Machine Learning

CS 485

Shai Ben-David

LaTeXed by Sibelius Peng

Preface

Disclaimer Much of the information on this set of notes is transcribed directly/indirectly from the lectures of CS 485 during Fall 2020 as well as other related resources. I do not make any warranties about the completeness, reliability and accuracy of this set of notes. Use at your own risk.

Since the course is online, we are watching recordings from a previous offering. Videos are available on `https://www.newworldai.com/understanding-machine-learning-course/`. The textbook for this course is available at `http://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning`.

Some notations:

- $D[A]$ denotes the probability hitting the set A.

For any questions, send me an email via `https://notes.sibeliusp.com/contact/`.

You can find my notes for other courses on `https://notes.sibeliusp.com/`.

Sibelius Peng

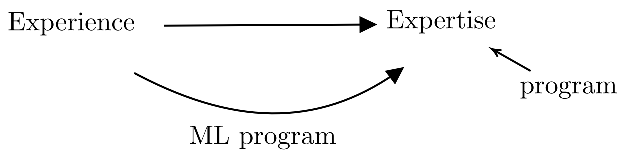
Contents

1 Introduction

Reading

Up to page 41 of the textbook.

What is learning?



Process takes us from experience and leads us to expertise. Expertise would be another program that can do something you need expertise to do. For example, develop a spam filter. The outcome program is the spam filter.

1.1 Learning in Nature

Bait Shyness: It's difficult to poison the rats with the bait food. The rats will find that the shape might be different. If they take a bite and feel sick, they will immediately associate the sickness with the food and then never touch it again. It's a clear example of learning from a single experience.

Spam filters: Inputs are emails which are labeled.

$$(\text{email1, spam}), (\text{email2, not spam}), \ldots$$

Then we have to come up with the program which filters the spam. The simplest way is to **memorize** all the emails that are spam. So what's wrong with such a program?

It does not generalize. We want **generalization**. Memorization is not enough. Generalization is sometimes called **inductive reasoning**: take previous cases and try to extend it to something new.

Pigeon Superstition: discovered by Skinner in 1947. He took a collection of pigeons and put them in the cage. Also he put different kinds of toys. Above the cage, there is some mechanism that can spread grains. Something interesting happens. When the birds get hungry, they pick around for worms. Suddenly there's a spread of food. The birds start to learn: maybe the toys the bird is picking at that particular moment had some influence on getting food. So the next time the bird is hungry, the bird is more likely to pick on this toy than others. Then the next time food spreads, it reinforces what the bird did. After several times, the birds are completely devoted to some specific toys.

This is silly generalization. For rats, it's important generalization making them survive.

Garcia 1996, looks at the rats again. He gave the rats the poisoned bait which smelled and looked exactly like the usual food they get. Then the question: does the rat learn the connection between sickness and the poisoned food? Rats fail to associate the bell ringing with the poison effect. Here note that unlike the Pavlov's dog experiment which did repeatedly many times, the rat only has one chance to learn.

The key point here is prior knowledge: the rat already knows the shape and smell of the food through generation. Why have this limitation, why not paying attention to everything? In terms of rats, if they feel sick, every experience/feed is special, then the rat don't know what to associate to. Therefore, the prior knowledge is very important.

If we have little prior knowledge, we need a lot of training. If we have much prior knowledge, maybe we can do without much experiences. ML is living somewhere between these two.

Why do we need Machine Learning?

1. Some tasks that we (animals) can carry out may be too complex to program. E.g., Spam filter, driving, speech recognition.
2. Tasks that require experience with amounts of data that are beyond human capabilities. E.g., ads placement, genetic data.
3. Adaptivity.

1.2 Many types of machine learning

1. Supervised vs. Unsupervised. Supervised: spam filter. Unsupervised: outlier detection, clustering.

   There's also an intermediate scenario called reinforcement learning.
2. Batch vs. Online. Batch: get all training data in advance. Online: need to response as you learn.
3. Cooperative → indifferent → adversarial. Teacher.
4. Passive vs. Active learner.

1.3 Relationships to other fields

AI: two important differences: We are going beyond what human/animals can do, not try to imitate; This area is rigorous, mathematical, nothing like "happens to be".

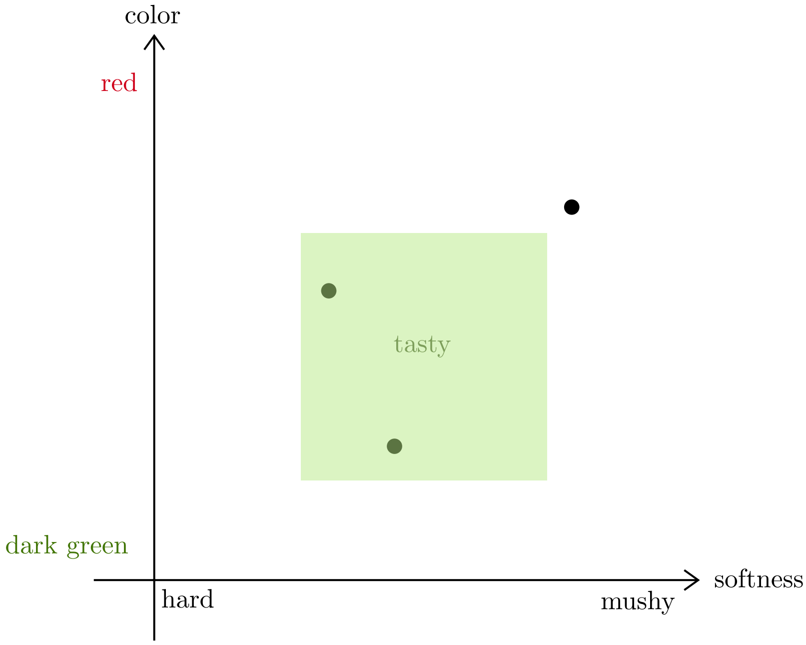
Also we need: Algorithms & complexity, statistics, linear algebra, combinatorics, optimization. However, there's something different with statistics in several ways.

1. algorithmic statistics
2. Distribution free. No clue on how spam is generated.
3. finite samples.

Outline of the course

- Principles: supervised, batch, ...
- Algorithmic paradigms.
- Other types of learning.

1.4 Papaya Tasting



Each papaya corresponds to a coordinate (c, s).

Training data: $(x_1, y_1), \dots (x_m, y_m) = S$, where $x_i \in \mathbb{R}^2$ and $y_i \in \{T, N\}$

Domain set: $[0, 1]^2$

Label set: $\{T, N\}$

Output: $f : [0, 1]^2 \to \{T, N\}$. Prediction rule.

Assumption about data generation:

1. Training data are randomly generated.
2. Reliability by rectangles. See the picture above.

Measure of success: probability of my predictor f to err on randomly generated papaya.

2

A Gentle Start

2.1 Formal model for learning

In the context of papaya example last time,

Domain set X	$[0,1]^2$
Label set y	$\{T, N\}$
Training input (sample) $S = ((x_1, y_1), \ldots, (x_m, y_m))$	set of already tasted papayas
Learners output $h : X \to Y$	Prediction rule for tasteness
The quality of such h is determined with respect to some data generating distribution and labeling rule	$L_{D,f}(h) = D[\{x : h(x) \neq f(x)\}]$ where L stands for loss, D is the distribution of papaya generated in the world, f is the function to determine the true tastefulness of papaya. So it determines the probability that our hypothesis h fails

The goal of the learner is given S to come up with h with small loss.

2.2 Empirical Risk Minimization

Basic learning strategy: **empirical risk minimization** (ERM) which minimizes the empirical loss.

We define **empirical loss (risk)** over a sample S:

$$L_S(h) = \frac{|\{i : h(x_i) \neq y_i\}|}{|S|}$$

A very simple rule for finding h with small empirical risk (ER)

$$h_S(x) \triangleq \begin{cases} y_i & \text{if } x = x_i \text{ for some } (x_i, y_i) \in S \\ N & \text{otherwise} \end{cases}$$

Then $L_S(h_S) = 0$.

Although the ERM rule seems very natural, without being careful, this approach may fail miserably. It **overfit**s our sample.

To guard against overfitting we introduce some prior knowledge (Inductive Bias).

There exists a good prediction rule that is some axis aligned rectangle. Let H denote a fixed collection of potential (candidate) prediction rules, i.e.,

$$H \subseteq \{f : X \to Y\} = X^Y,$$

and we call it **hypothesis class**. Then we have a revised learning rule: ERM_H - "pick $h \in H$ that minimizes $L_S(h)$", i.e.,

$$\text{ERM}_H(S) \in \underset{h \in H}{\text{argmin}}\{L_S(h)\}$$

Theorem 2.1

Let X be any set, $Y = \{0, 1\}$, and let H be a finite set of functions from X to Y. Assume:

1. The training sample S is generated by some probability distribution D over X and labeled some $f \in H$, and elements of S are picked i.i.d.[a]
2. **Realizability assumption**: $\exists h \in H$ such that $L_{D,f}(h) = 0$

Then ERM_H is guaranteed to come up with an h that has small true loss, given sufficiently large sample S.

[a] identically and independently distributed

Remark:
This is quite different from hypothesis testing. Unlike hypothesis testing, here we have assumptions after seeing the data. We are developing theories based on the data, and here H is a finite set.

Proof:
Confusing samples are those on which ERM_H may come up with a bad h.

Fix some success parameter $\varepsilon > 0$, and the set of confusing S's is

$$\{S : L_{D,f}(h_S) > \varepsilon\}$$

We wish to upper bound the probability of getting such a bad sample.

$$D^m[\{S|_X : L_{D,F}(h_S) > \varepsilon\}]$$

where $S|_X = x_1, \ldots, x_m$.

Consider $H_B = \{h \in H : L_{D,f}(h) > \varepsilon\}$ which is the set we want to avoid.

The misleading samples is the set of samples that may lead to an out come in H_B, formally:

$$M = \{S|_X : \exists h \in H_B \text{such that } L_S(h) = 0\}$$

We claim that $\{S|_X : L_{D,f}(h_S) > \varepsilon\} \subseteq M$. The former set is the cases that we select bad hypothesis, and M is the cases there exist bad hypothesis. So it is a subset. We might not have selected a bad hypothesis from M, then we cannot put an equal between these two sets. We want to upper bound a probability of the set we defined. Therefore, it suffices to upper bound $D^m(M)$.

$$D^m(M) = D^m\left[\bigcup_{h \in H_B} : \{S|_X : L_S(h) = 0\}\right]$$

Now we need two basic probability rules:

1. The union bound: For any two events A, B and any probability distribution P, $P(A \cup B) \leq P(A) + P(B)$.
2. If A and B are independent events then $P(A \cap B) = P(A) \cdot P(B)$.

For any fixed $h \in H_B$. Let us upper bound $D^m[\{S|_X : L_S(h) = 0\}]$. For a single one, the probability of h is doing wrong on X is at least ε, then

$$\begin{aligned} D^m[\{S|_X : L_S(h) = 0\}] &= D^m[\{S_X : h(x_1) = f(x_1) \wedge \cdots \wedge h(x_m) = f(x_m)\}] \\ &\leq (1 - \varepsilon)^m \end{aligned}$$

Then we conclude

$$D^m[\text{Bad } S] \leq D^m(M) = D^m[\cup_{h \in H_B}(L_S(h) = 0)] \leq |H_B| \cdot (1-\varepsilon)^m \leq |H| \cdot (1-\varepsilon)^m$$

Then ERM_H has small probability of failure as $m \to \infty$. $\square$

Note that here we call it paradigm not algorithm since it doesn't tell you which H to pick.

This time we use a different notation:

$$\Pr_{S \sim D^m}[L_{D,f}(\mathrm{ERM}_H(S)) > \varepsilon] \leq |H| \cdot (1-\varepsilon)^m$$

for every $\varepsilon \geq 0$ and for every m, where

$$L_{D,f}(h) = \Pr_{x \sim D}[h(x) \neq f(x)].$$

Trust that for every $1 > \varepsilon > 0$, m

$$(1-\varepsilon)^m \leq e^{-\varepsilon m}$$

Thus the probability of making an error is going down exponentially fast in the sample size.

Also recall the proof idea:

- Step 1: For any given $h : X \to \{0, 1\}$, $\Pr_{S \sim D^m}[L_S(h) = 0]$ is small and getting smaller with m, provided that $L_{D,f}(h) > \varepsilon$. We are looking at samples that make h look good in spite of h being bad.
- Step 2: Take union over all $h \in H$.

3

A Formal Learning Model

3.1 A formal notion of learnability

PAC learnability

We say that a class of predictors H is **PAC learnable** (Probably Approximately Correct) if there exists a function $m_H : (0,1) \times (0,1) \to \mathbb{N}$ such that there exists a learner

$$A : \bigcup_{m=1}^{\infty} (X \times \{0,1\})^m \to \{f | f : X \to \{0,1\}\}$$

that for every D probability distribution over X and every $f \in H$ and every $\varepsilon, \delta > 0$

$$\Pr_{S \sim D^m, f} [L_{D,f}(A(S)) > \varepsilon] < \delta$$

for every $m \geq m_H(\varepsilon, \delta)$.

Here we can think of ε as an accuracy parameter and δ as confidence parameter.

If we rephrase Theorem 2.1, we get

Theorem

Every finite H is PAC learnable with $m_H(\varepsilon, \delta) \leq \frac{\ln |H| + \ln(1/\delta)}{\varepsilon}$. Furthermore, any ERM_H learner will be successful.

Last time we have showed

$$\Pr_{S \sim D^m, f} [L_{D,f}(A(s)) > \varepsilon] \leq |H| \cdot e^{-\varepsilon m} \leq \delta$$

Then take ln,

$$\ln |H| - \varepsilon m \leq \ln(\delta)$$

Then we get results as desired.

Strength of the PAC definition is that we can guarantee the number of needed examples (for training) regardless of the data distribution D and of which $f \in H$ is used for labeling. We call this is a "**Distribution free guarantee**".

Weakness: If only works if the labeling rule f comes from H.

Relaxation The data is generated by some probability distribution D over $X \times Y$.

We still wish to output a labeling rule $h : X \to Y$.

Assume $Y = \{0, 1\}$, we claim the best predictor h should be

$$h^*(x) = \begin{cases} 1 & \text{if } D((x,1)|x) \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

which is called the Bayes rule. The problem is we *do not* know D. We only see a sample (generated by D).

Note that $L_D(h^*)$ in some cases may be high. If it's a coin flip, and the data generating process is completely random, then this rate will be half.

3.2 A More General Learning Model

Now redefine successful learning to have only a relative error guarantee.

Agnostic PAC learnability

A class of predictors H is **agnostic PAC learnable** if there exist some function $m_H(\varepsilon, \delta) : (0,1) \times (0,1) \to \mathbb{N}$, and a learner A (taking samples, outputting predictors) such that for every D over $X \times Y$ and every $\varepsilon, \delta > 0$

$$\Pr_{S \sim D^m}[L_D(A(S)) > \min_{h \in H}(L_D(h)) + \varepsilon] < \delta$$

whenever $m \geq m_H(\varepsilon, \delta)$.

Weaker notion of learner's success defined relative to some "benchmark" class of functions H. h is **ε-accurate** with respect to D, H if $L_D(h) \leq \min_{h' \in H}(L_D(h')) + \varepsilon$.

Some other learning tasks

1. **Multiclass prediction**.

   The set of labels Y could be larger than just two elements. For example, {Politics, Sports,Entertainment, Finance, ...}

2. **Real valued prediction (Regression)**.

   The set of labels is the real line. For example, predict tomorrow's max temperature.

3.3 More general setup for learning

- Domain set Z
- Set of models M
- Loss of some model: on a given instance z: $\ell(h, z)$

The data is generated by some unknown distribution over Z and we aim to find the best model for that distribution.

$$L_D(h) = \mathbb{E}_{z \sim D} \ell(h, z)$$

So far,

- $Z = X \times \{0, 1\}$
- M: functions from X to $\{0, 1\}$
- $\ell(h, \underbrace{(x, y)}_{z}) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$

$L_D(h)$ coincides with our previous definition $L_D(h) = \Pr_{(x,y)\in D}(h(x) \neq y)$

Let's consider our previous examples with this new setting.

1. Binary label prediction.

   $$Z = X \times \{0,1\}, \quad \ell(h,(x,y)) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$$

2. Multiclass prediction.

   $Z = X \times Y$ where Y is the set of topics from the previous example.

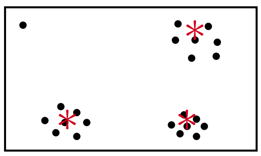
   $$\ell(h,(x,y)) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$$

   These are called 0-1 loss for the obvious reason.

3. Regression (Predicting temperature)

   $Z = X \times \mathbb{R}, \quad \ell(h,(x,t)) = (h(x)-t)^2$ which is called square loss

4. Representing data by k codewords.

   distributions of audio signals

   $Z = \mathbb{R}^d \qquad M =$ vectors of k members of $\mathbb{R}^d \qquad h = (c_1 \ldots c_k)$

   $\ell((c_1 \ldots c_k), z) = \min_{1\leq i\leq k} \|c_i - z\|^2$

3.4 Agnostic PAC-Learning as a game

Fix the domain set X and hypothesis class H.

$$\begin{tikzcd}
\text{Tester} & \text{Learner} \\
(\varepsilon,\delta) \in (0,1)^2 \arrow[r, "\varepsilon{,}\delta"] & m_H(\varepsilon,\delta) \arrow[dl, "m"] \\
D \arrow[d] & \\
S \sim D^m \arrow[r, "S"] & h = A(S) \arrow[dl, "h"] \\
L_D(h) - L_D(h^*) \overset{?}{>} \varepsilon &
\end{tikzcd}$$

How often the learner fails? more than δ?

where $h^* = \operatorname{argmin}_{h\in H} L_D(h)$.

4

Learning via Uniform Convergence

epsilon-representative sample

A sample $S = (z_1 \dots z_m)$ (or $S = (x_1, y_1) \dots (x_m, y_m)$) is **$\varepsilon$-representative** of a class H with respect to a distribution D if

$$\forall h \in H, \quad |L_S(h) - L_D(h)| \leq \varepsilon,$$

where $L_S(h) = \frac{1}{m} \sum_{z \in S} \ell(h, z)$, the Empirical Risk of h.

Note that if S is indeed representative of H with respect to D, then ERM_H is a good learning strategy.

Claim If S is ε-representative of H with respect to D then for any ERM_H function h_S

$$L_D(h_S) \leq \min_{h \in H}(L_D(h)) + 2\varepsilon$$

Proof:
By definition, since S is ε-representative and $h_S \in H$, then $L_D(h_S) \leq L_S(h_S) + \varepsilon$. Since h_S is ERM_H, then $L_S(h_S) \leq \min_{h \in H}[L_S(h)]$. Again since S is ε-representative, we have

$$L_D(h_S) \leq L_S(h_S) + \varepsilon \leq \min_{h \in H}[L_S(h) + \varepsilon] \leq \min_{h \in H}[L_D(h)] + \varepsilon + \varepsilon$$

□

4.1 Finite Classes Are Agnostic PAC Learnable

Next step Show that if a large enough S is picked at random by D then with hight probability, such S will be ε-representative of H with respect to D.

Suggestion Prove an upper bound for sample complexity of a specific algorithm, namely ERM: $A(S) = \operatorname{argmin}_{h \in H} L_D(h)$.

We then present the above claim as a lemma.

Lemma 4.1

If S is ε-rep, then $L_D\left(A^{\mathrm{ERM}}(S)\right) - L_D(h^*) \leq 2\varepsilon$.

sample complexity of uniform convergence

(w.r.t. H) $m_H^{UC}(\varepsilon, \delta)$: the minimum number m such that for every distribution D, if we pick $S \sim D^m$, then with probability at least $1-\delta$, S is ε-representative.

If we have $m_H^{UC}(\varepsilon, \delta)$ samples, then with high probability our sample S is ε-representative $\xrightarrow{\text{lemma}}$ ERM will work $\to$ it will be agnostic PAC-learnable.

Corollary 4.2

$$m_H(\varepsilon, \delta) \leq m_H^{UC}(\varepsilon/2, \delta)$$

New goal find upper-bound for $m_H^{UC}(\varepsilon, \delta)$ in the case $|H| < \infty$

Strategy:

- Step 1: for a single hypothesis $h \in H$, bound the number of samples to make sure that $L_D(h) \approx L_S(h)$ with "high probability".
- Step 2: use union bound to bound the probability that "any" of them fails.

Hoeffding's inequality

Assume $\theta_1, \theta_2, \ldots, \theta_m$ are iid random variables with mean μ that take values in $[a, b]$, then

$$\Pr\left[\left|\mu - \frac{1}{m}\sum \theta_i\right| > \varepsilon\right] < 2\exp\left(\frac{-2m\varepsilon^2}{(b-a)^2}\right)$$

Fix some $h \in H$.

$$\Pr[|L_D(h) - L_S(h)| \geq \varepsilon] = \Pr\left[\left|\mathop{\mathbb{E}}_{z\sim D} \ell(h,z) - \frac{1}{m}\sum_{z\in S} \ell(h,z)\right| \geq \varepsilon\right] \leq 2e^{\frac{-m\varepsilon^2}{(1-0)^2}} = 2e^{-m\varepsilon^2}$$

Proof of the main result:

$$\begin{aligned}
\Pr[S \text{ is not } \varepsilon\text{-representative w.r.t. } H] &= \Pr\left[\exists h \in H, \text{ s.t. } |L_D(h) - L_S(h)| > \varepsilon\right] \\
&\leq \sum_{h\in H} \Pr[|L_D(h) - L_S(h)| > \varepsilon] \qquad \text{by union bound} \\
&\leq \sum_{h\in H} 2e^{-m\varepsilon^2} \quad \text{by Hoeffding's ineq} \\
&= |H| \cdot 2e^{-m\varepsilon^2}
\end{aligned}$$

Then what can we say about $m_H^{UC}(\varepsilon, \delta)$?

$$|H| \cdot 2e^{-m\varepsilon^2} < \delta \implies m_H^{UC} > \frac{\ln(2|H|/\delta)}{2\varepsilon^2}$$

Corollary 4.3

$$m_H(\varepsilon, \delta) \leq \frac{2\ln(2|H|/\delta)}{\varepsilon^2}$$

4.2 PAC-learnable infinite class example

Do we have an infinite class that is PAC-learnable?

Let H^{thr} be the class of all thresholds on $[0, 1]$, that's

$$H^{thr} = \left\{ h_r : h_r(x) = \begin{cases} 0 & x \le r \\ 1 & x > r \end{cases}, \quad r \in [0, 1] \right\}$$

Practical Approach: Discretize

$$H^{thr}_{\alpha} = \left\{ h_r : h_r(x) = \begin{cases} 0 & x \le r \\ 1 & x > r \end{cases}, \quad r \in \left\{ 0, \frac{1}{\alpha}, \frac{2}{\alpha}, \dots, \frac{\alpha - 1}{\alpha}, 1 \right\} \right\}$$

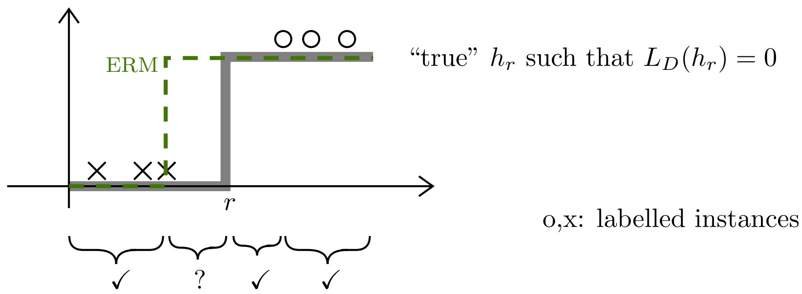
$|H^{thr}_{\alpha}| = \alpha + 1$

In theory, H^{thr}_{α} may not be a good approximation of H^{thr}

$$\min_{h \in H^{thr}} L_D(h) \ll \min_{h \in H^{thr}_{\alpha}} L_D(h)$$

for some specific distribution. Consider the case: $D[\{(\frac{3}{2\alpha}, 0)\}] = D[\{(\frac{7}{2\alpha}, 0)\}] = 0.5$

Is H^{thr} PAC-learnable (realizable case)?



Let A be the ERM algorithm that resolves ties in favor of smaller thresholds.

Let q_ε be the smallest number in $[0, r]$ that satisfies $D_{|x}\{x \in [q_\varepsilon, r]\} \le \varepsilon$.

Claim If sample $S_{|X}$ contains a point in $[q_\varepsilon, r]$ then $L_D(A(S)) \le \varepsilon$.

Proof:
Let t be such point in S. Then

$$\begin{aligned} L_D(A(S)) &\le D_{|X}\{x : x \in (t, r]\} \\ &\le D_{|X}\{x : x \in (q_\varepsilon, r]\} \quad \text{because } t \ge q_\varepsilon \\ &\le \varepsilon \end{aligned}$$

□

Proof of PAC-learnability of H^{thr}:

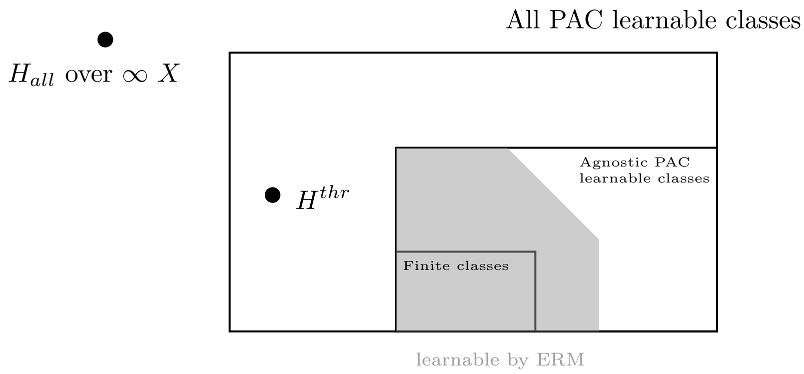
$$\begin{aligned} \Pr[L_D(A(S)) > \varepsilon] &\le D^m_{|X}\{s : \nexists x \in S|_X \text{ s.t. } x \in [q_\varepsilon, r]\} \\ &\overset{iid}{\le} \left(D_{|X}\{x : x \notin [q_\varepsilon, r]\}\right)^m \\ &\le (1 - \varepsilon)^m \le e^{-\varepsilon m} \end{aligned}$$

□

Remark:
So H^{thr} is PAC-Learnable with $m_{H^{thr}}(\varepsilon, \delta) \le \frac{\ln(1/\delta)}{\varepsilon}$.

We will not prove here, but H^{thr} is agnostic PAC-Learnable.

4.3 Summary



Agnostic PAC learnable is stronger than PAC learnable because within agnostic, we require for every distribution, you will be able to get close to the best classifier with respect to that distribution. In PAC, we only require this will hold for the distributions for which one of the element of H is a perfect classifier.

Finally, learnable by ERM, agnostic learnable, PAC learnable are going to be the same family of classes.

Example: non-ERM learnable class

Let $X = \mathbb{R}$. Let $H_{finite} = \{h_A : A \text{ is a finite subset of } \mathbb{R}\}$ where $h_A = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$

Let $H = H_{finite} \cup \{h_{one}\}$

Claim H_{finite} is not learnable (PAC) by ERM.

Proof:
Let P be the uniform distribution over $[0, 1]$. Pick as a labeling rule the all-1 function. We wish to show ERM may fail on this challenge. Pick any sample size m, and $S \sim P^m$, then

$$S = ((x_1, 1), \ldots, (x_m, 1))$$

An ERM algorithm may now pick h_A for $A = \{x_1, \ldots, x_m\}$, then $L_S(h_A) = 0$, but $L_P(h_A) = 1$. h_A fails on every test point $x \notin A$. $\square$

However $H_{thr} = \{h_x : x \in \mathbb{R}\}$ where $h_x(y) = \begin{cases} 1 & y \leq x \\ 0 & \text{otherwise} \end{cases}$

Under this setting, ERM is a successful PAC learner.

5 The Bias-Complexity Tradeoff

Tradeoff between "approximation error" and "estimation error".

$$\varepsilon_{app} = \min_{h \in H} L_D(h)$$
$$\varepsilon_{est} = L_D(h_S) - \varepsilon_{app}$$

NFL shows for large class, estimation error is large; for small class, approximation error is large.

5.1 The No-Free-Lunch Theorem

Theorem 5.1: No-Free-Lunch

Let X be a domain of size n i.e., $|X| = n$). Let H_n^{all} be the set of all possible labelings, i.e., $H_n^{all} = \{h : X \to \{0,1\}\}$ ($|H_n^{all}| = 2^n$). NFL proves that

$$m_{H_n^{all}}\left(\frac{1}{8}, \frac{1}{7}\right) \geq \frac{n}{2}$$

Proof:
Either see textbook or Lecture 8. □

Corollary 5.2

$$m_{H_\infty^{all}}\left(\frac{1}{8}, \frac{1}{7}\right) \geq \infty$$

Therefore, the class of labeling functions over an infinite domain is not PAC-Learnable.

Intuition Assume that $|S| = \frac{n}{2}$. Assume $D_{|n}$ is uniform over X. Then the learner can find out about the labels of points in $S_{|X}$. But for the points are not in the sample, it cannot do better than random guess. (Because every labeling is possible on them) Therefore, it fails on at least $\frac{1}{2}$ of the points in $X \setminus S_{|X}$ (in expectation). It will fail on 25% of the points (expected)

$$\mathop{\mathbb{E}}_{S \sim D^m} L_D(A(S)) \geq \frac{1}{4}$$

6

The VC-Dimension

6.1 Infinite-Size Classes Can Be Learnable

See section 4.2

6.2 The VC-Dimension

shatter

Let H be a class of $\{0,1\}$ functions over some domain X, and let $A \subseteq X$. H **shatters** A if for every $g : A \to \{0,1\}$, $\exists h \in H$ such that for any $x \in A$, $h(x) = g(x)$.

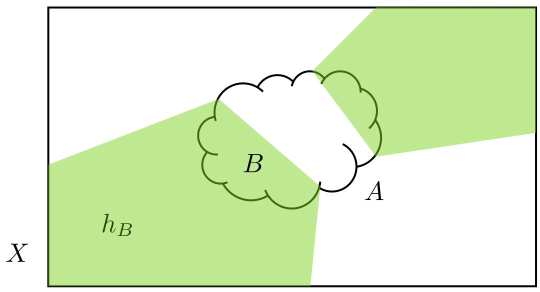
Example:

Let $X = \mathbb{R}$. Consider H_{thr}.

$A = \{7, 10\}$. We have 4 possible g's over A. However, we cannot get $g(7) = 0, g(10) = 1$ from H_{thr}. Thus H does not shatter A.

Note that there is equivalence between functions from X to $\{0,1\}$ and subsets of X. Given $h : X \to \{0,1\}$, define $A_h = \{x \in X : h(x) = 1\}$. Or going backwards, given $A \subseteq X$, define $h_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$

In terms of subsets H is a collection of subsets of X: A is shattered by H if for every $B \subseteq A$, $\exists h_B \in H$ such that $B = h_B \cap A$.



Example:

Let $X = \mathbb{R}^2$. Let $H = \{B_{(x,r)} : x \in \mathbb{R}^2, r \in \mathbb{R}^+\}$ where $B_{(x,r)} = \{y : \|y - x\| \leq r\}$

Any A of size 2 is shattered. Any set A consisting of 3 non-colinear points is shattered by H. Any set A consisting of 3 colinear points is not shattered by H.

VC-dimension

$$\mathrm{VCdim}\,(H) := \max\{|A| : A \text{ is shattered by } H\}$$

and it is ∞ if no maximal such A exists.

6.3 Examples

1. $\mathrm{VCdim}\,(H_{thr}) = 1$

   Proof:
   We have showed any set A of size ≥ 2 is not shattered by H_{thr}. So $\mathrm{VCdim}\,(H_{thr}) \leq 1$. The set $\{1\}$ is shattered by H_{thr}, therefore $\mathrm{VCdim}\,(H_{thr}) \geq 1$. Therefore $\mathrm{VCdim}\,(H_{thr}) = 1$. □

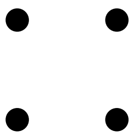
2. $\mathrm{VCdim}\,(H_{finite}) = \infty$

   Proof:
   Every finite A is shattered by H_{finite} because $\forall B \subseteq A$, $h_B \in H_{finite}$ and $h_B \cap A = B$. □

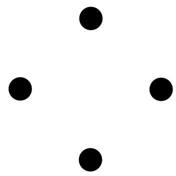
3. Let $X = \mathbb{R}$, $H_{interval} = \{h_{a,b} : a, b \in \mathbb{R}, a < b\}$ where $h_{a,b}(x) = \begin{cases} 1 & x \in [a,b] \\ 0 & \text{otherwise} \end{cases}$

   Every $A \subseteq R$ of size 2 is shattered by $H_{interval}$, then $\mathrm{VCdim}\,(H_{interval}) \geq 2$. Any ≥ 3 point set is not shattered so $\mathrm{VCdim}\,(()\,H_{interval}) = 2$.

4. Axis Aligned Rectangles. $H_{rect} = \{h_{(a,b,c,d)} : a, b, c, d \in \mathbb{R}\}$. $X = \mathbb{R}^2$. See the precise definition in 6.3.3 of textbook.

   not shattered

   shattered

   Claim $\mathrm{VCdim}\,(H_{rec}) \leq 4$.

   Proof:
   Given any set $A \subseteq \mathbb{R}^2$, let x_ℓ^A be the leftmost point of A, x_r^A be the rightmost point of A, x_b^A be the lowest point of A, x_t^A be the highest point of A. Every rectangle $h \in H_{rec}$ that captures $\{x_\ell^A, x_r^A, x_b^A, x_t^A\}$ captures all of A. If $|A| \geq 5$, A contains some $x^* \notin \{x_\ell^A, x_r^A, x_b^A, x_t^A\}$, we cannot get $B = \{x_\ell^A, x_r^A, x_b^A, x_t^A\}$. □

6.4 Some basic properties of VCdim

1. $\mathrm{VCdim}\,(H) \leq \log_2(|H|)$ or $|H| \geq 2^{\mathrm{VCdim}(H)}$

   It takes $2^{|A|}$ h's to shatter a set A.

   This ineq can be not tight. For example, $|H_{thr}| = \infty \gg 2^1$

2. If $H_1 \subseteq H_2$, then $\mathrm{VCdim}\,(H_1) \leq \mathrm{VCdim}\,(H_2)$

Lemma 6.1

If H has infinite VCdim then H is not PAC learnable.

Proof:
Follows from the NFL.

For the sake of contradiction, assume such H is PAC learnable, then for some $m_H : (0,1)^2 \to \mathbb{N}$, some A, for every distribution P over X and every $f \in H$ and every $\varepsilon, \delta > 0$,

$$\Pr_{S \sim P^m, f}[L_{P,f}(A(S)) > \varepsilon] < \delta$$

whenever $m \geq m_H(\varepsilon, \delta)$.

Consider $m_H(\frac{1}{8}, \frac{1}{8})$. By $\mathrm{VCdim}\,(H) = \infty$, $\exists W \subseteq X$ such that H shatters W and $|W| > 2m_H(\frac{1}{8}, \frac{1}{8})$. H induces every possible function from W to $\{0,1\}$. But by the NFL theorem, in such case,

$$m_H(1/8, 1/8) \geq \frac{|W|}{2} > m_H(1/8, 1/8)$$

contradiction. □

6.5 The Fundamental Theorem of PAC learning

Theorem 6.2: The Fundamental Theorem of Statistical Learning

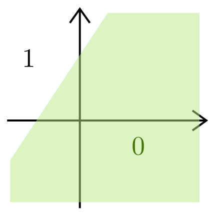
For every domain X and every class H of functions from X to $\{0,1\}$. TFAE:

1. H has the uniform convergence property.
2. ERM is a successful agnostic PAC learner for H.
3. H is agnostic PAC learnable.
4. ERM is a successful PAC learner for H.
5. H is PAC learnable.
6. $\mathrm{VCdim}\,(H)$ is finite.

Proof:
See 6.5 of textbook or Lecture 8.

Hard part is $6 \to 1$. □

A practical class H- the class of linear predictors over $\mathbb{R}^n$



The class HS^n. Let $X = \mathbb{R}^n$. Given some vector $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$. Let

$$h_{w,b}(x) = \mathrm{sign}(\langle w, x \rangle + b) = \begin{cases} +1 & \sum_i^n w_i x_i + b \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Then

$$HS^n = \{h_{w,b} : w \in \mathbb{R}^n, b \in \mathbb{R}\}$$

Index