] \\
&= \frac{1}{2}\mathop{\mathsf{E}}_{\sigma}\left[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i\in[m]} \sigma_i h(x_i)\right] \\
&= \frac{1}{2}\mathsf{Rad}_{S_x}(\mathcal{H})\,.
\end{aligned}$$

□

Proposition 6.9

If $\mathcal{H}$ is a class of functions $\mathcal{X} \to \{0, 1\}$ then $\mathsf{Rad}_S(\mathcal{L}) = \mathsf{Rad}_{S_x}(\mathcal{H})$.

Lemma 6.10

Let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty sets, and let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$. Let $\mathcal{L}$ be the loss class of $\mathcal{H}$ with respect to some loss function $\ell\colon \mathcal{Y}^2 \to [0, 1]$. Then for any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ and $m \in \mathbb{N}$, if $S \sim \mathcal{D}^m$,

$$\begin{aligned}
&\mathop{\mathsf{E}}_{S}\big[\Delta_S^+(\mathcal{H})\big] \leq 2\mathsf{Rad}_{\mathcal{D},m}(\mathcal{L})\,, \text{ and} \\
&\mathop{\mathsf{E}}_{S}\big[\Delta_S^-(\mathcal{H})\big] \leq 2\mathsf{Rad}_{\mathcal{D},m}(\mathcal{L})\,.
\end{aligned}$$

Proof. We will present the proof for the first inequality; the proof for the second inequality is similar. Note that

$$\mathop{\mathsf{E}}_{S}[L_S(h)] = \frac{1}{m}\sum_{i\in[m]} \mathop{\mathsf{E}}_{S}[\ell(h(x_i), y_i)] = \frac{1}{m}\sum_{i\in[m]} L_{\mathcal{D}}(h) = L_{\mathcal{D}}(h). \tag{6.2}$$

Hence, we can use the double sampling technique to express $\mathsf{E}_S\big[\Delta_S^+(\mathcal{H})\big]$ as an expectation concerning a finite sample:

$$\mathop{\mathsf{E}}_{S}\big[\Delta_S^+(\mathcal{H})\big] = \mathop{\mathsf{E}}_{S}\left[\sup_{h\in\mathcal{H}} (L_S(h) - L_{\mathcal{D}}(h))\right]$$

$$
\begin{aligned}
&= \mathop{\mathsf{E}}_{S}\left[\sup_{h\in\mathcal{H}}\left(L_S(h) - \mathop{\mathsf{E}}_{S'}[L_{S'}(h)]\right)\right] \\
&= \mathop{\mathsf{E}}_{S}\left[\sup_{h\in\mathcal{H}}\mathop{\mathsf{E}}_{S'}\left[\frac{1}{m}\sum_{i\in[m]}\left(\ell(h(x_i),y_i) - \ell(h(x_i'),y_i')\right)\right]\right] \\
&= \mathop{\mathsf{E}}_{S}\left[\sup_{h\in\mathcal{H}}\mathop{\mathsf{E}}_{S'}\left[\frac{1}{m}\sum_{i\in[m]}Q_i(h)\right]\right] \\
&\le \mathop{\mathsf{E}}_{S,S'}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i\in[m]}Q_i(h)\right],
\end{aligned}
$$

where we used the notation $Q_i(h) = \ell(h(x_i), y_i) - \ell(h(x_i'), y_i')$.

S and S' are independent and have the same distribution, and therefore we can view the samples from S and S' as being interchangeable. Formally, we use the following *symmetrization* technique (we already saw a variant of this technique in the previous unit). Each random variable $Q_i(h) = \ell(h(x_i), y_i) - \ell(h(x_i'), y_i')$ is equal in distribution to $-Q_i(h) = \ell(h(x_i'), y_i') - \ell(h(x_i), y_i)$, because flipping the sign corresponds to swapping the names of (x_i, y_i) with (x_i', y_i') for some i. Moreover, if we introduce Rademacher variables σ_i (that are independent and uniform over $\{\pm 1\}$), then the entire matrix $\{Q_i(h)\}_{i\in[m],h\in\mathcal{H}}$ is equal in distribution to the matrix $\{\sigma_i Q_i(h)\}_{i\in[m],h\in\mathcal{H}}$, for the same reason. Hence

$$
\begin{aligned}
\mathop{\mathsf{E}}_{S,S'}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i\in[m]}Q_i(h)\right] &= \mathop{\mathsf{E}}_{S,S',\sigma}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i\in[m]}\sigma_i Q_i(h)\right] \\
&= \mathop{\mathsf{E}}_{S,S',\sigma}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i\in[m]}\sigma_i\left(\ell(h(x_i),y_i) - \ell(h(x_i'),y_i')\right)\right] \\
&\le \mathop{\mathsf{E}}_{S,S',\sigma}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i\in[m]}\sigma_i\ell(h(x_i),y_i) + \sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i\in[m]}(-\sigma_i)\ell(h(x_i'),y_i')\right] \\
&= \mathop{\mathsf{E}}_{S,\sigma}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i\in[m]}\sigma_i\ell(h(x_i),y_i)\right] + \mathop{\mathsf{E}}_{S,\sigma}\left[\sup_{h\in\mathcal{H}}\frac{1}{m}\sum_{i\in[m]}(-\sigma_i)\ell(h(x_i'),y_i')\right] \\
&= 2\mathsf{Rad}_{\mathcal{D},m}(\mathcal{L}).
\end{aligned}
$$

□

As a corollary, we obtain the following PAC learning bounds.

Theorem 6.11

Let $\delta \in (0,1)$, let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty sets, let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$, and let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$. Let $\ell\colon \mathcal{Y}^2 \to [0,1]$. Let $S \sim \mathcal{D}^m$ for some $m \in \mathbb{N}$.

(i) With probability at least $1-\delta$,

$$
\forall h \in \mathcal{H}:\ L_{\mathcal{D}}(h) \le L_S(h) + 2\mathsf{Rad}_{\mathcal{D},m}(\mathcal{L}) + \sqrt{\frac{\log(1/\delta)}{2m}},
$$

where $\mathcal{L}$ is the loss class of $\mathcal{H}$ w.r.t. ℓ.

(ii) Assume $\mathcal{Y} = \{\pm 1\}$ and ℓ is the 0–1 loss. Then with probability at least $1 - \delta$,

$$\forall h \in \mathcal{H}\colon\ L_{\mathcal{D}}(h) \leq L_S(h) + \mathsf{Rad}_{\mathcal{D}_x,m}(\mathcal{H}) + \sqrt{\frac{\log(1/\delta)}{2m}}.$$

(iii) Assume $\mathcal{Y} = \{\pm 1\}$, ℓ is the 0-1 loss, and let h be the output of an $\mathrm{ERM}_{\mathcal{H}}$ algorithm executed on S. Then with probability at least $1 - \delta$,

$$L_{\mathcal{D}}(h) \leq \inf_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + 2\mathsf{Rad}_{\mathcal{D}_x,m}(\mathcal{H}) + \sqrt{\frac{2\log(2/\delta)}{m}}.$$

Proof.

(i) Denote $\varepsilon = \sqrt{\frac{\log(1/\delta)}{2m}}$. From Proposition 6.2,

$$\mathop{\mathsf{P}}_{S}\left[\Delta_S^-(\mathcal{H}) \geq \mathop{\mathsf{E}}_{S'}\left[\Delta_{S'}^-(\mathcal{H})\right] + \varepsilon\right] \leq \exp\left(-2m\varepsilon^2\right) = \delta.$$

Hence, with probability at least $1 - \delta$, it is the case that $\Delta_S(\mathcal{H}) \leq \mathsf{E}_{S'}[\Delta_{S'}\mathcal{H}] + \varepsilon$, and then for all $h \in \mathcal{H}$,

$$\begin{aligned} L_{\mathcal{D}}(h) &= L_S(h) + (L_{\mathcal{D}}(h) - L_S(h)) \\ &\leq L_S(h) + \Delta_S^-(\mathcal{H}) \\ &\leq L_S(h) + \mathop{\mathsf{E}}_{S'}\left[\Delta_{S'}^-(\mathcal{H})\right] + \varepsilon \\ &\leq L_S(h) + 2\mathsf{Rad}_{\mathcal{D},m}(\mathcal{L}) + \varepsilon && \text{(Lemma 6.10)} \end{aligned}$$

(ii) Follows from (i) and Proposition 6.8.

(iii) Conceptually, this follows from (ii) together with the fact that uniform convergence implies that any ERM algorithm is a PAC learner (Lemma 3.16). More fully, choosing $\varepsilon = \sqrt{\frac{2\log(2/\delta)}{m}}$, Proposition 6.2 implies that

$$\mathop{\mathsf{P}}_{S}\left[\Delta_S^+(\mathcal{H}) \geq \mathop{\mathsf{E}}_{S'}\left[\Delta_{S'}^+(\mathcal{H})\right] + \frac{\varepsilon}{2}\right] \leq \exp\left(-2m\left(\frac{\varepsilon}{2}\right)^2\right) = \frac{\delta}{2}, \tag{6.3}$$

$$\mathop{\mathsf{P}}_{S}\left[\Delta_S^-(\mathcal{H}) \geq \mathop{\mathsf{E}}_{S'}\left[\Delta_{S'}^-(\mathcal{H})\right] + \frac{\varepsilon}{2}\right] \leq \exp\left(-2m\left(\frac{\varepsilon}{2}\right)^2\right) = \frac{\delta}{2}. \tag{6.4}$$

Hence, with probability at least $1 - \delta$, for any $h^* \in \mathcal{H}$,

$$\begin{aligned} L_{\mathcal{D}}(h) &\leq \underbrace{L_{\mathcal{D}}(h) - L_S(h)}_{\leq \Delta_S^-(\mathcal{H})} + \underbrace{L_S(h) - L_S(h')}_{\leq 0} + \underbrace{L_S(h^*) - L_{\mathcal{D}}(h^*)}_{\leq \Delta_S^+(\mathcal{H})} + L_{\mathcal{D}}(h^*) \\ &\leq L_{\mathcal{D}}(h^*) + \mathop{\mathsf{E}}_{S'}\left[\Delta_{S'}^-(\mathcal{H})\right] + \mathop{\mathsf{E}}_{S'}\left[\Delta_{S'}^+(\mathcal{H})\right] + \varepsilon && \text{(from Equation 6.3 and Equation 6.4)} \\ &\leq L_{\mathcal{D}}(h^*) + 4\mathsf{Rad}_{\mathcal{D},m}(\mathcal{L}) + \varepsilon && \text{(Lemma 6.10)} \\ &\leq L_{\mathcal{D}}(h^*) + 2\mathsf{Rad}_{\mathcal{D}_x,m}(\mathcal{H}) + \varepsilon && \text{(Proposition 6.8)} \end{aligned}$$

as desired.

□

6.4 Bounding the Rademacher Complexity

Theorem 6.11 shows that to obtain PAC learning bounds, it suffices to bound the Rademacher complexity.

6.4.1 Estimating the Rademacher Complexity

In some cases, it is possible to show that the Rademacher complexity is small by estimating it empirically. Namely, one can take samples from the unknown distribution and compute the empirical Rademacher complexity. By McDiarmid's inequality, the empirical Rademacher complexity is close to the Rademacher complexity. This yields a version of Theorem 6.11 that contains the empirical Rademacher complexity instead of the Rademacher complexity.

Note that for a fixed sample S, the empirical Rademacher complexity is defined as an average over all possible assignments to the Rademacher variables, so computing it would appear to require exponential time in the number of samples in S. To over come this, one can instead estimate the empirical Rademacher complexity by sampling a small number of vectors of Rademacher variables uniformly at random, and taking the average only over these vectors. By Hoeffding's inequality, this estimate converges exponentially fast to the empirical Rademacher complexity, and this can again yield a bound similar to Theorem 6.11, that involves only the estimate of the empirical Rademacher complexity.

Unfortunately, the computational complexity can be prohibitive even if we attempt only to estimate the empirical Rademacher complexity as outlined above. This is because for any fixed sample S and vector of Rademacher variables σ, computing $\sup_{f \in \mathcal{H}} \frac{1}{m} \sum_{i \in [m]} \sigma_i f(x_i)$ is a combinatorial optimization problem that involves searching over the entire class $\mathcal{H}$. For many hypothesis classes, this optimization problem can be NP-hard.

6.4.2 Combinatorial Bound on the Rademacher Complexity

Another approach is to bound the Rademacher Complexity using the VC dimension. This is tantamount to saying that the average-case analysis offered by the Rademacher complexity is upper bounded by the worst-case analysis offered by the VC dimension

Lemma 6.12 (Maximal Inequality)
Let $n \in \mathbb{N}$, let $v > 0$, let $Z_1, \dots, Z_n$ be real-valued random variables, and assume that for all $i \in [n]$ and $\lambda > 0$,

$$\psi_{Z_i}(\lambda) \leq \frac{\lambda^2 v}{2}.$$

Then

$$\mathsf{E}[\max\{Z_1, \dots, Z_n\}] \leq \sqrt{2v \log(n)}.$$

Remark 6.13. The variables $Z_1, \dots, Z_n$ in the lemma might not be independent.

Proof of Lemma 6.12. For any $\lambda > 0$,

$$\begin{aligned}
\exp\Big(\lambda\, \mathsf{E}[\max\{Z_1, \dots, Z_n\}]\Big) &= \exp\left(\mathsf{E}\left[\lambda \max_{i \in [n]} Z_i\right]\right) \\
&\leq \mathsf{E}\left[\exp\left(\lambda \max_{i \in [n]} Z_i\right)\right] \\
&= \mathsf{E}\left[\max_{i \in [n]} \mathrm{e}^{\lambda Z_i}\right] \\
&\leq \mathsf{E}\left[\sum_{i \in [n]} \mathrm{e}^{\lambda Z_i}\right] \\
&= \sum_{i \in [n]} \mathsf{E}\left[\mathrm{e}^{\lambda Z_i}\right]
\end{aligned}$$

$$\leq \sum_{i\in[n]} \mathrm{e}^{\frac{\lambda^2 v}{2}} = n\mathrm{e}^{\frac{\lambda^2 v}{2}}.$$

Taking logarithms on both sides yields

$$\mathsf{E}[\max\{Z_1,\dots,Z_n\}] \leq \frac{\log(n)}{\lambda} + \frac{\lambda v}{2}.$$

Choosing $\lambda = \sqrt{\frac{2\log(n)}{v}}$, which minimizes the right hand side, we obtain

$$\mathsf{E}[\max\{Z_1,\dots,Z_n\}] \leq \sqrt{2v\log(n)}.$$

□

Lemma 6.14 (Finite Class, Massart)
Let $A \subseteq \mathbb{R}^m$ be a finite subset, and assume there exists $r \in \mathbb{R}$ such that for all $x \in A$, $\|x\|_2 \leq r$. Then

$$\mathsf{Rad}(A) \leq \frac{r\sqrt{2\log(|A|)}}{m}.$$

Proof. Write

$$\mathsf{Rad}(A) = \mathop{\mathsf{E}}_{\sigma}\left[\sup_{a\in A}\frac{\langle\sigma,a\rangle}{m}\right] = \frac{1}{m}\mathop{\mathsf{E}}_{\sigma}\left[\max_{a\in A} Z_a\right], \tag{6.5}$$

where $Z_a = \sum_{i\in[m]}\sigma_i a_i$, and we used the fact that A is finite. By **??**,

$$\psi_{\sigma_i a_i}(\lambda) \leq \frac{\lambda^2(2a_i)^2}{8} = \frac{\lambda^2 a_i^2}{2}.$$

Hence,

$$\psi_{Z_a} = \psi_{\sum_{i\in[m]}\sigma_i a_i}(\lambda) = \sum_{i\in[m]}\psi_{\sigma_i a_i}(\lambda) \leq \sum_{i\in[m]}\frac{\lambda^2 a_i^2}{2} \leq \frac{\lambda^2 r^2}{2}.$$

Lemma 6.12 implies that

$$\mathop{\mathsf{E}}_{\sigma}\left[\max_{a\in A} Z_a\right] \leq r\sqrt{2\log(|A|)}. \tag{6.6}$$

Combining Equation 6.5 and Equation 6.6 implies the lemma. □

6.4.3 Learning Bounds for VC Classes from Rademacher Complexity

As a corollary from Lemma 6.14, the Rademachare complexity is bounded by the VC dimension.

Theorem 6.15
Let $\mathcal{X}$ be a nonempty set, and let $\mathcal{H}$ be a set of functions $\mathcal{X} \to \{0,1\}$, and let $\mathcal{D}$ be a distribution over $\mathcal{X}$. Then for all $m \in \mathbb{N}$,

$$\mathsf{Rad}_{\mathcal{D},m}(\mathcal{H}) \leq \sup_{S\in\mathcal{X}^m}\mathsf{Rad}_S(\mathcal{H}) \leq \sqrt{\frac{2\log(\tau_{\mathcal{H}}(m))}{m}} \leq O\left(\sqrt{\frac{\log(m/d)}{m/d}}\right),$$

where the last inequality holds if $\mathsf{VC}(\mathcal{H}) = d \leq \infty$.

Proof. The first inequality is immediate from the definition of $\mathsf{Rad}_{\mathcal{D},m}(\mathcal{H})$. For the second inequality, for any

$S \in \mathcal{X}^m$,

$$\mathsf{Rad}_S(\mathcal{H}) = \mathsf{Rad}(\mathcal{H}(S)) \leq \max_{f \in \mathcal{H}} \|f(S)\|_2 \cdot \frac{\sqrt{2\log(|\mathcal{H}(S)|)}}{m} \leq \sqrt{\frac{2\log(\tau_{\mathcal{H}}(m))}{m}}.$$

We used the fact that

$$\max_{f \in \mathcal{H}} \|f(S)\|_2 \leq \|(1, \dots, 1)\|_2 = \sqrt{m}.$$

The final inequality in the statement follows from Sauer's Lemma, which states that $\tau_{\mathcal{H}}(m) \leq (\varepsilon m/d)^d$, and so $\sqrt{\frac{2\log(\tau_{\mathcal{H}}(m))}{m}} \leq \sqrt{\frac{2(\log(m/d)+1)}{m/d}}$. □

6.5 Discussion

Combining Theorem 6.15 and Theorem 6.11 (iii) yields the following learning bound for binary classification using an ERM algorithm with respect to the 0–1 loss:

$$\begin{aligned} L_{\mathcal{D}}(h) &\leq \inf_{h' \in \mathcal{H}} \left(L_{\mathcal{D}}(h') + 2\mathsf{Rad}_{\mathcal{D}_x,m}(\mathcal{H}) + \sqrt{\frac{2\log(2/\delta)}{m}} \right) \\ &\leq \inf_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + O\left(\sqrt{\frac{\log(m/d)}{m/d}} \right) + \sqrt{\frac{2\log(2/\delta)}{m}} \\ &\leq \inf_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + O\left(\sqrt{\frac{d\log(m/d) + \log(1/\delta)}{m}} \right). \end{aligned}$$

In particular this implies via a direct calculation that taking

$$m = O\left(\frac{d\log(1/\varepsilon) + \log(1/\delta)}{\varepsilon^2} \right)$$

samples is sufficient for agnostic PAC learning a class of binary functions of VC dimension d with accuracy ε and confidence $1 - \delta$. However, the fundamental theorem states a stronger bound of $O\left(\frac{d+\log(1/\delta)}{\varepsilon^2}\right)$. In the next unit we will use connections between Rademacher complexity and covering numbers to obtain that stronger bound.

7 Covering Numbers and Chaining

7.1 Definitions

Definition 7.1
A *pseudo-metric space* is a tuple (Ω, ρ) where Ω is a set and $\rho \colon \Omega^2 \to (0, \infty]$ is a function such that for every $x, z, y \in \Omega$ the following properties hold:

1. Identity: $\rho(x, x) = 0$.
2. Symmetry: $\rho(x, y) = \rho(y, x)$.
3. Triangle inequality: $\rho(x, z) \leq \rho(x, y) + \rho(y, z)$.

Remark 7.2. Pseudo-metric spaces differ from metric spaces in that a metric space satisfies the stronger *identity of indiscernables* property, $\rho(x, y) = 0 \iff x = y$. In other words, in a pseudo-metric space it is possible that $\rho(x, y) = 0$ for $x \neq y$, but that is not possible in a metric space. Additionally, note that the definition does not change if we allow $\rho \colon \Omega^2 \to \mathbb{R}$, because it is possible to deduce that ρ is non-negative from the other assumptions: $\forall x, y \in \Omega \colon 0 = \rho(x, x) \leq \rho(x, y) + \rho(y, x) = 2\rho(x, y)$.

Definition 7.3
Let (Ω, ρ) be a pseudo-metric space, let $X, C, P \subseteq \Omega$, and let $\varepsilon > 0$.

- We say that C *is an ε-cover of X* if
$$\text{for all } x \in X \text{ there exists } c \in C \text{ s.t.} \quad \rho(x, c) \leq \varepsilon.$$
- We say that C *is an internal ε-cover of X* if $C \subseteq X$ and C is an ε-cover of X.
- The *ε-cover number of X* is
$$N(X, \varepsilon, \rho) = \inf \{|C| : C \subseteq \Omega \wedge C \text{ is an } \varepsilon\text{-cover of } X.\}$$
- The *internal ε-cover number of X* is
$$N_{\text{in}}(X, \varepsilon, \rho) = \inf \{|C| : C \subseteq X \wedge C \text{ is an } \varepsilon\text{-cover of } X.\}$$
- We say that P *is an ε-packing of X* if $P \subseteq X$ and
$$\text{for all } x, y \in P, \quad \rho(x, y) > \varepsilon.$$
- The *ε-packing number of X* is
$$M(X, \varepsilon, \rho) = \sup \{|P| : P \subseteq X \wedge P \text{ is an } \varepsilon\text{-packing of } X\}.$$

When ρ is understood from context we will simply write $N(X, \varepsilon)$, $N_{\text{in}}(X, \varepsilon)$, and $M(X, \varepsilon)$.

All these numbers are closely related.

Proposition 7.4

Let (Ω, ρ) be a pseudo-metric space, let $X \subseteq \Omega$, and let $\varepsilon > 0$. Then

$$N(X, \varepsilon) \leq N_{\text{in}}(X, \varepsilon) \leq M(X, \varepsilon) \leq N\left(X, \frac{\varepsilon}{2}\right).$$

Definition 7.5

Let (Ω, ρ) be a pseudo-metric space, let $x \in \Omega$ and $\varepsilon \geq 0$. The *ε-ball centered at x* is $\mathbb{B}(x, \varepsilon) = \{y \in \Omega : \rho(x, y) \leq \varepsilon\}$.

Proof of Proposition 7.4. We prove this claim for the case where all terms are finite, otherwise the inequalities are trivially true.

The first inequality is immediate from the definitions of N and N_{in}.

For the second inequality, let $P \subseteq X$ be an ε-packing of X such that $|P| = M(X, \varepsilon)$. Then P is maximal in the sense that for any point $x \in X \setminus P$, the set $P \cup \{x\}$ is not an ε-packing of X. Namely, for every $x \in X$ there exists $p \in P$ such that $\rho(x, p) \leq \varepsilon$. Hence P is an internal ε-cover of X, and so $N_{\text{in}}(X, \varepsilon) \leq |P|$.

For the last inequality, let C be an $\frac{\varepsilon}{2}$ cover of X such that $|C| = N\left(X, \frac{\varepsilon}{2}\right)$. Let P be any ε-packing of X. We construct an injective function $f : P \to C$, and this completes the proof because it implies that $|P| \leq |C|$. For each $p \in P$, we define $f(p)$ to be an arbitrary $c \in C$ such that $p \in \mathbb{B}\left(c, \frac{\varepsilon}{2}\right)$; such a c always exists, because C is an $\frac{\varepsilon}{2}$ cover. To see that f is injective, assume for contradiction that there exist $p_1, p_2 \in P$ such that $p_1 \neq p_2$ and $f(p_1) = f(p_2) = c$. Then $\rho(p_1, p_2) \leq \rho(p_1, c) + \rho(c, p_2) \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon$, which is a contradiction to P being an ε-packing. □

7.2 Intuition for COvering and Packing Numbers

Example 7.6

Consider the metric space $(\mathbb{R}^d, \rho)$ where $\rho(x, y) = \|x - y\|$ and $\|\cdot\|$ is some ℓ_p norm. For any $\mathbb{B}(x, r) \subseteq \mathbb{R}^d$ of radius r centered at $x \in \mathbb{R}^d$, the Lebesgue measure $\lambda(\mathbb{B}(x, r))$ is given by the formula

$$\lambda(\mathbb{B}(x, r)) = C_{d,p} r^d$$

where $C_{d,p} \in \mathbb{R}$ is some constant that depends on p and d.

In this metric space, we can obtain the following bounds for the packing and covering numbers of a ball $\mathbb{B}(x, r)$:

- If $\varepsilon \leq r$ then $M(\mathbb{B}(x, r), \varepsilon) \leq \left(\frac{3r}{\varepsilon}\right)^d$. To see this, let $P \subseteq \mathbb{B}(x, r)$ be an ε-packing of $\mathbb{B}(x, r)$. Then for every $p_1, p_2 \in P$, the balls $\mathbb{B}\left(p_1, \frac{\varepsilon}{2}\right)$ and $\mathbb{B}\left(p_2, \frac{\varepsilon}{2}\right)$ are disjoint and contained in $\mathbb{B}\left(x, r + \frac{\varepsilon}{2}\right)$. Hence

  $$\lambda\left(\mathbb{B}\left(x, r + \frac{\varepsilon}{2}\right)\right) \geq \sum_{p \in P} \lambda\left(\mathbb{B}\left(p, \frac{\varepsilon}{2}\right)\right) = |P| \lambda\left(\mathbb{B}\left(0, \frac{\varepsilon}{2}\right)\right).$$

  So

  $$|P| \leq \frac{\lambda(\mathbb{B}(x, r + \frac{\varepsilon}{2}))}{\lambda(\mathbb{B}(0, \frac{\varepsilon}{2}))} = \frac{C_{d,p}\left(r + \frac{\varepsilon}{2}\right)^d}{C_{d,p}\left(\frac{\varepsilon}{2}\right)^d} \leq \left(\frac{r + \frac{r}{2}}{\frac{\varepsilon}{2}}\right)^d = \left(\frac{3r}{\varepsilon}\right)^d.$$

- $N\left(\mathbb{B}(x,r),\varepsilon\right) \geq \left(\frac{r}{\varepsilon}\right)^d$. Indeed, if $C \subseteq \Omega$ is an ε-cover of $\mathbb{B}(x,r)$, then

$$\lambda(\mathbb{B}(x,r)) \leq \sum_{c\in C} \lambda(\mathbb{B}(c,\varepsilon)) = |C|\,\lambda(\mathbb{B}(0,\varepsilon)),$$

and this implies that

$$|C| \geq \frac{\lambda(\mathbb{B}(x,r))}{\lambda(\mathbb{B}(0,\varepsilon))} = \frac{C_{d,p}r^d}{C_{d,p}\varepsilon^d} = \left(\frac{r}{\varepsilon}\right)^d.$$

Thus, for $\varepsilon \leq r$ we have that $\left(\frac{r}{\varepsilon}\right)^d \leq N\left(\mathbb{B}(x,r),\varepsilon\right) \leq M\left(\mathbb{B}(x,r),\varepsilon\right) \leq \left(\frac{3r}{\varepsilon}\right)^d$.

The above example demonstrates a fairly general phenomena. For sets in $\mathbb{R}^d$ bounded by a constant, the log of the packing and covering numbers, $\log(M(A,\varepsilon))$ and $\log(N(A,\varepsilon))$, tend to scale like $d\log\left(\frac{1}{\varepsilon}\right)$.[1] Perhaps surprisingly, a similar phenomena also holds for metric spaces of functions, if we replace the algebraic dimension with the VC dimension, as we will see in the next section.

7.3 Packing Numbers for VC Claasses

Definition 7.7
Let V be a vector space over $\mathbb{R}$. A *seminorm* is a function $p\colon V \to \mathbb{R}$ satisfying

- Triangle inequality. For all $u, v \in V$, $p(u+v) \leq p(u) + p(v)$.
- Absolute homogeneity. For all $v \in V$ and all $c \in \mathbb{R}$, $p(cv) = |c|\,p(v)$.

Remark 7.8. The definition of seminorm also implies non-negativity, namely $p(v) \geq 0$ for all $v \in V$. To see this, note that $0 = |0|\,p(u) = p(0u) = p(0) = p(v-v) \leq p(v) + p(-v) = 2p(v)$. A *norm* is a semi-norm where $p(v) = 0$ implies $v = 0$.

Definition 7.9
Let Ω be a set, let $\mathcal{F}$ be a class of functions $\Omega \to \mathbb{R}$, let $S = \{z_i\}_{i\in[m]} \in \Omega$ and let $p > 0$. The *empirical p-semi-norm of $\mathcal{F}$ with respect to S* is a function $\|\cdot\|_{S,p} : \mathcal{F} \to \mathbb{R}$ such that

$$\|f\|_{S,p} = \left(\frac{1}{m}\sum_{i\in[m]} |f(z_i)|^p\right)^{1/p}.$$

Additionally, for $p = \infty$ we define $\|f\|_{S,\infty} = \sup_{i\in[m]} |f(z_i)|$.

Notation 7.10
Let Ω be a set, let $\mathcal{F}$ be a class of functions $\Omega \to \mathbb{R}$, $S \in \Omega^m$, and $p \in (0,\infty]$. We write $\rho_{S,p}\colon \mathcal{F}^2 \to \mathbb{R}$ to denote $\rho_{S,p}(f_1, f_2) = \|f_1 - f_2\|_{S,p}$.

[1] One can in fact use this idea to define a notion of dimension in metric spaces that do not have an algebraic notion of dimension. This is called the Minkowski–Bouligand dimension and it is used, for example, to define the dimension of fractals.

Lemma 7.11
There exists a constant $c > 0$ as following. Let Ω be a set, let $\mathscr{F}$ be a class of functions $\Omega \to \{0, 1\}$ with $\mathsf{VC}(\mathscr{F}) = d < \infty$. Let $S \in \Omega^m$ and $p \in (0, \infty]$. Then $(\mathscr{F}, \rho_{S,p})$ is a pseudo-metric space, and for all $\varepsilon > 0$,

$$M(\mathscr{F}, \varepsilon) \leq \left(\frac{c}{\varepsilon^p} \log\left(\frac{1}{\varepsilon^p}\right)\right)^d .$$

Proof. To prove the lemma it suffices to show that

$$M\big(\mathscr{F}, \varepsilon, \rho_{S,1}\big) \leq \left(\frac{c}{\varepsilon} \log\left(\frac{1}{\varepsilon}\right)\right)^d . \tag{7.1}$$

To see that this suffices, note that because the functions $f \in \mathscr{F}$ are binary,

$$\|f\|_{S,p}^p = \frac{1}{m} \sum_{i \in [m]} |f(z_i)|^p = \frac{1}{m} \sum_{i \in [m]} |f(z_i)| = \|f\|_{S,1} ,$$

so $\|f\|_{S,p} = \varepsilon \iff \|f\|_{S,1} = \varepsilon^p$. That is, a subset $P \subseteq \mathscr{F}$ is an ε-packing of $\mathscr{F}$ in $(\mathscr{F}, \rho_{S,p})$ if and only if it is an ε^p-packing of $\mathscr{F}$ in $(\mathscr{F}, \rho_{S,1})$. Together with Equation 7.1 this implies that

$$M\big(\mathscr{F}, \varepsilon, rho_{S,p}\big) = M\big(\mathscr{F}, \varepsilon^p, \rho_{S,1}\big) \leq \left(\frac{c}{\varepsilon^p} \log\left(\frac{1}{\varepsilon^p}\right)\right)^d .$$

We now prove Equation 7.1. First, note that $M(\mathscr{F}, \varepsilon, \rho_{S,1}) \leq |\mathscr{F}|_S|$. To see this, assume for contradiction that there exists a suitable ε-packing P such that $|P| > |\mathscr{F}|_S|$. By the pigeonhole principle, this implies that there exist two functions $f, f' \in P$ such that $f \neq f'$ and $f|_S = f'|_S$. This implies that $\rho_{S,1}(f, f') = 0$ which is a contradiction to P being an ε-packing. Thus $M(\mathscr{F}, \varepsilon, \rho_{S,1})$ is finite and thus there exists a finite set $P \subseteq \mathscr{F}$ such that P is an ε-packing of $\mathscr{F}$ with respect to $\rho_{S,1}$ and $|P| = M = M(\mathscr{F}, \varepsilon, \rho_{S,1})$.

Second, note that for any $f, f' \in P$, and supposing that $z \sim \mathrm{Uni}(S)$,

$$\begin{aligned}\varepsilon < \rho_{S,1}(f, f') &= \frac{1}{m} \sum_{i \in [m]} \big|f(z_i) - f'(z_i)\big| = \frac{1}{m} \sum_{i \in [m]} \mathbb{1}\big(f(z_i) \neq f'(z_i)\big) \\ &= \mathop{\mathsf{P}}_{z}\big[f(z) \neq f'(z)\big] . \end{aligned} \tag{7.2}$$

Third, let $\widetilde{S} = \{\widetilde{z}_i\}_{i \in [m]} \in \Omega$ be a vector of elements chosen independently and uniformly from S, that is, $\widetilde{S} \sim (\mathrm{Uni}(S))^m$. Then for any $f, f' \in P$,

$$\begin{aligned}\mathop{\mathsf{P}}_{\widetilde{S}}\big[f|_{\widetilde{S}} = f'|_{\widetilde{S}}\big] &= \mathop{\mathsf{P}}_{\widetilde{S}}\left[\bigcap_{i \in [m]} \{f(\widetilde{z}_i) = f'(\widetilde{z}_i)\}\right] = \prod_{i \in [m]} \mathop{\mathsf{P}}_{\widetilde{S}}\big[f(\widetilde{z}_i) = f'(\widetilde{z}_i)\big] \\ &= \prod_{i \in [m]} \mathop{\mathsf{P}}_{z}\big[f(z) = f'(z)\big] = \prod_{i \in [m]} \Big(1 - \mathop{\mathsf{P}}_{z}\big[f(z) \neq f'(z)\big]\Big) \\ &< \prod_{i \in [m]} (1 - \varepsilon) \leq \mathrm{e}^{-\varepsilon m} . && \text{(From Equation 7.2.)}\end{aligned}$$

Fourth, fix $m \geq \frac{2}{\varepsilon} \log(M)$. Applying a union bound to the last inequality yields

$$\mathop{\mathsf{P}}_{\widetilde{S}}\big[\exists f, f' \in P \colon f|_{\widetilde{S}} = f'|_{\widetilde{S}}\big] \leq \binom{M}{2} \mathrm{e}^{-\varepsilon m} < M^2 \mathrm{e}^{-\varepsilon m} \leq 1.$$

Namely,

$$\mathop{\mathsf{P}}_{\widetilde{S}}\left[\forall f, f' \in P\colon f|_{\widetilde{S}} \neq f'|_{\widetilde{S}}\right] > 0,$$

nad in particular there exists an assignment to $\widetilde{S}$ such that $f|_{\widetilde{S}} \neq f'|_{\widetilde{S}}$ for all $f, f' \in P$. This implies that

$$\begin{aligned} M &\leq \left|\mathcal{F}|_{\widetilde{S}}\right| \leq \tau_{\mathcal{F}}(m) && \text{(definition of } \tau_{\mathcal{F}}\text{)} \\ &\leq \left(\frac{\varepsilon m}{d}\right)^d && \text{(Sauer's lemma)} \\ &\leq \left(\frac{2\mathrm{e}}{\varepsilon d}\log(M)\right)^d && \text{(choice of } m.\text{)} \end{aligned}$$

Equivalently, $M^{1/d} \leq \frac{2\mathrm{e}}{\varepsilon}\log\left(M^{1/d}\right)$. Finally, invoking **??** with $x = M^{1/d}$ and $y = \frac{2\mathrm{e}}{\varepsilon}$, we obtain

$$M^{1/d} \leq \frac{4\mathrm{e}}{\varepsilon}\log\left(\frac{2\mathrm{e}}{\varepsilon}\right),$$

and this completes the proof. □

7.4 Uniform Convergence via Covering Numbers

In Theorem 5.9 we saw that

$$\mathop{\mathsf{P}}_{S}[\exists h \in \mathcal{H}\colon |L_S(h) - L_{\mathcal{D}}(h)| > \varepsilon] \leq 4\tau_{\mathcal{H}}(2m)\exp\left(-\frac{\varepsilon^2 m}{8}\right). \tag{7.3}$$

Hence, if $\tau_{\mathcal{H}}$ grows sub-exponentially, then the class will satisfy uniform convergence and therefore the ERM algorithm learns successfully. Observe that we can think of $\tau_{\mathcal{H}}$ as a covering number. Specifically, assume the set of labels is $Y \subseteq \mathbb{R}$, $|Y| < \infty$, and fix $\varepsilon > 0$ such that for all $y, y' \in \mathcal{Y}$ if $y \neq y'$ then $|y - y'| > \varepsilon$. Let $X = \{x_i\}_{i\in[m]} \subseteq \mathcal{X}$. Then

$$\begin{aligned} |\mathcal{H}|_X| &= |\{h|_X : h \in \mathcal{H}\}| \\ &= \sup\left\{|H| : H \subseteq \mathcal{H} \wedge \forall h, h' \in \mathcal{H}, h \neq h' \implies h|_X \neq h'|_X\right\} \\ &= \sup\left\{|H| : H \subseteq \mathcal{H} \wedge \forall h, h' \in \mathcal{H}, h \neq h' \implies \rho_{X,\infty}(h, h') > \varepsilon\right\} \\ &= N_{\mathrm{in}}(\mathcal{H}, \varepsilon, \rho_{X,\infty}). \end{aligned}$$

Here we are doing a "sleight of hand" by considering X both as an unordered subset of $\mathcal{X}$ and an ordered sample (element of $\mathcal{X}^m$). Hence,

$$\tau_{\mathcal{H}}(m) = \sup_{\substack{X \subseteq \mathcal{X} \\ |X| = m}} |\mathcal{H}|_X| = \max_{X \in \mathcal{X}^m} N_{\mathrm{in}}(\mathcal{H}, \varepsilon, \rho_{X,\infty}).$$

This motivates the following definition.

Definition 7.12

Let Ω be a set, $\mathcal{F}$ be a set of functions $\Omega \to \mathbb{R}$, $p \in (0, \infty]$, and $\varepsilon > 0$. The *uniform ε-covering number for $\mathcal{F}$ with respect to the empirical p-semi-norm* is

$$N_p^{\mathrm{uniform}}(\mathcal{F}, \varepsilon, m) = \sup\left\{N_{\mathrm{in}}(\mathcal{F}, \varepsilon, \rho_{S,p})\colon S \in \Omega^m\right\}.$$

The following theorem shows that we can generalize the result of Equation 7.3 to other covering numbers beyond $\tau_{\mathcal{H}}$.

Theorem 7.13

Let $\mathcal{X}$ and $\mathcal{Y}$ be sets, let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \mathcal{Y}$, let $\ell\colon \mathcal{Y}^2 \to [0, c]$ be a loss function bounded by some positive $c \in \mathbb{R}$, and let $\mathcal{L}$ be the loss class of $\mathcal{H}$ with respect to ℓ. Then for any $\varepsilon > 0$, any $m \in \mathbb{N}$, and any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, if $S \sim \mathcal{D}^m$, then

$$\underset{S}{\mathsf{P}}[\exists h \in \mathcal{H}\colon\ |L_S(h) - L_{\mathcal{D}}(h)| > \varepsilon] \leq 4N_1^{\text{uniform}}\left(\mathcal{L}, \frac{\varepsilon}{8}, 2m\right)\exp\left(-\frac{\varepsilon^2 m}{32c^2}\right).$$

Proof. We modify the proof of Theorem 5.9. Recall that in the proof, we considered two independent samples $S = \{z_i\}_{i\in[m]}$, and $S' \sim \mathcal{D}^m$, $S' = \{z_i'\}_{i\in[m]}$, and showed that

$$\underset{S}{\mathsf{P}}[\exists h \in \mathcal{H}\colon\ |L_S(h) - L_{\mathcal{D}}(h)| > \varepsilon] \leq 2\underset{S,S'}{\mathsf{P}}\left[\exists h \in \mathcal{H}\colon \left|\frac{1}{m}\sum_{i\in[m]}\left(\ell_h(z_i) - \ell_h(z_i')\right)\right| \geq \frac{\varepsilon}{2}\right], \tag{7.4}$$

where for any $z = (x, y) \in \mathcal{X} \times \mathcal{Y}$, the function $\ell_h(z) = \ell(h(x), y)$ is the loss function in $\mathcal{L}$ associated with h. Let C be an internal $\frac{\varepsilon}{8}$-cover of $\mathcal{L}$ with respect to $\rho_{S\cup S',1}$ such that

$$|C| \leq N_1^{\text{uniform}}\left(\mathcal{L}, \frac{\varepsilon}{8}, 2m\right).$$

(A suitable cover C with this cardinality exists by the definition of N_1^{uniform}.) The idea of the next step is to replace each ℓ_h with an approximation $\ell_h' \in C$, that is, to quantize or approximate the functions in $\mathcal{L}$ using the coarser set of functions C. Specifically, for any $h \in \mathcal{H}$, let $\ell_h' \in C$ be a function such that $\rho_{S\cup S',1}\left(\ell_h, \ell_h'\right) \leq \frac{\varepsilon}{8}$. Then the expression in Equation 7.4 can be rewritten as follows.

$$\begin{aligned}
\frac{\varepsilon}{2} &\leq \left|\frac{1}{m}\sum_{i\in[m]}\left(\ell_h(z_i) - \ell_h(z_i')\right)\right| \qquad (7.5)\\
&= \left|\frac{1}{m}\sum_{i\in[m]}\left(\ell_h(z_i) \underbrace{-\ell_h'(z_i) + \ell_h'(z_i)}_{=0} \underbrace{-\ell_h'(z_i') + \ell_h'(z_i')}_{=0} - \ell_h(z_i')\right)\right|\\
&\leq \frac{1}{m}\sum_{i\in[m]}\left|\ell_h(z_i) - \ell_h'(z_i)\right| + \left|\frac{1}{m}\sum_{i\in[m]}\left(\ell_h'(z_i) - \ell_h'(z_i')\right)\right| + \frac{1}{m}\sum_{i\in[m]}\left|\ell_h(z_i') - \ell_h'(z_i')\right|\\
&\leq \frac{\varepsilon}{8} + \left|\frac{1}{m}\sum_{i\in[m]}\left(\ell_h'(z_i) - \ell_h'(z_i')\right)\right| + \frac{\varepsilon}{8},
\end{aligned}$$

and so Equation 7.5 implies $\left|\frac{1}{m}\sum_{i\in[m]}\left(\ell_h'(z_i) - \ell_h'(z_i')\right)\right| \geq \frac{\varepsilon}{4}$. Thus, if $\sigma \sim \mathrm{Uni}\left(\{\pm 1\}^m\right)$

$$\begin{aligned}
\underset{S,S'}{\mathsf{P}}\left[\exists h \in \mathcal{H}\colon\ |L_S(h) - L_{\mathcal{D}}(h)| > \frac{\varepsilon}{2}\right] &\leq 2\underset{S,S'}{\mathsf{P}}\left[\exists h \in \mathcal{H}\colon \left|\frac{1}{m}\sum_{i\in[m]}\left(\ell_h'(z_i) - \ell_h'(z_i')\right)\right| \geq \frac{\varepsilon}{4}\right]\\
&= 2\underset{S,S'}{\mathsf{E}}\left[\underset{\sigma}{\mathsf{P}}\left[\exists h \in \mathcal{H}\colon \left|\frac{1}{m}\sum_{i\in[m]}\sigma_i\left(\ell_h'(z_i) - \ell_h'(z_i')\right)\right| \geq \frac{\varepsilon}{4}\right]\right]. \qquad (7.6)
\end{aligned}$$

In the last equality we applied the symmetrization technique which we have seen in previous lectures, using the fact that $\ell_h'(z_i) - \ell_h'(z_i') \overset{d}{=} \left\{\sigma_i\left(\ell_h'(z_i) - \ell_h'(z_i')\right)\right\}_{i\in[m]}$.

Next, for each S, S',

$$\begin{aligned}
\mathop{\mathsf{P}}_{\sigma}\left[\bigcup_{h\in\mathcal{H}}\left\{\left|\frac{1}{m}\sum_{i\in[m]}\sigma_i\left(\ell'_h(z_i)-\ell'_h(z'_i)\right)\right|\geq\frac{\varepsilon}{4}\right\}\right] &= \mathop{\mathsf{P}}_{\sigma}\left[\bigcup_{\ell'\in C}\left\{\left|\frac{1}{m}\sum_{i\in[m]}\sigma_i\left(\ell'(z_i)-\ell'(z'_i)\right)\right|\geq\frac{\varepsilon}{4}\right\}\right] && (\forall h\in\mathcal{H}:\ell'_h\in C.)\\
&\leq \sum_{\ell'\in C}\mathop{\mathsf{P}}_{\sigma}\left[\left|\frac{1}{m}\sum_{i\in[m]}\sigma_i\left(\ell'(z_i)-\ell'(z'_i)\right)\right|\geq\frac{\varepsilon}{4}\right] && \text{(union bound)}\\
&= \sum_{\ell'\in C}\mathop{\mathsf{P}}_{\sigma}\left[\left|\frac{1}{m}\sum_{i\in[m]}Z_i\right|\geq\frac{\varepsilon}{4}\right] && (\text{let } Z_i=\sigma_i\left(\ell'(z_i)-\ell'(z'_i)\right))\\
&\leq \sum_{\ell'\in C}2\exp\left(-\frac{2\,(\varepsilon/4)^2\,m}{(2c)^2}\right) && \text{(Hoeffding's)}\\
&= 2\,|C|\exp\left(-\frac{\varepsilon^2 m}{32c^2}\right)\\
&\leq 2N_1^{\text{uniform}}\left(\mathcal{L},\frac{\varepsilon}{8},2m\right)\exp\left(-\frac{\varepsilon^2 m}{32c^2}\right). && (7.7)
\end{aligned}$$

Combining Equation 7.4, Equation 7.6, Equation 7.7 yields the theorem. □

7.5 Chaining

Chaining is a technique for bounding the Rademacher average. Roughly speaking, we can think of it as a sophisticated union bound.

Let $A\subseteq\mathbb{R}^m$ be a finite set such that $\|a\|_2\leq r$ for all $a\in A$. We are interested in upper-bounding the Rademacher average

$$\mathsf{Rad}(A)=\mathop{\mathsf{E}}_{\sigma}\left[\sup_{a\in A}\frac{1}{m}\sum_{i\in[m]}\sigma_i a_i\right].$$

Method 0: Union Bound

For any $t>0$,

$$\begin{aligned}
\mathop{\mathsf{P}}_{\sigma}\left[\sup_{a\in A}\frac{1}{m}\sum_{i\in[m]}a_i\sigma_i>t\right] &\leq |A|\sup_{a\in A}\mathop{\mathsf{P}}_{\sigma}\left[\frac{1}{m}\sum_{i\in[m]}a_i\sigma_i>t\right] && \text{(union bound)}\\
&\leq 2\,|A|\exp\left(-\frac{mt^2}{2r^2}\right). && \text{(Hoeffding's inequality)}
\end{aligned}$$

Hence

$$\begin{aligned}
\mathsf{Rad}(A) &= \mathop{\mathsf{E}}_{\sigma}[Z] && (Z=\sup_{a\in A}\frac{1}{m}\sum_{i\in[m]}\sigma_i a_i)\\
&= \mathsf{P}[Z\leq t]\,\mathsf{E}[Z\mid Z\leq t]+\mathsf{P}[Z>t]\,\mathsf{E}[Z\mid Z>t]\\
&\leq t+\mathsf{P}[Z>t]\,\frac{r}{\sqrt{m}} && (Z\leq\sup_{a,\sigma}\frac{\langle a,\sigma\rangle}{m}\leq\sup_{a,\sigma}\frac{\|a\|_2\,\|\sigma\|_2}{m}\leq\frac{r}{\sqrt{m}}\ \mathbb{P}\text{-a.e.})
\end{aligned}$$

$$\leq t + \frac{2r}{\sqrt{m}} |A| \exp\left(-\frac{mt^2}{2r^2}\right).$$

Choosing $t = \frac{2r}{\sqrt{m}}$, we obtain

$$\mathsf{Rad}(A) \leq \frac{2r}{\sqrt{m}} + \frac{2r}{\sqrt{m}} |A| \exp(-2) = O\left(\frac{|A| r}{\sqrt{m}}\right).$$

This bound is very weak, but it was also very simple to prove – the tools we used were the union bound and Hoeffding's inequality.

7.5.1 Method 1: Massart's Lemma

In Unit 6 we proved Massart's lemma, which states that

$$\mathsf{Rad}(A) \leq \frac{r\sqrt{2\log(|A|)}}{m}.$$

This is considerably stronger than the bound from Method 0 above. The proof we presented for Massart's lemma used the Maximal Inequality lemma, which was based on the Chernoff method for proving concentration bounds. However, in the homework we will see that Massart's lemma can also be proved in a manner very similar to Method 0 above. From that point of view, Massart's lemma is basically a clever application of the union bound with Hoeffding's inequality.

7.5.2 Method 2: ε-Cover

If $|A| = \infty$ (or A is finite but very large) then the bound from Massart's lemma is not useful. We can overcome this problem by approximating the large set $|A|$ with a much smaller set C, which is an ε-cover for A. On the one hand, C is small and so Massart's lemma gives a good bound on $\mathsf{Rad}(C)$, and on the other hand C is a "good enough" approximation of A, such that a good bound for $\mathsf{Rad}(C)$ implies a good bound for $\mathsf{Rad}(A)$.

More fully, let $C \subseteq A$ be an internal ε-cover of A such that $|C| = N_{\text{in}}(A, \varepsilon, \rho)$, where $\rho(x, y) = \|x - y\|_2$. For each $\pi \in A$, let $\pi(a) = c$ for $c \in C$ such that $\rho(a, c) \leq \varepsilon$. By linearity of the inner product, for any $a \in A$ and $\sigma \in \{\pm 1\}^m$,

$$\langle \sigma, a \rangle = \langle \sigma, \pi(a) \rangle + \langle \sigma, a - \pi(a) \rangle \leq \langle \sigma, \pi(a) \rangle + \|\sigma\|_2 \|a - \pi(a)\|_2 \leq \langle \sigma, \pi(a) \rangle + \varepsilon\sqrt{m}. \tag{7.8}$$

Therefore

$$\begin{aligned}
\mathsf{Rad}(A) &= \mathop{\mathsf{E}}_{\sigma}\left[\sup_{a \in A} \frac{\langle \sigma, a \rangle}{m}\right] \\
&\leq \mathop{\mathsf{E}}_{\sigma}\left[\sup_{a \in A} \frac{\langle \sigma, \pi(a) \rangle + \varepsilon\sqrt{m}}{m}\right] && \text{(by Equation 7.8)} \\
&= \frac{\varepsilon}{\sqrt{m}} + \mathop{\mathsf{E}}_{\sigma}\left[\sup_{a \in A} \frac{\langle \sigma, \pi(a) \rangle}{m}\right] \\
&= \frac{\varepsilon}{\sqrt{m}} + \mathsf{Rad}(C) \\
&\leq \frac{\varepsilon}{\sqrt{m}} + \frac{r\sqrt{2\log(|C|)}}{m} && \text{(Massart's lemma)} \\
&= \frac{\varepsilon}{\sqrt{m}} + \frac{r\sqrt{2\log(N_{\text{in}}(A, \varepsilon, \rho))}}{m}. && (7.9)
\end{aligned}$$

If $\varepsilon = 0$ this bound is the same as in Method 1. However, if we choose a value $\varepsilon > 0$ that minimizes Equation 7.9, we can get a better bound.

Method 3: Chaining

Instead of committing to a particular ε-cover, chaining is a technique that uses a countable number of ε-covers with $\varepsilon \to 0$. We can think of this as a recursive application of Method 2, where we first approximate A by a coarse ε-cover, and then repeatedly improve the approximation with finer and finer covers. Formally, the result is as follows.

Theorem 7.14 (Dudley)
Let $r > 0$ and let $A \subseteq \mathbb{R}^m$ be a set such that $\|a\|_2 \leq r$ for all $a \in A$. Then

$$\mathsf{Rad}(A) \leq \frac{12}{m} \int_0^r \sqrt{\log(N(A, \varepsilon, \rho))} \mathrm{d}\varepsilon.$$

Furthermore, if $r \geq 1$ then

$$\mathsf{Rad}(A) \leq \frac{12r}{m} \int_0^r \sqrt{\log(N(A, \varepsilon, \rho))} \mathrm{d}\varepsilon.$$

Remark 7.15.

- In particular, $r \geq 1$ if A is a set of boolean vectors, for instance if A is the 0-1 loss class for some class of hypotheses.
- A small modification of the proof yields a bound of the form

  $$\mathsf{Rad}(A) \leq \inf_{\alpha \in [0, \frac{r}{2}]} \left(4\alpha + \frac{12}{m} \int_\alpha^r \sqrt{\log(N(A, \varepsilon, \rho))} \mathrm{d}\varepsilon\right).$$

  In some cases, this bound has the advantage that the integral $\int_\alpha^r$ converges even though $\int_0^r$ does not converge (e.g., if $\sqrt{\log(N(A, \varepsilon, \rho))} > \frac{1}{\varepsilon}$ in some neighborhood of 0).

Proof of Theorem 7.14. Let $\rho(x, y) = \|x - y\|_2$. For any $k \geq 0$, let $C_k \subseteq \mathbb{R}^n$ be an ε_k-cover of A where $\varepsilon_k = \frac{r}{2^k}$ and $|C_k| = N(A, \varepsilon_k, \rho)$. In particular, we can take $C_0 = \{0\}$, because $\rho(a, 0) = \|a\|_2 \leq r = \varepsilon_0$ for all $a \in A$. For any k and any $a \in A$, let $\pi_k(a) = c$ such that $c \in C_k$ and $\rho(a, c) \leq \varepsilon_k$. Furthermore, let $\Delta_k(a) = \pi_k(a) - \pi_{k-1}(a)$. For any $a \in A$,

$$a = \pi_0(a) + \sum_{k \in \mathbb{N}} \Delta_k(a) = \sum_{k \in \mathbb{N}} \Delta_k(a).$$

Hence,

$$\begin{aligned} \mathsf{Rad}(A) &= \mathop{\mathsf{E}}_{\sigma}\left[\sup_{a \in A} \frac{\langle \sigma, a \rangle}{m}\right] \\ &= \mathop{\mathsf{E}}_{\sigma}\left[\sup_{a \in A} \frac{\langle \sigma, \sum_{k \in \mathbb{N}} \Delta_k(a) \rangle}{m}\right] \\ &\leq \sum_{k \in \mathbb{N}} \mathop{\mathsf{E}}_{\sigma}\left[\sup_{a \in A} \frac{\langle \sigma, \Delta_k(a) \rangle}{m}\right] \\ &= \sum_{k \in \mathbb{N}} \mathsf{Rad}(\Delta_k), \end{aligned} \tag{7.10}$$

where $\Delta_k = \{\Delta_k(a) \colon a \in A\}$. Notice that

$$\begin{aligned}|\Delta_k| &= |\{\pi_k(a) - \pi_{k-1}(a) \colon a \in A\}| \\ &\leq |\{\pi_k(a) \colon a \in A\}|\,|\{\pi_{k-1}(a)\, colona \in A\}| \\ &\leq N(A, \varepsilon_k, \rho) N(A, \varepsilon_{k-1}, \rho) \\ &\leq N(A, \varepsilon_k, \rho)^2,\end{aligned}$$

and for all $a \in A$,

$$\|\Delta_k(a)\|_2 \leq \|\pi_k(a) - a\|_2 + \|a - \pi_{k-1}(a)\|_2 \leq 3\varepsilon_k.$$

Thus by Massart's lemma,

$$\mathsf{Rad}(\Delta_k) \leq \frac{3\varepsilon_k \sqrt{2\log\left(N(A, \varepsilon_k, \rho)^2\right)}}{m} = \frac{6\varepsilon_k \sqrt{\log(N(A, \varepsilon_k, \rho))}}{m}.$$

Plugging into Equation 7.10 yields

$$\begin{aligned}\mathsf{Rad}(A) &\leq \sum_{k \in \mathbb{N}} \frac{6\varepsilon_k \sqrt{\log(N(A, \varepsilon_k, \rho))}}{m} \\ &= \frac{6}{m} \sum_{k \in \mathbb{N}} \varepsilon_k \log(N(A, \varepsilon_k, \rho)) \\ &= \frac{12}{m} \sum_{k \in \mathbb{N}} \left(\frac{r}{2^k} - \frac{r}{2^{k+1}}\right) \sqrt{\log\left(N\left(A, \frac{r}{2^k}, \rho\right)\right)} \\ &\leq \frac{12}{m} \int_0^r \sqrt{\log(N(A, \varepsilon, \rho))} \mathrm{d}\varepsilon.\end{aligned}$$

In the last inequality we used trivial bounds on Riemann sums. This completes the proof of the first part of the statement.

For the second part of the statement, notice that if $r \geq 1$ then $N\left(A, \frac{r}{2^k}, \rho\right) \leq N\left(A, \frac{1}{2^k}, \rho\right)$ and so Lemma 7.11 is upper bounded by

$$\frac{12r}{m} \int_0^1 \sqrt{\log(N(A, \varepsilon, \rho))} \mathrm{d}\varepsilon.$$

□

7.6 Learning Bounds via Chaining

We now have all the ingredients to prove the tight sample complexity bound that appears in Theorem 5.12.

Proposition 7.16

Let $\mathcal{X}$ be a set, and let $\mathcal{H}$ be a class of functions $\mathcal{X} \to \{0, 1\}$ with $\mathsf{VC}(\mathcal{H}) = d$. Then $\mathcal{H}$ is agnostic PAC learnable with sample complexity

$$m = O\left(\frac{d + \log\left(\frac{1}{\delta}\right)}{\varepsilon^2}\right).$$

Proof. From Theorem 6.11, the hypothesis h selected by an ERM algorithm that uses a sample of size m satisfies

$$L_{\mathfrak{D}}(h) \leq L_{\mathfrak{D}}(\mathcal{H}) + 2\mathsf{Rad}_{\mathfrak{D}_x, m}(\mathcal{H}) + \sqrt{\frac{2\log\left(\frac{2}{\delta}\right)}{m}}. \tag{7.11}$$

Let S be the sample $A = \{h(S) \colon h \in \mathcal{H}\}$. Then $\|a\|_2 \leq \sqrt{m}$ for all $a \in A$. Let $\rho(x, y) = \|x - y\|_2$. Dudley's theorem implies

$$\mathsf{Rad}_{\mathfrak{D}_x, m}(\mathcal{H}) \leq \frac{12}{\sqrt{m}} \int_0^1 \sqrt{\log(N(A, \varepsilon, \rho))} \mathrm{d}\varepsilon = \frac{12}{\sqrt{m}} \int_0^1 \sqrt{\log\bigl(N\bigl(\mathcal{H}, \varepsilon, \rho_{S,2}\bigr)\bigr)} \mathrm{d}\varepsilon. \tag{7.12}$$

Note that

$$\begin{aligned}
\log\bigl(N\bigl(\mathcal{H}, \varepsilon, \rho_{S,2}\bigr)\bigr) &\leq \log\bigl(M\bigl(\mathcal{H}, \varepsilon, \rho_{S,2}\bigr)\bigr) && \text{(Proposition 7.4)} \\
&\leq d \log\left(\frac{4\mathrm{e}}{\varepsilon^2} \log\left(\frac{2\mathrm{e}}{\varepsilon^2}\right)\right) && \text{(Lemma 7.11)} \\
&\leq d \log\left(\frac{8\mathrm{e}}{\varepsilon^4}\right) = d \left(\log(8\mathrm{e}) + 4 \log\left(\frac{1}{\varepsilon}\right)\right). && (7.13)
\end{aligned}$$

Combining Theorem 7.13 and Theorem 7.14 and using numerical integration yields

$$\mathsf{Rad}_{\mathfrak{D}_x, m}(\mathcal{H}) \leq 12 \sqrt{\frac{d}{m}} \int_0^1 \sqrt{\log(8\mathrm{e}) + 4 \log\left(\frac{1}{\varepsilon}\right)} \mathrm{d}\varepsilon \leq 31 \sqrt{\frac{d}{m}}.$$

Plugging this into Definition 7.12, we obtain

$$L_{\mathfrak{D}}(h) \leq L_{\mathfrak{D}}(\mathcal{H}) + 62 \sqrt{\frac{d}{m}} + \sqrt{\frac{2 \log\left(\frac{2}{\delta}\right)}{m}} \leq L_{\mathfrak{D}}(\mathcal{H}) + O\left(\sqrt{\frac{d + \log\left(\frac{1}{\delta}\right)}{m}}\right).$$

Thus taking m as in the statement is sufficient to ensure that $L_{\mathfrak{D}}(h) \leq L_{\mathfrak{D}}(\mathcal{H}) + \varepsilon$. □

8 Boosting

Boosting is an algorithmic paradigm that grew out of a theoretical question and became a very practical machine learning tool. The boosting approach uses a generalization of linear predictors to address two major issues that have been raised earlier in the book. The first is the bias-complexity tradeoff. We have seen that the error of an ERM learner can be decomposed into a sum of approximation error and estimation error. The more expressive the hypothesis class the learner is searching over, the smaller the approximation error is, but the larger the estimation error becomes. A learner is thus faced with the problem of picking a good tradeoff between these two considerations. The boosting paradigm allows the learner to have smooth control over this tradeoff. The learning starts with a basic class (that might have a large approximation error), and as it progresses the class that the predictor may belong to grows richer.

The second issue that boosting addresses is the computational complexity of learning. For many interesting concept classes the task of finding an ERM hypothesis may be computationally infeasible. A boosting algorithm amplifies the accuracy of weak learners. Intuitively, one can think of a weak learner as an algorithm that uses a simple “rule of thumb” to output a hypothesis that comes from an easy-to-learn hypothesis class and performs just slightly better than a random guess. When a weak learner can be implemented efficiently, boosting provides a tool for aggregating such weak hypotheses to approximate gradually good predictors for larger, and harder to learn, classes.

In this chapter we will describe and analyze a practically useful boosting algorithm, AdaBoost (a shorthand for Adaptive Boosting). The AdaBoost algorithm outputs a hypothesis that is a linear combination of simple hypotheses. In other words, AdaBoost relies on the family of hypothesis classes obtained by composing a linear predictor on top of simple classes. We will show that AdaBoost enables us to control the tradeoff between the approximation and estimation errors by varying a single parameter.

8.1 Weak Learnability

We return to the non-agnostic setting for now.

Definition 8.1 (Weak Learnability)
Let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty sets, let $\mathcal{F}$ be the set of all functions $\mathcal{X} \to \mathcal{Y}$, let $\mathcal{H} \subseteq \mathcal{F}$ be a class of functions, and let $f \in \mathcal{F}$ be a target function. We say that a (possibly randomized) algorithm A is a *weak PAC learner for* $\mathcal{H}$ if there exists a sample complexity function $m\colon \left(0, \frac{1}{2}\right) \times [0,1] \to \mathbb{N}$ such that for every precision parameter $\varepsilon \in \left(0, \frac{1}{2}\right)$, every confidence parameter $\delta \in (0,1)$, and every distribution $\mathcal{D}$ over $\mathcal{X}$, if A receives as input the parameters ε and δ and a sample S of size $m = m(\varepsilon, \delta$ such that $S = \{(x_i, f(x_i))\}_{i \in [m]}$ where $\{x_i\}_{i \in [m]} \overset{\text{i.i.d.}}{\sim} \mathcal{D}$, then A halts and outputs a hypothesis $h \in \mathcal{F}$ such that

$$\underset{S}{\mathsf{P}}\left[L_{\mathcal{D}}(h) \le \frac{1}{2} - \varepsilon\right] \ge 1 - \delta.$$

We say that a class $\mathcal{H} \subseteq \mathcal{F}$ is *weakly PAC learnable* if there eixsts an algorithm that is a weak PAC learner for $\mathcal{H}$.

This definition is almost identical to the definition of PAC learning, which here we will call strong learning, with one crucial difference: Strong learnability implies the ability to find an arbitrarily good classifier (with error rate at most ε, for arbitrarily small $\varepsilon > 0$). In weak learnability, however, we only need to output a hypothesis whose error rate is at most $\frac{1}{2} - \varepsilon$, namely, whose error rate is slightly better than what a random labeling would give us. The hope is that it may be easier to come up with efficient weak learners than with efficient (full) PAC learners.

The fundamental theorem of learning says that if a hypothesis class has a VC dimension d, then the sample complexity of PAC learning $\mathcal{H}$ satisfies $m(\varepsilon, \delta) = O\left(\frac{d+\log(1/\delta)}{\varepsilon}\right)$. Applying this with $\varepsilon \leftarrow \frac{1}{2} - \varepsilon$, we immediately obtain that if $d = \infty$ then $\mathcal{H}$ is not ε-weak-learnable. Thus from the statistical perspective (i.e., if we ignore computational complexity), weak learnability is also characterized by the VC dimension of $\mathcal{H}$ and therefore is just as hard as PAC (strong) learning. However, when we do consider computational complexity, the potential advantage of weak learning is that maybe there is an algorithm that satisfies the requirements of weak learning and can be implemented efficiently.

One possible approach is to take a "simple" hypothesis class, denoted B, and to apply ERM with respect to B as the weak learning algorithm. For this to work, we need that B will satisfy two requirements:

- ERM_B is efficiently implementable.
- For every sample that is labeled by some hypothesis from $\mathcal{H}$, any $\mathrm{ERM}_{\mathcal{B}}$ hypothesis will have an error of at most $\frac{1}{2} - \varepsilon$.

Then, the immediate question is whether we can boost an efficient weak learner into an efficient strong learner. In the next section we will show that this is indeed possible.

8.2 AdaBoost

Algorithm 1 AdaBoost

```
Input: Training set $S = \{(x_i, y_i)\}_{i \in [m]}$
Input: Weak learner WL
Input: Horizon $T \in \mathbb{N}$
Output: Hypothesis $h\colon \mathcal{X} \to \{0, 1\}$
  $\mathcal{D}^{(1)} \leftarrow \mathrm{Uni}(S)$
  for $t \in [T]$ do
    $h_t \leftarrow \mathrm{WL}\left(\mathcal{D}^{(t)}, S\right)$
    $\varepsilon_t \leftarrow \mathcal{D}^{(t)}(\{(x, y) \in S : y \neq h_t(x)\})$
    $w_t \leftarrow \frac{1}{2} \log\left(\frac{1}{\varepsilon_t} - 1\right)$
    for $i \in [m]$ do
      $\mathcal{D}^{(t+1)}(\{i\}) \leftarrow \frac{\mathcal{D}^{(t)}(\{i\}) \exp(-w_t y_i h_t(x_i))}{\sum_{j \in [m]} \mathcal{D}^{(t)}(\{j\}) \exp(-w_t y_j h_t(x_j))}$
  return $x \mapsto \mathrm{sign}\left(\sum_{t \in [T]} w_t h_t(x)\right)$
```

The following theorem shows that the training error of the output hypothesis decreases exponentially fast with the number of boosting rounds.

Theorem 8.2

Let S be a training set and assume that at each iteration of AdaBoost, the weak learner returns a hypothesis for which $\varepsilon_t \leq \frac{1}{2} - \varepsilon$. Then the training error of the output hypothesis h is at most

$$L_S(h) = \frac{1}{m} \sum_{i \in [m]} \mathbb{1}(h_t(x_i) \neq y_i) \leq \exp\left(-2\varepsilon^2 T\right).$$

Proof. For each t, denote $f_t = \sum_{s \in [t]} w_s h_s$. Therefore, the output of AdaBoost is f_T. In addition, denote

$$Z_t = \frac{1}{m} \sum_{i \in [m]} \mathrm{e}^{-y_i f_t(x_i)}.$$

Note that for any hypothesis we have $\mathbb{1}(h(x) \neq y) \leq \mathrm{e}^{-yh(x)}$. Therefore, $L_S(f_T) \leq Z_T$, so it suffices to show that $Z_T \leq \mathrm{e}^{-2\varepsilon^2 T}$. To upper bound Z_T we rewrite it as

$$Z_T = \frac{Z_T}{Z_0} = \prod_{t \in [T]} \frac{Z_t}{Z_{t-1}},$$

where we used the fact that $f_0 = 0$ so $Z_0 = 1$. Thus it suffices to show that for every t,

$$\frac{Z_{t+1}}{Z_t} \leq \mathrm{e}^{-2\varepsilon^2}.$$

One shows by induction that

$$\mathcal{D}^{(t+1)}(\{i\}) = \frac{\mathrm{e}^{-y_i f_t(x_i)}}{\sum_{j \in [m]} \mathrm{e}^{-y_j f_t(x_j)}}.$$

Hence

$$\begin{aligned}
\frac{Z_{t+1}}{Z_t} &= \frac{\sum_{i \in [m]} \mathrm{e}^{-y_i f_{t+1}(x_i)}}{\sum_{j \in [m]} \mathrm{e}^{-y_j f_t(x_j)}} \\
&= \frac{\sum_{i \in [m]} \mathrm{e}^{-y_i f_t(x_i)} \mathrm{e}^{-y_i w_{t+1} h_{t+1}(x_i)}}{\sum_{j \in [m]} \mathrm{e}^{-y_j f_t(x_j)}} \\
&= \sum_{i \in [m]} \mathcal{D}^{(t+1)}(\{i\})\, \mathrm{e}^{-y_i w_{t+1} h_{t+1}(x_i)} \\
&= \mathrm{e}^{-w_{t+1}} \mathcal{D}^{(t+1)}(\{i : y_i h_{t+1}(x_i) = 1\}) + \mathrm{e}^{w_{t+1}} \mathcal{D}^{(t+1)}(\{i : y_i h_{t+1}(x_i) = -1\}) \\
&= \mathrm{e}^{-w_{t+1}} (1 - \varepsilon_{t+1}) + \mathrm{e}^{w_{t+1}} \varepsilon_{t+1} \\
&= \frac{1 - \varepsilon_{t+1}}{\sqrt{\frac{1}{\varepsilon_{t+1}} - 1}} + \varepsilon_{t+1} \sqrt{\frac{1}{\varepsilon_{t+1}} - 1} \\
&= (1 - \varepsilon_{t+1}) \sqrt{\frac{\varepsilon_{t+1}}{1 - \varepsilon_{t+1}}} + \varepsilon_{t+1} \sqrt{\frac{1 - \varepsilon_{t+1}}{\varepsilon_{t+1}}} \\
&= 2\sqrt{\varepsilon_{t+1}(1 - \varepsilon_{t+1})}.
\end{aligned}$$

By our assumption, $\varepsilon_{t+1} \leq \frac{1}{2} - \varepsilon$. By monotonicity of the function $x \mapsto x(1-x)$ over $\left[0, \frac{1}{2}\right]$,

$$2\sqrt{\varepsilon_{t+1}(1 - \varepsilon_{t+1})} \leq 2\sqrt{\left(\frac{1}{2} - \varepsilon\right)\left(\frac{1}{2} + \gamma\right)} = \sqrt{1 - 4\varepsilon^2} \leq \mathrm{e}^{-\frac{4\varepsilon^2}{2}} = \mathrm{e}^{-2\varepsilon^2}.$$

□

Each iteration of AdaBoost involves O(m) operations as well as a single call to the weak learner. Therefore, if the weak learner can be implemented efficiently (as happens in the case of ERM with respect to decision stumps) then the total training process will be efficient.

Remark 8.3. The proof of the theorem assumes that at each iteration of AdaBoost, the weak learner returns a hypothesis with weighted sample error of at most $\frac{1}{2} - \varepsilon$. According to the definition of a weak learner, it can fail with probability δ. By union bound, the probability that the weak learner will not fail at all of the iterations is at least

$1 - \delta T$. The dependence of the sample complexity on δ can always be logarithmic in $\frac{1}{\delta}$, and therefore invoking the weak learner with a very small δ is not problematic. We can therefore assume that δT is also small. Furthermore, since the weak learner is only applied with distributions over the training set, in many cases we can implement the weak learner so that it will have a zero probability of failure (i.e., $\delta = 0$).

8.3 Linear Combinations of Base Hypotheses

The output of AdaBoost will be a member of the following class:

$$L(B, T) = \left\{x \mapsto \operatorname{sign}\left(\sum_{t \in [T]} w_t h_t(x)\right) : w \in \mathbb{R}^T, h_t \in B \text{ for all } t \in [T]\right\}$$

In this section we analyze the estimation error of $L(B, T)$ by bounding the VC dimension of $L(B, T)$ in terms of the VC dimension of B and T. We will show that, up to logarithmic factors, the VC dimension of $L(B, T)$ is bounded by T times the VC dimension of B. It follows that the estimation error of AdaBoost grows linearly with T. On the other hand, the empirical risk of AdaBoost decreases with T. In fact, as we demonstrate later, T can be used to decrease the approximation error of $L(B, T)$. Therefore, the parameter T of AdaBoost enables us to control the bias-complexity tradeoff.

Lemma 8.4

Let B be a base class, $T \geq 3$, and $\mathsf{VC}(B) \geq 3$. Then

$$\mathsf{VC}(L(B, T)) \leq T\left(\mathsf{VC}(B) + 1\right)\left(3\log(T\left(\mathsf{VC}(B) + 1\right)) + 2\right).$$

Proof. Denote $d = \mathsf{VC}(B)$. Let $C = \{x_i\}_{i \in [m]}$ be a set shattered by $L(B, T)$. Each labeling of C by $h \in L(B, T)$ is obtained by first choosing $h_1, \ldots, h_T \in B$ and then applying a halfspace hypothesis over the vector $(h_1(x), \ldots, h_T(x))$. By Sauer's lemma, there are at most $\left(\frac{em}{d}\right)^d$ different dichotomies induced by B over C. Therefore, we need to choose T hypotheses, out of at most $\left(\frac{em}{d}\right)^T$ different hypotheses. There are at most $\left(\frac{em}{d}\right)^{dT}$ ways to do it. Next, for each such choice, we apply a linear predictor, which yields at most $\left(\frac{em}{T}\right)^T$ dichotomies. Therefore, the number of dichotomies we can construct is upper bounded by

$$\left(\frac{em}{d}\right)^T\left(\frac{em}{T}\right)^T \leq m^{(d+1)T}$$

where we use the assumption that $d \geq 3$ and $T \geq 3$. Since we assume C is shattered, we must have

$$2^m \leq \left(\frac{em}{d}\right)^T\left(\frac{em}{T}\right)^T \leq m^{(d+1)T}.$$

Therefore

$$m \leq \log(m)\,\frac{(d+1)T}{\log(2)}.$$

A necessary condition is

$$m \leq (d+1)T\left(3\log((d+1)T) + 2\right)$$

as desired. □

9 Non-Uniform Learning

The notions of PAC learnability discussed so far allow the sample sizes to depend on the accuracy and confidence parameters, but they are uniform with respect to the labeling rule and the underlying data distribution. Consequently, classes that are learnable in that respect are limited (they must have a finite VC-dimension). In this chapter we consider more relaxed, weaker notions of learnability. We discuss the usefulness of such notions and provide characterization of the concept classes that are learnable using these definitions.

9.1 Nonuniform Learnability

Nonuniform learnability allows the sample size to be nonuniform with respect to the different hypotheses with which the learner is competing.

Definition 9.1 (Competitiveness)
A (possibly randomized) hypothesis h is (ε, δ)-competitive with another hypothesis h' if

$$\mathsf{P}\big[L_{\mathcal{D}}(h) \le L_{\mathcal{D}}(h') + \varepsilon\big] \ge 1 - \delta.$$

In PAC learnability, this notion of "competitiveness" is not very useful, as we are looking for a hypothesis with an absolute low risk (in the realizable case) or with a low risk compared to the minimal risk achieved by hypotheses in our class (in the agnostic case). Therefore, the sample size depends only on the accuracy and confidence parameters. In nonuniform learnability, however, we allow the sample size to be of the form $m(\varepsilon, \delta, h)$; namely, it depends also on the h with which we are competing. Formally,

Definition 9.2
Let $\mathcal{X}$ and $\mathcal{Y}$ be nonempty sets, let $\mathcal{F}$ be the set of all functions $\mathcal{X} \to \mathcal{Y}$, and let $\mathcal{H} \subseteq \mathcal{F}$ be a class of functions. We say that a (possibly randomized) algorithm A is an *nonuniform PAC learner for* $\mathcal{H}$ if there exists a sample complexity function $m\colon (0,1)^2 \times \mathcal{H} \to \mathbb{N}$ such that for every precision parameter $\varepsilon \in (0,1)$, every confidence parameter $\delta \in (0,1)$, every comparison function $h_c \in \mathcal{H}$, and every distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, if A receives as input the parameters ε and δ and a sample S of size $m = m(\varepsilon, \delta, h_c)$ such that $S = \{(x_i, y_i)\}_{i \in [m]}$ where for each $i \in [m]$, the pair (x_i, y_i) is sampled independently from $\mathcal{D}$, then A halts and outputs a hypothesis $h \in \mathcal{F}$ such that

$$\underset{S}{\mathsf{P}}[L_{\mathcal{D}}(h) \le L_{\mathcal{D}}(h_c) + \varepsilon] \ge 1 - \delta.$$

In both types of learnability, we require that the output hypothesis will be (ε, δ)-competitive with every other hypothesis in the class. But the difference between these two notions of learnability is the question of whether the sample size m may depend on the hypothesis h_c to which the error of h is compared. Note that that nonuniform learnability is a relaxation of agnostic PAC learnability. That is, if a class is agnostic PAC learnable then it is also nonuniformly learnable.

9.2 Structural Risk Minimization

So far, we have encoded our prior knowledge by specifying a hypothesis class $\mathcal{H}$, which we believe includes a good predictor for the learning task at hand. Yet another way to express our prior knowledge is by specifying preferences over hypotheses within $\mathcal{H}$. In the Structural Risk Minimization (SRM) paradigm, we do so by first assuming that $\mathcal{H}$ can be written as $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$ and then specifying a weight function, $w \colon \mathbb{N} \to [0, 1]$, which assigns a weight to each hypothesis class, $\mathcal{H}_n$, such that a higher weight reflects a stronger preference for the hypothesis class. In this section we discuss how to learn with such prior knowledge. In the next sections we describe a couple of important weighting schemes, including Minimum Description Length.

Concretely, let $\mathcal{H}$ be a hypothesis class that can be written as $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$. Assume that for each n, the class $\mathcal{H}_n$ has the uniform convergence property with a sample complexity function $m_n^{\mathrm{UC}}(\varepsilon, \delta)$. Let us also define the function $\varepsilon_n \colon \mathbb{N} \times (0, 1) \to (0, 1)$ by

$$\varepsilon_n(m, \delta) = \inf\left\{\varepsilon \in (0, 1) \colon m_n^{\mathrm{UC}}(\varepsilon, \delta) \leq m\right\}.$$

In words, we have a fixed sample size m, and we are interested in the lowest possible upper bound on the gap between empirical and true risks achievable by using a sample of m examples.

From the definition of uniform convergence and ε_m, it follows that for every m and δ,

$$\mathop{\mathsf{P}}_{S}[\forall h \in \mathcal{H}_n \colon\ |L_{\mathcal{D}}(h) - L_S(h)| \leq \varepsilon_n(m, \delta)] \geq 1 - \delta.$$

Let $w \colon \mathbb{N} \to [0, 1]$ be a function such that $\sum_{n \in \mathbb{N}} w(n) \leq 1$. We say w is a *weight function* over the hypothesis classes $\{\mathcal{H}_n\}_{n \in \mathbb{N}}$. Such a weight function can reflect the importance that the learner attributes to each hypothesis class, or some measure of the complexity of different hypothesis classes. If $\mathcal{H}$ is a finite union of N hypothesis class, one can simply assign the same weight of N^{-1} to all hypothesis classes. Of course, if one believes (as prior knowledge) that a certain hypothesis class is more likely to contain the correct target function, then it should be assigned a larger weight, reflecting this prior knowledge. When $\mathcal{H}$ is a (countable) infinite union of hypothesis classes, a uniform weighting is not possible but many other weighting schemes can work. For example, one can choose $w(n) = \frac{6}{\pi^2 n^2}$ or $w(n) = 2^{-n}$. Later in this chapter we will provide another convenient way to define weighting functions using description languages.

The SRM rule follows a "bound minimization" approach. This means that the goal of the paradigm is to find a hypothesis that minimizes a certain upper bound on the true risk. The bound that the SRM rule wishes to minimize is given in the following theorem.

Theorem 9.3

Let $w \colon \mathbb{N} \to [0, 1]$ be a function such that $\sum_{n \in \mathbb{N}} w(n) \leq 1$. Let $\mathcal{H}$ be a hypothesis class that can be written as $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$, where for each n, $\mathcal{H}_n$ satisfies the uniform convergence property with a sample complexity m_n^{UC}. Let ε_n be defined as above. Then for every $\delta \in (0, 1)$ and distribution $\mathcal{D}$,

$$\mathop{\mathsf{P}}_{S}[\forall n \in \mathbb{N}\ \forall h \in \mathcal{H}_n \colon\ |L_{\mathcal{D}}(h) - L_S(h)| \leq \varepsilon_n(m, \delta w(n))] \geq 1 - \delta.$$

Therefore, for every $\delta \in (0, 1)$ and distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$,

$$\mathop{\mathsf{P}}_{S}\left[\forall h \in \mathcal{H} \colon L_{\mathcal{D}}(h) \leq L_S(h) + \inf_{\substack{n \in \mathbb{N} \\ h \in \mathcal{H}_n}} \varepsilon_n(m, \delta w(n))\right] \geq 1 - \delta.$$

Proof. For each n define $\delta_n = \delta w(n)$. Applying the assumption that uniform convergence holds for all n, if we fix n in advance,

$$\mathop{\mathsf{P}}_{S}[\forall h \in \mathcal{H}_n \colon\ |L_{\mathcal{D}}(h) - L_S(h)| \leq \varepsilon_n(m, \delta_n)] \geq 1 - \delta_n.$$

Applying the union bound over over $n \in \mathbb{N}$, we get

$$\mathop{\mathsf{P}}_{S}[\forall n \in \mathbb{N}\ \forall h \in \mathcal{H}_n \colon\ |L_{\mathcal{D}}(h) - L_S(h)| \leq \varepsilon_n(m, \delta_n)] \geq 1 - \sum_{n \in \mathbb{N}} \delta_n$$

$$= 1 - \delta \sum_{n \in \mathbb{N}} w(n)$$
$$\geq 1 - \delta$$

which concludes the proof. □

Denote

$$n(h) = \inf\{n \in \mathbb{N} \colon h \in \mathcal{H}_n\}.$$

Then the bound is

$$L_{\mathcal{D}}(h) \leq L_S(h) + \varepsilon_{n(h)}(m, \delta w(n(h))).$$

The SRM paradigm searches for h that minimizes this bound:

Algorithm 2 Structural Risk Minimization (SRM)

```
Input: A hypothesis class $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$ s.t. $\mathcal{H}_n$ has uniform convergence property with sample complexity $m_n^{\mathrm{UC}}$
Input: Training set $S \sim \mathcal{D}^m$
Input: Confidence parameter $\delta \in (0, 1)$
Output: Hypothesis $h \colon \mathcal{X} \to \mathcal{Y}$ s.t. SRM nonuniformly PAC learns $\mathcal{H}$
  return $\operatorname{argmin}_{h \in \mathcal{H}} \left(L_S(h) + \varepsilon_{n(h)}(m, \delta w(n(h)))\right)$
```

Unlike the ERM paradigm discussed in previous chapters, we no longer just care about the empirical risk, $L_S(h)$, but we are willing to trade some of our bias toward low empirical risk with a bias toward classes for which $\varepsilon_{n(h)}(m, \delta w(n(h)))$ is smaller, for the sake of a smaller estimation error.

Next we show that the SRM paradigm can be used for nonuniform learning of every class, which is a countable union of uniformly converging hypothesis classes.

Theorem 9.4

Let $\mathcal{H}$ be a hypothesis class such that $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$, where each $\mathcal{H}_n$ has the uniform convergence property with sample complexity m_n^{UC}. Let $w \colon \mathbb{N} \to [0, 1]$ such that $w(n) = \frac{6}{\pi^2 n^2}$. Then $\mathcal{H}$ is nonuniformly learnable using the SRM rule with rate

$$m^{\mathrm{NUL}}(\varepsilon, \delta, h) \leq m_{n(h)}^{\mathrm{UC}}\left(\frac{\varepsilon}{2}, \frac{6\delta}{\pi^2 n(h)^2}\right).$$

Proof. Let A be the SRM algorithm with respect to the weighting function w. For every $h \in \mathcal{H}$, ε, and δ, let $m \geq m_{n(h)}^{\mathrm{UC}}(\varepsilon, \delta w(n(h)))$. Using the fact that $\sum_{n \in \mathbb{N}} w(n) = 1$, we use the previous result to get that

$$\mathop{\mathsf{P}}_{S}\left[\forall h' \in \mathcal{H} \colon L_{\mathcal{D}}(h') \leq L_S(h') + \varepsilon_{n(h')}(m, \delta w(n(h')))\right].$$

The preceding holds in particular for the hypothesis $h = A(S, \delta)$ returned by the SRM rule. By the definition of SRM we obtain that

$$L_{\mathcal{D}}(h) \leq \inf_{h' \in \mathcal{H}} \left(L_S(h') + \varepsilon_{n(h')}(m, \delta w(n(h')))\right) \leq L_S(h_c) + \varepsilon_{n(h_c)}(m, \delta w(n(h_c))).$$

Finally, if $m \geq m_{n(h_c)}^{\mathrm{UC}}\left(\frac{\varepsilon}{2}, w(n(h_c))\right)$ then clearly $\varepsilon_{n(h_c)}(m, \delta w(n(h_c))) \leq \frac{\varepsilon}{2}$. From the uniform convergence property of each $\mathcal{H}_n$ we have that

$$\mathop{\mathsf{P}}_{S}\left[L_S(h_c) \leq L_{\mathcal{D}}(h_c) + \frac{\varepsilon}{2}\right] \geq 1 - \delta.$$

Combining all the preceding we obtain that

$$\mathop{\mathsf{P}}_{S}[L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(h_c) + \varepsilon] \geq 1 - \delta$$

which concludes our proof. □

Theorem 9.5

A hypothesis $\mathcal{H}$ of binary classifiers is nonuniformly learnable if and only if it is a countable union of agnostic PAC learnable hypothesis classes.

Proof. First assume that $\mathcal{H} = \bigcup_{n\in\mathbb{N}} \mathcal{H}_n$ where each $\mathcal{H}_n$ is agnostic PAC learnable. Using the fundamental theorem of statistical learning, it follows that each $\mathcal{H}_n$ has the uniform convergence property. Therefore from the previous theorem we obtain that $\mathcal{H}$ is nonuniformly learnable.

For the other direction, assume that $\mathcal{H}$ is nonuniformly learnable using some algorithm A. For every $n \in \mathbb{N}$, let $\mathcal{H}_n = \{h \in \mathcal{H} : m^{\mathrm{NUL}}(\frac{1}{8}, \frac{1}{7}, h) \leq n\}$. Clearly $\mathcal{H} = \bigcup_{n\in\mathbb{N}} \mathcal{H}_n$. In addition, using the definition of m^{NUL}, we know that for any distribution $\mathcal{D}$ that satisfies the realizability assumption with respect to $\mathcal{H}_n$, if $S \sim \mathcal{D}^n$ and $h = A(S, \delta)$,

$$\mathop{\mathsf{P}}_{S}\left[L_{\mathcal{D}}(h) \leq \frac{1}{8}\right] \geq \frac{6}{7}.$$

Using the fundamental theorem of statistical learning, this implies that the VC dimension of $\mathcal{H}_n$ must be finite, and therefore $\mathcal{H}_n$ is agnostic PAC learnable. □

The following example shows that nonuniform learnability is a strict relaxation of agnostic PAC learnability; namely, there are hypothesis classes that are nonuniform learnable but are not agnostic PAC learnable.

Remark 9.6 (No Free Lunch for Nonuniform Learnability). We have shown that any countable union of classes of finite VC dimension is nonuniformly learnable. It turns out that, for any infinite domain set, $\mathcal{X}$, the class of all binary valued functions over $\mathcal{X}$ is not a countable union of classes of finite VC dimension.

It follows that, in some sense, the No Free Lunch theorem holds for nonuniform learning as well: namely, whenever the domain is not finite, there exists no nonuniform learner with respect to the class of all deterministic binary classifiers (although for each such classifier there exists a trivial algorithm that learns it – ERM with respect to the hypothesis class that contains only this classifier).

It is interesting to compare the nonuniform learnability result given to the task of agnostic PAC learning any specific $\mathcal{H}_n$ separately. The prior knowledge, or bias, of a nonuniform learner for $\mathcal{H}$ is weaker – it is searching for a model throughout the entire class $\mathcal{H}$, rather than being focused on one specific $\mathcal{H}_n$. The cost of this weakening of prior knowledge is the increase in sample complexity needed to compete with any specific $h \in \mathcal{H}_n$.

9.3 Minimum Description Length and Occam's Razor

Let $\mathcal{H}$ be a countable hypothesis class. Then we can write $\mathcal{H}$ as a countable union of singleton classes, namely $\mathcal{H} = \bigcup_{n\in\mathbb{N}} \{h_n\}$. By Hoeffding's inequality, each singleton class has the uniform convergence property with rate $m^{\mathrm{UC}}(\varepsilon, \delta) = \frac{\log(2/\delta)}{2\varepsilon^2}$. Therefore, the function ε_n is

$$\varepsilon_n(m, \delta) = \sqrt{\frac{2\log(2/\delta)}{2m}}$$

and the SRM rule becomes

$$\mathop{\mathrm{argmin}}_{h_n\in\mathcal{H}} \left(L_S(h) + \sqrt{\frac{\log(2/\delta) - \log(w(n))}{2m}}\right).$$

Equivalently, we can think of w as a function from $\mathcal{H}$ to $[0, 1]$, and then the SRM rule becomes

$$\mathop{\mathrm{argmin}}_{h_n\in\mathcal{H}} \left(L_S(h) + \sqrt{\frac{\log(2/\delta) - \log(w(h))}{2m}}\right).$$

It follows that in this case, the prior knowledge is solely determined by the weight we assign to each hypothesis. We assign higher weights to hypotheses that we believe are more likely to be the correct one, and in the learning algorithm we prefer hypotheses that have higher weights.

In this section we discuss a particular convenient way to define a weight function over $\mathcal{H}$, which is derived from the length of descriptions given to hypotheses. Having a hypothesis class, one can wonder about how we describe, or represent, each hypothesis in the class. We naturally fix some description language. This can be English, or a programming language, or some set of mathematical formulas. In any of these languages, a description consists of finite strings of symbols (or characters) drawn from some fixed alphabet. We shall now formalize these notions.

Fix an alphabet $\Sigma = \{0, 1\}$ and use the Kleene-star notation Σ^* to denote the set of finite strings. A description language for $\mathcal{H}$ is a function $d : \mathcal{H} \to \Sigma^*$ mapping each member h of $\mathcal{H}$ to a string $d(h)$, called the "description of h". By an abuse of notation $|h| = |d(h)|$.

We require that the description language be *prefix-free*; that is, for every $h \neq h'$, $d(h)$ is not a prefix of $d(h')$ (or vice versa). We do not allow that any string $d(h)$ is exactly the first $|h|$ symbols of any longer string $d(h')$. Prefix-free collections of strings enjoy the following combinatorial property.

Lemma 9.7 (Kraft Inequality)
If $S \subseteq \{0, 1\}^*$ is a prefix-free set of strings, then

$$\sum_{\sigma \in S} \frac{1}{2^{|\sigma|}} \leq 1.$$

In light of Kraft's inequality, any prefix-free description language of a hypothesis class, $\mathcal{H}$, gives rise to a weighting function w over that hypothesis class – we will simply set $w(h) = 2^{-|h|}$. This observation immediately yields the following:

Theorem 9.8
Let $\mathcal{H}$ be a hypothesis class and let $d : \mathcal{H} \to \{0, 1\}^*$ be a prefix-free description language for $\mathcal{H}$. Then, for any sample size, m, every confidence parameter $\delta > 0$, and every probability distribution cD, if $S \sim \mathcal{D}^m$ then

$$\mathop{\mathsf{P}}_{S}\left[\forall h \in \mathcal{H} : L_{\mathcal{D}}(h) \leq L_S(h) + \sqrt{\frac{|h| + \log(2/\delta)}{2m}}\right].$$

Proof. Choose $w(h) = 2^{-|h|}$, apply the earlier theorem with $\varepsilon_n(m, \delta) = \sqrt{\frac{\log(2/\delta)}{2m}}$, and note that $\log\left(2^{-|h|}\right) = -|h| \log(2) > -|h|$. □

As was the case with the earlier theorem, this result suggests a learning paradigm for $\mathcal{H}$ – given a training set S, suggest for a hypothesis $h \in \mathcal{H}$ that minimizes the bound $L_S(h) + \sqrt{\frac{|h|+\log(2/\delta)}{2m}}$. In particular, it suggests trading off empirical risk for saving description length. This yields the Minimum Description Length learning paradigm.

Algorithm 3 Minimum Description Length (MDL)

```
Input: A countable hypothesis class $\mathcal{H}$.
Input: A prefix-free description of $\mathcal{H}$, $d : \mathcal{H} \to \{0, 1\}^*$.
Input: A training set $S \sim \mathcal{D}^m$.
Input: A confidence level $\delta$.
  return $\operatorname{argmin}_{h \in \mathcal{H}} \left(L_S(h) + \sqrt{\frac{|d(h)|+\log(2/\delta)}{2m}}\right)$
```

9.4 PAC-Bayes Bounds

The Minimum Description Length (MDL) and Occam's razor principles allow a potentially very large hypothesis class but define a hierarchy over hypotheses and prefer to choose hypotheses that appear higher in the hierarchy. In this section we describe the PAC-Bayesian approach that further generalizes this idea. In the PAC-Bayesian approach, one expresses the prior knowledge by defining prior distribution over the hypothesis class.

As in the MDL paradigm, we define a hierarchy over hypotheses in our class $\mathcal{H}$. Now, the hierarchy takes the form of a prior distribution over $\mathcal{H}$. That is, we assign a measure P over $\mathcal{H}$. Following the Bayesian reasoning approach, the output of the learning algorithm is not necessarily a single hypothesis. Instead, the learning process defines a posterior probability measure over $\mathcal{H}$, which we denote by Q. In the context of a supervised learning problem, where $\mathcal{H}$ contains functions from $\mathcal{X}$ to $\mathcal{Y}$, one can think of Q as defining a randomized prediction rule as follows. Whenever we get a new instance x, we randomly predict a hypothesis $h \in \mathcal{H}$ according to Q and predict $h(x)$. We define the loss of Q on an example z to be

$$\ell(Q, z) = \mathop{\mathsf{E}}_{h\sim Q}[\ell(h, z)].$$

By the linearity of expectation, the generalization loss and training loss of Q can be written as

$$L_D(Q) = \mathop{\mathsf{E}}_{h\sim Q}[L_{\mathcal{D}}(h)] \quad \text{and} \quad L_S(Q) = \mathop{\mathsf{E}}_{h\sim Q}[L_S(h)].$$

The following theorem tells us that the difference between the generalization loss and the empirical loss of a posterior Q is bounded by an expression that depends on the Kullback-Leibler divergence between Q and the prior distribution P. The Kullback-Leibler is a natural measure of the distance between two distributions. The theorem suggests that if we would like to minimize the generalization loss of Q, we should jointly minimize both the empirical loss of Q and the Kullback-Leibler distance between Q and the prior distribution. We will later show how in some cases this idea leads to the regularized risk minimization principle.

Theorem 9.9

Let $\mathcal{D}$ be an arbitrary distribution over an example domain Z. Let $\mathcal{H}$ be a hypothesis class and let $\ell\colon \mathcal{H} \times Z \to [0, 1]$ be a loss function. Let P be a prior distribution over $\mathcal{H}$ and let $\delta \in (0, 1)$. Then, if $S = \{z_i\}_{i\in[m]} \sim \mathcal{D}$,

$$\mathop{\mathsf{P}}_{S}\left[\forall Q \text{ distribution over } \mathcal{H}\colon L_{\mathcal{D}}(Q) \leq L_S(Q) + \sqrt{\frac{D(Q \parallel P) + \log\left(\frac{m}{\delta}\right)}{2(m-1)}}\right],$$

where

$$D(Q \parallel P) = \mathop{\mathsf{E}}_{h\sim Q}\left[\log\left(\frac{Q(h)}{P(h)}\right)\right]$$

is the Kullback-Leibler divergence.

Proof. For any function $f(S)$, using Markov's inequality,

$$\mathop{\mathsf{P}}_{S}[f(S) \geq \varepsilon] = \mathop{\mathsf{P}}_{S}\left[\mathrm{e}^{f(S)} \geq \mathrm{e}^{\varepsilon}\right] \leq \frac{\mathsf{E}_S\left[\mathrm{e}^{f(S)}\right]}{\mathrm{e}^{\varepsilon}}.$$

Let $\Delta(h) = L_D(h) - L_S(h)$. We use this bound with the function

$$f(S) = \sup_{Q}\left(2(m-1)\mathop{\mathsf{E}}_{h\sim Q}\left[\Delta(h)^2\right] - D(Q \parallel P)\right).$$

We now turn to bound $\mathsf{E}_S\left[\mathrm{e}^{f(S)}\right]$. The main trick is to upper bound $f(S)$ by an expression in terms of P and in particular not Q. To do this, fix S and note that by definition of $D(Q \parallel P)$,

$$\begin{aligned}
2(m-1)\mathop{\mathsf{E}}_{h\sim Q}\left[\Delta(h)^2\right] - D(Q \parallel P) &= \mathop{\mathsf{E}}_{h\sim Q}\left[\log\left(\mathrm{e}^{2(m-1)\Delta(h)^2}\frac{P(h)}{Q(h)}\right)\right] \\
&\leq \log\left(\mathop{\mathsf{E}}_{h\sim Q}\left[\mathrm{e}^{2(m-1)\Delta(h)^2}\frac{P(h)}{Q(h)}\right]\right) \\
&= \log\left(\mathop{\mathsf{E}}_{h\sim P}\left[\mathrm{e}^{2(m-1)\Delta(h)^2}\right]\right).
\end{aligned}$$

Therefore

$$\mathop{\mathsf{E}}_{S}\left[\mathrm{e}^{f(S)}\right] \leq \mathop{\mathsf{E}}_{S}\left[\mathop{\mathsf{E}}_{h\sim P}\left[\mathrm{e}^{2(m-1)\Delta(h)^2}\right]\right] = \mathop{\mathsf{E}}_{h\sim P}\left[\mathop{\mathsf{E}}_{S}\left[\mathrm{e}^{2(m-1)\Delta(h)^2}\right]\right].$$

Now we claim for all h we have $\mathsf{E}_S\left[\mathrm{e}^{2(m-1)\Delta(h)^2}\right] \leq m$. To do so, recall that Hoeffding's inequality tells us that

$$\mathop{\mathsf{P}}_{S}[\Delta(h) \geq \varepsilon] \leq \mathrm{e}^{-2m\varepsilon^2}.$$

This implies that

$$\begin{aligned}
\mathop{\mathsf{E}}_{S}\left[\mathrm{e}^{2(m-1)\Delta(h)^2}\right] &= \int_0^\infty \mathop{\mathsf{P}}_{S}\left[\mathrm{e}^{2(m-1)\Delta(h)^2} \geq t\right]\mathrm{d}t \leq \int_0^\infty \mathop{\mathsf{P}}_{S}\left[\Delta(h) \geq \sqrt{\frac{\log(t)}{2(m-1)}}\right]\mathrm{d}t \leq \int_0^\infty \exp\left(-2m\frac{\log(t)}{2(m-1)}\right)\mathrm{d}t \\
&= \int_0^\infty \exp\left(-\log(t)\frac{m}{m-1}\right)\mathrm{d}t = \int_0^\infty t^{-\frac{m}{m-1}}\,\mathrm{d}t \\
&\leq m.
\end{aligned}$$

Combining this with the earlier results, we get

$$\mathop{\mathsf{P}}_{S}[f(S) \geq \varepsilon] \leq \frac{m}{\mathrm{e}^{\varepsilon}}.$$

Denote the right-hand side of the above δ, thus $\varepsilon = \log\left(\frac{m}{\delta}\right)$ and we therefore obtain

$$\mathop{\mathsf{P}}_{S}\left[\forall Q \text{ distribution over } \mathcal{H}: 2(m-1)\mathop{\mathsf{E}}_{h\sim Q}\left[\Delta(h)^2\right] - D(Q \parallel P) \leq \log\left(\frac{m}{\delta}\right)\right] \geq 1-\delta.$$

Rearranging the inequality and using Jensen's inequality again,

$$\mathop{\mathsf{E}}_{h\sim Q}[\Delta(h)]^2 \leq \mathop{\mathsf{E}}_{h\sim Q}\left[\Delta(h)^2\right] \leq \frac{\log\left(\frac{m}{\delta}\right) + D(Q \parallel P)}{2(m-1)}.$$

□

The PAC-Bayes bound leads to the following learning rule: Given a prior P, return a posterior Q that minimizes the function

$$L_S(Q) + \sqrt{\frac{D(Q \parallel P) + \log\left(\frac{m}{\delta}\right)}{2(m-1)}}.$$

This rule is similar to the *regularized risk minimization* principle. That is, we minimize the empirical loss of Q on the sample and the Kullback-Leibler "distance" between Q and P.

10 Regularization and Stability

The new learning paradigm we introduce in this chapter is called Regularized Loss Minimization, or RLM for short. In RLM we minimize the sum of the empirical risk and a regularization function. Intuitively, the regularization function measures the complexity of hypotheses. Indeed, one interpretation of the regularization function is the structural risk minimization paradigm we discussed in the last chapter. Another view of regularization is as a stabilizer of the learning algorithm. An algorithm is considered stable if a slight change of its input does not change its output much. We will formally define the notion of stability (what we mean by "slight change of input" and by "does not change much the output") and prove its close relation to learnability. Finally, we will show that using the squared ℓ^2 norm as a regularization function stabilizes all convex-Lipschitz or convex-smooth learning problems. Hence, RLM can be used as a general learning rule for these families of learning problems.

10.1 Regularized Loss Minimization

Regularized Loss Minimization (RLM) is a learning rule in which we jointly min- imize the empirical risk and a regularization function. Formally, a regularization function is a real-valued functional, and the regualrized loss minimization rule outputs a hypothesis in

$$\operatorname*{argmin}_{f} \left(L_S(f) + R(f) \right).$$

Regularized loss minimization shares similarities with minimum description length algorithms and structural risk minimization. Intuitively, the "complexity" of hypotheses is measured by the value of the regularization function, and the algorithm balances between low empirical risk and "simpler" or "less complex" hypotheses.

There are many possible regularization functions one can use, reflecting some prior belief about the problem (similarly to the description language in Minimum Description Length). Throughout this section we will focus on one of the most simple regularization functions: $R(f_w) = \lambda \left\| w \right\|_2^2$, where $\lambda > 0$ is a scalar. This yields the learning rule:

$$A(S, \varepsilon, \delta) = \operatorname*{argmin}_{w} \left(L_S(f_w) + \lambda \left\| w \right\|_2^2 \right).$$

This type of regularization function is called *Tikhonov regularization*.

One interpretation of this equation is using structural risk minimization, where the norm of w is a measure of its "complexity." Recall that we introduced the notion of bounded hypothesis classes. Therefore, we can define a sequence of hypothesis classes, $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \mathcal{H}_3, \ldots$, where $\mathcal{H}_i = \{w \colon \left\| w \right\|_2 \leq i\}$. If the sample complexity of each $\mathcal{H}_i$ depends on i then the RLM rule for linear regression with the squared loss.

10.1.1 Ridge Regression

Applying the RLM rule with Tikhonov regularization to linear regression with the squared loss, we obtain the following learning rule:

$$\operatorname*{argmin}_{w \in \mathbb{R}^d} \left(\frac{1}{m} \left\| Xw - y \right\|_2^2 + \lambda \left\| w \right\|_2^2 \right).$$

Performing linear regression using this method is called *ridge regression*.

To solve this equation we compute the gradient:

$$\nabla_w \left(\frac{1}{m} \left\| Xw - y \right\|_2^2 + \lambda \left\| w \right\|_2^2 \right) = 2\frac{Xw - y}{m} + 2\lambda w \overset{\text{set}}{=} 0 \implies Xw - y + \lambda m w = 0 \implies w = (X + \lambda m I)^{-1} y.$$

In the next section we formally show how regularization stabilizes the algorithm and prevents overfitting. In particular, the analysis presented in the next sections will yield

Theorem 10.1

Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times [-1, 1]$, where $\mathcal{X} = \mathbb{B}_2(0_d, 1)$. Let $\mathcal{H} = \mathbb{B}_2(0_d, B)$. For any $\varepsilon \in (0, 1)$, let $m \geq \frac{150B^2}{\varepsilon^2}$. Then, applying the ridge regression algorithm with parameter $\lambda = \frac{\varepsilon}{3B^2}$ satisfies

$$\mathop{\mathsf{E}}_{S}[L_{\mathcal{D}}(h)] \leq \inf_{w \in \mathcal{H}} L_{\mathcal{D}}(w) + \varepsilon.$$

Remark 10.2. The preceding theorem tells us how many examples are needed to guarantee that the expected value of the risk of the learned predictor will be bounded by the approximation error of the class plus ε. In the usual definition of agnostic PAC learning we require that the risk of the learned predictor will be bounded with probability of at least $1 - \delta$. An algorithm with bounded expected risk can be used to construct an agnostic PAC learner by doing the same algorithm many times independently.

10.2 Stable Rules Do Not Overfit

Intuitively, a learning algorithm is stable if a small change of the input to the algorithm does not change the output of the algorithm much. Of course, there are many ways to define what we mean by "a small change of the input" and what we mean by "does not change the output much". In this section we define a specific notion of stability and prove that under this definition, stable rules do not overfit.

The algorithm A suffers from overfitting if the difference between the true risk of its output, $L_{\mathcal{D}}(h)$, and the empirical risk of its output, $L_S(h)$, is large. Throughout this chapter we focus on the expectation with respect to the choice of S of this quantity, namely, $\mathsf{E}_S[L_{\mathcal{D}}(h) - L_S(h)]$.

We next define the notion of stability. Given a trianing set S and a training example z', let $S^{(i)}$ be the training set obtained by relacing the i^{th} sample of S with z'; namely $S^{(i)} = \{z_1, \ldots, z_{i-1}, z', z_{i+1}, \ldots, z_m\}$. Let $h^{(i)} = A(S^{(i)}, \varepsilon, \delta)$. We measure the effect of this small change of the input on the output of A by comparing the loss of the hypothesis h on z_i to the loss of the hypothesis $h^{(i)}$ on z_i. Intuitively, a good learning algorithm will have $\ell(h^{(i)}, z_i) - \ell(h, z_i) \geq 0$, since in the first term the learning algorithm does not observe the sample z_i, while in the second term z_i is indeed observed. If the preceding difference is very large we suspect that the learning algorithm might overfit. This is because the learning algorithm drastically changes its prediction on z_i if it observes it in the training set. This is formalized in the following theorem.

Theorem 10.3

Let $\mathcal{D}$ be a distribution. Let $S = \{z_i\}_{i \in [m]} \sim \mathcal{D}^m$ be an i.i.d. sequence of examples and let z_i' be another i.i.d. example. Then for any learning algorithm:

$$\mathop{\mathsf{E}}_{S}[L_{\mathcal{D}}(h) - L_S(h)] = \mathop{\mathsf{E}}_{(S,z') \sim \mathcal{D}^{m+1}, i \sim \mathrm{Uni}([m])}\left[\ell(h^{(i)}, z_i) - \ell(h, z_i)\right].$$

Proof. Since S and z' are both drawn i.i.d. from $\mathcal{D}$, we have that for every i,

$$\mathop{\mathsf{E}}_{S}[L_{\mathcal{D}}(h)] = \mathop{\mathsf{E}}_{S,z'}\left[\ell(h, z')\right] = \mathop{\mathsf{E}}_{S,z'}\left[\ell(h^{(i)}, z_i)\right].$$

On the other hand, we have

$$\mathop{\mathsf{E}}_{S}[L_S(h)] = \mathop{\mathsf{E}}_{S,i}[\ell(h, z_i)].$$

Combining these equations concludes the proof. □

10.3 Controlling the Fitting-Stability Tradeoff

We can rewrite the expected risk of a leraning algorithm as

$$\mathop{\mathsf{E}}_{S}[L_{\mathcal{D}}(h)] = \mathop{\mathsf{E}}_{S}[L_S(h)] + \mathop{\mathsf{E}}_{S}[L_{\mathcal{D}}(h) - L_S(h)] \,.$$

The first term reflects how well h fits the training set while the second term reflects the difference between the true and empirical risks of h. As we have shown, the second term is equivalent to the stability of A. Since our goal is to minimize the risk of the algorithm, we need that the sum of both terms will be small.

By imposing regularity conditions on the loss, we can bound the stability term. We can show that the stability term decreases as the regularization parameter λ increases. We therefore face a tradeoff between fitting and overfitting. This tradeoff is quite similar to the bias-complexity tradeoff we discussed previously.