Course Notes for Machine Learning and Mathematical Logic

Maxwell Levine

Acknowledgments

These notes are currently a work on progress. I am very grateful to my friend Ben Fish for advice and recommended reading on machine learning. The material that is purely about machine learning uses the textbooks of Ben-David and Shalev-Shwartz and of Mohri, Rostamizadeh, and Talwalkar, both of which are available for free online.

Contents

Contents **1**

1 Some Basic Notions of Machine Learning **3**

1.1 The Framework of Machine Learning 3

1.1.1 Conceptual Issues in Machine Learning 3

1.1.2 A Quick Primer on Probability 6

1.1.3 Formal Models 9

1.1.4 Our First Solid Example: Empirical Risk Minimization 11

1.2 PAC Learning 13
1.2.1 Definitions and Examples 13
1.2.2 Agnostic Learnability 15
1.2.3 The No Free Lunch Theorem 15
1.3 VC Dimension 18
1.3.1 Definition and Examples 19
1.3.2 Theoretical Implications of VC Dimension 20

Bibliography **26**

Chapter 1

Some Basic Notions of Machine Learning

October 16, 2025

The goal of this course is to understand certain mathematical aspects of machine learning. The material will not be computational, and it will not be (at least directly) useful for implementation. We specifically want to understand enough of the mathematical aspects to understand their connections with mathematical logic. This means that we will omit many potentially interesting subjects within machine learning because our specific goals and the limitations on our time.

1.1 The Framework of Machine Learning

We will start by developing that formal language with which we can discuss machine learning. We will give some motivation, provide some definitions, and start looking at basic mathematical examples.

1.1.1 Conceptual Issues in Machine Learning

What is learning in general, framed in what might be an excessively abstract way? And how does this bring us to the subject at hand?

- Wikipedia says that *learning* is "the process of acquiring new understanding, knowledge, behaviors, skills, values, attitudes, and preferences." This is too broad for what we need. Ben-David says that learning is about "converting experience into expertise." Coldly

speaking, we can equate a lot of experience with data. Understanding and knowledge seems harder to mathematicize.

- *Artifical intelligence* is the capacity of machines to mimic human intelligence and understanding. This is also quite broad, and its full extent is outside the scope of this course. But in this case, we should expect understanding to be something roughly programmable.

- *Machine learning* is basically about converting data into functions. According to Mohri, Rostamizadeh, and Talwalkar, "Machine learning consists of designing efficient and accurate prediction algorithms." Now we really have something mathematical. Ben-David and Shelev-Shwartz say that, "machine learning is not not trying to build automated imitation of intelligent behavior, but rather to use the strengths and special abilities of computers to complement human intelligence, often performing tasks that fall way beyond human capabilities."

Some natural questions come up.

- *Can we expect computers to think?* Well, maybe this is not the point. Computers can process far more information than human beings, and they can do it with far more speed. Whether they are "thinking" or not is a separate question.

- *Okay, why not just let computers go ahead and crunch the numbers? Who needs our input?* Consider the following pitfall mentioned by Ben-David: The famous psychologist B.F. Skinner performed an experiment on pigeons. The pigeons were confined to their cages, and they were given food at random with no reference to their behavior. At first, the pigeons would peck around, as pigeons do. When food arrived, they would peck more often in the areas where they were pecking when food arrived. This, in turn, increased the probability that they would be pecking in those areas when food arrived again. Eventually, the pigeons "learned" to search for food in specific areas in their cages, even though this had nothing to do with their feeding schedule. So we need to be aware of the limitations of inductive reasoning.

- *Is there a general lesson from that example?* Zooming out, we need to be aware that an algorithm depends to a large degree on the framing

of the problem. This goes to the notion of *prior knowledge*. We will be making some assumptions—assuming what we already know to be true—as we consider various learning algorithms.

Concrete examples of learning tasks include, but are not limited to: spam filters, image recognition, speech processing, and document classification. There are a lot of options.

Going back to the mathematical perspective, here are some learning tasks framed abstractly:

- *Classification*: These are problems in which data needs to be separated. To use a classic example (from Valiant's paper defining PAC learning! [Val84]), imagine that there are a bunch of objects that are either robots or elephants. The problem would be to accurately predict whether something is a robot or elephant.

- *Regression*: Like what you may be familiar with from statistics, this is where the problem is to predict a value for an item. For example, maybe one wants to predict the price of strawberries given the date.

- *Ranking*: This is where items need to be ranked. This does not mean e.g. ranking the size of elephants. A non-obvious example would be search engine rankings.

- *Clustering*: This is where data need to be gathered into clusters. This is somewhat like classification, but in this case the clusters are not pre-defined. Think of blocs voting in elections.

Since machine learning is such an interdisciplinary field, there is no one model for all scenarios. In fact, we will consider at least three types of machine learning. With this in mind, there are a number of potential scenarios in machine learning that need to be considered:

- *Supervised versus unsupervised*: We say that learning is *supervised* if the learner is obtaining labeled data. This is like the elephant example, in which the status of an object—elephant or not—is given to the learner. Learning is otherwise *unsupervised*.

- *Batch learning versus online learning*: If the learner is given a bunch of data first, and then must derive a predictor, then we call it *batch*

learning. Otherwise, if the learner is given a stream of data and makes successive predictions, then this is *online* learning.

- *Cooperative versus adversarial learning (or in between)*: This has to do with the context in which the learner is obtaining information. An example of adversarial learning would be the detection of spam emails. Generally, spam emails will be disguised at something that is not spam, and would be attempting to fool the learner.
- *Active learning versus passive learning*: Learnings is *active* if the learner interacts with the training data in some way.

This is meant to illustrate some breadth, but we will not study all of these scenarios carefully.

Back to pigeons: It turns out that they can be train to recognize words [SBUR+16].

1.1.2 A Quick Primer on Probability

We need some basic concepts from probability, so let us review them now.

Definition 1.1.1. Let X be a set. A *σ-algebra* (often referred to in a probability context as a *σ-field* is a subset $\mathcal{F} \subseteq \mathcal{P}(X)$ of the powerset of X with the following properties:

1. If $A \subseteq X$ and $A \in \mathcal{F}$, then $X \setminus A \in \mathcal{F}$.
2. If $\langle A_i : i < \theta \rangle$ is an at most countable sequence of subsets of X that are in $\mathcal{F}$, then $\bigcup_{i<\theta} A_i$ and $\bigcap_{i<\theta} A_i$ are both in $\mathcal{F}$.

Definition 1.1.2 (Kolmogorov)**.** A *probability space* is a triple $(\Omega, \mathcal{F}, P)$ consisting of an underlying set Ω, a σ-algebra $\mathcal{F}$ on Ω, and and a function $P : \mathcal{F} \to [0, 1]$ with the following properties:

1. $P(\emptyset) = 0$ and $P(S) = 1$,
2. If $\langle A_i : i < \theta \rangle$ is a sequence of at most countably many mutually disjoint sets (i.e. $A_i \cap A_j = \emptyset$ if $i \neq j$) then the equation

$$P\left(\bigcup_{i<\theta} A_i\right) = \sum_{i<\theta} P(A_i)$$

holds.

We may refer to a probability space in terms of the triple $(\Omega, \mathcal{F}, P)$.

We get a simple proposition which is easily provable from these requirements.

Proposition 1.1.3. *Suppose $(\Omega, \mathcal{F}, P)$ is a probability space. Then the following are true for all $A, B \in \mathcal{F}$:*

1. $P(S \setminus A) = 1 - P(A)$.
2. *If $A \subseteq B$ then $P(A) \leq P(B)$.*
3. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.

Definition 1.1.4. Given a probability space $(\Omega, \mathcal{F}, P)$, a *random variable* over $(\Omega, \mathcal{F}, P)$ is a function $X : \Omega \to \mathbb{R}$.

Example 1.1.5. Suppose our space Ω is just the set representing the outcome of flipping a coin twice: $\{HH, TT, HT, TH\}$. Then a random variable X could be the number of heads: $X(HH) = 2$, $X(HT) = X(TH) = 1$, and $X(TT) = 0$.

If we write e.g. $P(X \leq 1)$ then we mean that probability of the event that $X \leq 1$, i.e. $3/4$.

The notion of a random variable allows us to formalize quantities that depend on the probability space.

Definition 1.1.6. Suppose $(\Omega, \mathcal{F}, P)$ is a probability space and X a random variable over $(\Omega, \mathcal{F}, P)$, the *distribution* $\mathcal{D}_X$ of X is the induced function $\mathcal{D} : \mathbb{R} \to [0, 1]$ given by

$$\mathcal{D}_X(z) = P(X^{-1}\{z\}) = P(X = z).$$

The subscript for X is dropped if the context is clear.

If $f : \Omega \to \{\mathsf{true}, \mathsf{false}\}$, then $\mathbb{P}_{z \sim \mathcal{D}}[f(z)]$ will denote $\mathcal{D}(\{z \mid f(z) = \mathsf{true}\})$.

Example 1.1.7. For flipping a coin twice: Take X from the previous example. Then $\mathcal{D}$ is the function such that $\mathcal{D}(0) = 1/4$, $\mathcal{D}(1) = 1/2$, $\mathcal{D}(2) = 1/4$.

Definition 1.1.8. A distribution $\mathcal{D}$ is *discrete* if its range is finite or countable.

If $\{x_1, \ldots, x_N\}$ enumerates the values of the distribution, then the *expected value* is the sum

$$\sum_{i=1}^{N} x_i \cdot P(X = x_i).$$

For a countably infinite distribution, the expected value of a random variable is

$$\mathbb{E}[Z] = \int_0^\infty \mathbb{P}[Z \geq x]dx,$$

at least when this integral converges.

If the distribution is composed with a function f, the expected value is denoted $\mathbb{E}_{z\sim\mathcal{D}}[f(z)]$.

Corrected from lecture.

Definition 1.1.9. Two random variables X, Y over $(\Omega, \mathcal{F}, P)$, are *independent* if

$$P(X \leq x, Y \leq y) = P(X \leq x) \cdot P(Y \leq y)$$

for all $x, y \in \mathbb{R}$.

A sequence of random variables $X_1, \ldots, X_n$ are independent if for all $z_1, \ldots, z_n \in \mathbb{R}$,

$$P(X_1 \leq x_1, \ldots, X_n \leq x_n) = P(X_1 \leq x_1) \cdot \ldots \cdot P(X_n \leq x_n).$$

Example 1.1.10. Going back to the coin-flip example: A random variable X_1 giving the result of the first flip (1 if heads and 0 if tails) will be independent of a random variable giving the result of the second flip.

Definition 1.1.11. Two random variables X, Y are *identically distributed* if they have the same distribution, i.e. if $\mathcal{D}_X = \mathcal{D}_Y$.

Example 1.1.12. Again with the coin-flip example: Consider random variable X given the number of heads over two flips and the random variable $Y = 1 - X$ giving the number of tails. Then $\mathcal{D}_X = \mathcal{D}_Y$ even though the random variables are not equal!

Definition 1.1.13. We say that random variables X and Y are *independently and identically distributed* or *i.i.d.* if they are independent and have the same distribution.

Example 1.1.14 (Independent but not identically distributed)**.** Flip a coin once and then roll a twenty-sided die. Corrected from lecture

Suppose that X is the coin-flip variable, where heads is given with the outcome 1, and Y is the die-roll variable. Let $\mathcal{D}_X$ and $\mathcal{D}_Y$ be their respective distributions. Then we can express the independence through the conditional probability

$$1/2 = \mathcal{D}_X(X = 1|Y = 5) = \frac{\mathcal{D}_X(X = 1 \wedge Y = 5)}{\mathcal{D}_Y(Y = 5)} = \frac{(1/2) \cdot (1/20)}{1/20} = 1/2.$$

Example 1.1.15 (Identically distributed but not independent)**.** Roll a six-sided die ten times. Let X be the number of times that a one is rolled, and let Y be the number of times a two is rolled.

Example 1.1.16 (Independent *and* identically distributed)**.** Roll a die twice. Let X be the value of the first roll and let Y be the value of the second roll.

1.1.3 Formal Models

We need some language to discuss particular learning scenarios.

Definition 1.1.17. Here are our most basic definitions, together with the typically-employed notation.

1. The *domain set* $\mathcal{X}$ is a set of objects that are to be labeled. The objects in $\mathcal{X}$ may be referred to as *examples* or *instances*, and $\mathcal{X}$ itself may be referred to as the *input space*.

2. The set of *labels* or *target values* may be denoted as $\mathcal{Y}$. It will often be the case that $\mathcal{Y} = \{0, 1\}$, which corresponds to a *binary classification*.

3. We will typically consider a subset $\mathcal{X} \times \mathcal{Y}$ of *training data*. A subset $S \subseteq \mathcal{X} \times \mathcal{Y}$ may be referred to as a *sample*.

4. A *concept* is a function $c : \mathcal{X} \to \mathcal{Y}$. A set of concepts $\mathcal{C}$ is called a *concept class*. We will often refer to a restriction of $\mathcal{C}$ called the *hypothesis set* and denote it $\mathcal{H}$.

5. We consider outputs $h \in \mathcal{H}$. An output may be referred to as a *prediction rule*, *predictor*, or *classifier*.

Definition 1.1.18 (Informal!)**.** A *learning algorthm* $\mathcal{A}$ is a process for processing training data and turning it into a predictor. The definition of the term *algorithm* is known to be contentious, but we will take it to be any precise operation that is performed in discrete steps.

Example 1.1.19. Picture some red and blue dots in the upper right quadrant of $\mathbb{R}^2$. Suppose we are trying to classify points between red and blue. One algorithm might return a rectangle with the smallest possible diameter that includes all red points.

Given a concept class $\mathcal{C}$ and a set of training data, we will want to find a reasonably good h that fits the data. The assessment of what a good fit can be is given by the notion of error.

Definition 1.1.20 (Generalization Error)**.** Given a hypothesis set $\mathcal{H}$, a target concept $\mathcal{C}$, and a probability distribution $\mathcal{D}$, the *generalization error* or *risk* of a predictor $h \in \mathcal{H}$ is defined as

$$L_{\mathcal{D},c}(h) = \mathbb{P}_{x\sim\mathcal{D}}[h(x) \neq c(x)] = \mathcal{D}(\{x : h(x) \neq c(x)\}).$$

This function may also be referred to as the *risk* or *loss* of h.

In other words, c represents what is actually happening, h is a guess, and $L_{\mathcal{D},c}(h)$ gives the probability (in the context of $\mathcal{D}$ that h is wrong. This is in reference to some framing of "reality." We also want a concept of error that pertains to a specific data set. After all, the learner does not know what $\mathcal{D}$ and c are.

Definition 1.1.21 (Empirical Error)**.** Given a training set of the form $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ and predictor h, then

$$L_S(h) = \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$$

empirical error or *training error.*

Going back to specter of bad outcomes, let us first consider how one of these models can mislead us.

Example 1.1.22 (Overfitting)**.** Suppose we have a classification problem where a space $\mathcal{X} = \mathbb{R}^2$ is given one of two values. More precisely, suppose

that the target concept is $c : (x, y) \mapsto \{0, 1\}$ as given by $c(x, y) = 1$ if and only if $y > 0$. Consider a sample $S = \{\vec{a}_1, \ldots, \vec{a}_5, \vec{b}_1, \ldots, \vec{b}_5\}$ where the $\vec{a}$'s are all above the x-axis and the $\vec{b}$'s are all below. Let $h_S(\vec{z}) = 1$ if and only if $\vec{z} = \vec{a}_i$ for $1 \leq i \leq 5$. Then $L_S(h_S) = 0$ even though h_S is obviously a bad predictor.

This phenomenon is called *overfitting*. This is why we are interested in i. i. d. variables.

1.1.4 Our First Solid Example: Empirical Risk Minimization

Now we are close to actually doing some math. With just a couple more definitions, we can say something positive about a learning problem.

Definition 1.1.23 (Empirical Risk Minimization)**.** This is any algorithm that returns a predictor h such that $L_S(h)$ is minimized. He call such a predictor h an ERM hypothesis.

Definition 1.1.24 (The Realizability Assumption)**.** Given a distribution $\mathcal{D}$ and a labeling function f, there exists $h^\star$ such that $L_{\mathcal{D},f}(h^\star) = 0$.

Proposition 1.1.25 (see Shalev-Shwartz/Ben-David)**.** *Assume that $\mathcal{H}$ is a finite hypothesis class, that $\delta \in (0, 1)$, that $\epsilon > 0$, and let m be an integer such that*

$$m \geq \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}.$$

Then for any labeling function f, any distribution $\mathcal{D}$ for which the realizability assumption holds, for any i.i.d. sample S of size at least m, for any ERM *hypothesis h_S, we have the bound*

$$L_{(\mathcal{D},f)}(h_S) \leq \epsilon$$

with probability at least $1 - \delta$.

Note that the finiteness of $\mathcal{H}$ is used directly in that we can express such a formula for the threshold of m.

Proof. Fix m as in the hypothesis of the proposition. Consider the set

$$W = \{S \in [\mathcal{X} \times \mathcal{Y}]^m \mid L_{\mathcal{D},f}(h_S) > \epsilon\}$$

consisting of "wrong" sample sets. We want to bound this set.

We also have a set of "bad" hypotheses

$$\mathcal{H}_B := \{h \in \mathcal{H} \mid L_{\mathcal{D},f}(h) > \epsilon\}$$

and a set of "misleading" samples given by

$$M = \{S \restriction_{\mathcal{X}} \mid S \in [\mathcal{X} \times \mathcal{Y}]^m, \exists h \in \mathcal{H}_B, L_S(h) = 0\}.$$

If $S \in W$, then in particular $h_S \in \mathcal{H}_B$. By the realizability hypothesis, it must be the case that if $S \in W$ then $L_S(h_S) = 0$: there is some predictor h such that $L_S(h)$ is zero, therefore the learning algorithm can only output something with L_S equal to zero. Therefore $S \in M$. So $W \subseteq M$.

By the definition of M, we can write

$$M = \bigcup_{h \in \mathcal{H}_B} \{S_{\mathcal{X}} \restriction S \in [\mathcal{X} \times \mathcal{Y}]^m, L_S(h) = 0\}.$$

Therefore

$$\mathcal{D}^m(W) \leq \mathcal{D}^m(M) = \mathcal{D}^m\left(\bigcup_{h \in \mathcal{H}_B} \{S \restriction_{\mathcal{X}} \mid S \in [\mathcal{X} \times \mathcal{Y}]^m, L_S(h)\} = 0\right) \leq$$
$$\leq \sum_{h \in \mathcal{H}_B} \mathcal{D}^m(\{S \restriction_{\mathcal{X}} \mid S \in [\mathcal{X} \times \mathcal{Y}]^m, L_S(h) = 0\})$$

The statement $L_S(h) = 0$ is equivalent to the statement that for all $x_i \in S$, $h(x_i) = f(x_i)$. Applying the i. i. d. assumption, we have that for all $h \in \mathcal{H}_B$,

$$\mathcal{D}^m(\{S_{\mathcal{X}} \mid S \in [\mathcal{X} \times \mathcal{Y}]^m, L_S(h) = 0\}) = \prod_{1 \leq i \leq m} \mathcal{D}(\{x_i \mid h(x_i) = f(x_i)\}) =$$
$$= (1 - L_{\mathcal{D},f}(h))^m \leq (1 - \epsilon)^m$$

Claim. *The inequality* $1 - x \leq e^{-x}$ *holds for* $x \geq 0$.

Proof. The inequality is an equality if $x = 0$. If we look at $f(x) = 1 - x$ and $g(x) = e^{-x}$ and compare derivatives, we have $f'(x) = -1$ versus $g'(x) = -e^{-x} = -1/e^x > -1$ (for $x \geq 0$). This implies that the inequality will hold for $x > 0$ as follows. Let $h(x) = g(x) - f(x)$, so we are claiming that $h(x) \geq 0$ for $x \geq 0$. Otherwise there would be some $a > 0$ such that $h(a) < 0$. Then the Mean Value Theorem implies that there is some $b \in (0, a)$ such that $0 > h'(b) = g'(b) - f'(b)$, and hence $f'(b) > g'(b)$. But this contradicts the fact that $f'(x) = -1 < g'(x)$ for $x \geq 0$. □

Applying the claim to the previous inequality, we obtain

$$\mathcal{D}^m(\{S \restriction_{\mathcal{X}} | \ S \in [\mathcal{X} \times \mathcal{Y}]^m, L_S(h) = 0\}) \leq e^{\epsilon m}$$

and therefore we can combine this with the sum equation above to get

$$\mathcal{D}^m(W) \leq |\mathcal{H}_B| e^{\epsilon m} \leq |\mathcal{H}| e^{\epsilon m} \leq |\mathcal{H}| e^{\epsilon \cdot \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}} = \delta.$$

This completes the proof. □

1.2 PAC Learning

The preceding examples fits under the general template of PAC-learnability, which is might be the most widely-used notion of learnability.

1.2.1 Definitions and Examples

Definition 1.2.1 (PAC Learnability)**.** A hypothesis class $\mathcal{H}$ is PAC *Learnable* if there exist a function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm $\mathcal{A}$ with the following property: For every $\epsilon, \delta \in (0,1)$, every distribution $\mathcal{D}$ over $\mathcal{X}$, and every function $f : \mathcal{X} \to \{0,1\}$, if the realizability assumption holds with respect to $\mathcal{H}, \mathcal{D}, f$, then when running $\mathcal{A}$ on m-many i. i. d. samples with $m \geq m_{\mathcal{H}}$ which are generated by $\mathcal{D}$ and labeled by f, $\mathcal{A}$ returns a hypothesis h such that, with probability at least $1 - \delta$, $L_{\mathcal{D},f}(h) \leq \epsilon$.

If such an $\mathcal{A}$ exists, then it is called a PAC *learning algorithm* for $\mathcal{H}$. The function $m_{\mathcal{H}}(\cdot, \cdot)$ that chooses the *minimal* such m is the *sample complexity*.

Remark 1.2.2. In other contexts we might demand that $m_{\mathcal{H}}$ is a polynomial. If it is, we will say that $\mathcal{H}$ is *efficiently* PAC learnable.

Example 1.2.3. Proposition 1.1.25 showed that finite hypothesis classes are PAC-learnable!

Example 1.2.4 (Learning axis-aligned rectangles)**.** Let $\mathcal{X} = \mathbb{R}^2$ and let the concept class $\mathcal{C}$ be the set of axis-aligned rectangles, that is, the rectangles whose sides are parellel to the x- or y-axis. Assume a continuous probability distribution.[1] So each concept c will be a function indicating whether a

[1] We make this assumption for ease, but in theory we want to have the result for all probability distributions.

point in $\mathbb{R}^2$ is in a particular rectangle (say 0 is negative and 1 is positive). The learning problem is to determine a particular rectangle R. A hypothesis will take the form of some axis-aligned rectangle R'.

We describe a learning algorithm $\mathcal{A}$ as follows: Given a sample $S = \{(x_0, y_0, \iota_0), \ldots, (x_n, y_n, \iota_n)$ where the ι's are 1 is the coordinate is in R and 0 otherwise, $\mathcal{A}$ will return the tightest axis-aligned rectangle fitting that sample. In other words, the sample will contain a largest value y_t where $\iota_t = 1$, a smallest such y_b with $\iota_b = 1$, a leftmost such x_l, and a rightmost x_r. So the R_S that $\mathcal{A}$ returns will be the set of (x, y) such that $x_l \leq x \leq x_r$ and $y_b \leq y \leq y_t$. Observe that $\mathcal{A}$ does not produce false positives by the convexity of rectangles.

Fix some ϵ and δ from which we will find a way to choose m. Let $\mathcal{D}_{\mathsf{R}}$ denote the distribution of R, i.e. the probability of choosing a point in R. We can assume that $\mathcal{D}(\mathsf{R}) > \epsilon$ because otherwise we have the bound automatically; errors will only occur from points inside R. We can choose four rectangular regions inside R that R_S would have to avoid for there to be an error, and where we also assume $\mathcal{D}(r_i) = \epsilon/4$ for $i \in [1, 4]$ (this is where we use continuity).

Then we have

$$\begin{aligned}\mathcal{D}(\{S : L_{\mathcal{D},f}(\mathsf{R}_S) > \epsilon\}) &\leq \mathcal{D}(\cup_{i\in[4]}\{S : \mathsf{R}_S \cap r_i = \emptyset\}) \leq \\ &\leq \sum_{i\in[4]} \mathcal{D}(\{S : \mathsf{R}_S \cap r_i = \emptyset\}) \leq 4(1-\epsilon/4)^m \leq \\ &\leq 4e^{-m\epsilon/4}.\end{aligned}$$

So if we want $\mathcal{D}(\{S : L_{\mathcal{D},f}(\mathsf{R}_S) > \epsilon\}) \leq \delta$ it is sufficient to choose

$$m \geq \frac{4}{\epsilon}\log\frac{4}{\delta}.$$

(*Remark:* The algorithm depicted is not the only PAC learning algorithm for this problem. One can also choose the largest axis-aligned rectangle that avoids false values from the sample.)

Remark 1.2.5. We will see a non-example—a problem that is not PAC-learnable—below in the No Free Lunch Theorem (Theorem 1.2.10).

1.2.2 Agnostic Learnability

Definition 1.2.6 (True Error)**.** [2] For a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the *true error* of a prediction rule $h : \mathcal{X} \to \mathcal{Y}$ is

$$L_{\mathcal{D}}(h) = \mathcal{D}(\{(x, y) : h(x) \neq y\}).$$

Remark 1.2.7. The true error generalizes the generalization error. First, any concept $c : \mathcal{X} \to \mathcal{Y}$, together with a probability distribution $\mathcal{D}_0$ on $\mathcal{X}$, induces a probability distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$. The true error also generalizes the empirical error under a uniform probability distribution.

In other words, the true error generalizes the generalization error and empirical error.

Definition 1.2.8 (Agnostic PAC Learnability)**.** A hypothesis class $\mathcal{H}$ is *agnostic PAC Learnable* if there exist a function $m_{\mathcal{H}} : (0, 1)^2 \to \mathbb{N}$ and a learning algorithm $\mathcal{A}$ with the following property: For every $\epsilon, \delta \in (0, 1)$, every distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, then when running $\mathcal{A}$ on m-many i. i. d. samples with $m \geq m_{\mathcal{H}}$ which are generated by $\mathcal{D}$, $\mathcal{A}$ returns a hypothesis h such that, with probability at least $1 - \delta$,

$$L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon.$$

As with PAC learnability, we have a notion of effectiveness when we require $m_{\mathcal{H}}$ to be a polynomial.

Example 1.2.9 (Agnostic PAC Learnability for Finite Classes)**.** There is a proof here, but we will derive an equivalence below (Theorem 1.3.6) from which the result will follow.

1.2.3 The No Free Lunch Theorem

The No Free Lunch Theorem essentially shows that there is no universal learner. In particular, it provides us with a hypothesis class that is not PAC-learnable (Corollary 1.2.11 below).

[2] Ben-David and Shalev-Shwartz call this the "redefined" true error [SSBD14, page 45].

Theorem 1.2.10 (No Free Lunch)**.** *Let $\mathcal{A}$ be any learning algorithm for the task of binary classification over a domain $\mathcal{X}$. Let m be any number smaller than $|\mathcal{X}|/2$ (where $|\mathcal{X}|$ may be infinite). Then there exists a distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$ such that the following hold:*

1. *There exists a function $f : \mathcal{X} \to \{0,1\}$ with $L_{\mathcal{D}}(f) = 0$.*
2. *With probability of at least $1/7$, we have that $L_{\mathcal{D}}(\mathcal{A}(S)) \geq 1/8$ over an* i. i. d. *sample S of size m.*

Edit: agnostic case

Corollary 1.2.11. *Let $\mathcal{X}$ be an infinite domain set and let $\mathcal{H}$ be the set of all functions from $\mathcal{X}$ to $\{0,1\}$. Then $\mathcal{H}$ is not agnostic* PAC *learnable.*

Proof. Assume for contradiction that the class is actually learnable. Choose some $\epsilon < 1/8$ and $\delta < 1/7$. By the definition of PAC learnability, there is an algorithm $\mathcal{A}$ and an integer m such that for any distribution $\mathcal{D}$ over $\mathcal{X} \times \{0,1\}$, if there is some function $f : \mathcal{X} \to \{0,1\}$ with $L_{\mathcal{D}}(f) = 0$, then we have $L_{\mathcal{D}}(\mathcal{A}(S)) \leq \epsilon$. But the No Free Lunch Theorem contradicts this. □

First we need a couple lemmas.

Proposition 1.2.12 (Markov's Inequality)**.** *If Z is a non-negative random variable, then for all $a \geq 0$,*

$$\mathbb{P}[Z \geq a] \leq \frac{\mathbb{E}[Z]}{a}.$$

Proof. We have

$$\mathbb{E}[Z] \geq \int_0^a \mathbb{P}[X \geq x]dx \geq \int_0^a \mathbb{P}[X \geq a]dx \geq a \cdot \mathbb{P}[X \geq a].$$

□

Lemma 1.2.13. *Let X be a random variable that takes values in $[0,1]$ and let $\mu = \mathbb{E}[X]$. Then for any $a \in (0,1)$, then*

$$\mathbb{P}[X > 1 - a] \geq \frac{\mu - (1-a)}{a}$$

and

$$\mathbb{P}[X > a] \geq \frac{\mu - a}{1-a} \geq \mu - a.$$

Proof. Homework (use Markov's Inequality). □

Lemma 1.2.14. *Let X be a random variable that takes values in $[0,1]$ and whose expected value satisfies $\mathbb{E}[X] \geq 1/4$. Then $\mathbb{P}[X \geq 1/8] \geq 1/7$.*

Proof. Homework (use Lemma 1.2.13). □

Proof of Theorem 1.2.10. Let C be a subset of $\mathcal{X}$ of size $2m$. Since the learning algorithm will only observe half of the examples, we have the flexibility to come up with a target function f that contradicts the samples.

There are $\ell = 2^{2m}$-many possible functions from C to $\{0,1\}$. Enumerate these functions $f_1, \ldots, f_\ell$. Given f_i, let $\mathcal{D}_i$ be the distribution over $C \times \{0,1\}$ given by

$$\mathcal{D}_i(\{x,y\}) = \begin{cases} 1/|C| & \text{if } y = f_i(x) \\ 0 & \text{otherwise.} \end{cases}$$

Furthermore, $L_{\mathcal{D}_i}(f_i) = \mathcal{D}(\{(x,y) \mid f_i(x) \neq y\}) = 0$ for all i.

By Lemma 1.2.14, it is sufficient to show that for every algorithm $\mathcal{A}$ that receives a training set of m samples from $C \times \{0,1\}$ and returns $\mathcal{A}(S) : C \to \{0,1\}$, we have

$$\max_{i \in [\ell]} \mathbb{E}_{S \sim \mathcal{D}_i^m}[L_{\mathcal{D}_i}(\mathcal{A}(S))] \geq 1/4.$$

There are $k = (2m)^m$-many possible sequences from C. Enumerate these as $\langle S_j : 1 \leq j \leq k \rangle$. Given $S_j = (x_1, \ldots, x_m)$, let S_j^i be the sequence $((x_1, f_i(x_1)), \ldots, (x_m, f_i(x_m)))$. For distribution $\mathcal{D}_i$, the possible sets of training data are S_j^i for $1 \leq j \leq k$, all with the same probability of being sampled. Therefore we have

$$\mathbb{E}_{S \sim \mathcal{D}_i^m}[L_{\mathcal{D}_i}(\mathcal{A}(S))] = \frac{1}{k} \cdot \sum_{j=1}^{k} L_{\mathcal{D}_i}(\mathcal{A}(S_j^i)).$$

We also have

$$\max_{i \in [\ell]} \frac{1}{k} \sum_{j=1}^{k} L_{\mathcal{D}_i}(\mathcal{A}(S_j^i)) \geq \frac{1}{\ell} \cdot \sum_{i=1}^{\ell} \frac{1}{k} \sum_{j=1}^{k} L_{\mathcal{D}_i}(\mathcal{A}(S_j^i)) \geq \min_{j \in [k]} \frac{1}{\ell} \sum_{j=1}^{\ell} L_{\mathcal{D}_i}(\mathcal{A}(S_j^i)).$$

Fixing some j and $S_j = (x_1, \ldots, x_m)$, let $v_1, \ldots, v_p$ enumerate the points not appearing in C. (If C is infinite, then just take p larger than m.) Note that $p \geq m$. Therefore, for every $h : C \to \{0, 1\}$ and i we have

$$L_{\mathcal{D}_i}(h) = \frac{1}{2m} \sum_{x \in C} \mathbb{1}_{[h(x) \neq f_i(x)]} \geq \frac{1}{2m} \sum_{r=1}^{p} \mathbb{1}_{[h(v_r) \neq f_i(v_r)]} \geq \frac{1}{2p} \sum_{r=1}^{p} \mathbb{1}_{[h(v_r) \neq f_i(v_r)]}$$

because $|C| = 2m$. We also have

$$\frac{1}{\ell} \sum_{j=1}^{\ell} L_{\mathcal{D}_i}(\mathcal{A}(S_j^i)) \geq \frac{1}{\ell} \sum_{j=1}^{\ell} \frac{1}{2p} \sum_{r=1}^{p} \mathbb{1}_{[\mathcal{A}(S_j^i)(v_r) \neq f_i(v_r)]} \geq$$

$$\geq \frac{1}{2\ell} \min_{r \in [p]} \sum_{j=1}^{\ell} \mathbb{1}_{[\mathcal{A}(S_j^i)(v_r) \neq f_i(v_r)]}.$$

As the final aspect to consider, fix some $r \in [p]$. We can partition all functions from the enumeration $f_1, \ldots, f_\ell$ into $\ell/2$-many pairs $(f_i, f_{i'})$ such that for every c, $f_i(c) \neq f_{i'}(c)$ if and only if $c = v_r$. Therefore it follows that

$$\mathbb{1}_{[\mathcal{A}(S_j^i)(v_r) \neq f_i(v_r)]} + \mathbb{1}_{[\mathcal{A}(S_j^{i'})(v_r) \neq f_{i'}(v_r)]} = 1$$

and thus by symmetry we have

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \mathbb{1}_{[\mathcal{A}(S_j^i)(v_r) \neq f_i(v_r)]} = \frac{1}{2}.$$

Combining all of this information from the various inequalities, we have

$$\max_{i \in [\ell]} \mathbb{E}_{S \sim \mathcal{D}_i^m}[L_{\mathcal{D}_i}(\mathcal{A}(S))] = \max_{i \in [\ell]} \frac{1}{k} \cdot \sum_{j=1}^{k} L_{\mathcal{D}_i}(\mathcal{A}(S_j^i)) \geq$$

$$\geq \min_{j \in [k]} \frac{1}{\ell} \sum_{j=1}^{\ell} L_{\mathcal{D}_i}(\mathcal{A}(S_j^i)) \geq \frac{1}{2} \min_{j \in [k]} \min_{r \in [p]} \frac{1}{\ell} \sum_{j=1}^{\ell} \mathbb{1}_{[\mathcal{A}(S_j^i)(v_r) \neq f_i(v_r)]} = \frac{1}{2} \cdot \frac{1}{2}.$$

□

1.3 VC Dimension

All of theses examples are understandable through the notion of VC-dimension, which will give us a clear way to assess which hypothesis classes are PAC-learnable.

1.3.1 Definition and Examples

Definition 1.3.1 (VC Dimension)**.** Let $\mathcal{H}$ be a hypothesis class of functions from $\mathcal{X}$ to $\{0,1\}$.

1. Let $C = \{x_1, \ldots, x_m\} \subseteq \mathcal{X}$. Then the *restriction* of $\mathcal{H}$ to C is
$$\mathcal{H}_C = \{((x_1, h(x_1)), \ldots, (x_m, h(x_m))) \mid h \in \mathcal{H}\}.$$
2. We say that $\mathcal{H}$ *shatters* a finite $C \subseteq \mathcal{X}$ if $\mathcal{H}_C = 2^{|C|}$, that is, it $\mathcal{H}_C$ is the set of all functions from C into $\{0,1\}$.
3. The *Vapnik-Chervonenkis dimension* or *VC dimension* of $\mathcal{H}$, which we denote $\text{VCdim}(\mathcal{H})$, is the maximal size of a set $C \subseteq \mathcal{X}$ that is shattered by $\mathcal{H}$. If $\mathcal{H}$ shatters arbitrarily large finite sets then we say that the VC dimension is infinite.

Example 1.3.2. Let $\mathcal{H} = \{(0,0)\}$. Then the VC dimension of $\mathcal{H}$ is zero because no set can be shattered by it. (It is also obviously PAC learnable.)

Example 1.3.3. If $\mathcal{H}$ has cardinality $d < \aleph_0$ (i.e. is finite) then $\mathcal{H}$ has VC dimension at most d.

Example 1.3.4. Consider the set of hyperplanes in $\mathbb{R}^2$, i.e. lines in $\mathbb{R}^2$, formatted as classifications. So for every line ℓ, we have some h_ℓ^+ classifying all points "above" the line (this word can be tweaked) with 1 and all points "below" the line to be 0, and h_ℓ^-, the other way around. (Let us assume we have some consistent distinction for vertical lines.) Then we can argue that the VC dimension is 3.

First, we show that there are sets of size 3 that are shattered. Consider any three non-colinear points $\vec{x}$, $\vec{y}$, and $\vec{y}$. If we want them to all have the same classification, then we put them on one side of a line. If we want one to differ in classification from the other two, e.g. we want an h with $h(\vec{x}) = 0$ and $h(\vec{y}) = h(\vec{z}) = 1$, then we find a line separating $\vec{x}$ from the other two and then choose $h = h_\ell^+$ or $h = h_\ell^-$ accordingly.

Then we want to argue that four points can *never* be shattered. If three points are colinear, then the set cannot be shattered, so we can ignore such cases. Given this restriction, we first consider the case where the four points outline a convex set. To oversimplify, suppose $\vec{w} = (0,0)$, $\vec{x} = (1,0)$, $\vec{y} = (0,1)$, and $\vec{z} = (1,1)$. Then the pairs $\vec{w}$-$\vec{z}$ cannot be given a classification

opposite that of $\vec{x}$-$\vec{y}$. The other case is where the fourth point sits in a convex set outlined by the other three points. Then the inside point cannot be given a classification opposite the outside points.

Example 1.3.5. Consider again the case of axis-aligned rectangles. We can show that there is a set of size 4 which is shattered: Consider a set of points in a diamond configuration, i.e. $\{(-1,0),(1,0),(0,1),(0,-1)\}$. We can isolate points across from each other by a sufficiently "skinny" or "squat" rectangle. We can definitely get each point by itself in a rectangle, and we can exclude any particular point, and we can catch "adjacent" points as well.

Then we need to argue that there is no set of size 5 that can be shattered. Let $C = \{c_1, c_2, c_3, c_4, c_5\}$. Then there is a (not necessarily unique) "topmost," "leftmost," "rightmost," and "bottom-most" point. Let d be the point that is not counted among these. Consider the assignment that colors each $c \neq d$ with 1 and d with 0. This assignment is impossible to realize, because any rectangle containing the c's for $c \neq d$ must necessarily contain d.

When we prove the Fundamental Theorem of Statistical Learning (Theorem 1.3.6 below), we will have an accurate proof (minus the continuity assumption from before) that the hypothesis class of axis-aligned rectangles is PAC learnable.

1.3.2 Theoretical Implications of VC Dimension

We can now show that a hypothesis class is PAC-learnable if and only if it has finite VC-dimension.

Theorem 1.3.6 (The Fundamental Theorem of Statistical Learning)**.** *Let $\mathcal{H}$ be a hypothesis class of functions $\mathcal{X} \to \{0,1\}$. The following are equivalent:*

1. *$\mathcal{H}$ is agnostic* PAC*-learnable.*
2. *$\mathcal{H}$ is* PAC*-learnable.*
3. *$\mathcal{H}$ has finite* VC *dimension.*

Proposition 1.3.7. *If $\mathcal{H} \subseteq \mathcal{X} \times \{0,1\}$ is agnostic* PAC*-learnable, then it is* PAC*-learnable.*

Proof. We are using the realizability assumption. Let $f : \mathcal{X} \to \{0,1\}$ be a labeling function and $\mathcal{D}$ a distribution on $\mathcal{X}$ and let h be the function returned by the agnostic PAC algorithm.

We let $\mathcal{D}'$ be the distribution on $\mathcal{X} \times \{0,1\}$ given by

$$\mathcal{D}'(A) = \mathcal{D}(\{x \in \mathcal{X} \mid (x, f(x)) \in A\})$$

so that we have

$$\begin{aligned} L_{\mathcal{D},f}(h) = \mathcal{D}(\{x \mid f(x) \neq h(x)\}) = \mathcal{D}'(\{(x,y) \mid y \neq h(x)\}) = \\ = L_{\mathcal{D}'}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}'}(h') + \epsilon = \min_{h' \in \mathcal{H}} L_{\mathcal{D},f}(h') + \epsilon = 0 + \epsilon. \end{aligned}$$

because the realizability assumption allows us to set $\min_{h' \in \mathcal{H}} L_{\mathcal{D},f}(h') = 0$. □

Proof of 1. to 3. for Theorem 1.3.6. This follows from the proof of the No Free Lunch Theorem (Theorem 1.2.10): Specifically, from the point where we observe that there are 2^{2m}-many possible functions from C to $\{0,1\}$, we do not need to use the hypotheses any further. We could have instead argued from a hypothesis class $\mathcal{H}$ which shatters some C of cardinality 2^m. □

For the last part of the loop, we need to develop more terminology.

Definition 1.3.8. Let $\mathcal{H} \subseteq \mathcal{X} \times \mathcal{Y}$ be a hypothesis class. The *growth function* of $\mathcal{H}$, is the function $\mathbb{N} \to \mathbb{N}$ given by

$$\Pi_{\mathcal{H}}(m) = \max_{C \subseteq \mathcal{X}, |C| = m} |\mathcal{H}_C|.$$

Lemma 1.3.9 (Sauer's Lemma)**.** *If $\mathcal{H}$ is a hypothesis set with* $\mathrm{VCdim}(\mathcal{H}) = d$*, then for all $m \in \mathbb{N}$, we have*

$$\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}.$$

This lemma was derived independently by Shelah and Perles. It will use a weak version of:

Proposition 1.3.10. *If $\mathcal{H} \subseteq \mathcal{H}'$ then* $\mathrm{VCdim}(\mathcal{H}) \leq \mathrm{VCdim}(\mathcal{H}')$.

Proof. Homework. □

Proof. We establish the proof by induction on $m + d$. The left side of the inequality is zero if $m = 0$. If $m = 1$ and $d \in \{0, 1\}$, the left side is one and the right size includes either one summand of one or else two.

Assume that the statement holds for $(m-1, d-1)$ and $(m-1, d)$. Fix a set $S = \{x_1, \ldots, x_m\}$ witnessing that $\text{VCdim}(\mathcal{H}) = d$, that is, a set such that $\mathcal{H}_S$ is cardinality $\Pi_{\mathcal{H}}(m)$. Let $\mathcal{G} := \mathcal{H}_S$.

Let $S' = \{x_1, \ldots, x_{m-1}\}$. We let $\mathcal{G}_1 := \mathcal{G}_{S'}$ (noting that all of these functions have domain S', there being no proper subsets). Then we let $\mathcal{G}_2$ be the set of functions g with domain S' such that both $g^\frown\langle x_m, 0\rangle$ and $g^\frown\langle x_m, 1\rangle$ are in $\mathcal{G}$. If we want to count the functions in $\mathcal{G}$, then we can first count the functions in $\mathcal{G}_1$, and then ask ourselves which of those functions need to be counted again, and then we count the functions in $\mathcal{G}_2$. In other words, $|\mathcal{G}| = |\mathcal{G}_1| + |\mathcal{G}_2|$.

We have $\text{VCdim}(\mathcal{G}_1) \leq \text{VCdim}(\mathcal{G}_2) \leq d$. Therefore, the inductive hypothesis gives us

$$|\mathcal{G}_1| \leq \Pi_{\mathcal{G}_1}(m-1) = \sum_{i=0}^{d} \binom{m-1}{i}.$$

By the definition of $\mathcal{G}_2$, if a set $Z \subseteq S'$ is shattered by $\mathcal{G}_2$, then the set $Z \cup \{x_m\}$ is shattered by $\mathcal{G}$. Therefore $\text{VCdim}(\mathcal{G}_2) \leq \text{VCdim}(\mathcal{G}) - 1 = d - 1$. Therefore the inductive hypothesis gives us

$$|\mathcal{G}_2| \leq \Pi_{\mathcal{G}_2}(m-1) \leq \sum_{i=0}^{d-1} \binom{m-1}{i}.$$

Therefore,

$$|\mathcal{G}| \leq \sum_{i=0}^{d} \binom{m-1}{i} + \sum_{i=0}^{d-1} \binom{m-1}{i} \leq \sum_{i=0}^{d} \left(\binom{m-1}{i} + \binom{m-1}{i-1} \right) = \\ = \sum_{i=0}^{d} \binom{m}{i}$$

where the last equality is Pascal's rule. □

Corollary 1.3.11. *Let $\mathcal{H}$ be a hypothesis class with* $\mathrm{VCdim}(\mathcal{H}) = d$. *Then for all $m \geq d$,*

$$\Pi_{\mathcal{H}}(m) \leq \left(\frac{em}{d}\right)^d.$$

Proof. We have

$$\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} \leq \sum_{i=0}^{d} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i} \leq \sum_{i=0}^{m} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i} \leq$$
$$\leq \left(\frac{m}{d}\right)^d \sum_{i=0}^{m} \binom{m}{i} \left(\frac{d}{m}\right)^i = \left(\frac{m}{d}\right)^d \left(1 + \frac{d}{m}\right)^m \leq \left(\frac{m}{d}\right)^d e^d.$$

The inequalities follows from: Sauer's Lemma; the fact that $(m/d) \geq 1$; summing over more non-negative terms; pulling out a constant; and $(1 - x) \leq e^x$ where $x = d/m$. The last equality is the bonomial theorem. □

We have made progress towards proving that finite VC dimension implies pack learnability. The next step is to rephrase the requirements of what we need to prove.

Definition 1.3.12. Consider a training set S and over a hypothesis class $\mathcal{H}$.

1. S is *ϵ-representative* (with respect to a distribution $\mathcal{D}$) if

$$\forall h \in \mathcal{H}, |L_S(h) - L_{\mathcal{D}}(h)| \leq \epsilon.$$

2. We say that $\mathcal{H}$ has the *uniform convergence property* if there exists a function $m_{\mathcal{H}}^{\mathsf{UC}} : (0,1)^2 \to \mathbb{N}$ such that for every $\epsilon, \delta \in (0,1)$ and every i. i. d. distribution $\mathcal{D}$, if S is a sample of size $m \geq m_{\mathcal{H}}^{\mathsf{UC}}(\epsilon, \delta)$, then with probability at least $1 - \delta$, S is ϵ-representative.

Proposition 1.3.13. *If $\mathcal{H}$ has a uniform convergence property as witnessed by a minimal $m_{\mathcal{H}}^{\mathsf{UC}}$, then $\mathcal{H}$ is agnostically* PAC *learnable with sample complexity $m_{\mathcal{H}}(\epsilon, \delta) \leq m_{\mathcal{H}}^{\mathsf{UC}}(\epsilon/2, \delta)$. Furthermore, the* $\mathrm{ERM}_{\mathcal{H}}$ *paradigm gives a learning algorithm.*

Proof. Consider any $h \in \mathcal{H}$ and let h_S by an $\mathrm{ERM}_{\mathcal{H}}$ output. Since $|L_S(h) - L_{\mathcal{D}}(h)| \leq \epsilon/2$, we have

$$L_{\mathcal{D}}(h_S) \leq L_S(h_S) + \epsilon/2 \leq L_S(h) + \epsilon/2 \leq L_{\mathcal{D}}(h) + \epsilon/2 + \epsilon/2 = L_{\mathcal{D}}(h) + \epsilon$$

where we are applying $\epsilon/2$-uniform convergence in the first inequality, the $\mathrm{ERM}_{\mathcal{H}}$ property for the second, and the $\epsilon/2$-uniform convergence again in the third. □

Now we know that it will be enough to prove that $\mathcal{H}$ has the uniform convergence property.

Lemma 1.3.14. *Let $\mathcal{H}$ be a hypothesis class with growth function $\Pi_{\mathcal{H}}$. Then for every distribution $\mathcal{D}$ and every $\delta \in (0,1)$, we have*

$$|L_{\mathcal{D}}(h) - L_S(h)| \leq \frac{4 + \sqrt{\log(\Pi_{\mathcal{H}}(2m))}}{\delta\sqrt{2m}}$$

with probability at least $1-\delta$ from an i. i. d. *sample S of size m.*

Before proving the lemma, which is fairly technical, we will show how to prove the rest of the theorem.

We will also use an item from the homework:

Proposition 1.3.15. *Let $a \geq 1$ and $b > 0$. Then $x \geq 4a\log(2a) + 2b \implies x \geq a\log(x) + b$.*

Proof of 3. to 1. for Theorem 1.3.6. We know that it suffices to show that a finite VC-dimension implies the uniform convergence property. Specifically, we will show that it suffices to take m large than

$$4\frac{16d}{(\delta\epsilon)^2}\log\left(\frac{16d}{(\delta\epsilon)^2}\right) + \frac{16d\log(2e/d)}{(\delta\epsilon)^2}$$

and large enough that $\sqrt{d\log(2em/d)} \geq 4$ in order to bound $|L_S(h) - L_{\mathcal{D}}(h)|$ by epsilon.

By Proposition 1.3.15, if we have

$$m \geq 4 \cdot \frac{2d}{(\delta\epsilon)^2} \cdot \log\left(\frac{2d}{(\delta\epsilon)^2}\right) + \frac{4d\log(2e/d)}{(\delta\epsilon)^2}, \tag{1.1}$$

then it follows that

$$m \geq \frac{2d\log(m)}{(\delta\epsilon)^2} + \frac{2d\log(2e/d)}{(\delta\epsilon)^2}.$$

So let us assume that 1.1 holds and that m is . Then, from the corollary of Sauer's Lemma, we have

$$|L_S(h) - L_{\mathcal{D}}(h)| \leq \frac{4 + \sqrt{d \log(2em/d)}}{\delta\sqrt{2m}} \leq \frac{1}{\delta}\sqrt{\frac{2d \log(2em/d)}{m}} \leq \epsilon$$

and so we are done. □

Now to prove the technical lemma.

Proof of Lemma 1.3.14. □

Bibliography

[BH19] Joseph K Blitzstein and Jessica Hwang. *Introduction to probability*. Chapman and Hall/CRC, 2019.

[GS06] Charles Miller Grinstead and James Laurie Snell. *Grinstead and Snell's introduction to probability*. Chance Project, 2006.

[Har19] Klaas Pieter Hart. Machine learning and the continuum hypothesis. *arXiv preprint arXiv:1901.04773*, 2019.

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[SBUR+16] Damian Scarf, Karoline Boy, Anelisie Uber Reinert, Jack Devine, Onur Güntürkün, and Michael Colombo. Orthographic processing in pigeons (columba livia). *Proceedings of the National Academy of Sciences*, 113(40):11272–11276, 2016.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[Val84] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Vau11] Jennifer Wortman Vaughan. CS260: Machine learning theory lecture 2: Introduction to the PAC model, September 2011. `https://www.jennwv.com/courses/F11/lecture2.pdf`.