Appendix VC Dimension for Numeric Parameters and Sample Complexity

A basic question in learning is *how many examples are necessary to learn an accurate representation?* In Section 4.1, we saw a formal version of Occam's Razor, showing that a number of examples scaling with the encoding length of a representation suffices for accurate learning. But, when representations may use real or rational-valued parameters and there is no a priori bound on the representation size of these parameters, Occam's Razor does not enable us to learn such representations. Actually, Occam's Razor is needlessly pessimistic, and many such representations are learnable. In this chapter, we will introduce a parameter of families of properties that asymptotically captures the number of examples that are necessary and sufficient for learning. This is the *Vapnik-Chervonenkis (VC) dimension*. In particular, we will see that some families of properties have an infinite VC-dimension, and hence are not learnable with any finite amount of data.

We will also discuss the related question of *how the accuracy of learned representations scale with the number of examples seen*. Although this may sound like the same question, a subtle difference in the way it is posed leads to a very different answer: in many cases, the long-run accuracy grows much more quickly than the VC-dimension bounds (or the Occam's Razor bound) seem to suggest, approaching perfect accuracy at an *exponential* rate. We will seek to demystify this seemingly paradoxical state of affairs.

Finally, we will use these enriched representations to show that efficient learnability is an all-or-nothing property of a family of representations: that is, either there is an efficient learning algorithm that obtains high accuracy, or else no efficient algorithm can be guaranteed to find representations that agree with the property better than chance. This is established by giving an efficient algorithm to *boost* the accuracy of learned representations from slightly better than chance to near-perfect, by combining multiple of these "weak" representations.

A.1 Intoducing the VC-dimension.

We can motivate the VC-dimension by first considering when generalization is not possible. Let's suppose that I choose a property uniformly at random: that is, each possible world independently satisfies the property (or not) with probability 1/2. Since these choices are independent, even if you are given examples indicating that

some possible worlds have the property and others do not, the remaining worlds still each possess the property with probability 1/2. You cannot predict whether or not they will have the property any better than by making an uninformed guess, say tossing a coin yourself. The Chernoff/Hoeffding bounds tell us that your probability of getting substantially more than half of these predictions correct is exponentially small. Generalization is thus impossible, and the only way to learn is to observe most of the possible worlds—assuming there are even finitely many of these. (We will revisit this argument more carefully in the next section.)

The VC-dimension of a family of properties captures how many possible worlds can have their membership determined independently. More precisely:

Definition A.1 For a class of representations $\mathcal{H}$ and a set s of possible worlds, the *set of dichotomies of s induced by $\mathcal{H}$*, denoted $\Pi_{\mathcal{H}}(s)$, is the set of properties on s obtainable by members of $\mathcal{H}$, i.e., $\{h|_s \subseteq s : \exists h \in \mathcal{H} \text{ s.t. } h|_s = h \cap s\}$.

We are interested in how large a set of possible worlds s can be arbitrarily set in or out of the induced property by choosing an appropriate member of $\mathcal{H}$, i.e.:

Definition A.2 We say that a set of possible worlds s is *shattered* by the class of representations $\mathcal{H}$ if $\Pi_{\mathcal{H}}(s) = \mathcal{P}(s)$, the set of all subsets of s.

Definition A.3 The *Vapnik-Chervonenkis (VC) dimension* of a class of representations $\mathcal{H}$ is the size of the largest set S of possible worlds shattered by $\mathcal{H}$. If sets of all sizes are shattered by $\mathcal{H}$, then we say its VC-dimension is ∞.

Aside: One of the main motivations for the VC dimension is to study properties with numeric parameters and/or with numerical attributes. In this book, we have primarily focused on attributes taking Boolean values, where there are a finite number of possible worlds, and hence there was no question that we could define all probability distributions by simply assigning a nonnegative probability value to each possible world. (That is, using the formalism of "discrete probability".) Unfortunately, probability in continuous domains is generally not so simple, and requires considerations of the "measurability" of events under consideration. However, for our purposes, we are only interested in parameters and attributes with finite representations, hence belonging to some countable universe. In particular, if the representation size of the attributes is finite in expectation, Markov's inequality implies that for any $\epsilon > 0$, $1 - \epsilon$ fraction of the probability mass is covered by representations of bounded length. We are therefore again justified in simply assigning positive probabilities to the possible worlds. To stress this, we will generally refer to our properties as subsets of the rational vectors $\mathbb{Q}^n$, but we can work with any countable set of numbers.

The VC dimension of a family of representations can be quite challenging to determine. On the one hand, it is often straightforward to devise a set S of d possible worlds that is shattered. What is tricky is to show that *no* set of $d + 1$ possible worlds can be shattered: that no matter which $d + 1$ points are chosen, some property of those points cannot be represented. We'll next consider some examples of families for which we can determine the VC dimension without too much difficulty.

Example A.1: Suppose $\mathcal{H}$ is a singleton set, $\mathcal{H} = \{h\}$. Then for any possible world x, $\Pi_{\mathcal{H}}(\{x\})$ is either $\{x\}$, if x satisfies the property represented by h, or $\varnothing$ if it does not. Thus $\{x\}$ is *not* shattered by $\mathcal{H}$, so the VC dimension is 0. Note that in this case, learning is trivial: The only property $\mathcal{H}$ can represent is h, so if $\mathcal{H}$ can represent the target c, for every possible world x we know $x \in c$ iff $x \in h$. Thus, we can predict membership in c for any possible world x, even without seeing any examples.

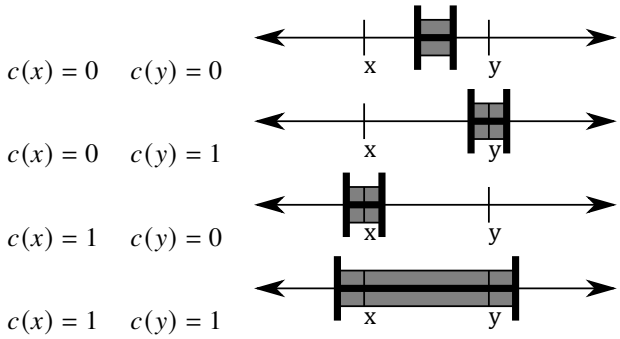
Example A.2: Let $\mathcal{H}$ represent $\mathcal{P}(\{0,1\}^n)$, the set of all possible worlds on n Boolean attributes. Then for every property $c \subseteq \{0,1\}^n$, there is some $h \in \mathcal{H}$ such that for all $x \in \{0,1\}^n$, $x \in h$ iff $x \in c$. That is, for the set $S = \{0,1\}^n$, $\Pi_{\mathcal{H}}(S) = \mathcal{P}(\{0,1\}^n)$, so S is shattered by $\mathcal{H}$. Since $\mathcal{H}$ is only representing subsets of $\{0,1\}^n$ and there are exactly 2^n members of $\{0,1\}^n$, we know that

- Some set of size 2^n, namely $\{0,1\}^n$, is shattered by $\mathcal{H}$.
- No set of size larger than 2^n can be shattered by $\mathcal{H}$.

And hence, the VC dimension of this family is 2^n.

Example A.3: Let $\mathcal{H}$ represent $\mathcal{P}(\mathbb{N})$, the set of all possible properties of the nonnegative integers. Now, for every $n \in \mathbb{N}$, let's consider the set $S = \{1, \ldots, n\}$. We find that $\Pi_{\mathcal{H}}(S) = \mathcal{P}(\{1, \ldots, n\})$, so this set S of size n is shattered by $\mathcal{H}$. Thus there is no bound on the size of sets shattered by $\mathcal{H}$, and the VC dimension is infinite.

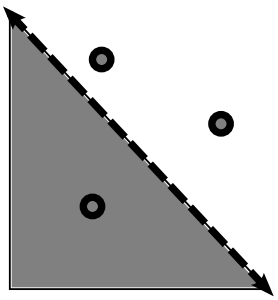
Example A.4: Let $\mathcal{H}$ represent closed intervals on $\mathbb{Q}$, $\{[l, r] : l, r \in \mathbb{Q}\}$. Now, for any two distinct points $x < y$, we can find intervals that correspond to each of the four subsets:



The intervals therefore have VC-dimension at least two. Now, on the other hand, for any three distinct points $x < y < z$, observe that if both x and z lie in the interval, y must also lie inside. Therefore, no interval can give the labeling $c(x) = 1$, $c(y) = 0$,

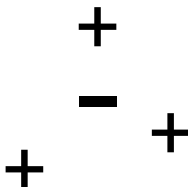
and $c(z) = 1$. Since this is true for any choice of three points, no set of three points is shattered. Hence the VC dimension of the intervals is exactly two.

Example A.5: Let $\mathcal{H}$ represent halfspaces in $\mathbb{Q}^2$. Consider any three points $(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)})$, $(x^{(3)}, y^{(3)})$ that do not all lie on a line. It's easy to obtain the all-0 and all-1 labelings, and the other six label one point different from the others:

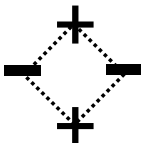


which is also clearly achievable. So we see that sets of three points are shattered by the halfspaces, and the VC dimension is at least three. Now if we consider sets of four points, there are three cases to consider:

1. If three of the points lie on a line, then as with our intervals earlier, one of the points must lie between the other two. Then if the other two points are in the halfspace, that one must be as well, so the points cannot be shattered.
2. Suppose one point lies inside the convex hull of the other three. Then, similarly, if the points bounding the convex hull lie in the halfspace, the point on the interior cannot lie on the other side, so this labeling cannot be achieved:



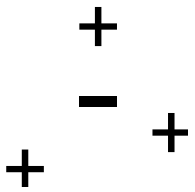
   and hence, the points cannot be shattered.
3. Finally, we have the case that no point lies inside the convex hull of any of the others; we can now consider a labeling of the points so that the neighbors on the convex hull have opposite labels:



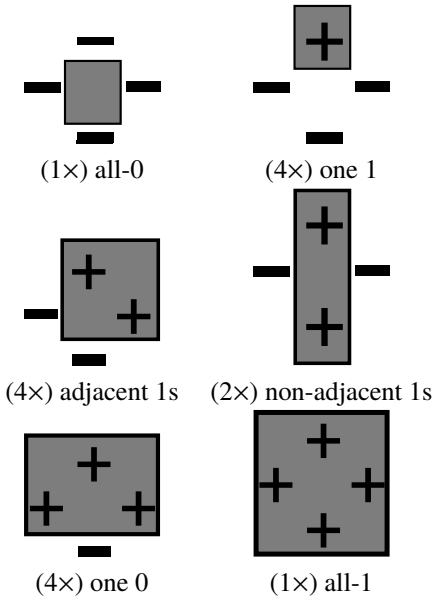
This labeling cannot be realized by a halfspace: indeed, suppose it contains both of the positive labels. Every point on the segment between these two points must also lie inside the halfspace. If the two negative points were outside the halfspace, the segment joining them must also lie outside the halfspace. But notice, since none of the points lie inside the rest, the segment joining the two negative points must cross the segment joining the positive points, where the intersection cannot be on both sides of the halfspace. Therefore, no such labeling exists.

Thus, we find that no set of four points can be shattered. The halfspaces in $\mathbb{Q}^2$ therefore have VC-dimension three.

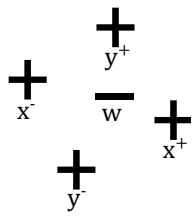
Example A.6: Let $\mathcal{H}$ represent the class of axis-aligned boxes in $\mathbb{Q}^2$, i.e., sets of the form $[l, r] \times [b, t]$ for $l \leq r$ and $b \leq t$ in $\mathbb{Q}$. Now, there exist some sets of four points that cannot be shattered, for example again:



is an unrealizable labeling if one point lies in the interior of the other three. But, to obtain a lower bound on the VC dimension, it is sufficient for some adequate set of points to exist. So, we may instead consider a set of four points such as $\{(1, 0), (-1, 0), (0, 1), (0, -1)\}$. Now, up to reflection and rotation, we may obtain the sixteen different labelings as follows:



And thus, we see that the VC dimension of axis-aligned boxes in $\mathbb{Q}^2$ is at least four. To see that it is exactly four, we must consider all possible sets of five points. For any set of five points, we can choose one point $\vec{x}^-$ with minimum first coordinate, $\vec{x}^+$ with maximum first coordinate, $\vec{y}^-$ with minimum second coordinate, and $\vec{y}^+$ with maximum second coordinate. (These might not be distinct.) Since there are five points and the set $\{\vec{x}^-, \vec{x}^+, \vec{y}^-, \vec{y}^+\}$ includes at most four points, there must be at least one point $\vec{w}$ that is not among these points. Now, we claim that no axis-aligned box can give label 0 to $\vec{w}$ and label 1 to the rest of the points:



Indeed, if $\{\vec{x}^-, \vec{x}^+, \vec{y}^-, \vec{y}^+\} \subset [l, r] \times [b, t]$ then $l \leq x_1^- \leq w_1 \leq x_1^+ \leq r$ and $b \leq y_2^- \leq w_2 \leq y_2^+ \leq t$, so $\vec{w} \in [l, r] \times [b, t]$ as well. Therefore, this arbitrary set of five points is not shattered by the axis-aligned boxes, and thus their VC-dimension is exactly four.

A.2 Sample complexity lower bound for accuracy ("No Free Lunch").

By returning to our motivating discussion, it is relatively easy to connect the VC-dimension to the amount of data required. Classes of representation with VC-dimension d are rich enough to capture random – hence, unpredictable – properties on some collection of d possible worlds. Here, rather than the target property c being fixed, we think of it as having been chosen uniformly at random from a set of properties. These properties will be those represented by the members of $\mathcal{H}$ that enable $\mathcal{H}$ to shatter some set S of d possible worlds, as guaranteed by the VC-dimension. We now cannot predict whether any of these d worlds satisfy the randomly chosen c with probability greater than 1/2, no matter how many of the other worlds we have observed: each is independent of the rest. Then, unless we have seen enough examples to observe most of the distribution, we cannot attain high accuracy. Formally, this gives the following sample-size requirement for learning:

Theorem A.4 ("No free lunch") *For any class of properties of VC-dimension d, there is a distribution D and a constant $\delta_0 \in (0, 1)$ such that for any algorithm that uses $O(d/\epsilon)$ examples, there is some $c^* \in C$ for which the algorithm outputs a representation h with $\Pr_D[c^*(x) = h(x)] \geq 1 - \epsilon$ with probability at most $1 - \delta_0$.*

Proof We'll first show $d/2$ examples are not sufficient to obtain $\epsilon < \frac{1}{8}$ with probability $6/7$, then extend the construction to obtain the bound for algorithms using $O(d/\epsilon)$ examples.

Since C has VC-dimension d, there exists a set of d possible worlds $\{\vec{x}^{(1)}, \ldots, \vec{x}^{(d)}\} \subseteq \mathcal{X}$ that is shattered by C. This means that for any $\vec{b} \in \{0,1\}^d$, there is some $c_{\vec{b}} \in C$ giving $c_{\vec{b}}(\vec{x}^{(i)}) = b_i$ for $i = 1, \ldots, d$. Suppose $C \in C$ is chosen at random with the following distribution: it is $c_{\vec{b}}$ with probability $1/2^d$ for each $\vec{b} \in \{0,1\}^d$. Notice, for each $\vec{b}$, $(C(\vec{x}^{(1)}), \ldots, C(\vec{x}^{(d)})) = \vec{b}$ with probability $1/2^d$, which is the distribution of d independent Bernoulli random variables.

We now consider the distribution D_0 that is uniform over $\{\vec{x}^{(1)}, \ldots, \vec{x}^{(d)}\}$, and consider any function that takes a set S of at most $d/2$ examples from D_0 and produces some $H(S) : \mathcal{X} \to \{0,1\}$. Letting $S^c = \{\vec{x}^{(1)}, \ldots, \vec{x}^{(d)}\} \setminus S$, we observe that given C is independent of S, $H(S)$ is independent of $C(\vec{x})$ for every $\vec{x} \in S^c$. Therefore, the expected error of $H(S)$ is at least $\frac{1}{2}\frac{d}{2} = \frac{d}{4}$. Let p be the probability that the error of $H(S)$ is less than $\frac{1}{8}$, and observe that p satisfies

$$\frac{1}{4} \leq p \cdot \frac{1}{8} + (1-p) \cdot 1 \Leftrightarrow p \leq \frac{6}{7}.$$

In particular, for any algorithm using so few examples (giving a distribution on $H(S)$), there is some $c^* = c_{\vec{b}^*}$ for which the probability that $H(S)$ has error less than $1/8$ is at most $6/7$, or else the average value (attained by C) would be lower.

Now, for our final distribution, let D be the distribution that is $\vec{x}^{(d)}$ with probability $1 - 8\epsilon$ and otherwise is equal to each $\vec{x}^{(i)}$ for $i = 1, \ldots, d-1$ with probability $\frac{8\epsilon}{d-1}$. Now, if the set of examples S contains fewer than $\frac{d-1}{2}$ of the first $d-1$ examples, S^c contains at least $\frac{d-1}{2}$ of the examples, and again any $H(S)$ is independent of $C(\vec{x})$ for every $\vec{x} \in S^c$. When this occurs, h is wrong on at least $\frac{d-1}{4}$ in expectation. The probability that $H(S)$ is wrong on fewer than $\frac{d-1}{8}$ of these examples (attaining error ϵ) is similarly at most $\frac{6}{7}$:

$$\frac{d-1}{4} \leq p \cdot \frac{d-1}{8} + (1-p) \cdot (d-1) \Leftrightarrow p \leq \frac{6}{7}$$

that is, with probability at least $1/7$, it has error at least ϵ.

It thus finally suffices to show that a sample of $O(\frac{d}{\epsilon})$ examples from D contains fewer than $\frac{d-1}{2}$ of the first $d-1$ examples with constant probability. Each draw is equal to $\vec{x}^{(d)}$ with probability $1 - 8\epsilon$, so in a sample of size $\frac{1}{14}\frac{d-1}{2}\frac{1}{8\epsilon}$, the expected number of draws that are not $\vec{x}^{(d)}$ is $\frac{1}{14}\frac{d-1}{2}$. By Markov's inequality, it is only at least $\frac{d-1}{2}$ with probability at most $1/14$. Thus we see that the total probability of the algorithm obtaining $H(S)$ with error less than ϵ is indeed at most $1 - 1/14$. Finally, as before, observe that there must be some c^* for which algorithm has probability at most $1-1/14$ of producing a satisfactory h (or else the average would be higher). □

Theorem A.4 is called a "no free lunch" theorem because it implies that learning is only possible for classes of representations with bounded VC-dimension. By contrast, the classes of all functions on $\mathbb{N}^n$, $[0,1]^n$, etc. all have infinite VC-dimension. Generalization is impossible to guarantee on such domains unless we are given some kind of restriction on the functions that can be represented. Even the class of all functions on $\{0,1\}^n$ has VC-dimension 2^n, and hence $\Omega(2^n)$ examples is necessary—essentially, we must observe enough examples to memorize the domain in this case. No

points in the shattered set allow us to infer the labeling of any others that we have not yet observed. Thus, in order for sample-efficient learning to be possible, the properties must be "simple", in the sense captured by low VC-dimension. Low VC-dimension means precisely that there are some labelings that cannot be represented by the class.

A.3 Sample complexity upper bound (Sauer's lemma and all that).

Perhaps surprisingly, we can also give a bound on the number of examples that is sufficient for learning in terms of the VC-dimension. That is, we can state an analogue of Occam's Razor in which the VC-dimension appears in place of the representation length. We will start by generalizing the Occam's Razor analysis.

Definition A.5 The *error regions of $\mathcal{H}$ w.r.t. a property c*, denoted $\Delta_{\mathcal{H}}(c)$, is $\{h\Delta c : h \in \mathcal{H}\}$ where Δ is the symmetric difference operator on sets: $x \in h\Delta c$ iff $h(x) \neq c(x)$. The *ϵ-error regions of $\mathcal{H}$ w.r.t. c and a distribution D*, denoted $\Delta_{\mathcal{H},\epsilon}(c)$, is the subset of $\Delta_{\mathcal{H}}(c)$ with probability at least ϵ under D, i.e., $\{r \in \Delta_{\mathcal{H}}(c) : \Pr_{x\in D}[x \in r] \geq \epsilon\}$.

In general, we can at best choose *some* representation in $\mathcal{H}$ that is consistent with the examples we have. This strategy succeeds (up to ϵ error) precisely when we have an example from each of the ϵ-error regions. In such a case, the set of examples is said to be an "ϵ-net" for the error regions:

Definition A.6 A set of possible worlds is an *ϵ-net* for a set of sets of possible worlds R w.r.t. a distribution D if for every $r \in R$ with $\Pr_{x\in D}[x \in r] \geq \epsilon$, $s \cap r \neq \varnothing$.

We can phrase our Occam's Razor analysis in the language of ϵ-nets as follows: suppose h is bad, i.e., $\Pr_{x\in D}[h(x) \neq c(x)] > \epsilon$. This event refers to an ϵ-error region r of some $h \in \mathcal{H}$; the probability that we miss r after m draws is at most $(1-\epsilon)^m$. So by a union bound over the set of ϵ-error regions $\Delta_{\mathcal{H},\epsilon}$, the probability that the data set is not an ϵ-net is at most $|\Delta_{\mathcal{H},\epsilon}|(1-\epsilon)^m \leq |\Delta_{\mathcal{H}}|(1-\epsilon)^m$. It turns out that $|\Delta_{\mathcal{H}}|$ is equal to $|\mathcal{H}|$, so we will obtain the Occam's Razor bound again in this way:

Lemma A.7 *The map $h \mapsto h\Delta c$ is a bijection between $\mathcal{H}$ and $\Delta_{\mathcal{H}}$. Similarly, letting $c|_s = c \cap s$ for any set s, $h|_s \mapsto h|_s \Delta c|_s$ is a bijection between $\Pi_{\mathcal{H}}(s)$ and $\Pi_{\Delta_{\mathcal{H}}}(s)$.*

Proof Indeed, the map is its own inverse: $(h\Delta c)\Delta c = h$. It is therefore a bijection. The same holds for the members of $\Pi_{\mathcal{H}}(s)$ and $\Pi_{\Delta_{\mathcal{H}}}(s)$ that are simply the intersections with s. □

The main trick in the VC-dimension analysis is that instead of considering the set of possible properties – which may be infinite – we consider the set of induced properties on a sample, which is finite:

Definition A.8 The *growth function* $\Pi_{\mathcal{H}} : \mathbb{N} \to \mathbb{N}$ is the size of the largest set of dichotomies of a set of the given size m induced by $\mathcal{H}$, $\max_{s\subseteq\mathcal{X}:|s|=m} |\Pi_{\mathcal{H}}(s)|$.

Even though the number of induced properties may be finite, as we saw in the lower bound, if we have too few examples compared to the VC-dimension, the induced property may be unconstrained and learning is not possible. Indeed, this is an immediate consequence of the definition:

Lemma A.9 *If $\mathcal{H}$ has VC-dimension d, then for all $m \leq d$, $\Pi_{\mathcal{H}}(m) = 2^m$.*

Proof If $\mathcal{H}$ has VC-dimension d, then a set s of size d is shattered by $\mathcal{H}$. Consider any $s' \subseteq s$ of size m. Then since s is shattered, for every $s'' \subseteq s'$, since $s'' \subseteq s$ as well, we know that s'' is induced by some member of $\mathcal{H}$. Thus, s' is a set of size m that is also shattered by $\mathcal{H}$, and hence $\Pi_{\mathcal{H}}(m) = 2^m$. □

The surprising fact is that when the number of examples exceeds the VC-dimension, *the number of dichotomies grows polynomially (not exponentially) in the number of examples*. This lemma is the key to the VC-dimension bound: the probability of obtaining a faulty estimate of the quality of an induced property shrinks exponentially with the number of examples drawn, while the number of induced properties only grows polynomially. We'll therefore be able to control the probability of failure by collecting more examples.

Lemma A.10 (Sauer's Lemma) *If $\mathcal{H}$ has VC-dimension d, then* $\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} = O(m^d)$.

Proof We first show by induction on d and m that $\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}$. The base case is $d = 0$ or $m = 0$: in the former case, no points are shattered, and $\mathcal{H}$ must contain only a single set, so $\Pi_{\mathcal{H}}(m) = 1 = \binom{m}{0}$. In the latter case, the empty set is the only property, and so again $\Pi_{\mathcal{H}}(0) = 1 = \sum_{i=0}^{d} \binom{0}{i}$.

So, assume for induction that the claim holds for all pairs $d' \leq d$, $m' \leq m$ such that at least one of these inequalities is strict. Let s be a set of size m and $x \in s$ be given. We partition $\Pi_{\mathcal{H}}(s)$ into three sets: the first two, $\Pi_{\mathcal{H}|_1}(s)$ and $\Pi_{\mathcal{H}|_0}(s)$, consist of the respective pairs of members of $\Pi_{\mathcal{H}}(s)$ that only differ in x—the h with $x \in h$ is in $\mathcal{H}|_1$ and the h with $x \notin h$ is in $\mathcal{H}|_0$ (with $h \cap s \setminus \{x\}$ identical in the pair). The third, $\Pi_{\mathcal{H}|_-}(s)$ consists of the remaining members of $\Pi_{\mathcal{H}}(s)$. Note that for each $h|_s \in \Pi_{\mathcal{H}|_-}(s) \cup \Pi_{\mathcal{H}|_0}(s)$, if $x \notin h|_s$, then $h|_s$ is in $\Pi_{\mathcal{H}}(s \setminus \{x\})$ or, for those $h|_s \in \Pi_{\mathcal{H}|_-}(s)$ for which the unique $h|_s \in \Pi_{\mathcal{H}}(s)$ has $x \in h|_s$, $h|_s \setminus \{x\}$ is in $\Pi_{\mathcal{H}}(s \setminus \{x\})$ (and $h|_s \setminus \{x\}$ was not counted in $\Pi_{\mathcal{H}|_-}(s) \cup \Pi_{\mathcal{H}|_0}$ in the first case). So, $|\Pi_{\mathcal{H}|_-}(s) \cup \Pi_{\mathcal{H}|_0}(s)| \leq |\Pi_{\mathcal{H}}(s \setminus \{x\})| \leq \sum_{i=0}^{d} \binom{m-1}{i}$ by the induction hypothesis since $|s \setminus \{x\}| = m - 1$.

It thus remains to count the contribution from $\Pi_{\mathcal{H}|_1}(s)$. Observe that $\mathcal{H}|_1$ must have VC-dimension at most $d - 1$: since all $h \in \mathcal{H}|_1$ have $x \in h$, no set containing x can be shattered by $\mathcal{H}|_1$, so any set s' shattered by $\mathcal{H}|_1$ is also shattered by $\mathcal{H}|_0$. But then $s' \cup \{x\}$ is shattered by $\mathcal{H}$. Thus, $|s'| + 1 = |s' \cup \{x\}| \leq d$ by hypothesis, so indeed $\mathcal{H}|_1$ has VC-dimension at most $d - 1$. So now, by the induction hypothesis,

$|\Pi_{\mathcal{H}|_1}(s)| = |\Pi_{\mathcal{H}|_1}(s \setminus \{x\})| \leq \sum_{i=0}^{d-1} \binom{m-1}{i}$. All together, we find that

$$\begin{aligned}
|\Pi_{\mathcal{H}}(s)| &= |\Pi_{\mathcal{H}|_0}(s) \cup \Pi_{\mathcal{H}|_-}(s)| + |\Pi_{\mathcal{H}|_1}(s)| && \text{sets partition } \Pi_{\mathcal{H}}(s) \\
&\leq \sum_{i=0}^{d} \binom{m-1}{i} + \sum_{j=0}^{d-1} \binom{m-1}{j} && \text{bounds obtained above} \\
&= 1 + \sum_{i=1}^{d} \left[\binom{m-1}{i} + \binom{m-1}{i-1} \right] && \text{pair index } i \text{ with } j = i-1 \\
&= \sum_{i=0}^{d} \binom{m}{i} && \text{binomial identity}
\end{aligned}$$

completing the induction.

Finally, for $d < m$, $0 < \frac{d}{m} < 1$, so

$$\begin{aligned}
\left(\frac{d}{m}\right)^d \sum_{i=0}^{d} \binom{m}{i} &< \sum_{i=0}^{d} \left(\frac{d}{m}\right)^i \binom{m}{i} && \text{since } \frac{d}{m} < 1 \\
&< \sum_{i=0}^{m} \left(\frac{d}{m}\right)^i \binom{m}{i} && \text{terms are positive} \\
&= (1 + \frac{d}{m})^m && \text{Binomial Theorem} \\
&\leq e^d && \text{since } 1 + \frac{d}{m} \leq e^{d/m}.
\end{aligned}$$

Thus, dividing both sides by $\left(\frac{d}{m}\right)^d$ gives $\sum_{i=0}^{d} \binom{m}{i} < \left(\frac{me}{d}\right)^d = O(m^d)$. □

We now show that VC-dimension bounds the amount of data required to learn:

Theorem A.11 *There is a constant c_0 such that for any class of representations $\mathcal{H}$, any distribution D on possible worlds, any property c, any accuracy $\epsilon \in (0, 1/2]$, and any confidence $\delta \in (0, 1/2]$, if $\mathcal{H}$ has VC-dimension d then with probability $1 - \delta$, any $h \in \mathcal{H}$ that is consistent with c on $m \geq \frac{c_0}{\epsilon}(d \log \frac{1}{\epsilon} + \log \frac{1}{\delta})$ examples drawn independently from D satisfies $\Pr_{x \in D}[h(x) = c(x)] \geq 1 - \epsilon$.*

This bound is analogous to the Occam's Razor bound, and its proof follows the same basic outline, with some complications. The basic issue is that we can only use the VC-dimension to bound the number of labelings that may occur on a set of examples. The central idea is to first draw a data set, thus fixing the labels, and then randomly split it into "training" and "test" sets. It is very unlikely that a labeling of the data set induced by any $h \in \mathcal{H}$ could disagree with c on many of the examples, and that all of these examples fall into the "test" set:

Lemma A.12 *Let s be a set of $2m$ possible worlds, $r \subseteq s$, and suppose s is split into s_{train} and s_{test}, each of size m, uniformly at random. Then*

$$\Pr_{\substack{\text{split of } s \text{ into} \\ (s_{\text{train}}, s_{\text{test}})}} \left[r \cap s_{\text{train}} = \varnothing \wedge \frac{|r \cap s_{\text{test}}|}{m} \geq \frac{\epsilon}{2} \middle| s \right] \leq \frac{1}{2^{m\epsilon/2}}.$$

Proof Note that if $|r \cap s| > m$ we cannot have $r \cap s_{\text{train}} = \varnothing$. Let us just consider which set the members of $r \cap s$ are placed in during the split of s. Concretely, suppose we place them in random order, and take the first m elements to be s_{train}, and the final m to be s_{test}. There are $\binom{2m}{|r\cap s|}$ ways the list could be distributed among $r \cap s$ and $s \setminus r$. Out of these, only $\binom{m}{|r\cap s|}$ place all of $r \cap s$ into s_{test}. Since the order is chosen uniformly at random, the probability is exactly

$$\begin{aligned}\frac{\binom{m}{|r\cap s|}}{\binom{2m}{|r\cap s|}} &= \frac{m!}{|r\cap s|!(m-|r\cap s|)!}\frac{|r\cap s|!(2m-|r\cap s|)!}{(2m)!} \\ &= \prod_{i=0}^{|r\cap s|-1}\frac{m-i}{2m-i} \\ &\leq \frac{1}{2^{|r\cap s|}}\end{aligned}$$

Of course, if $|r \cap s| < \frac{\epsilon}{2}m$, the probability is zero since then s_{test} cannot have $\frac{\epsilon}{2}m$ members of $r \cap s$. Thus in any case, the probability is at most $1/2^{m\epsilon/2}$. □

Yet, the members of these sets are still independent draws from the distribution D. The fraction of examples in which a representation h disagrees with c on the "test" set is thus a good estimate of the true error of h at approximating c. We will therefore be able to conclude that it is unlikely that any of the $h \in \mathcal{H}$ with high error at approximating c could be consistent with the "training" set:

Lemma A.13 *Let two sets of* $m \geq \frac{8}{\epsilon}\ln 2$ *examples,* s_{train} *and* s_{test} *be drawn independently from* D*. Then*

$$\begin{aligned}\Pr_{s_{\text{train}},s_{\text{test}}}\left[\exists r \in \Delta_{\mathcal{H},\epsilon}(c) : r \cap s_{\text{train}} = \varnothing \wedge \frac{|s_{\text{test}} \cap r|}{m} \geq \frac{\epsilon}{2}\right] \\ \geq \frac{1}{2}\Pr_{s_{\text{train}}}\left[\exists r \in \Delta_{\mathcal{H},\epsilon}(c) : r \cap s_{\text{train}} = \varnothing\right]\end{aligned}$$

Proof We first use conditional probability to write

$$\Pr_{s_{\text{train}},s_{\text{test}}}\left[\exists r \in \Delta_{\mathcal{H},\epsilon}(c) : r \cap s_{\text{train}} = \varnothing \wedge \frac{|s_{\text{test}} \cap r|}{m} \geq \frac{\epsilon}{2}\right]$$

as

$$\begin{aligned}\Pr_{\substack{s_{\text{train}},\\ s_{\text{test}}}}\left[\exists r \in \Delta_{\mathcal{H},\epsilon}(c) : r \cap s_{\text{train}} = \varnothing \wedge \frac{|s_{\text{test}} \cap r|}{m} \geq \frac{\epsilon}{2}\middle| \exists r \in \Delta_{\mathcal{H},\epsilon}(c) : r \cap s_{\text{train}} = \varnothing\right] \\ \cdot \Pr_{s_{\text{train}}}\left[\exists r \in \Delta_{\mathcal{H},\epsilon}(c) : r \cap s_{\text{train}} = \varnothing\right].\end{aligned}$$

Now, for any $r \in \Delta_{\mathcal{H},\epsilon}(c)$, the probability that $\frac{|s_{\text{test}}\cap r|}{m} < \frac{\epsilon}{2}$ can be bounded as follows: $\mathbb{E}_{x\in D}[x \in r] \geq \epsilon$ by definition of $\Delta_{\mathcal{H},\epsilon}(c)$. Thus, a Chernoff bound gives

$$\Pr_{s_{\text{test}}}\left[\frac{|s_{\text{test}} \cap r|}{m} < \frac{\epsilon}{2}\right] < e^{-\frac{\epsilon m(1/2)^2}{2}}.$$

This quantity is less than $1/2$ when $m \geq \frac{8}{\epsilon} \ln 2$. Thus, supposing that any $r \in \Delta_{\mathcal{H},\epsilon}(c)$ has $r \cap s_{\text{train}} = \varnothing$, the probability that this r also satisfies $\frac{|s_{\text{test}} \cap r|}{m} \geq \frac{\epsilon}{2}$ is at least $1/2$. The claim follows. □

We now give the full formal version of the argument:

Proof of Theorem A.11 First observe that by Lemma A.13, if we can show

$$\Pr_{s_{\text{train}}, s_{\text{test}}}\left[\exists r \in \Delta_{\mathcal{H},\epsilon}(c) : r \cap s_{\text{train}} = \varnothing \wedge \frac{|s_{\text{test}} \cap r|}{m} \geq \frac{\epsilon}{2}\right] \leq \frac{\delta}{2},$$

then $\Pr_{s_{\text{train}}}[\exists r \in \Delta_{\mathcal{H},\epsilon}(c) : r \cap s_{\text{train}} = \varnothing] \leq \delta$, and so with probability $1 - \delta$ any consistent h thus satisfies $\Pr_{x \in D}[h(x) = c(x)] \geq 1 - \epsilon$. We thus turn to establishing the first bound.

We produce the two m example sets s_{train} and s_{test} by first drawing a set s of $2m$ examples from D and then splitting it into s_{train} and s_{test} uniformly at random. Note that since the examples are still i.i.d. draws from D, this has the same distribution as simply drawing two m element sets from D independently. Thus,

$$\begin{aligned}\Pr_{s_{\text{train}}, s_{\text{test}}}&\left[\exists r \in \Delta_{\mathcal{H},\epsilon}(c) : r \cap s_{\text{train}} = \varnothing \wedge \frac{|s_{\text{test}} \cap r|}{m} \geq \frac{\epsilon}{2}\right] \\ &= \mathbb{E}_s\left[\Pr_{\substack{\text{split of } s \text{ into} \\ (s_{\text{train}}, s_{\text{test}})}}\left[\exists r \in \Delta_{\mathcal{H},\epsilon}(c) : r \cap s_{\text{train}} = \varnothing \wedge \frac{|r \cap s_{\text{test}}|}{m} \geq \frac{\epsilon}{2} \middle| s\right]\right]\end{aligned}$$

where now we can equivalently consider the set of dichotomies of s induced by $\Delta_{\mathcal{H},\epsilon}(c)$ instead of $\Delta_{\mathcal{H},\epsilon}(c)$ itself; i.e., this expression is also equivalent to

$$\mathbb{E}_s\left[\Pr_{\substack{\text{split of } s \text{ into} \\ (s_{\text{train}}, s_{\text{test}})}}\left[\exists r \in \Pi_{\Delta_{\mathcal{H},\epsilon}(c)}(s) : r \cap s_{\text{train}} = \varnothing \wedge \frac{|r \cap s_{\text{test}}|}{m} \geq \frac{\epsilon}{2} \middle| s\right]\right].$$

By a union bound, thus is at most

$$\mathbb{E}_s\left[\sum_{r \in \Pi_{\Delta_{\mathcal{H},\epsilon}(c)}(s)} \Pr_{\substack{\text{split of } s \text{ into} \\ (s_{\text{train}}, s_{\text{test}})}}\left[r \cap s_{\text{train}} = \varnothing \wedge \frac{|r \cap s_{\text{test}}|}{m} \geq \frac{\epsilon}{2} \middle| s\right]\right],$$

and Lemma A.12 bounds the inner probability by $2^{-m\epsilon/2}$. Now the terms of the sum do not depend on r, and so we are simply picking up $2^{-m\epsilon/2}\,|\Pi_{\Delta_{\mathcal{H},\epsilon}(c)}(s)|$ times: i.e., the quantity is now at most

$$\mathbb{E}_s[|\Pi_{\Delta_{\mathcal{H},\epsilon}(c)}(s)| 2^{-m\epsilon/2}].$$

By Lemma A.7, $|\Pi_{\Delta_{\mathcal{H},\epsilon}(c)}(s)| \leq |\Pi_{\Delta_{\mathcal{H}}(c)}(s)| = |\Pi_{\mathcal{H}}(s)|$; $|\Pi_{\mathcal{H}}(s)|$ is in turn by definition at most the growth function for $\mathcal{H}$, $\Pi_{\mathcal{H}}(2m)$. Sauer's Lemma gives that this is at most $\left(\frac{2me}{d}\right)^d$. So in summary, it suffices to obtain

$$\left(\frac{2me}{d}\right)^d 2^{-m\epsilon/2} \leq \frac{\delta}{2}.$$

Taking logs and rearranging, this is equivalent to

$$\frac{m\epsilon}{2} \geq d \log_2 \frac{2me}{d} + \log_2 \frac{2}{\delta}. \tag{A.1}$$

We can remove the vexing "$\log_2 me$" expression as follows: since $1 + x \leq e^x$,

$$1 + (m\frac{\epsilon \ln 2}{4d} - 1) \leq e^{m\frac{\epsilon \ln 2}{4d} - 1}.$$

Taking logs again and rearranging,

$$\begin{aligned} \ln m + \ln \frac{\epsilon \ln 2}{4d} &\leq \frac{m\epsilon \ln 2}{4d} - \ln e \\ \Rightarrow \log_2(me) &\leq \log_2 \frac{4d}{\epsilon \ln 2} + \frac{m\epsilon}{4d}. \end{aligned}$$

We therefore find that if

$$m \geq \frac{4}{\epsilon}(d \log_2 \frac{8}{\epsilon \ln 2} + \log_2 \frac{2}{\delta}),$$

$$\begin{aligned} \frac{m\epsilon}{4} &\geq d \log_2 \frac{8}{\epsilon \ln 2} + \log_2 \frac{2}{\delta} \\ \Rightarrow \frac{m\epsilon}{2} &= d\left(\log_2 \frac{4d}{\epsilon \ln 2} + \frac{m\epsilon}{4d} + \log_2 \frac{2}{d}\right) + \log_2 \frac{2}{\delta} \\ &\geq d \log_2 \frac{2me}{d} + \log_2 \frac{2}{\delta} \qquad \text{by above} \end{aligned}$$

yielding Equation A.1. Now, we observe that $c_0 = 16$ suffices since then if

$$m \geq \frac{16}{\epsilon}(d \log_2 \frac{1}{\epsilon} + \log_2 \frac{1}{\delta}),$$

$$\begin{aligned} m &\geq \frac{4}{\epsilon}(d \log_2 \frac{1}{\epsilon} + 3d \log_2 \frac{1}{\epsilon} + \log_2 \frac{1}{\delta} + 3 \log_2 \frac{1}{\delta}) \\ &> \frac{4}{\epsilon}(d \log_2 \frac{1}{\epsilon} + d \log_2 \frac{8}{\ln 2} + \log_2 \frac{1}{\delta} + \log_2 2) \end{aligned}$$

which is thus sufficient. □

We note that the bound obtained in Theorem A.11 is *not* optimal: actually, $O(\frac{1}{\epsilon}(d + \log \frac{1}{\delta}))$ examples suffice to learn properties of VC-dimension d, which asymptotically matches the lower bound we obtain using Theorem A.4 and Exercise A.7. The proof of this sharper bound is unfortunately much more complicated, and we do not present it here. We include references in the chapter notes.

A.4 Learning and optimization

VC-dimension quantifies how much data is necessary and sufficient for learning a given family of properties. It is, in particular, sometimes significantly sharper than Occam's Razor (Theorem 4.2). Perhaps the most important example of the difference between

the two is that only the VC-dimension bounds give generalization guarantees for linear threshold functions (also known as halfspaces), introduced in Definition 4.24: that is, properties defined by a linear inequality of the features, $h = \{\vec{x} \in \mathbb{Q}^n : \langle \vec{w}, \vec{x} \rangle \geq t\}$ for $\vec{w} \in \mathbb{Q}^n$ and $t \in \mathbb{Q}$. Even when we focus on rational parameters rather than real values (as we will for this discussion), there are still infinitely many halfspaces on our n attributes, and hence no finite bound on the representation length. Occam's razor thus cannot establish learnability of halfspaces on rational points. Nevertheless, halfspaces on n attributes have VC-dimension $n + 1$ (Exercise A.2), and therefore Theorem A.11 allows us to conclude that in spite of the lack of a prior bound on the representation size, it suffices to find a halfspace that is consistent with $O(\frac{1}{\epsilon}(n \log \frac{1}{\epsilon} + \log \frac{1}{\delta}))$ examples.[1]

We now turn to the problem of finding the parameters of a halfspace that is consistent with a set of examples. We will see that this problem is tightly connected to a fundamental problem in optimization known as *linear programming*:

Definition A.14 *Linear programming* is the following search problem: given a matrix $A \in \mathbb{Q}^{m \times n}$ and a vector $\vec{b} \in \mathbb{Q}^m$, find $\vec{x} \in \mathbb{Q}^n$ such that for all i $\langle \vec{a}_i, \vec{x} \rangle \geq b_i$ (which we will write as $A\vec{x} \geq \vec{b}$) or return $\perp$ if no such *feasible* $\vec{x}$ exists. (The associated decision problem is known as the *linear program feasibility* problem.)

We remark that the problem is usually formulated with a linear objective function that we wish to minimize or maximize over the feasible region. This seemingly more sophisticated variant may be solved by reduction to the above "feasibility" problem in polynomial time, but we will not discuss this here. Polynomial time algorithms for linear programming are known, but they are nontrivial; see the chapter notes for further reading on this problem.

We will connect this problem to the problem of learning halfspaces, as captured by solving $\text{Consis}_{\text{Halfspace}}$: the problem of finding a halfspace that is consistent with a given set of examples. By Theorem A.11 this is sufficient and we recall from Theorem 4.11 that finding a consistent halfspace is also necessary for proper learning. Intuitively, the problem of finding a consistent halfspace is a problem with variables $w_1, \ldots, w_n$ and t, where for each positive example $\vec{x}^{(i)}$, we should have $\langle \vec{w}, \vec{x}^{(i)} \rangle \geq t$ and for each negative example $\vec{x}^{(i')}$ we have $\langle \vec{w}, \vec{x}^{(i')} \rangle < t$. We would like to write a linear program in which we have such a constraint for each example. The difficulty is that linear programming does not use strict inequality constraints. If we simply relaxed the strict inequality to nonstrict (from $<$ to $\leq$), the resulting linear program has a trivial solution in which each w_i and t are all 0, that incorrectly labels all examples as positive.

We can address this issue by enforcing a positive "margin." The *margin* of a halfspace $\langle \vec{w}, \vec{x} \rangle \geq t$ on a given data set $(\vec{x}^{(1)}, \ldots, \vec{x}^{(m)})$ is $\min_i |\langle \vec{w}, \vec{x}^{(i)} \rangle - t|$. If the halfspace has a positive margin, *both* inequalities hold strictly. We observe that we can rescale all of the parameters of the halfspace by a common factor without changing the set of points: $\langle \vec{w}, \vec{x} \rangle \geq t$ iff $\langle s\vec{w}, \vec{x} \rangle \geq st$ for any $s > 0$; and the result of this is that the margin is also rescaled by s: $\min_i |\langle s\vec{w}, \vec{x}^{(i)} \rangle - st| = s \cdot \min_i |\langle \vec{w}, \vec{x}^{(i)} \rangle - t|$ so the

[1] Again, actually, it is known that $O(\frac{1}{\epsilon}(n + \log \frac{1}{\delta}))$ examples suffice; see chapter notes.

margin is only meaningful if we have fixed a scale. Or, in our formulation, we can find a set of parameters scaled to give us any fixed positive margin, for example, 1.

Given our labeled data set $(\vec{x}^{(1)}, c(\vec{x}^{(1)})), \ldots, (\vec{x}^{(m)}, c(\vec{x}^{(m)}))$ we will use the following linear program:

$$\begin{aligned} \langle(\vec{w}, t), (\vec{x}^{(i)}, -1)\rangle &\geq 1 \qquad & i : c(\vec{x}^{(i)}) = 1 \\ \langle(\vec{w}, t), (-\vec{x}^{(i)}, 1)\rangle &\geq 1 \qquad & i : c(\vec{x}^{(i)}) = 0 \end{aligned} \tag{A.2}$$

Observe that any solution to this linear program satisfies $\langle \vec{w}, \vec{x}^{(i)} \rangle \geq t + 1 > t$ for positive examples $\vec{x}^{(i)}$ and $\langle \vec{w}, \vec{x}^{(i)} \rangle \leq t - 1 < t$ for negative examples. Thus, it is a solution to $\text{Consis}_{\text{Halfspace}}$. At the same time, if there is a solution to $\text{Consis}_{\text{Halfspace}}$, then there must exist a solution to the above linear program: since the negative examples must be satisfied strictly, there must be a positive margin for these examples, $s = \min_{i:c(\vec{x}^{(i)})=0}\{t - \langle \vec{w}, \vec{x}^{(i)} \rangle - t\} > 0$. Now, consider $\tilde{t} = \frac{2}{s}t - 1$ and $\tilde{\vec{w}} = \frac{2}{s}\vec{w}$. On the negative examples, we still have

$$\min_{i:c(\vec{x}^{(i)})=0}\{\tilde{t} - \langle \tilde{\vec{w}}, \vec{x}^{(i)} \rangle\} = \frac{2}{s} \cdot s - 1 = 1$$

and on the positive examples we have

$$\min_{i:c(\vec{x}^{(i)})=1}\{\langle \tilde{\vec{w}}, \vec{x}^{(i)} \rangle - \tilde{t}\} = \min_{i:c(\vec{x}^{(i)})=1}\{\frac{2}{s}\langle \vec{w}, \vec{x}^{(i)} \rangle - \frac{2}{s}t + 1\} \geq 1$$

since $\min_{i:b^{(i)}=1}\{\frac{2}{s}\langle \vec{w}, \vec{x}^{(i)} \rangle - \frac{2}{s}t\} \geq 0$. Hence indeed, $\tilde{\vec{w}}$ and $\tilde{t}$ are solutions. So we can solve $\text{Consis}_{\text{Halfspace}}$ by generating the above linear program and passing it to an algorithm for linear programming. This is a "mapping reduction" (it generates a single instance of linear programming, for which the solutions may be used verbatim) that is obtained by a simple syntactic transformation of the examples into constraints.

There is also a slightly more complex reduction from linear programming to $\text{Consis}_{\text{Halfspace}}$, that similarly involves solving a single instance of $\text{Consis}_{\text{Halfspace}}$. We are given $A \in \mathbb{Q}^{m \times n}$ and $\vec{b} \in \mathbb{Q}^m$, and we will create a $n + 1$-dimensional data set as follows: we first create a positive example for each linear constraint

$$((-\vec{a}_{(\cdot,j)}, b_j), 1) \qquad j = 1, \ldots, m \tag{A.3}$$

Next, we create a negative example $((\vec{0}, 1), 0)$ and a positive example $((\vec{0}, 0), 1)$. Supposing that the linear program has a feasible solution $\vec{x}^*$, the linear threshold function with $\vec{w} = (-\vec{x}^*, -1)$ and $t = 0$ obtains $\langle(-\vec{a}_{(\cdot,j)}, b_j), (-\vec{x}^*, -1)\rangle = \langle \vec{a}_{(\cdot,j)}, \vec{x}^* \rangle - b_j \geq 0$ for $j = 1, \ldots, m$, and hence is easily seen to be consistent with this set of examples. We thus solve $\text{Consis}_{\text{Halfspace}}$ on this set of examples to obtain the parameters of a consistent halfspace $(\vec{w}, \vec{t})$ and return the solution $\vec{x}$ with $x_i = \frac{w_i}{w_{n+1}}$ for all i.

Together, the constraints ensure that we obtain a linear threshold function given by $\vec{w}$ and t such that

$$\sum_{i=1}^{n} w_i a_{(i,j)} - w_{n+1} b_j \leq -t$$

and $w_{n+1} < t \leq 0$. So, if we divide by $w_{n+1} < 0$, we obtain

$$\sum_{i=1}^{n} \frac{w_i}{w_{n+1}} a_{(i,j)} - b_j \geq \frac{t}{w_{n+1}} > 0$$

so indeed, our choice of $\vec{x}$ satisfies $A\vec{x} > \vec{b}$. In conclusion we have

Theorem A.15 *There are polynomial-time reductions between linear programming and* $\text{Consis}_{\text{Halfspace}}$ *that use a single instance of the respective problems.*

More strongly, we see that learning halfspaces and linear programming are essentially the same problem, with examples corresponding directly to constraints. In terms of representation power, halfspaces are surprisingly powerful: they can express decision lists (see Exercise 4.5) and indeed, combined with various appropriate choices of feature spaces (e.g., Exercise 4.4) they are essentially the most expressive class of representations known to be efficiently learnable, cf. Figure 4.2.

A.5 Learning curves and exponential rates

We now consider the question of *how the accuracy of a learned representation scales as a function of the number of examples seen*. Plots of this function are known as *learning curves*, and are often estimated in experiments. The VC-dimension bounds, Theorem A.11 (or more precisely, Equation A.1) and Theorem A.4, seem to suggest that as we draw m examples, at a fixed confidence level, we should obtain an error rate between $O(\frac{d}{m} \log \frac{m}{d})$ and $\Omega(\frac{d}{m})$ for a family of representations of VC-dimension d. In general, this may not be correct: we may see the error shrink at a rate of $\frac{1}{e^{\Omega(m)}}$ in expectation.

How this may occur is most easily seen if we return to the setting where the possible worlds are described by n Boolean attributes. Because of the way the question is now framed, there is a *fixed* probability distribution on the 2^n possible worlds. Of the actual possibilities, i.e., occurring with nonzero probability, there is a minimum probability – and hence, the probability that we do not see even these unlikely worlds drops exponentially as we draw more examples. And, if we manage to see all of the worlds, we obtain perfect accuracy. This will be enough to show that the expected error rate is decaying exponentially with the number of examples: the error is only nonzero in an event of exponentially decaying probability.

Before we revisit this argument in formal terms, let's understand why the VC-dimension bounds obtain such a different rate. The VC-dimension bound and Occam's Razor present a guarantee that says *no matter what probability distribution on the possible worlds the examples are drawn from*, a given number of examples is enough to attain accuracy $1 - \epsilon$ with confidence $1 - \delta$. The No Free Lunch Theorem, Theorem A.4, says that we cannot significantly improve such a guarantee. But, it does so by considering a *different distribution for each possible error rate*. Here, since we are instead supposing that we are interested in some particular, *fixed* probability

distribution on possible worlds, the argument that established the No Free Lunch Theorem fails to hold and we obtain a faster rate.

Theorem A.16 *Let $\mathcal{H}$ be any class of representations and D be any distribution on a finite set of possible worlds $\mathcal{X}$. Let $p_{\min}$ denote the smallest nonzero probability D assigns to any possible world x that satisfies some $h_1 \in \mathcal{H}$ and does not satisfy some other $h_0 \in \mathcal{H}$. Then for m examples drawn from D and labeled by $c \in \mathcal{H}$, any $h \in \mathcal{H}$ that is consistent with these has error at most $(1 - p_{\min})^m$ in expectation.*

Proof For any $h \in \mathcal{H}$, we can write the error rate of h at representing c on D as

$$\sum_{x \in \mathcal{X}} \Pr_D[x] I[h(x) \neq c(x)]$$

where $I[h(x) \neq c(x)] = \begin{cases} 1 & \text{if } h(x) \neq c(x) \\ 0 & \text{otherwise} \end{cases}$. Now, we have restricted our attention to h that are consistent with our m examples. For any $x \in \mathcal{X}$ where some $h \in \mathcal{H}$ can disagree with the given $c \in \mathcal{H}$, the probability that x does not appear in the set of m independent examples drawn from D is precisely $(1 - \Pr_D[x])^m \leq (1 - p_{\min})^m$. Now, let H be a random variable that is a member of $\mathcal{H}$ that is consistent with the sample of m examples attaining the highest error rate for c under D. We have, for any fixed world x, $\Pr_H[H(x) \neq c(x)] \leq (1 - p_{\min})^m$. Therefore,

$$\begin{aligned}
\mathop{\mathbb{E}}_{x^{(1)},\ldots,x^{(m)}}\Big[\max_{\substack{h \in \mathcal{H}: \\ h(x^{(1)})=c(x^{(1)}) \\ \vdots \\ h(x^{(m)})=c(x^{(m)})}} [\Pr[h(x) \neq c(x)]]\Big] &= \mathop{\mathbb{E}}_{x^{(1)},\ldots,x^{(m)}}[\Pr_D[H(x) \neq c(x)]] \\
&= \mathop{\mathbb{E}}_{x^{(1)},\ldots,x^{(m)}}\left[\sum_{x \in \mathcal{X}} \Pr_D[x] I[H(x) \neq c(x)]\right] \\
&= \sum_{x \in \mathcal{X}} \Pr_D[x] \mathop{\mathbb{E}}_{x^{(1)},\ldots,x^{(m)}}[I[H(x) \neq x(x)]] \\
&\leq \sum_{x \in \mathcal{X}} \Pr_D[x] \mathop{\mathbb{E}}_{x^{(1)},\ldots,x^{(m)}}[(1 - p_{\min})^m] \\
&= (1 - p_{\min})^m
\end{aligned}$$

□

Actually, by instead simply using $\Pr_H[H(x) \neq c(x)] \leq (1 - \Pr_{X \in D}[X = x])^m$, we could obtain the slightly sharper bound $\mathbb{E}_{x \in D}[(1 - \Pr_{X \in D}[X = x])^m]$, where $p_{\min}(1 - p_{\min})^m$ is the asymptotically dominant term in the sum. In general, this bound is asymptotically tight:

Theorem A.17 *Let $\mathcal{H}$ be any class of representations of VC-dimension at least two on a set of possible worlds $\mathcal{X}$, and let H_m denote a random function that, given m labeled possible worlds $(x^{(1)}, b^{(1)}), \ldots, (x^{(m)}, b^{(m)})$ from $\mathcal{X}$ returns some member of $\mathcal{H}$. Then for every $p_{\min} \in (0, 1/2)$ there is some $c \in \mathcal{H}$ and distribution D on $\mathcal{X}$ such that $p_{\min}$*

is the minimum nonzero probability of some x under D for which $h_0(x) \neq h_1(x)$ for some $h_0, h_1 \in \mathcal{H}$ and

$$\mathop{\mathbb{E}}_{x^{(1)},\ldots,x^{(m)}}[\Pr_D[H_m(x) \neq c(x)]] \geq \frac{p_{\min}}{2}(1 - p_{\min})^m.$$

Proof Since $\mathcal{H}$ has VC-dimension at least two, there exists a shattered set S of at least two possible worlds, x and y. There exist $c_0 \in \mathcal{H}$ such that $c_0(x) = 0$ and $c_0(y) = 0$, and c_1 such that $c_1(x) = 1$ and $c_1(y) = 0$.

Let D be a probability distribution that gives x with probability $p_{\min}$ and y with probability $1 - p_{\min}$. $p_{\min} \leq 1/2 \leq 1 - p_{\min}$, so indeed $p_{\min}$ is the minimum nonzero probability of this distribution and by construction, $c_0(x) \neq c_1(x)$.

Now, with probability $(1 - p_{\min})^m$, m independent draws from D give y every time, where $c_0(y) = c_1(y) = 0$. H_m, when given such a set of labeled examples, cannot agree with both c_0 and c_1 on x with probability greater than $1/2$. For one of these, it therefore must give an error rate on D of at least $\frac{p_{\min}}{2}$. Since the error rate is always nonnegative, the expected error rate is thus at least $\frac{p_{\min}}{2}(1 - p_{\min})^m$. □

Although these rather general bounds essentially settle the theoretical question for our discrete domains, they are not very relevant in practice. $p_{\min}$ is usually negligibly small—and thus, in turn, $(1 - p_{\min})^m$ is quite large until $m > 1/p_{\min}$, which is usually enormous. But, exponential learning curves are sometimes observed in practice, even for moderate sample sizes.

We briefly mention that in the case of infinite domains, it is possible that the error rate only decays at a rate of $1/\Omega(m)$, or even arbitrarily slowly, but that these three – exponential, linear, or arbitrarily slow rates – are the only possible asymptotic rates. (See the chapter notes for further reading on this topic.)

A.6 Boosting: improper learnability is all-or-nothing

Our focus in this chapter so far has been almost entirely on the sample complexity of learning. In particular, we found that representations using numeric parameters don't need to suffer a cost in their sample complexity that grows with the representation size of those numbers, but rather generally only need a number of examples that grows with the VC-dimension of the representations. In this section, we'll make use of these representations to answer a fundamental question about computational learnability.

We may motivate the question by the following line of reasoning: we know that in order to learn families of bounded VC-dimension, it suffices to find a representation that is consistent with a sufficiently large set of examples. But, as we saw in Section 4.7, for representations that are seemingly only slightly more powerful than halfspaces, finding such a consistent representation is computationally intractable given standard assumptions. If we view the problem of finding a consistent representation as a solution to an optimization problem – finding a representation with a minimum number of errors – then we may be inspired by the success of approximation algorithms for such problems. For many intractable problems like 3SAT, there are efficient approximation

algorithms with nontrivial guarantees, such as satisfying a constant fraction of the clauses. We may hope to at least design an efficient algorithm that can analogously find a representation that agrees with the labels of the given examples some nontrivial fraction of the time, even if we cannot get arbitrarily close to perfect prediction. We know that for Boolean properties, either at least half of the examples must be true or at least half must be false; this may be viewed as giving a trivial "1/2-approximation" using the constant properties (all possible worlds or none of them). *When can we do better than this?*

We may also approach this question from the opposite direction. In Section 4.7 we also saw that hard learning problems could be used to obtain encryption schemes. Indeed, in general, if it is *not* possible for a polynomial-time algorithm to obtain error rates significantly lower than 50% for some family of representations C, then by agreeing on some $c \in C_n$ for some n, Alice and Bob may communicate bits by exchanging points in X^n sampled from some suitably hard distribution for C_n, interpreting $c(\vec{x}) \in \{0,1\}$ as a 1-bit message. The scheme would then be semantically secure as a private-key scheme since any adversary algorithm that violated semantic security for such encryption (as in Definition 4.16, but without a public key) would predict c with error measurably below 50%. *Which learning problems are hard enough for such cryptographic uses?*

The answer may be surprising: the first approach yields nothing, in that whenever it is possible to efficiently learn representations that give error rates below 50%, it is possible to combine these representations to obtain one with error rate as close to zero as desired. In other words, for any class of representations, either there is an efficient PAC-learning algorithm, or else the learning task is hard enough that the representations could (in principle) be used as keys for encryption. There is no middle ground.

We begin by formalizing the property of interest, *"weak"* learnability:

Definition A.18 Let C_n and $\mathcal{H}_n$ be classes of representations on the same n attributes. We say that C_n is *weakly learnable with advantage* $\alpha(n)$ using $\mathcal{H}_n$ in time $t(n, \frac{1}{\delta})$ and sample complexity $m(n, \frac{1}{\delta})$ if there is an algorithm such that for any m examples drawn independently from a distribution D on n attribute possible worlds and labeled by some $c \in C_n$, given these examples and the parameter δ, with probability at least $1-\delta$ outputs a representation $h \in \mathcal{H}_n$ such that

$$\Pr_{\vec{x} \in D}[h(\vec{x}) = c(\vec{x})] \geq \frac{1}{2} + \alpha(n).$$

If C_n is weakly learnable with advantage $1/p(n)$ for some strictly positive polynomial $p(n)$ (i.e., $p(n) > 0$ for all n) in polynomial time and sample complexity by any class $\mathcal{H}_n$ for which evaluation can be computed in polynomial time, then we simply say that C_n is *efficiently weakly learnable*.

With this definition in hand, we can formally pose the question of whether there is a difference between the family of representations that are efficiently PAC-learnable and

the properties that are efficiently weakly learnable. As we alluded to above, the answer is that there is no difference:

Theorem A.19 *For any class of representations C, if C is efficiently weakly learnable, it is efficiently PAC-learnable. In particular, if it is efficiently weakly learnable using the family of representations $\mathcal{H}$, it is efficiently PAC-learnable using* $\mathrm{THR} \circ \mathcal{H}$.

Naturally, this is established by giving a reduction from PAC-learning to weak learning. These are known as *"boosting"* algorithms, since they "boost" the accuracy from $1/2 + \alpha$ for some small but nonnegligible $\alpha > 0$ to $1 - \epsilon$ for any desired $\epsilon > 0$. Interestingly, the following algorithm, *AdaBoost* (Algorithm 24), is actually quite successful in practical settings. Note that we'll use $\tilde{c}$ to denote a $\{-1, 1\}$-valued version of c, given by mapping 0 to -1 and 1 to 1, and similarly for other representations.

Input: Examples $((\vec{x}^{(1)}, c(\vec{x}^{(1)})), \ldots, (\vec{x}^{(m)}, c(\vec{x}^{(m)}))$, integer parameter T, weak learner L for C using $\mathcal{H}$.
Output: h in $\mathrm{THR} \circ \mathcal{H}_n$.
begin
  Initialize D_1 to the uniform distribution on the input examples.
  for $t = 1, \ldots, T$ **do**
    Sample $(\vec{x}^{(1,t)}, c(\vec{x}^{(1,t)})), \ldots, (\vec{x}^{(1,t)}, c(\vec{x}^{(m,t)}))$ independently from D_t.
    $\tilde{r}_t \leftarrow L((\vec{x}^{(1,t)}, c(\vec{x}^{(1,t)})), \ldots, (\vec{x}^{(1,t)}, c(\vec{x}^{(m,t)})))$ $(\tilde{r}_i : \mathcal{X}^n \to \{-1, 1\})$
    Let ϵ_t denote $\frac{1}{m} \sum_{i=1}^{m} D_t(\vec{x}^{(i)}, c(\vec{x}^{(i)}))[r_t(\vec{x}^{(i)}) \neq c(\vec{x}^{(i)})]$
    Put $w_t \leftarrow \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$
    Put $h_t(\vec{x}) \leftarrow [\sum_{i=1}^{t} w_i \tilde{r}_i(\vec{x}) \geq 0]$
    Put D_{t+1} equal to the distribution that assigns $(x^{(i)}, c(x^{(i)}))$ probability $D_t(x^{(i)}, c(x^{(i)}))e^{-w_t \tilde{r}_t(\vec{x}^{(i)})\tilde{c}(\vec{x}^{(i)})}/Z_t$ for $Z_t \leftarrow 2\sqrt{\epsilon_t(1-\epsilon_t)}$.
  end
  return h_T.
end

Algorithm 24: AdaBoost Algorithm for converting weak learners to PAC-learners

At a higher level, we can informally describe and motivate AdaBoost as follows:

1. Starting from the empirically observed distribution on examples D_1, invoke the weak learner on the current distribution D_t to obtain a weak representation r_t.
2. Choose a weight w_t for $\tilde{r}_t$ in h_t to minimize the following convex upper bound on $\frac{1}{m} \sum_{i=1}^{m} [h_t(\vec{x}^{(i)}) \neq c(\vec{x}^{(i)})]$, i.e., the empirical error:

$$\frac{1}{m} \sum_{i=1}^{m} e^{-\tilde{c}(\vec{x}^{(i)}) \sum_{j=1}^{t} w_j \tilde{r}_j(\vec{x}^{(i)})} \tag{A.4}$$

Note that it is indeed an upper bound since if h_t disagrees with c on $\vec{x}^{(i)}$, $-\tilde{c}(\vec{x}^{(i)}) \sum_{j=1}^{t} w_j \tilde{r}_j(\vec{x}^{(i)}) \geq 0$ which gives $e^{-\tilde{c}(\vec{x}^{(i)}) \sum_{j=1}^{t} w_j \tilde{r}_j(\vec{x}^{(i)})} \geq 1$.

3. Choose D_{t+1} to eliminate the advantage of r_t by scaling up the total probability of incorrectly classified points to $1/2$, so that the weak learning algorithm must obtain a different representation.
4. Repeat the above for T rounds.

Regarding step 2, we first observe that the distribution D_t may be unwound to obtain $D_t(x^{(i)}, c(x^{(i)})) = e^{-\sum_{j=1}^{t} w_j \tilde{r}_j(\vec{x}^{(i)})\tilde{c}(\vec{x}^{(i)})} / \prod_{j=1}^{t} Z_j$. So, if we set the derivative of our convex upper bound w.r.t. w_t to 0, we may rewrite it in terms of ϵ_t as follows:

$$\frac{1}{m}\sum_{i:\tilde{c}(\vec{x}^{(i)})=\tilde{r}_t(\vec{x}^{(i)})} -e^{-\tilde{c}(\vec{x}^{(i)})\sum_{j=1}^{t} w_j \tilde{r}_j(\vec{x}^{(i)})} + \frac{1}{m}\sum_{i:\tilde{c}(\vec{x}^{(i)})\neq\tilde{r}_t(\vec{x}^{(i)})} e^{-\tilde{c}(\vec{x}^{(i)})\sum_{j=1}^{t} w_j \tilde{r}_j(\vec{x}^{(i)})} = 0$$

$$\frac{1}{m}\sum_{i:\tilde{c}(\vec{x}^{(i)})=\tilde{r}_t(\vec{x}^{(i)})} -e^{-w_t} D_t(x^{(i)}, c(x^{(i)})) \prod_{j=1}^{t} Z_j + \frac{1}{m}\sum_{i:\tilde{c}(\vec{x}^{(i)})\neq\tilde{r}_t(\vec{x}^{(i)})} e^{w_t} D_t(x^{(i)}, c(x^{(i)})) \prod_{j=1}^{t} Z_j = 0$$

$$-(1-\epsilon_t)e^{-w_t} + \epsilon_t e^{w_t} = 0$$

$$\Rightarrow 2w_t = \log\frac{1-\epsilon_t}{\epsilon_t}$$

Note that for Step 3 in the informal description, first observe D_{t+1} is indeed a distribution since D_t is and hence

$$\begin{aligned}
\sum_{i=1}^{m} D_{t+1}(x^{(i)}, c(x^{(i)})) &= \sum_{i:\tilde{c}(\vec{x}^{(i)})=\tilde{r}_t(\vec{x}^{(i)})} D_t(x^{(i)}, c(x^{(i)}))\frac{e^{-w_t}}{Z_t} + \sum_{i:\tilde{c}(\vec{x}^{(i)})\neq\tilde{r}_t(\vec{x}^{(i)})} D_t(x^{(i)}, c(x^{(i)}))\frac{e^{w_t}}{Z_t} \\
&= \frac{1}{Z_t}[(1-\epsilon_t)e^{-w_t} + \epsilon_t e^{w_t}] \\
&= \frac{1}{Z_t}[\sqrt{(1-\epsilon_t)\epsilon_t} + \sqrt{\epsilon_t(1-\epsilon_t)}] \qquad \text{(By choice of } w_t\text{)} \\
&= 1.
\end{aligned}$$

It satisfies the stated condition since the first equation in the calculation for the previous step may also be rewritten in terms of D_{t+1} as

$$\frac{1}{m}\sum_{i:\tilde{c}(\vec{x}^{(i)})=\tilde{r}_t(\vec{x}^{(i)})} -D_{t+1}(x^{(i)}, c(x^{(i)})) \prod_{j=1}^{t} Z_j + \frac{1}{m}\sum_{i:\tilde{c}(\vec{x}^{(i)})\neq\tilde{r}_t(\vec{x}^{(i)})} D_{t+1}(x^{(i)}, c(x^{(i)})) \prod_{j=1}^{t} Z_j = 0$$

which immediately gives

$$\sum_{i=1}^{m} D_{t+1}(x^{(i)}, c(x^{(i)}))[\tilde{c}(\vec{x}^{(i)}) \neq \tilde{r}_t(\vec{x}^{(i)})] = \sum_{i=1}^{m} D_{t+1}(x^{(i)}, c(x^{(i)}))[\tilde{c}(\vec{x}^{(i)}) = \tilde{r}_t(\vec{x}^{(i)})]$$

i.e., r_t has a 50% error rate under D_{t+1}.

Furthermore, it is also easy to see that our convex upper bound on the error – and hence the empirical error itself – is shrinking exponentially across rounds:

Lemma A.20 *Suppose the weak learner always obtains advantage $\alpha > 0$. Then the bound $\frac{1}{m}\sum_{i=1}^{m} e^{-\tilde{c}(\vec{x}^{(i)})\sum_{j=1}^{t} w_j \tilde{r}_j(\vec{x}^{(i)})}$ is at most $\frac{1}{m}e^{-2t\alpha^2}$.*

Proof Using our formula for D_t, we find that the bound is at most

$$\frac{1}{m}\sum_{i=1}^{m} e^{-\tilde{c}(\vec{x}^{(i)})\sum_{j=1}^{t} w_j \tilde{r}_j(\vec{x}^{(i)})} = \frac{1}{m}\sum_{i=1}^{m} D_t(x^{(i)}, c(x^{(i)})) \cdot \prod_{j=1}^{t} Z_j = \frac{1}{m}\prod_{j=1}^{t} Z_j$$

since $\sum_{i=1}^{m} D_t(x^{(i)}, c(x^{(i)})) = 1$; since $\epsilon_j \leq \frac{1}{2} - \alpha$ for all j, $Z_j \leq 2\sqrt{(\frac{1}{2}-\alpha)(\frac{1}{2}+\alpha)} = \sqrt{1-4\alpha^2}$, and thus

$$\prod_{j=1}^{t} Z_j \leq (1-4\alpha^2)^{t/2} \leq e^{-2t\alpha^2}.$$

□

The only remaining issue is that the complexity of the combined hypothesis grows with the number of rounds of boosting, so we need to take care that we can ensure generalization. We first show how to bound the VC-dimension for members of the combined class THR $\circ\, \mathcal{H}$ given a fixed number of rounds T:

Lemma A.21 *If $\mathcal{H}$ has VC-dimension $d \geq 2$ and $T \geq 2$, $\mathrm{THR}_T \circ \mathcal{H}$ has VC-dimension at most $2(d+1)T\log_2 2(d+1)T$.*

Proof For the VC-dimension d' of $\mathrm{THR}_T \circ \mathcal{H}$, we know that we must obtain all $2^{d'}$ dichotomies on some shattered set of d' points (Lemma A.9). Each of the T members of $\mathcal{H}$ have VC-dimension d, and hence by Sauer's Lemma induce at most $\Pi_{\mathcal{H}}(d') \leq \sum_{i=0}^{d}\binom{d'}{i} \leq \left(\frac{d'e}{d}\right)^d$ dichotomies on any set of d' points. Note that we may show (see Exercise A.2) that THR_T has VC-dimension T, and so by Sauer's Lemma again, the combining threshold function has at most $\left(\frac{d'e}{T}\right)^T$ dichotomies on the d' points obtained by the image of any fixed set of T representations from $\mathcal{H}$. Thus, the total number of dichotomies of the composition class is at most $\left(\frac{d'e}{d}\right)^{dT}\left(\frac{d'e}{T}\right)^T$. Since this must be at least $2^{d'}$, we find

$$\begin{aligned}
2^{d'} &\leq \left(\frac{d'e}{d}\right)^{dT}\left(\frac{d'e}{T}\right)^T \\
d' &\leq dT\log_2\frac{d'e}{d} + T\log_2\frac{d'e}{T} \\
&\leq (d+1)T\log_2\frac{d'e}{2} \text{ Since we assumed } d, T \geq 2
\end{aligned}$$

We can address the presence of d' on the RHS similarly to how we handled Equation A.1: we know $1+\frac{d'\ln 2}{2(d+1)T} - 1 \leq e^{\frac{d'\ln 2}{2(d+1)T}-1}$, so taking logs, $\ln\frac{d'\ln 2}{2(d+1)T} \leq \frac{d'\ln 2}{2(d+1)T} - 1$; we may thus obtain

$$\begin{aligned}
d' &\leq (d+1)T\frac{1}{\ln 2}\left(\ln\frac{(d+1)T}{\ln 2} + \frac{d'\ln 2}{2(d+1)T}\right) \\
\Rightarrow d' &\leq 2(d+1)T\log_2 2(d+1)T.
\end{aligned}$$

□

It is then straightforward to establish the equivalence between PAC-learning and weak learning:

Theorem A.22 *There is a constant c_1 such that if there is an efficient weak learner for the class of representations $\mathcal{C}$ using a class $\mathcal{H}$ of VC-dimension d, obtaining advantage $\alpha(n)$, then given $m \geq \frac{c_1}{\epsilon}(\frac{d}{\alpha(n)^2}\log\frac{d}{\alpha(n)}\log\frac{1}{\epsilon}+\log\frac{1}{\delta})$ examples, running AdaBoost for $T = \frac{1}{2\alpha(n)^2}$ rounds with the given weak learner is an efficient PAC-learning algorithm for $\mathcal{C}$ using* $\mathrm{THR} \circ \mathcal{H}$.

Proof We first observe that once the convex upper bound is below $1/m$, AdaBoost must have zero empirical error. Hence, taking $T \geq \frac{1}{2\alpha(n)^2}$ in Lemma A.20 suffices for AdaBoost to find a consistent representation in $\mathrm{THR} \circ \mathcal{H}$.

Now, Lemma A.21 gives us that for this choice of T, the VC-dimension of the composed class is at most $\frac{d+1}{\alpha(n)^2}\log_2\frac{d+1}{\alpha(n)^2}$ so the claim follows from Theorem A.11. □

We remark that in Lemma A.20, we assumed that the weak learner always succeeds. Since we are considering a known distribution on a fixed set of examples, this is generally not an issue. In any case, we can always check whether the weak learning algorithm has obtained an error rate sufficiently less than $1/2$, and re-run it if it fails.

Chapter notes

The VC-dimension was introduced by Vapnik and Chervonenkis (1971), and the contents of Sections A.1 and A.3 were essentially first presented in that work. Strictly speaking, Vapnik and Chervonenkis were concerned about uniform bounds on the accuracy of sampling-based estimates of the expected value of a family of events. Subsequently, Blumer et al. (1989) applied these VC-dimension generalization bounds to the error indicator functions of a class of representations, thus interpreting the bounds of Vapnik and Chervnonenkis as sample complexity bounds for supervised learning, as we present here. Specifically, the application of VC-dimension bounds to learning halfspaces using linear programming in Section A.4 is from Blumer et al. (1989). Sauer's Lemma is from Sauer (1972); Vapnik and Chervonenkis (1971) and Shelah (1972) all independently obtain essentially the same bound. Our presentation of the sample complexity upper bound in Section A.3 follows the structure of Anthony and Bartlett (1999, Chapter 4). As we mentioned, this upper bound is not tight; the $\log\frac{1}{\epsilon}$ factor was removed from the VC-dimension term by Hanneke (2016). The analogous sample complexity upper bound for agnostic learning (Exercise A.8, originally from Anthony and Bartlett (1999, Chapter 4)) is also not tight; the optimal bound in this case was given by Talagrand (1994). The sample complexity lower bound of Theorem A.4 ("No free lunch") is from Ehrenfeucht et al. (1989), though Vapnik and Chervonenkis (1971) had established a lower bound on the expected error (as opposed to a high-probability bound). The lower bound for confidence (Exercise A.7) is from Blumer et al. (1989). We note that Anthony and Bartlett (1999, Chapter 5) establish the optimal lower bound for the agnostic setting.

Section A.5 is essentially adapted from Schuurmans (1997). Empirical observations of exponential learning curves were first reported by Cohn and Tesauro (1992). Although the analysis we present here only holds for discrete classifiers, exponential rates were established (at more modest sample sizes) for minimizing a regularized empirical error on linear classifiers under margin assumptions (Audibert and Tsybakov, 2007; Koltchinskii and Beznosova, 2005) and for kernel classifiers obtained via stochastic gradient descent (Pillaud-Vivien et al., 2018; Nitanda and Suzuki, 2019). Finally, Bousquet et al. (2021) established a trichotomous classification of the possible learning curves: whereas the sample complexity bounds have only two possible regimes, polynomial (finite VC-dimension) and diverging (infinite VC-dimension), learning curves may be exponentially decaying with sample size, as in discrete cases; linearly decaying, as VC-dimension bounds appear naïvely to predict; or arbitrarily slowly converging, corresponding to infinite VC-dimension. These bounds have been further sharpened into a linearly-decaying class-dependent component and a separate, possibly exponentially-decaying, distribution-dependent component by Bousquet et al. (2023), clarifying the separate roles played by the distribution and the class of representations. We note that Hanneke and Moran (2026) have also extended this learning curve analysis to the agnostic setting.

The first boosting algorithm was proposed by Schapire (1990). It recursively filtered the set of examples to obtain a classifier structured as a circuit of majority-of-3 gates. Although it established the equivalence of weak and strong learnability, it was relatively inefficient both in theory and in practice. Freund (1995) improved the bounds and collapsed the representation to a single majority vote of weak classifiers. Finally, AdaBoost (Algorithm 24) was proposed and analyzed by Freund and Schapire (1997), and Section A.6 is based on this work. Beyond providing nice theoretical bounds with an elegant analysis, AdaBoost was also very effective in practice, and its behavior has been extensively studied. See the book by Schapire and Freund (2012) for a comprehensive overview of this work.

Exercise A.10 is adapted from a result of Shalev-Shwartz et al. (2011).

Exercises

A.1 *VC-dimension of triangles.* We can formalize triangles, and indeed much geometric reasoning, in analytic terms as follows. A *convex combination* of a set of points $S = \{x^{(1)}, \ldots, x^{(m)}\} \subset \mathbb{Q}^n$ is any point $x' \in \mathbb{R}^n$ given by $x' = \sum_i \lambda^{(i)} x^{(i)}$ where $\lambda^{(1)}, \ldots, \lambda^{(m)}$ are nonnegative real numbers satisfying $\sum_i \lambda^{(i)} = 1$. The *convex hull* of S, $\text{conv}(S)$, is then the set of all convex combinations of S. A *triangle* in two dimensions is defined by the convex hull of three points $\vec{x}^{(1)}, \vec{x}^{(2)}, \vec{x}^{(3)} \in \mathbb{Q}^2$. Compute the VC-dimension of the class of triangles in $\mathbb{Q}^2$.

A.2 *Compute the VC dimension of halfspaces in high dimension.* Prove that the VC-dimension of halfspaces in $\mathbb{Q}^n$ is precisely $n + 1$.
While any proof is acceptable, we suggest the following approach. It is not hard to show that if all of the points of a set $S \subset \mathbb{Q}^n$ lie on one side of the halfspace (i.e., receive the

same label), then the entire convex hull of S also receives the same label. The approach is based on using the following theorem:

Theorem A.23 (Radon) *Let $S \subset \mathbb{R}^n$ of size $n+2$ be given. Then there is a partition of S into two subsets S_1 and S_2 – i.e., $S_1 \cup S_2 = S$ and $S_1 \cap S_2 = \varnothing$ – such that* $\text{conv}(S_1) \cap \text{conv}(S_2) \neq \varnothing$.

(a) Prove Radon's Theorem. *(**Hint:** To start, show that given $n+2$ points, you can find real numbers $\lambda^{(1)}, \ldots, \lambda^{(n+2)}$ such that (1) some $\lambda^{(i)}$ is nonzero and yet (2) $\sum_{i=1}^{n+2} \lambda^{(i)} x^{(i)} = 0$ and (3) $\sum_{i=1}^{n+2} \lambda^{(i)} = 0$, that is, $x^{(1)}, \ldots, x^{(n+2)}$ have an "affine dependence." Then consider: what does affine dependence have to do with the convex hulls of subsets of $x^{(1)}, \ldots, x^{(n+2)}$?)*
(b) Prove that Radon's Theorem implies that no set of $n+2$ points is shattered by halfspaces.
(c) Finally, prove that a set of $n+1$ points is shattered by halfspaces.

A.3 *Compute the VC dimension of k-decision lists.* Prove that the VC-dimension of k-decision lists for a fixed constant k is $\Theta(n^k)$. You may use the result of Exercise A.2.

A.4 *VC dimension of unions.* In this exercise, we will examine how the complexity of learning grows as we build larger hypothesis classes using the set union operation.
(a) *Bounding the VC dimension of unions.* Consider any sequence of k hypothesis classes $\mathcal{H}_1, \ldots, \mathcal{H}_k$ that all have VC dimension at most d, and let $\mathcal{H}_\cup = \bigcup_{i=1}^k \mathcal{H}_i$. Prove that $\mathcal{H}_\cup$ has VC-dimension at most $4d \log(2d) + 2 \log k$. *(**Hint**: similar to Lemma A.21, for a set of points shattered by $\mathcal{H}_\cup$, bound the number of dichotomies possible under the k classes using Sauer's Lemma and compare this to the number of dichotomies obtained by $\mathcal{H}_\cup$.)*
(b) *Learnability is not closed under countable union.* Prove that there is a sequence of hypothesis classes $\{\mathcal{H}_i\}_{i=1}^\infty$ such that every finite union $\bigcup_{i=1}^k \mathcal{H}_i$ is learnable, but that the limit $\bigcup_{i=1}^\infty \mathcal{H}_i$ is not learnable.

A.5 *VC dimension of conjunction.* Suppose $\mathcal{H}_1$ has VC-dimension d_1 and $\mathcal{H}_2$ has VC-dimension d_2, where $d_1, d_2 \geq 2$. Let $\mathcal{H}_\wedge = \{h_1 \wedge h_2 : h_1 \in \mathcal{H}_1, h_2 \in \mathcal{H}_2\}$. Prove that $\mathcal{H}_\wedge$ has VC-dimension at most $2(d_1 + d_2) \log(2(d_1 + d_2))$. *(**Hint**: again similar to Lemma A.21, for a set of points shattered by $\mathcal{H}_\wedge$, bound the number of dichotomies possible under pairs from $\mathcal{H}_1$ and $\mathcal{H}_2$ using Sauer's Lemma and compare this to the number of dichotomies obtained by $\mathcal{H}_\wedge$.)*

A.6 *VC dimension of sine.* Although the VC dimension of a class is often equal to the number of parameters defining the class (in a natural parameterization), this is not always true. Consider the class $\mathcal{H}_{\sin}$ consisting of the sets $\{x \in \mathbb{Q} : \sin(\omega x) \geq 0\}$ for all possible $\omega \in \mathbb{Q}$. Prove that $\mathcal{H}_{\sin}$ has infinite VC-dimension. *(**Hint**: show that the sets of m points 2^{-i} for $i = 1, \ldots, m$ are shattered for all m.)*

A.7 *Sample complexity lower bound for confidence.* Theorem A.4 shows that $\Omega(d/\epsilon)$ examples are necessary to guarantee an error rate of ϵ with confidence $1 - \delta_0$ for

some constant $\delta_0 > 0$. In this problem we will extend this lower bound to include a dependence on δ when we wish to learn to arbitrarily high confidence $1 - \delta$.
(a) Prove that for any class of VC-dimension at least 2, $\Omega(\frac{1}{\epsilon} \log \frac{1}{\delta})$ examples are necessary for an algorithm to identify a representation that has error at most ϵ with probability at least $1 - \delta$.
(b) Using part (a) and Theorem A.4, prove that for any algorithm to find a representation with error at most ϵ with probability $1 - \delta$ when the data is labeled by a class of VC-dimension d requires $\Omega(\frac{1}{\epsilon}(d + \log \frac{1}{\delta}))$ examples (as $d \to \infty, \epsilon \to 0, \delta \to 0$).

A.8 *VC-dimension sample complexity bounds for agnostic learning.* We can derive sample complexity bounds that are analogous to Theorem A.11 in cases where no representation h from the class $\mathcal{H}$ fits the data perfectly. Suppose a class $\mathcal{H}$ of representations of VC-dimension d is given. Prove that if we take a sample of

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

examples from a distribution D, then with probability $1 - \delta$ ($\delta < 1/2$) every representation $h \in \mathcal{H}$ has an empirical error $\sum_{j=1}^{m} \frac{1}{m} I[h(x^{(j)}) \neq b^{(j)}]$ within an additive ϵ ($\epsilon < 1/2$) of its true error $\Pr_D[h(x) \neq b]$. (Here $I[\varphi]$ denotes the 0/1 valued indicator function that is 1 iff φ is true.) *(**Hint:** Try to follow an analogous argument except that instead of choosing a set of m out of $2m$ elements uniformly at random as in Lemma A.12, consider a pair of m-element samples in which you independently swap the ith element in the two samples with probability 1/2.)*

A.9 *Relating agnostic learning of conjunctions to learning DNF.* Prove that if there is a polynomial-time improper agnostic learner for conjunctions (Definition 6.3), DNF is efficiently PAC-learnable. *(**Hint**: use boosting.)*

A.10 *Halfspaces are not agnostically learnable.* Prove that if the γ-approximate shortest vector problem is intractable, then so is improper agnostic learning of halfspaces. *(**Hint**: use boosting.)*