Computational Learning Theory
Probably Approximately Correct (PAC) Learning

Dimitris Diochnos
School of Computer Science
University of Oklahoma

Outline

Table of Contents

Probably Approximately Correct (PAC) Learning

- PAC learning was introduced by **Leslie Valiant** in 1984 [12].
  - Received the **Turing award** (highest distinction in Computer Science) in 2010 because of several contributions, including PAC learning.
  - Wikipedia entry on Leslie Valiant
- To this day, the majority of provable results in machine learning is related to this model.
- Several good resources on the topic.
  - Tom Mitchell has a good brief description in a chapter devoted to computational learning theory in his book [8, Ch. 7].
  - An Introduction to Computational Learning Theory [7].
  - Foundations of Machine Learning [9].
  - Understanding Machine Learning - From Theory to Algorithms [11].
  - Certainly more books that I forget at the moment...

Reminder: (True) Risk and Empirical Risk

Definition 1 (Risk)

Given a hypothesis $h \in \mathcal{H}$, a target concept $c \in \mathcal{C}$, and an underlying distribution $\mathcal{D}$, the **risk** of h is defined by

$$R_{\mathcal{D}}\left(h, c\right) = \mathbf{Pr}_{x\sim\mathcal{D}}\left(h(x) \neq c(x)\right) = \mathbf{E}_{x\sim\mathcal{D}}\left[\mathbf{1}\left\{h(x) \neq c(x)\right\}\right] .$$

- $\mathbf{1}\left\{\mathcal{A}\right\}$ returns 1 if the event $\mathcal{A}$ holds, o.w. returns 0.

Definition 2 (Empirical Risk)

Given a hypothesis $h \in \mathcal{H}$, a target concept $c \in \mathcal{C}$, and a sample $S = (x_1, \ldots, x_m)$, the **empirical risk** of h is defined by

$$\widehat{R}_S\left(h, c\right) = \frac{1}{m} \cdot \sum_{i=1}^{m} \mathbf{1}\left\{h(x_i) \neq c(x_i)\right\} .$$

Motivating our Discussion on PAC Learning

- Overfitting happens because the empirical risk is a bad estimate of the true risk.
  Q: Can we infer something about the true risk (generalization error) from the empirical risk (training error)?

- Overfitting happens when the learner doesn't see "enough" examples.
  Q: Can we estimate how many examples are enough?
  On a related note:
  Q: Can we estimate how many examples are necessary?

Other Related Questions

- In general, what kind of concepts are easy or hard to learn?
- Which algorithm will we use to process the examples?
  - Does it matter which algorithm we select?
- How frequently will our solution make mistakes during prediction?
  - How confident are we about such a claim?

The Main Goal of PAC Learning

Find a **good approximation** of a function with **high probability**

At the End of the Day

Find a **good approximation** of a function with **high probability**

Two Questions Need to Be Resolved

1. **Statistical.** How many examples are sufficient (or necessary)?
2. **Computational.** Algorithm that solves the problem efficiently?

Basic Terminology for PAC Learning

Goal (Good Approximation with High Probability)
There is a function c over a space $\mathcal{X}$. One wants to come up (in a reasonable amount of time) with a function h such that h is a *good approximation* of c with *high probability*.

Description 1 (Parameters and Terminology)

- $\mathcal{X}$: *Instance Space (say, $\{0,1\}^n$)* $\mathcal{Y}$: *Labels (say, $\{+,-\}$)*
- $c \in \mathcal{C}$: *Target concept belonging to a concept class*
- $h \in \mathcal{H}$: *Hypothesis belonging to a hypothesis class*
- *Good Approximation: Small Risk (Error)* ε
- *High Probability: Confidence* $1-\delta$
- *Reasonable Amount of Time: Polynomial w.r.t. input parameters*

- Realizability assumption: $(\forall c \in \mathcal{C})(\exists h \in \mathcal{H})(\forall x \in \mathcal{X})\,[h(x) = c(x)]$
  ($\mathcal{H}$ is at least as expressive as $\mathcal{C}$; we will see examples later)

PAC Learning

Definition 3 (PAC Learning)

A concept class $\mathcal{C}$ is said to be **PAC-learnable** if there exists an algorithm $\mathcal{A}$ and a polynomial function $\text{poly}(\cdot,\cdot,\cdot,\cdot)$ such that for any $\varepsilon > 0$ and $\delta > 0$, for all distributions $\mathcal{D}$ on $\mathcal{X}$ and for any target concept $c \in \mathcal{C}$, the following holds for any sample size $m \geq \text{poly}(1/\varepsilon, 1/\delta, n, \mathit{size}(c))$:

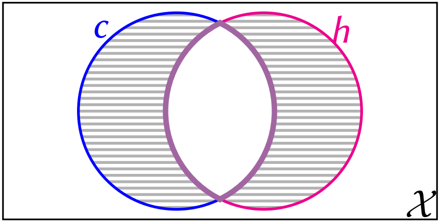
$$\mathbf{Pr}_{S\sim\mathcal{D}^m}\left(R_{\mathcal{D}}\left(h, c\right) \leq \varepsilon\right) \geq 1 - \delta$$

If $\mathcal{A}$ further runs in $\text{poly}(1/\varepsilon, 1/\delta, n, \mathit{size}(c))$, then $\mathcal{C}$ is said to be **efficiently PAC-learnable**. When such an algorithm $\mathcal{A}$ exists, it is called a **PAC-learning algorithm** for $\mathcal{C}$.

- $\mathit{size}(c)$ denotes the maximal cost for the representation of $c \in \mathcal{C}$.
  Example: Representing a monotone conjunction as a list of the k variables that pose the constraints, takes space $\mathcal{O}\left(k \log n\right)$.

PAC Learning (Summary)

- There is an *arbitrary, unknown* distribution $\mathcal{D}$ over $\mathcal{X}$.
- Learn from *poly* $\left(\frac{1}{\varepsilon}, \frac{1}{\delta}\right)$ many examples $(x, c(x))$, where $x \sim \mathcal{D}$.
- The risk is defined as $\boldsymbol{R}_{\boldsymbol{\mathcal{D}}}(\boldsymbol{h}, \boldsymbol{c}) = \mathbf{Pr}_{x\sim\mathcal{D}}(h(x) \neq c(x))$.



Goal 1 (PAC Criterion)

$$\mathbf{Pr}_{S\sim\mathcal{D}^m}\left(R_{\mathcal{D}}(h, c) \leq \varepsilon\right) \geq 1 - \delta\ .$$

Agnostic PAC Learning

Definition 4 (Agnostic PAC Learning)

Let $\mathcal{H}$ be a hypothesis space. Algorithm $\mathcal{A}$ is an **agnostic PAC-learning algorithm** if there exists a polynomial function $\mathrm{poly}(\cdot,\cdot,\cdot,\cdot)$ such that for any $\varepsilon > 0$, $\delta > 0$, for all distributions $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the following holds for any sample size $m \geq \mathrm{poly}(1/\varepsilon, 1/\delta, n, size(c))$:

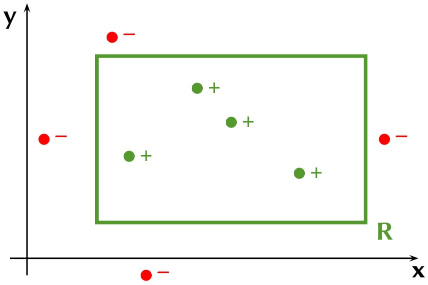
$$\mathbf{Pr}_{S\sim\mathcal{D}^m}\left(R_{\mathcal{D}}\left(h, c\right) \leq \min_{h^\star\in\mathcal{H}}\left\{R_{\mathcal{D}}\left(h^\star, c\right)\right\} + \varepsilon\right) \geq 1 - \delta$$

If $\mathcal{A}$ further runs in $\mathrm{poly}(1/\varepsilon, 1/\delta, n, size(c))$, then it is said to be an **efficient agnostic PAC-learning algorithm**.

Remark 1

We have a more general scenario (stochastic) since $\mathcal{D}$ is defined on $\mathcal{X} \times \mathcal{Y}$. (The label of the point is not unique.)

Example: PAC Learning Axis-Aligned Rectangles

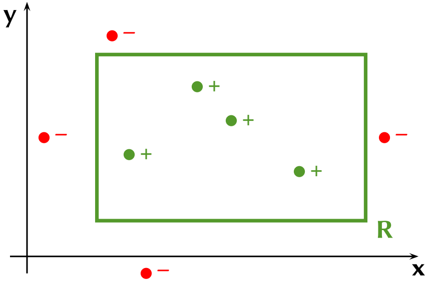


Problem. We want to learn an unknown rectangle R in the Euclidean plane $\mathbb{R}^2$ whose sides are parallel to the coordinate axes.
Information. Points $p \in \mathbb{R}^2$ drawn from some fixed probability distribution $\mathcal{D}$ over $\mathbb{R}^2$ together with their labels.

+: point contained in R

−: point not contained in R

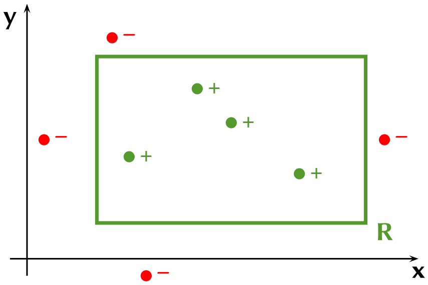
Example: PAC Learning Axis-Aligned Rectangles (cont'd)



Goal. Use as few examples as possible and as little computation as possible to pick a hypothesis (rectangle) R' which is a close approximation of R.

Informally. The player's knowledge of R is tested by picking a new point at random from the same probability distribution $\mathcal{D}$ and checking whether the player can correctly decide whether the point falls inside or outside of R.

Example: PAC Learning Axis-Aligned Rectangles (cont'd)

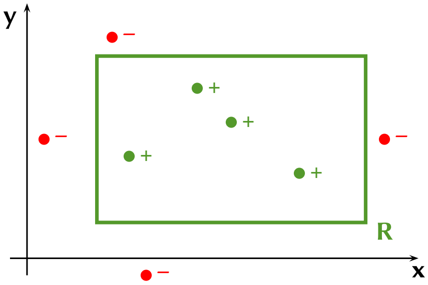


Goal. Use as few examples as possible and as little computation as possible to pick a hypothesis (rectangle) R' which is a close approximation of R.

Formally. We measure the risk (error rate) of R' as the probability that a randomly chosen point from $\mathcal{D}$ falls in the region

$$R \triangle R' = (R \setminus R') \cup (R' \setminus R)$$

Example: PAC Learning Axis-Aligned Rectangles (cont'd)

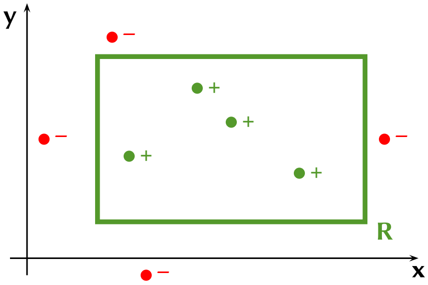


Motivation. For example: "men of medium build".

- Say, [5' 6" – 6'] $\times$ [150 – 200 pounds]

Assumption. Points are drawn according to the same probability distribution $\mathcal{D}$ as during the training phase.

Example: PAC Learning Axis-Aligned Rectangles (cont'd)

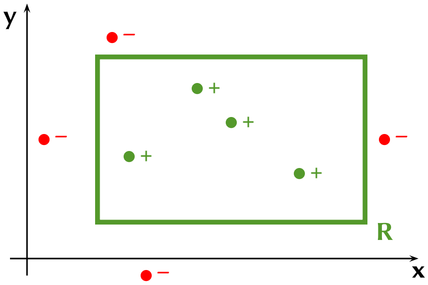


To show: For any target rectangle R, and any distribution $\mathcal{D}$, and for any small values ε and δ ($0 < \varepsilon, \delta < 1/2$), for a suitably chosen value of sample size m, then

$$\mathbf{Pr}_{S\sim\mathcal{D}^m}\left(R_{\mathcal{D}}\left(\mathrm{R}, \mathrm{R}'\right) \leq \varepsilon\right) \geq 1 - \delta\,.$$

(remark: $R_{\mathcal{D}}\left(\mathrm{R}, \mathrm{R}'\right) = \mathbf{Pr}_{\mathcal{D}}\left(\mathrm{R} \triangle \mathrm{R}'\right)$)

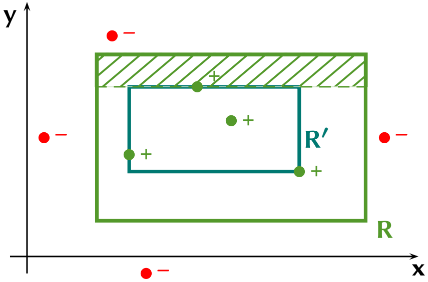
Example: PAC Learning Axis-Aligned Rectangles (cont'd)



What is a good strategy to solve this problem?

Hint: FIND-S

Find-S on Axis-Aligned Rectangles

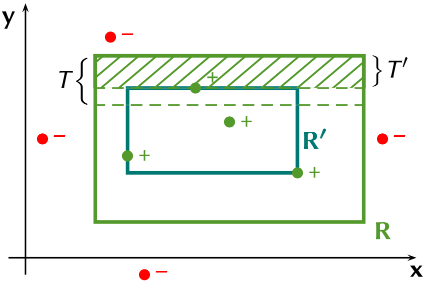


$$R' \subseteq R \Rightarrow R \triangle R' = R \setminus R' = \text{union of 4 rectangular strips}$$

Can we guarantee that each strip has weight under $\mathcal{D}$ at most $\varepsilon/4$?

(Then, the error of R' is at most $4(\varepsilon/4) = \varepsilon$.)

FIND-S on Axis-Aligned Rectangles (cont'd)

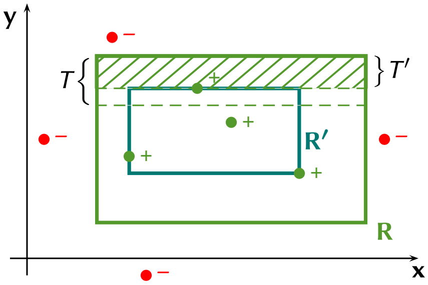


Define T to be rectangular strip along the inside top of R that encloses weight *exactly* $\varepsilon/4$ under $\mathcal{D}$. (Sweep the top edge of R downwards until we have swept out weight $\varepsilon/4$.)

Bad Situation. $T' \supseteq T \Rightarrow \mathbf{Pr}_{\mathcal{D}}\left(T'\right) \geq \varepsilon/4$.

- Will happen only if no point in T appears in S. (Note that the particular point is *positive*.

FIND-S on Axis-Aligned Rectangles (cont'd)



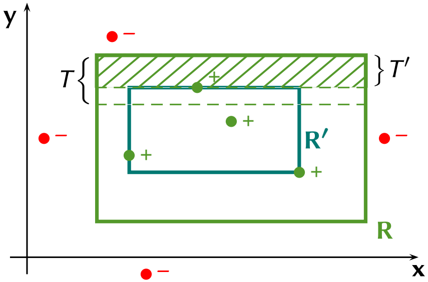
By definition of T, a single draw from $\mathcal{D}$ will miss the region T with probability exactly $1 - \varepsilon/4$.

$\Longrightarrow$ m independent draws from $\mathcal{D}$ all miss T with probability

$$\left(1 - \frac{\varepsilon}{4}\right)^m$$

- same analysis for the other three strips.

Find-S on Axis-Aligned Rectangles (cont'd)



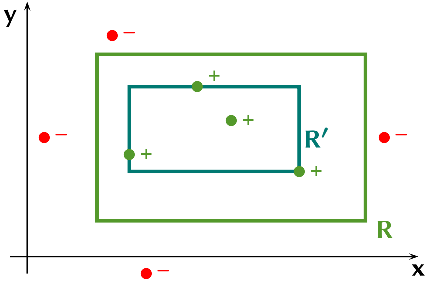
[Union Bound] The probability that *any of the four* rectangular strips of $\mathrm{R} \setminus \mathrm{R}'$ has weight greater than $\varepsilon/4$ is at most

$$4\left(1 - \varepsilon/4\right)^m .$$

- Want $4(1-\varepsilon/4)^m \leq \delta$. Enough if

$$4(1-\varepsilon/4)^m \leq 4e^{-\varepsilon m/4} \leq \delta \Longrightarrow \boxed{m \geq \frac{4}{\varepsilon} \cdot \ln\left(\frac{4}{\delta}\right)}$$

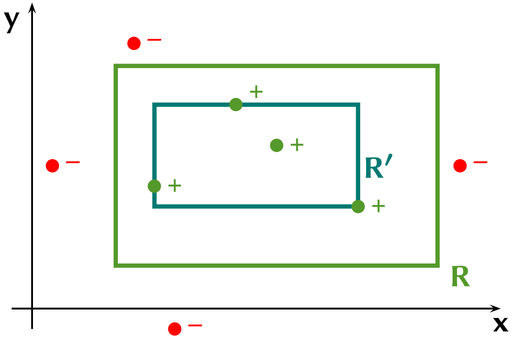
Find-S on Axis-Aligned Rectangles (cont'd)



$$m \geq \frac{4}{\varepsilon} \cdot \ln\left(\frac{4}{\delta}\right)$$

- Analysis holds for any $\mathcal{D}$ (only independence was used)
- The bound behaves as expected (accuracy, confidence)
- The algorithm is efficient
  - m is a slowly growing function of ε, δ
  - tightest fit is easy to compute.

FIND-S on Axis-Aligned Rectangles (cont'd)



Theorem 5

The concept class of axis-aligned rectangles over the Euclidean plane $\mathbb{R}^2$ is efficiently PAC learnable.

Example: PAC Learning Boolean Conjunctions

Problem. Learn $\mathcal{C}_n$: the class of all conjunctions of literals over $x_1, \ldots, x_n$.
(literal: variable x_i, or its negation)
$\mathcal{X}_n = \{0, 1\}^n$
$a \in \mathcal{X}_n$ is a truth assignment (a_i is the i-th bit)
For example,

$$x_1 \wedge \overline{x}_3 \wedge x_4 = \{a \in \{0, 1\}^n : a_1 = 1, a_3 = 0, a_4 = 1\} .$$

$size(c) \leq 2n$ for any $c \in \mathcal{C}$
(binary encoding of any $c \in \mathcal{C}$ has length $\mathcal{O}(n \lg n)$)

Theorem 6

The representation class of conjunctions of Boolean literals is efficiently PAC learnable.

Can you guess the algorithm?

Find-S on PAC Learning Boolean Conjunctions

Let $\mathcal{X}_n = \{0, 1\}^6$ and $c = x_1 \wedge \overline{x}_3 \wedge x_4$.

1. Start with $h = x_1 \wedge \overline{x}_1 \wedge \cdots \wedge x_n \wedge \overline{x}_n = \textsc{False}$.
2. Request m examples and look at the positive ones.
3. Delete the variables that are falsified by the positive examples.

A Study of Thinking [5]

example	hypothesis h
	$x_1 \wedge \overline{x}_1 \wedge x_2 \wedge \overline{x}_2 \wedge x_3 \wedge \overline{x}_3 \wedge x_4 \wedge \overline{x}_4 \wedge x_5 \wedge \overline{x}_5 \wedge x_6 \wedge \overline{x}_6$
$((110101), +)$	$x_1 \wedge x_2 \wedge \overline{x}_3 \wedge x_4 \wedge \overline{x}_5 \wedge x_6$
$((110111), +)$	$x_1 \wedge x_2 \wedge \overline{x}_3 \wedge \wedge x_4 \wedge x_6$
$((100111), +)$	$x_1 \wedge \overline{x}_3 \wedge x_4 \wedge x_6$

- h will never err on a negative example of c (h is more specific than c)
- Let z be a literal in $h \setminus c$. Then z causes h to err only on those positive examples of c in which $z = 0$.

$$p(z) = \mathbf{Pr}_{a \sim \mathcal{D}}\left(c(a) = 1 \wedge z \text{ is a } 0 \text{ in } a\right)$$

Find-S on PAC Learning Boolean Conjunctions (cont'd)

$$p(z) = \mathbf{Pr}_{a\sim\mathcal{D}}\left(c(a) = 1 \wedge z \text{ is a } 0 \text{ in } a\right)$$

- Every mistake of h can be "blamed" on at least one literal z of h. By the definition of risk and the union bound we have:

$$R_{\mathcal{D}}\left(h, c\right) = \mathbf{Pr}_{a\sim\mathcal{D}}\left(h(a) \neq c(a)\right) \leq \sum_{z\in h} p(z).$$

- Define a **literal** to be **bad** if $p(z) \geq \frac{\varepsilon}{2n}$.
  - If h contains no bad literals, then

$$R_{\mathcal{D}}\left(h, c\right) \leq \sum_{z\in h} p(z) \leq 2n \cdot \left(\frac{\varepsilon}{2n}\right) = \varepsilon\,.$$

Find-S on PAC Learning Boolean Conjunctions (cont'd)

Bad literal z: $p(z) \geq \frac{\varepsilon}{2n}$, where $p(z) = \mathbf{Pr}_{a\sim\mathcal{D}}\left(c(a) = 1 \wedge z \text{ is a } 0 \text{ in } a\right)$.

We want to upper bound the probability that a bad literal will appear in h.

- For any fixed bad literal z, the probability that this literal is not deleted from h after m examples is at most

$$\left(1 - \frac{\varepsilon}{2n}\right)^m \leq e^{-\varepsilon m/(2n)}$$

$\Longrightarrow$ By the union bound, the probability that there is *some* bad literal that is not deleted from h after m examples, is at most

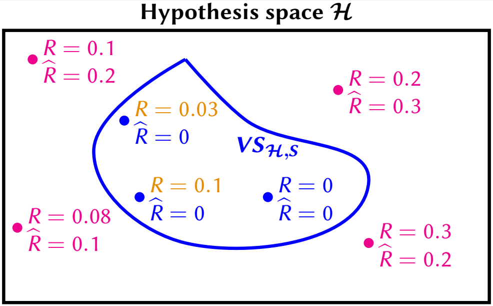
$$2n \cdot e^{-\frac{\varepsilon m}{2n}}$$

Thus,

$$m \geq \left\lceil \frac{2n}{\varepsilon} \cdot \ln\left(\frac{2n}{\delta}\right)\right\rceil$$

examples are enough to guarantee with probability at least $1 - \delta$ that h will have risk at most ε with respect to c and $\mathcal{D}$. (forward to slide 32)

Version Spaces Revisited



Is there a general strategy for PAC learning a concept class?

YES! Occam algorithms:

- Draw a large enough sample S so that (w.h.p.) we can eliminate all those hypotheses that have high risk.
- Any h that survives in $VS_{\mathcal{H},S}$ must have low true risk since it is consistent with S.
- Pick *any* such function from the version space. *(Find-S is your friend...)*

How Many Examples are Enough?

Theorem 7 (PAC Learning of Finite Concept Classes; [3])

Assume that we want to learn a $c \in \mathcal{C}$ using a hypothesis space $\mathcal{H}$ that contains a ***finite amount*** *$|\mathcal{H}|$ of functions, in the* ***realizable*** *case. For any distribution $\mathcal{D}$, drawing $m \geq \frac{1}{\varepsilon} \cdot \left(\ln|\mathcal{H}| + \ln\frac{1}{\delta}\right)$ examples are* ***enough*** *to guarantee that* ***any consistent*** *hypothesis h satisfies $\mathbf{Pr}\left(R_{\mathcal{D}}\left(h, c\right) \leq \varepsilon\right) \geq 1 - \delta$.*

Proof.

Call a hypothesis h **bad** if $R_{\mathcal{D}}\left(h, c\right) > \varepsilon$. Then for such a bad h,

$$\begin{aligned} \mathbf{Pr}\left(h \text{ is consistent with the first training example}\right) &< (1-\varepsilon) \\ \mathbf{Pr}\left(h \text{ is consistent with all } m \text{ training examples}\right) &< (1-\varepsilon)^m \end{aligned}$$

Let $h_1, h_2, \ldots, h_k$ be all the $\boldsymbol{k}$ hypotheses from $\boldsymbol{\mathcal{H}}$ that are **bad**. For each such bad hypothesis h_i with $i \in \{1, \ldots, k\}$, consider the bad event

$$B_i \equiv h_i \text{ is consistent with all } m \text{ training examples}$$

$\mathbf{Pr}\left(B_1 \vee \ldots \vee B_k\right) \leq \sum_{i=1}^{k} \mathbf{Pr}\left(B_i\right) < k \cdot (1-\varepsilon)^m \leq |\mathcal{H}|\,(1-\varepsilon)^m \leq |\mathcal{H}| \cdot e^{-\varepsilon \cdot m}.$ □

Applications of Occam's Razor

Occam Bound. $m \geq \frac{1}{\varepsilon} \cdot \left(\ln\left(|\mathcal{H}|\right) + \ln\left(\frac{1}{\delta}\right)\right)$

Corollary 8 (PAC Learning PlayTennis)

PlayTennis is PAC learnable to risk 0.2 *with probability at least* 0.9 *using* $m = 46$ *examples.*

Proof.

- 1 attribute with 3 values (Sky: Sunny, Cloudy, Rainy)
- 5 attributes with 2 values (AirTemp, Humidity, Wind, Water, Forecast)

Language: Conjunction of atttributes, or null concept:
$|\mathcal{H}| = 4 \cdot 3^5 + 1 = 973$. Therefore, plugging-in the above values we get:

$$m \geq \left\lceil \frac{1}{0.2} \cdot \left(\ln\left(973\right) + \ln(0.1)\right)\right\rceil = \lceil 45.914 \rceil = 46.$$

Note that there are $3 \cdot 2^5 = 96$ different instances. □

Applications of Occam's Razor

Occam Bound. $m \geq \frac{1}{\varepsilon} \cdot \left(\ln\left(|\mathcal{H}|\right) + \ln\left(\frac{1}{\delta}\right)\right)$

Corollary 9 (PAC Learning Conjunctions)

Conjunctions are PAC learnable using $\mathcal{O}\left(\frac{1}{\varepsilon}(n + \ln\left(\frac{1}{\delta}\right))\right)$ *examples.*

Proof.

$|\mathcal{H}| = 3^n + 1 < 3^{n+1}$. By Theorem 7, $m \geq \left\lceil \frac{1}{\varepsilon} \cdot \left((n+1)\ln(3) + \ln\left(\frac{1}{\delta}\right)\right)\right\rceil$.
Note: Even if we use the bound from a general argument, nevertheless *we have actually improved the sample size by a logarithmic factor* compared to the case where we were *blaming* bad literals for the mistakes.
(see slide 28) □

PAC Learning Conjunctions with Few Relevant Variables

Algorithm Based on Set-Covering.

1. Run FIND-S and form a preliminary hypothesis h.
2. Use the literals that appear in h above as a base to cover the negative examples.
   (we can always form a cover because h is a specialization of c)
3. For each literal $z \in h$, let
   $$S_z = \{\langle x, \ominus\rangle \in S \mid z(x) = 0\}\,.$$
   (that is, S_z is the set of negative examples for which $z = 0$)
4. Find a collection of z's (greedy) such that the z's are literals of h and S_z's cover the set of negative examples of S.
5. Let h' be the conjunction of all such literals.

Then, $|h'| = \mathcal{O}\left(|c| \ln m\right) \approx \mathcal{O}\left(|c| \ln\left(\frac{n}{\varepsilon}\right)\right)$.

Set Cover Problem

Given an input collection S of subsets of $U = \{1, 2, \ldots, m\}$, find a subcollection $T \subseteq S$ such that $|T|$ is minimized and the sets in T form a cover of U:

$$\bigcup_{t \in T} t = U$$

- Assumption: S is itself a cover.
- $opt(S)$ denotes the number of sets in a minimum cardinality cover.
- Set-cover decision problem *"is there a cover of size at most k?"* is *NP*-complete.
- However: efficient greedy heuristic to find a cover $\mathfrak{R}$ of cardinality at most $\mathcal{O}\left(opt(S) \cdot \ln m\right)$.

Set Cover Problem (cont'd)

Want to cover $U = \{1, 2, \ldots, m\}$ using sets from the collection S.

Greedy Heuristic for Set Cover.

1. $\mathfrak{R} = \emptyset$
2. $s^\star = \text{argmax}_{\{s \in S\}} |s|$
3. $\mathfrak{R} = \mathfrak{R} \cup \{s^\star\}$
4. For each set $s \in S$: $s = s \setminus s^\star$
5. If $\mathfrak{R}$ is a cover done; else goto 2.

Let $U^\star \subseteq U$. Then, $\exists t \in S$ such that

$$|t \cap U^\star| \geq \frac{|U^\star|}{opt(S)}$$

since $U^\star$ has a cover of size at most $opt(S)$ (since U does) and at least one of the sets in the optimal cover must cover a $1/opt(S)$ fraction of $U^\star$.

Let $U_i \subseteq U$ be the elements not covered after i steps. Then,

$$|U_{i+1}| \leq |U_i| - |U_i| / opt(S) = |U_i| \cdot (1 - 1/opt(S)) .$$

$$\Rightarrow |U_i| \leq (1 - 1/opt(S))^i \cdot |U_0| = (1 - 1/opt(S))^i \cdot m$$

Want $(1 - 1/opt(S))^i m < 1$. Enough if $e^{-i/opt(S)} m < 1 \Rightarrow \boxed{i > opt(S) \ln m}$

PAC Learning under the Realizability Assumption

- This is a reminder to discuss about a result that Steve Hanneke has achieved in recent years, when the realizability assumption holds.

- However, we need the notion of the VC-dimension in order to understand the result.

Agnostic PAC Learning using a Finite Hypothesis Space

Theorem 10 (Agnostic PAC Learning using a Finite Hypothesis Space)

*Let $\mathcal{H}$ contain a **finite amount** $|\mathcal{H}|$ of functions. For every distribution $\mathcal{D}$, drawing $m \geq \frac{2}{\varepsilon^2} \cdot \ln\left(\frac{2|\mathcal{H}|}{\delta}\right)$ examples are **enough** to guarantee that an ERM (empirical risk minimization) algorithm $\mathcal{A}$ will return a hypothesis h that satisfies $\mathbf{Pr}\left(R_{\mathcal{D}}(h, c) \leq \min_{h^\star \in \mathcal{H}}\left\{R_{\mathcal{D}}(h^\star, c)\right\} + \varepsilon\right) \geq 1 - \delta$.*

Proof Sketch.

1. Compute the risk of each $h \in \mathcal{H}$ within $\varepsilon/2$ of its true value, except with probability at most $\delta/|\mathcal{H}|$. (Hint: Hoeffding's Bound)
2. Argue that: *(free space on next two slides)*

$$R_{\mathcal{D}}(h, c) \leq \widehat{R}_S(h, c) + \varepsilon/2 \leq \widehat{R}_S(h^\star, c) + \varepsilon/2 \leq R_{\mathcal{D}}(h^\star, c) + \varepsilon . \qquad \square$$

Reminder on Hoeffding's Bound

Proposition 1 (Hoeffding's Bound)

Let $X_1, \ldots, X_m$ be m independent random variables, each taking values in the range $\mathfrak{I} = [\alpha, \beta]$. Let $X = \frac{1}{m}\sum_{i=1}^{m} X_i$ and $\mu = \mathbf{E}[X]$ denote the mean of their expectations. Then,

$$\mathbf{Pr}\left(|X - \mu| \geq \epsilon\right) \leq 2e^{-2R\epsilon^2/(\beta-\alpha)^2} .$$

Slide Intentionally Left Blank

What if $|\mathcal{H}| = \infty$?

- We will deal with $|\mathcal{H}| = \infty$ later and prove similar results to what we have just seen.

- But for now we will continue with finite hypotheses spaces.

Can we Learn a Disjunction of $k \geq 2$ Conjunctions?

- Say $k = 3$. Then a function looks like $(x_1 \wedge x_5) \vee (\overline{x_2} \wedge x_4 \wedge x_7) \vee (x_3 \wedge \overline{x}_4 \wedge \overline{x_5} \wedge x_7 \wedge \overline{x_8})$.
- Then, $|\mathcal{C}| \leq (3^n + 1) \cdot (3^n + 1) \cdot (3^n + 1) \leq 3^{n+1} \cdot 3^{n+1} \cdot 3^{n+1} = 3^{3n+3}$.
- The previous theorem implies $m = \left\lceil \frac{1}{\varepsilon} \cdot \ln\left(\frac{3^{3n+3}}{\delta}\right)\right\rceil = \left\lceil \frac{3n+3}{\varepsilon} \cdot \ln\left(\frac{3}{\delta}\right)\right\rceil$ training examples are more than enough for PAC learning the class.

So the question becomes:

Is there an algorithm for efficiently PAC learning such functions?

The answer is quite surprising!

- Assuming $NP \neq RP$, we cannot do that efficiently if we use $\mathcal{H} = \mathcal{C}$. (proper learning)
- However, we can PAC learn $\mathcal{C}$ efficiently if we use a larger class of functions as our hypothesis space $\mathcal{H}$. (representation-independent learning)

The Complexity Class *RP*

Randomized Polynomial (*RP*) time. Complexity class of problems for which a non-deterministic Turing machine:

- runs in poly-time w.r.t. the input size,
- if the correct answer is *NO* it returns *NO*,
- if the correct answer is *YES* it returns *YES* with probability $p \geq 1/2$.

(a *YES* answer is always correct!)

- For correct answer being *YES*, we get misleading k consecutive *NO*'s in k runs with probability $\leq 2^{-k}$.

(Receiving a *YES* would change our evaluation.)

- Class ***co-RP***: *NO* is always correct; *YES* might be incorrect.
- It holds: $P \subseteq RP \subseteq NP$.

Alternative definition: In *RP* the NTM accepts a constant fraction of the computation paths. (In *NP* we only need one accepting path.) This immediately shows that $RP \subseteq NP$.

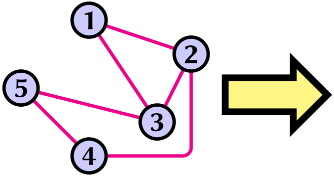
- Let us return to our problem now.

An Intractability Result

Theorem 11

If RP $\neq$ NP, the representation class of k-term DNF formulae is not efficiently PAC learnable for any $k \geq 2$.

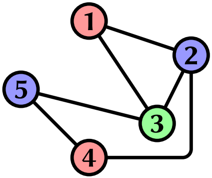
Proof Idea: Reduce Graph 3-Coloring problem to the problem of finding a consistent 3-term DNF formula with a sample $S_G = S_G^+ \cup S_G^-$.



$\langle(0,1,1,1,1),+\rangle \langle(0,0,1,1,1),-\rangle$
$\langle(1,0,1,1,1),+\rangle \langle(0,1,0,1,1),-\rangle$
$\langle(1,1,0,1,1),+\rangle \langle(1,0,0,1,1),-\rangle$
$\langle(1,1,1,0,1),+\rangle \langle(1,0,1,0,1),-\rangle$
$\langle(1,1,1,1,0),+\rangle \langle(1,1,0,1,0),-\rangle$
$\langle(1,1,1,0,0),-\rangle$

- Positive examples encode the vertices of the given graph.
- Negative examples encode the edges of the given graph.
- Show: G is 3-colorable *iff* S_G is consistent with some 3-term DNF.

G is 3-colorable $\Rightarrow S_G$ consistent with some 3-term DNF



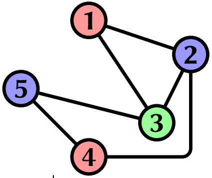
$$\begin{array}{ll}
\langle(0,1,1,1,1),+\rangle & \langle(0,0,1,1,1),-\rangle \\
\langle(1,0,1,1,1),+\rangle & \langle(0,1,0,1,1),-\rangle \\
\langle(1,1,0,1,1),+\rangle & \langle(1,0,0,1,1),-\rangle \\
\langle(1,1,1,0,1),+\rangle & \langle(1,0,1,0,1),-\rangle \\
\langle(1,1,1,1,0),+\rangle & \langle(1,1,0,1,0),-\rangle \\
 & \langle(1,1,1,0,0),-\rangle
\end{array}$$

node	*color*
1	*red*
2	*blue*
3	*green*
4	*red*
5	*blue*

$$\Longrightarrow \begin{cases} T_r & = & x_2 \wedge x_3 \wedge x_5 \\ T_b & = & x_1 \wedge x_3 \wedge x_4 \\ T_g & = & x_1 \wedge x_2 \wedge x_4 \wedge x_5 \end{cases} \Longrightarrow \varphi = T_r \vee T_b \vee T_g$$

- Consider a positive example $v(i) \in S_G^+$. Let $color(node\ i) = red$ (similar argument for other colors). Then, T_r is a conjunction of non-red nodes, so $v(i)$ satisfies T_r (and therefore φ).

G is 3-colorable $\Rightarrow S_G$ consistent with some 3-term DNF



$$
\begin{array}{rr}
\langle(0,1,1,1,1),+\rangle & \langle(0,0,1,1,1),-\rangle \\
\langle(1,0,1,1,1),+\rangle & \langle(0,1,0,1,1),-\rangle \\
\langle(1,1,0,1,1),+\rangle & \langle(1,0,0,1,1),-\rangle \\
\langle(1,1,1,0,1),+\rangle & \langle(1,0,1,0,1),-\rangle \\
\langle(1,1,1,1,0),+\rangle & \langle(1,1,0,1,0),-\rangle \\
 & \langle(1,1,1,0,0),-\rangle
\end{array}
$$

node	*color*
1	*red*
2	*blue*
3	*green*
4	*red*
5	*blue*

$$
\Longrightarrow \left\{ \begin{array}{lcl} T_r & = & x_2 \wedge x_3 \wedge x_5 \\ T_b & = & x_1 \wedge x_3 \wedge x_4 \\ T_g & = & x_1 \wedge x_2 \wedge x_4 \wedge x_5 \end{array} \right. \qquad \Longrightarrow \varphi = T_r \vee T_b \vee T_g
$$

- Let $e(i,j) \in S_G^-$. A valid 3-coloring with nodes i and j connected by an edge implies that they have a different color. But $e(i,j)$ will falsify at least one of the variables in the term (say T_r) since at least one of the two nodes must have color other than red and is therefore included in the term T_r.

S_G consistent with some 3-term DNF $\Rightarrow$ G is 3-colorable

Let $\varphi = \boldsymbol{T_r} \vee \boldsymbol{T_b} \vee \boldsymbol{T_g}$ be consistent with S_G.

We claim that the following coloring is valid:

- color node i red if $v(i) \in S_G^+$ satisfies T_r.
- color node i blue if $v(i) \in S_G^+$ satisfies T_b.
- color node i green if $v(i) \in S_G^+$ satisfies T_g.
- (break ties arbitrarily if $v(i) \in S_G^+$ satisfies more than one term)

Since φ is consistent with S_G, every $v(i) \in S_G^+$ satisfies some term $\Rightarrow$ every node is assigned a color.

- Suppose nodes i and j are assigned the same color (say red). Then both $v(i)$ and $v(j)$ satisfy term T_r. $\Rightarrow$ $x_i \notin T_r$ and moreover $\overline{x_i} \notin T_r$ because these two vectors satisfy T_r and their i-th bit is 0 in one case and 1 in the other case.
- But $e(i,j)$ and $v(j)$ differ only in their i-th bit and if $v(j)$ satisfies T_r, so does $e(i,j)$. But then this means $e(i,j) \notin S_G^-$ since φ is consistent with S_G. Therefore, (i,j) is not an edge in G as required.

Why the Reduction is About *RP*?

- PAC learning should work for every small ε and every small δ.
- Work against this definition.
- If we have a sample S of m training examples (say, all distinct), a PAC learning algorithm should also be able to learn these m examples to error $\varepsilon = \frac{1}{m+1}$ even when the distribution on these points is uniform; i.e., for every $(x, y) \in S$ it holds $\mathbf{Pr}_{x \sim \mathcal{D}}(x) = \frac{1}{m}$.
- But then this means that the algorithm should create a **consistent** hypothesis with the training examples.

  (Otherwise the risk would be very large.)
- Per the PAC criterion, a consistent hypothesis will be created with high probability.
- This explains why we care about *RP*.

Learning 3-Term DNF Formulae using 3-CNF Formulae

- We use the fact:

$$(u \wedge v) \vee (w \wedge z) = (u \vee w) \wedge (u \vee z) \wedge (v \vee w) \wedge (v \vee z)$$

- So, a 3-term DNF formula can be represented as a 3-CNF formula; i.e., a CNF formula where each clause has at most 3 literals.

$$T_1 \vee T_2 \vee T_3 = \bigwedge_{u \in T_1, v \in T_2, w \in T_3} (u \vee v \vee w)$$

- In general, this construction can take a k-term DNF formula and represent it with a k-CNF formula.
- Reduce the problem of learning a k-CNF formula to learning conjunctions:
  - For every triple (u, v, w) over the original variables $\{x_1, \ldots, x_n\}$, create a variable $y_{u,v,w}$ corresponding to this triple.
  - Hence number of variables $y_{u,v,w}$ is at most $(2n)^3$, which is $O(n^3)$.

(For k-term DNF the corresponding y's will be $O(n^k)$ in total.)

Learning 3-Term DNF Formulae using 3-CNF Formulae

- 3-CNF over $\{x_1, \ldots, x_n\}$ is equivalent to a 3-CNF over the new variables $\{y_{u,v,w}\}$.

So:

- A truth assignment $\sigma \in \{0, 1\}^n$ corresponding to the variables $\{x_1, \ldots, x_n\}$ can be converted in time $O(n^3)$ to a truth assignment corresponding to the variables $\{y_{u,v,w}\}$.
- So, we can run our algorithm for learning conjunctions in polynomial time over the variables $\{y_{u,v,w}\}$.
  - Find-S may run in time $O(mn)$; for m examples of bitsize n each.
  - In the new setting: $n' \mapsto (2n)^3$ and $m' \approx O(n') = O(n^3)$.
- Once we are done learning, we can convert the solution that uses the variables $\{y_{u,v,w}\}$ back to $\{x_1, \ldots, x_n\}$ by simply expanding each variable $\{y_{u,v,w}\}$ to the clause $(u \vee v \vee w)$.

Learning 3-Term DNF Formulae using 3-CNF Formulae

Finally, we need to argue that the solution that we compute indeed has low risk.

- Let c be the target 3-CNF and $\mathcal{D}$ the target distribution over $\{0, 1\}^n$.
- Let c' be the target 3-CNF using the variables $\{y_{u,v,w}\}$ and $\mathcal{D}'$ the (induced) distribution over the assignments to the $\{y_{u,v,w}\}$ variables.
- We need to argue that if h' has risk less than ε, so does h.
  - For $\sigma_1, \sigma_2 \in \{0, 1\}^n$ with $\sigma_1 \neq \sigma_2$, it follows that we have $\sigma_1' \neq \sigma_2'$.
  - So, $h'(\sigma') \neq c'(\sigma') \Rightarrow$ there is a *unique preimage* $\sigma \in \{0, 1\}^n$ such that $h(\sigma) \neq c(\sigma)$ and the weight of σ under $\mathcal{D}$ is the same as that of σ' under $\mathcal{D}'$.

(We have used the fact that *our algorithm learns under any distribution*.)

- For example, let $\mathcal{D}$ be the uniform distribution over $\{0, 1\}^n$; i.e., each variable in the truth assignment is satisfied with probability $1/2$.
- Under $\mathcal{D}'$, a variable $y_{u,v,w}$ corresponding to the clause $(u \vee v \vee w)$ is satisfied with probability $7/8$. Similarly, $y_{u,u,u}$ is satisfied with probability $1/2$, or $y_{u,u,\overline{u}}$ is satisfied with probability 1.

What if $|\mathcal{H}| = \infty$?

We need the VC-Dimension in order to answer that.

Background on the Vapnik-Chervonenkis (VC) Dimension

- The study of the VC dimension and its relevance to distribution-free results is due to the work of Vladimir Vapnik and Alexey Chervonenkis [13]. It is due to the last names of these two that the particular combinatorial parameter received its name.

- The connection that the VC dimension has with PAC learning was popularized by the work of Alselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred Warmuth, in [4]

- Similar ideas extend to situations where we have more than two labels; i.e., for multi-class classification, where the relevant combinatorial parameter there is what is called the *Natarajan dimension* [10] due to Balas Natarajan; see also, [2].

Dichotomies and Different Classifications of a Sample

Definition 12 (Dichotomy)

A **dichotomy** of a set S is a partition of S into two disjoint subsets.

Definition 13 (Number of Classifications of a Sample S)

For any hypothesis space $\mathcal{H}$, for all finite sets $S \subseteq \mathcal{X}$:

$$\Pi_{\mathcal{H}}(S) = \{h \cap S \mid h \in \mathcal{H}\} .$$

- In other words, we want to be able to enumerate all the possible labelings $(h(x_1), h(x_2), \ldots, h(x_m))$ that we can give to the set S, as h runs through $\mathcal{H}$. That is, how many dichotomies $\mathcal{H}$ can induce on S.
- Thus, $\Pi_{\mathcal{H}}(S)$ is the set of all the **behaviors** or **dichotomies** on S that are induced or **realized** by $\mathcal{H}$.

Note that $\Pi_{\mathcal{H}}(S) \leq 2^m$.

Growth Function

Definition 14 (Growth Function)

For any natural number m,

$$\Pi_{\mathcal{H}}(m) = \max\{|\Pi_{\mathcal{H}}(S)| \; : \; S \subseteq \mathcal{X} \; \wedge \; |S| = m\} \,.$$

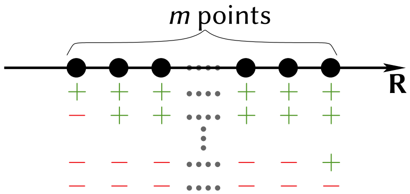
- Measure of complexity for a hypothesis space
- Suppose $d = \mathit{VC\text{-}dim}(\mathcal{H})$. Then,
  - $m \leq d \Longrightarrow \Pi_{\mathcal{H}}(m) = 2^m$.
  - $m > d \Longrightarrow \Pi_{\mathcal{H}}(m) < 2^m$.

Example 15

Rays on a line:

$$h_{\vartheta}(x) = \begin{cases} + & , \quad \text{if } x \geq \vartheta \\ - & , \quad \text{otherwise} \end{cases}$$

$$\Pi_{\mathcal{H}}(m) = m + 1 \,.$$



Shattering

Definition 16

A set of instances $S \in \mathcal{X}^m$ is **shattered** by a hypothesis space $\mathcal{H}$ (or, $\mathcal{H}$ shatters S) if and only if for every dichotomy of S there exists some hypothesis in $\mathcal{H}$ consistent with this dichotomy.

- In other words, if $|\Pi_{\mathcal{H}}(S)| = 2^{|S|}$, then S is shattered by $\mathcal{H}$.
- Further rephrasing: in a set of instances $S \subseteq \mathcal{X}$ ($|S| = m$), $\mathcal{H}$ can give all 2^m possible labelings.

The Vapnik-Chervonenkis Dimension

Definition 17 (*VC* Dimension)

The Vapnik-Chervonenkis dimension, $\mathit{VC\text{-}dim}(\mathcal{H})$, of a hypothesis space $\mathcal{H}$ defined over the instance space $\mathcal{X}$ is **the size of the largest finite subset of $\mathcal{X}$ shattered by $\mathcal{H}$**. If arbitrarily large finite sets of $\mathcal{X}$ can be shattered by $\mathcal{H}$, then $\mathit{VC\text{-}dim}(\mathcal{H}) = \infty$. In other words,

$$\mathit{VC\text{-}dim}(\mathcal{H}) = \max\{m \ : \ \Pi_{\mathcal{H}}(m) = 2^m\}$$

- Lower Bound $\Longrightarrow$ Explicit construction that achieves 2^m.
- Upper Bound $\Longrightarrow$ For *any* set S of size m we cannot achieve 2^m labelings.

More Examples on the VC-Dimension

- Our ray example has *VC-dim* (Rays) ... equal to 1. To see this, recall that a hypothesis h calculates:

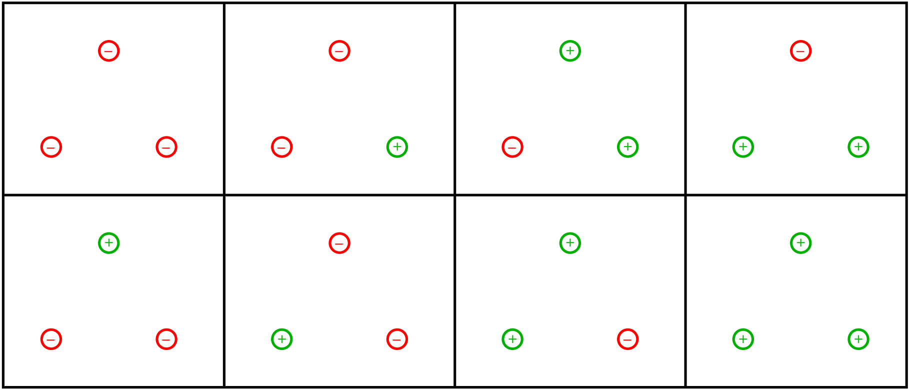
$$h_\vartheta(x) = \mathbf{1}\left\{x \geq \vartheta\right\} \ .$$

  Therefore,
  - One point is shattered.
  - Two points cannot be shattered $(+, -)$

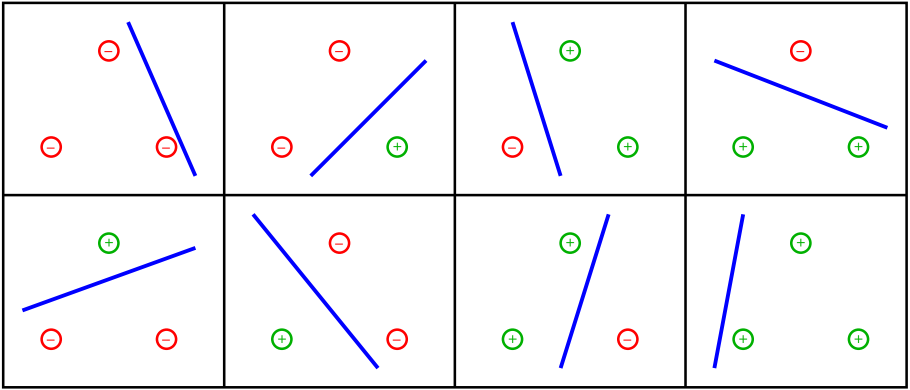
- Axis-aligned rectangles (AAR) in $\mathbb{R}^2$?
  - *VC-dim* (AAR) ≥ 4.
  - *VC-dim* (AAR) < 5. It is impossible to shatter 5 instances.

What about *HALFSPACES*?

Configurations of 3 Points in 2D



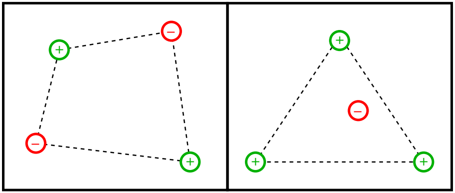
Halfspaces Shatter 3 Points in 2D



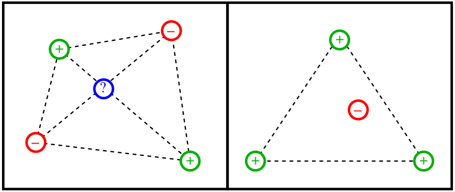
Question 1

Can we shatter 4 points ?

Can Halfspaces Shatter 4 Points in 2D?



Halfspaces *cannot* Shatter 4 Points in 2D



Theorem 18 (Radon)

Any set of $d+2$ points in $\mathbf{R}^d$ can be partitioned into two (disjoint) sets whose convex hulls intersect.

Corollary 19

- $VC\text{-}dim\,(HALFSPACES) = 3$ *in* 2 *dimensions.*
- $VC\text{-}dim\,(HALFSPACES) = d+1$ *in* $d \geq 1$ *dimensions.*

Learning *HALFSPACES*

Do we have an algorithm for learning *HALFSPACES*?

Perceptron

Some Typical Functions Used for Learning

Monotone Conjunctions/Monomials (Boolean AND of some variables chosen from $\{x_1, x_2, \ldots, x_n\}$)

e.g., $c = x_2 \wedge x_5 \wedge x_8$ (sometimes simply write $c = x_2 x_5 x_8$)

- $|\mathcal{H}| = 2^n$.

Conjunctions/Monomials (allow negated variables)

e.g., $c = x_2 \wedge \overline{x_5} \wedge x_8$ $\qquad$ ($c = x_2\overline{x_5}x_8$)

- $|\mathcal{H}| = 3^n + 1$. (including the constant FALSE function.)
- FALSE function can be represented: e.g., $c' = x_1 \wedge \overline{x_1}$.

Halfspaces $\quad$ e.g., $c = sgn(w_0 + w_1 \cdot x_1 + w_2 \cdot x_2 + \ldots + w_n \cdot x_n)$

$$sgn(z) = \begin{cases} +1 & \text{, if } z > 0 \\ -1 & \text{, if } z \leq 0 \end{cases}$$

- $|\mathcal{H}| = \infty$.

Why Are These Functions Used as Toy Examples?

- Exhibit **bias**.
- (Monotone) conjunctions is one of the most basic ways of selecting/combining features/constraints in a prediction mechanism.
- Building blocks for richer classes of functions that are less understood; e.g., general DNF formulae.
  (e.g., learning monotone DNF formulae over the uniform distribution is an open problem.)
- Directly or indirectly, applications to logic, circuit complexity, etc.
- Typical benchmarks as they usually provide interesting, but non-trivial insights of the definitions, the bounds that we should expect to get, etc.
- Can also be useful in contexts of other disciplines (e.g., psychology)

VC Dimension of Finite Hypothesis Spaces

Theorem 20

If $|\mathcal{H}| < \infty$, then VC-dim$(\mathcal{H}) \leq \lg(|\mathcal{H}|)$.

Proof.

The VC dimension of $\mathcal{H}$ is the largest integer d for which we can admit all 2^d possible labelings on a set of instances of size d. That is, $\Pi_{\mathcal{H}}(d) = 2^d$.

However, the number of classifications by a finite hypothesis space $\mathcal{H}$, is at most the number of distinct hypotheses in $\mathcal{H}$. Hence, for any integer m, it holds $\Pi_{\mathcal{H}}(m) \leq |\mathcal{H}|$. In particular,

$$2^d = \Pi_{\mathcal{H}}(d) \leq |\mathcal{H}| \ .$$

Thus, $d \leq \lg(|\mathcal{H}|)$. $\square$

Example: Monotone Conjunctions

Theorem 21

The VC dimension of monotone conjunctions using at most n variables, is exactly n.

Proof.

Upper Bound. $|\mathcal{H}| = 2^n \overset{\text{(Thm 20)}}{\Longrightarrow} VC\text{-}dim\,(\mathcal{H}) \leq n$.
Lower Bound. The following instances give $VC\text{-}dim\,(\mathcal{H}) \geq n$.

$$n \left\{ \begin{array}{cccccccc} 0 & 1 & 1 & 1 & \ldots & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & \ldots & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & \ldots & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & \ldots & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & \ldots & 1 & 1 & 0 \end{array} \right.$$

$\square$

The Φ Function

Definition 22

Define $\Phi_d(m) = \Phi_d(m-1) + \Phi_{d-1}(m-1)$, with $\Phi_d(0) = \Phi_0(m) = 1$. $(m, d \in \mathbb{N} = \{0, 1, \ldots\})$

Lemma 23

$\Phi_d(m) = \sum_{i=0}^{d} \binom{m}{i}$

Proof.

Base cases. If $d = 0$, $\binom{m}{0} = 1 = \Phi_0(m)$. If $m = 0$, $\sum_{i=0}^{d} \binom{0}{d} = \binom{0}{0} = 1$.
Inductive Step. We have the following

$$
\begin{aligned}
\Phi_d(m) &= \Phi_d(m-1) + \Phi_{d-1}(m-1) & \\
&= \sum_{i=0}^{d} \binom{m-1}{i} + \sum_{i=0}^{d-1} \binom{m-1}{i} & \text{(Induction Hypothesis)} \\
&= \sum_{i=0}^{d} \left[\binom{m-1}{i} + \binom{m-1}{i-1} \right] & \text{(define } \tbinom{m-1}{-1} = 0\text{)} \\
&= \sum_{i=0}^{d} \binom{m}{i} & \text{(Pascal's triangle)}
\end{aligned}
$$

$\square$

Polynomial Bound

Lemma 24

For all $m \geq d \geq 1$, $\sum_{i=0}^{d} \binom{m}{i} = \Phi_d(m) \leq \left(\frac{em}{d}\right)^d$

Proof.

We have $0 \leq \dfrac{d}{m} < 1$. We can write

$$\begin{array}{rcll} \left(\frac{d}{m}\right)^d \sum_{i=0}^{d} \binom{m}{i} & \leq & \sum_{i=0}^{d} \left(\frac{d}{m}\right)^i \binom{m}{i} & \\ & \leq & \sum_{i=0}^{m} \left(\frac{d}{m}\right)^i \binom{m}{i} & \\ & \leq & \left(1 + \frac{d}{m}\right)^m & \text{(Binomial Theorem)} \\ & \leq & e^d & \text{(see Lemma 33)} \end{array}$$

Thus, $\sum_{i=0}^{d} \binom{m}{i} = \Phi_d(m) \leq e^d \left(\frac{m}{d}\right)^d = \left(\frac{em}{d}\right)^d$.

Binomial Theorem: $(a+b)^n = \sum_{i=0}^{n} \binom{n}{i} a^i b^{n-i}$

$\square$

Sauer-Shelah Lemma (1972)

Lemma 25 (Sauer-Shelah Lemma)

Let $d \geq 0$ and $m \geq 1$ be given integers and let $\mathcal{H}$ be a hypothesis space such that VC-dim$(\mathcal{H}) = d$. Then,

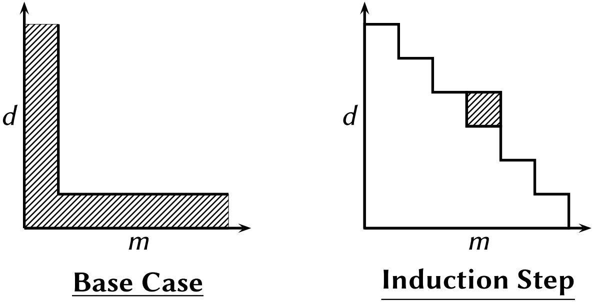
$$\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i} \;=\; \Phi(d, m) = \mathcal{O}\left(m^d\right)$$

- The bound is tight. Examples:
  - Rays in a line: $\Pi_{\mathcal{H}}(m) = m + 1 = 1 + \binom{m}{1} = \Phi_1(m)$,
  - Intervals in a line: $\Pi_{\mathcal{H}}(m) = 1 + \binom{m}{1}\binom{m}{2} = \Phi_2(m)$,
  - and others ...

Has been proved by:

- Sauer and Shelah independently of each other in 1972.
- Vapnik and Chervonenkis also independently proved this lemma slightly earlier.

Proof of Sauer-Shelah Lemma $\left(\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d}\binom{m}{i}\right)$



The proof will be a **complete induction on $m + d$**.

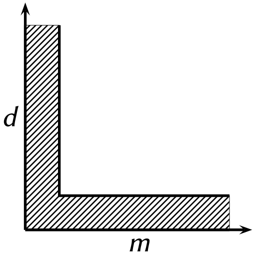
Base Case: Holds for any d and $m = 0$ and for any m and $d = 0$.

Induction Step: Holds for any m, d with $m + d = k$ assuming it holds for all m, d, s.t., $m + d < k$.

Facts that will be used.

- $\binom{m}{k} = \binom{m-1}{k} + \binom{m-1}{k-1}$ (used for constructing Pascal's triangle)
- $\binom{m}{k} = 0$, if $k < 0$ or $k > m$.

Proof of Sauer-Shelah Lemma $\left(\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d}\binom{m}{i}\right)$



Base Case

Case where $d = 0$. Then, we cannot shatter even one instance. Hence, only one labeling can be assigned to any set. In other words, $\Pi_{\mathcal{H}}(m) = 1 = \binom{m}{0} = \sum_{i=0}^{d}\binom{m}{i}$.

Case where $m = 0$. This is a degenerate case where we want to label the empty set. $\Pi_{\mathcal{H}}(m) \leq 1 = \sum_{i=0}^{d}\binom{0}{i}$. (Only one subset of the empty set.)

Perhaps it is simpler to accept the base case when $m = 1$:
In this case, either $\textit{VC-dim}(\mathcal{H}) \geq 1$, in which case we can give 2 labels to a single instance, or it is the case that $\textit{VC-dim}(\mathcal{H}) = 0$ and only one behavior is possible. Either way, it holds that $\Pi_{\mathcal{H}}(m) \leq 2 = 1 + 1 = \binom{1}{0} + \binom{1}{1} = \sum_{i=0}^{d}\binom{0}{i}$
(recall that $\binom{1}{d} = 0$ for $d \geq 2$)

Proof of Sauer-Shelah Lemma $\left(\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}\right)$

Induction Step.

- The main step of the proof is the construction of two new hypothesis spaces $\mathcal{H}_1$ and $\mathcal{H}_2$ to which we can apply our induction hypothesis.
- Given $S = \{x_1, x_2, \ldots, x_m\}$ we want to show $\Pi_{\mathcal{H}}(S) \leq \Phi_d(m)$.

$\mathcal{H}$	x_1	x_2	$\cdots$	x_{m-1}	x_m	
h_1	0	1	1	0	0	$\rightarrow$
h_2	0	1	1	0	1	$\nearrow$
h_3	0	1	1	1	0	$\rightarrow$
h_4	1	0	0	1	0	$\rightarrow$
h_5	1	0	0	1	1	$\nearrow$
h_6	1	1	0	0	1	$\rightarrow$

$\mathcal{H}_1$	x_1	x_2	$\cdots$	x_{m-1}
h_1	0	1	1	0
h_3	0	1	1	1
h_4	1	0	0	1
h_6	1	1	0	0

	$\mathcal{H}_2$	x_1	x_2	$\cdots$	x_{m-1}
$\searrow$	h_2	0	1	1	0
$\searrow$	h_5	1	0	0	1

$\mathcal{H}_1$: Defined by $\mathcal{H}$ restricted on the domain of the first $m-1$ instances of the set S.

$\mathcal{H}_2$: Defined by $\mathcal{H}$ restricted on the domain of the first $m-1$ instances of the set S but have the property that they give a different label in x_m compared to the functions that belong to $\mathcal{H}_1$ and give the same labels as those in $\mathcal{H}_2$ in the set $S_1 = \{x_1, x_2, \ldots, x_{m-1}\}$.

Proof of Sauer-Shelah Lemma $\left(\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}\right)$

Induction Step (cont'd).

Claim 1. *VC-dim* $(\mathcal{H}_1) \leq$ *VC-dim* $(\mathcal{H}) = d$.
(since all sets shattered by $\mathcal{H}_1$, will also be shattered by $\mathcal{H}$)
$\Longrightarrow$ By induction $|\Pi_{\mathcal{H}_1}(S_1)| \leq \Phi_d(m-1)$.

Claim 2. *VC-dim* $(\mathcal{H}_2) \leq d - 1$.
(T shattered by $\mathcal{H}_2 \Rightarrow T \cup \{x_m\}$ shattered by $\mathcal{H}$)
$\Longrightarrow$ By induction $|\Pi_{\mathcal{H}_1}(S_1)| \leq \Phi_{d-1}(m-1)$.

Proof of Sauer-Shelah Lemma $\left(\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d}\binom{m}{i}\right)$

Induction Step (cont'd).

Therefore, we have:

$$\begin{aligned}
|\Pi_{\mathcal{H}}(S)| &= |\Pi_{\mathcal{H}_1}(S_1)| + |\Pi_{\mathcal{H}_2}(S_1)| \\
&= |\mathcal{H}_1| + |\mathcal{H}_2| \\
&\leq \sum_{i=0}^{d}\binom{m-1}{i} + \sum_{i=0}^{d-1}\binom{m-1}{i} && \text{(Induction Hyp.)} \\
&= \sum_{i=0}^{d}\binom{m-1}{i} + \sum_{i=0}^{d}\binom{m-1}{i-1} && \left(\text{since } \binom{m-1}{-1} = 0\right) \\
&= \sum_{i=0}^{d}\left[\binom{m-1}{i} + \binom{m-1}{i-1}\right] \\
&= \sum_{i=0}^{d}\binom{m}{i}
\end{aligned}$$

Fundamental Theorem of Learning Theory

Notation.

- $M(h, S)$: # mistakes h makes on S
- $B \equiv [\exists h \in \mathcal{H} \colon (h \text{ consistent on } S) \wedge R_{\mathcal{D}}(h, c) > \varepsilon]$
- $B' \equiv \left[\exists h \in \mathcal{H} \colon (h \text{ consistent on } S) \wedge M(h, S') \geq \dfrac{m\varepsilon}{2}\right]$
- S : sample of $m > \frac{8}{\varepsilon}$ instances chosen independently from $\mathcal{D}$.
- S' : "ghost sample" of m instances drawn iid from $\mathcal{D}$.
- "Double-sample trick": take the mistakes on S' as a proxy for a hypothesis's generalization error.

Goal. $\mathbf{Pr}(B) \leq \delta$.
Subgoals to prove:

1. $\mathbf{Pr}(B' \mid B) \geq 1/2$
2. $\mathbf{Pr}(B') \leq \delta/2$

Note that: $\mathbf{Pr}(B') \geq \mathbf{Pr}(B' \wedge B) = \mathbf{Pr}(B' \mid B) \cdot \mathbf{Pr}(B) \geq \frac{1}{2} \cdot \mathbf{Pr}(B)$.
So, subgoals (1) and (2) from above imply the theorem.

Fundamental Theorem of Learning Theory

Theorem 26 (Fundamental Theorem of Learning Theory)

Assume that we want to learn a $c \in \mathcal{C}$ using a hypothesis space $\mathcal{H}$ such that $\mathcal{H}$ has finite $\text{VC-dim}(\mathcal{H}) = d \geq 1$ and the realizability assumption holds. Moreover let $0 < \delta, \varepsilon < 1$. Then,

$$m \geq \left\lceil \frac{4}{\varepsilon} \cdot \left(d \cdot \lg\left(\frac{12}{\varepsilon}\right) + \lg\left(\frac{2}{\delta}\right) \right) \right\rceil$$

samples guarantee that for any consistent hypothesis h it holds

$$\mathbf{Pr}_{\mathcal{D}^m}\left(R_{\mathcal{D}}\left(h, c\right) \leq \varepsilon\right) \geq 1 - \delta\,.$$

- We still need an efficient algorithm to efficiently PAC-learn the class.

Fundamental Theorem of Learning Theory

Instead, we will prove in class the following theorem and you will conclude the proof that connects the two statements as an exercise.

Theorem 27

For any $h \in \mathcal{H}$ that is consistent with all $m > \frac{8}{\varepsilon}$ examples that are sampled independently from distribution $\mathcal{D}$, then

$$\mathbf{Pr}_{\mathcal{D}^m}\left(R_{\mathcal{D}}\left(h, c\right) \leq 2 \cdot \frac{\lg \Pi_{\mathcal{H}}(2m) + \lg(2/\delta)}{m}\right) \geq 1 - \delta$$

Fundamental Theorem of Learning Theory (Proof)

Subgoal 1. Want to show: $\mathbf{Pr}\left(B' \mid B\right) \geq 1/2$. Recall:

- $B \equiv [\exists h \in \mathcal{H}\colon (h \text{ consistent on } S) \wedge R_{\mathcal{D}}\left(h, c\right) > \varepsilon]$
- $B' \equiv \left[\exists h \in \mathcal{H}\colon (h \text{ consistent on } S) \wedge M(h, S') \geq \dfrac{m\varepsilon}{2}\right]$

$c(x)$	+	−	+	+	−	−	−	+	+	−	−	−
	○	○	○	○	○	○	○	○	○	○	○	○
$h(x)$	+	−	+	+	−	−	+	−	−	−	−	−
							↑	↑	↑			

mistakes

Suppose B holds.

- Then there exists an $h \in \mathcal{H}$ such that h is consistent on S (first half) and $R_{\mathcal{D}}\left(h, c\right) > \varepsilon$.
- In that case we have $\mathbf{E}\left[M(h, S')\right] = |S'| \cdot R_{\mathcal{D}}\left(h, c\right) > m\varepsilon$. By Lemma 2 of the handout (Tools for Bounding Probabilities), we have $\mathbf{Pr}\left(M(h, S') < \frac{\varepsilon m}{2}\right) \leq 1/2$.
- Hence, $\mathbf{Pr}\left(B' \mid B\right) \geq 1/2$.

Fundamental Theorem of Learning Theory (Proof)

Subgoal 2. Want to show: $\mathbf{Pr}\left(B'\right) \leq \delta/2$. Recall:

- $B' \equiv \left[\exists h \in \mathcal{H}\colon (h \text{ consistent on } S) \wedge M(h, S') \geq \frac{m\varepsilon}{2}\right]$

Consider the following two experiments.
Experiment 1. Choose S, S' iid from $\mathcal{D}$.
Experiment 2. Choose S, S' iid from $\mathcal{D}$ but for $i \in \{1, 2, \ldots, m\}$ swap $x_i \in S$ with $x_i' \in S'$ with probability $1/2$ and call the resulting samples T and T'.

Note. T and T' have the same distribution as S, S'.
Define

$$\begin{aligned} B'' &\equiv \left[\exists h \in \mathcal{H}\colon (h \text{ consistent on } T) \wedge (M(h, T') \geq \tfrac{m\varepsilon}{2})\right] \\ &\equiv \left[\exists h \in \mathcal{H}\colon (M(h, T) = 0) \wedge (M(h, T') \geq \tfrac{m\varepsilon}{2})\right] \end{aligned}$$

Observation 1. It holds that $\mathbf{Pr}\left(B''\right) = \mathbf{Pr}\left(B'\right)$.

Fundamental Theorem of Learning Theory (Proof)

Define

$$b(h) \equiv \left[h \text{ consistent with } T \wedge M(h, T') \geq \tfrac{m\varepsilon}{2}\right]$$

Observation 2. We have $\mathbf{Pr}\left(b(h) \mid S, S'\right) \leq 2^{-m\varepsilon/2}$.

Note that $b(h)$ is asking about the event that all ℓ mistakes that h will make on both T and T', arise only in T'. Then this probability is

$$\frac{\binom{m}{\ell}}{\binom{2m}{\ell}} = \prod_{i=0}^{\ell-1} \frac{(m-i)}{(2m-i)} \leq \prod_{i=0}^{\ell-1} \left(\frac{1}{2}\right) = 2^{-\ell}\,.$$

- One can also prove this with a case-by-case analysis.

Fundamental Theorem of Learning Theory (Proof)

Recall that

$$\begin{cases} b(h) & \equiv & \left[h \text{ consistent with } T \wedge M(h, T') \geq \frac{m\varepsilon}{2}\right] \\ B'' & \equiv & \left[\exists h \in \mathcal{H} \colon (M(h, T) = 0) \wedge (M(h, T') \geq \frac{m\varepsilon}{2})\right] \end{cases}$$

Observation 3. It holds that $\mathbf{Pr}\left(B''\right) \leq \Pi_{\mathcal{H}}(2m) \cdot 2^{-m\varepsilon/2}$.
The number of behaviors we can have on the $2m$ instances in T, T' is finite, given by $\Pi_{\mathcal{H}}(2m)$. For each behavior we select a single representative hypothesis $h \in \mathcal{H}$ giving that behavior, thus creating a set $\mathcal{H}(S, S')$ of $\Pi_{\mathcal{H}}(2m)$ representative hypotheses.
We have:

$$\begin{aligned} \mathbf{Pr}\left(B''\right) &= \mathbf{Pr}\left(\exists h \in \mathcal{H} \colon b(h)\right) \qquad \left(= \mathbf{E}_{S,S'}\left[\mathbf{Pr}\left(B'' \mid S, S'\right)\right]\right) \\ &= \mathbf{E}_{S,S'}\left[\mathbf{Pr}\left(\exists h \in \mathcal{H} \colon b(h) \mid S, S'\right)\right] \text{ (marginalization)} \\ &= \mathbf{E}_{S,S'}\left[\mathbf{Pr}\left(\exists h \in \mathcal{H}(S, S') \colon b(h) \mid S, S'\right)\right] \end{aligned}$$

Fundamental Theorem of Learning Theory (Proof)

In other words, we have:

$$\begin{aligned}
\mathbf{Pr}\left(B''\right) &= \mathbf{E}_{S,S'}\left[\mathbf{Pr}\left(\exists h \in \mathcal{H}(S,S') \colon b(h) \mid S, S'\right)\right] \\
&\leq \mathbf{E}_{S,S'}\left[\sum_{h\in\mathcal{H}(S,S')} \mathbf{Pr}\left(b(h) \mid S, S'\right)\right] \quad \text{(union bound)} \\
&\leq \mathbf{E}_{S,S'}\left[\Pi_{\mathcal{H}}(2m)\cdot 2^{-m\varepsilon/2}\right] \\
&= \Pi_{\mathcal{H}}(2m)\cdot 2^{-m\varepsilon/2}
\end{aligned}$$

Therefore, we have finally proved that

$$\mathbf{Pr}\left(B\right) \leq 2\mathbf{Pr}\left(B'\right) = 2\mathbf{Pr}\left(B''\right) \leq 2\cdot\Pi_{\mathcal{H}}(2m)\cdot 2^{-m\varepsilon/2},$$

which we require to be upper bounded by δ and therefore we get

$$\varepsilon \geq \frac{2}{m}\cdot\left(\lg\left(\Pi_{\mathcal{H}}(2m)\right) + \lg\left(2/\delta\right)\right).$$

QED

VC Dimension: How Many Examples are Necessary for Learning (Distribution Independently)?

Theorem 28

Any algorithm for PAC-learning a concept class of VC dimension d with parameters $\varepsilon < 1/16$ and $\delta \leq 1/15$, must use

$$m > \frac{d-1}{64\varepsilon}$$

training examples in the worst case.

Proving the Lower Bound

Let $X = \{x_1, \ldots, x_d\}$ be shattered by $\mathcal{C}$.

- Construct a pathological distribution that forces any algorithm to take many examples.
- $supp(\mathcal{D}) = X \Rightarrow$ w.l.o.g. $\mathcal{C} = \mathcal{C}(X)$, so $\mathcal{C}$ is a finite class, $|\mathcal{C}| = 2^d$.
- Choosing a c from $\mathcal{C}$ is equivalent to tossing a fair coin d times to determine the labeling on X.
- Suppose there is a learning algorithm $\mathcal{A}$ that uses at most $m = \left\lceil \frac{d-1}{64\varepsilon} \right\rceil$ training examples producing a hypothesis h.
- Want to show: $(\exists \mathcal{D} \text{ on } \mathcal{X})(\exists c \in \mathcal{C}) \left[\mathbf{Pr}_{S \sim \mathcal{D}^m}\left(R_{\mathcal{D}}\left(h, c\right) > \varepsilon\right) > 1/15\right]$.

Proving the Lower Bound

- Define $\mathcal{D}$ independently of $\mathcal{A}$:

$$\begin{cases} \mathbf{Pr}\left(x_1\right) & = & 1 - 16\varepsilon \\ \mathbf{Pr}\left(x_2\right) & = & \mathbf{Pr}\left(x_3\right) = \ldots = \mathbf{Pr}\left(x_d\right) = \frac{16\varepsilon}{d-1} \end{cases}$$

- Let $\mathcal{X}' = \{x_2, x_3, \ldots, x_d\}$.
- Let $R'_{\mathcal{D}}\left(h, c\right) = \mathbf{Pr}_{x\sim\mathcal{D}}\left(h(x) \neq c(x) \wedge x \in \mathcal{X}'\right)$.

Note that

$$\begin{aligned} R_{\mathcal{D}}\left(h, c\right) & = \mathbf{Pr}_{x\sim\mathcal{D}}\left(h(x) \neq c(x)\right) \\ & \geq \mathbf{Pr}_{x\sim\mathcal{D}}\left(h(x) \neq c(x) \wedge x \in \mathcal{X}'\right) \\ & = R'_{\mathcal{D}}\left(h, c\right) . \end{aligned}$$

- It is easier to prove $\mathbf{Pr}_{S\sim\mathcal{D}^m}\left(R'_{\mathcal{D}}\left(h, c\right) > \varepsilon\right) > 1/15$.
  - But then the result follows from the above observation.

Proving the Lower Bound

Probabilistic argument: Pick a random $c \in \mathcal{C}$ and show that c is hard to learn for $\mathcal{A}$ with positive probability. This implies that there is at least one $c \in \mathcal{C}$ that is hard to learn for $\mathcal{A}$.

Idea: Argue that the sample S containing m iid examples from $\mathcal{D}$, will miss more than half of the points from $\mathcal{X}'$.

- h will be 'guessing' the labels for these points $\Rightarrow$ inevitable to have large risk under $\mathcal{D}$.
- Expected # of instances from $\mathcal{X}'$ appearing in S: $\mu = \left[\frac{16\varepsilon}{d-1} \cdot (d-1)\right] \cdot \left(\frac{d-1}{64\varepsilon}\right) = \frac{d-1}{4}$.
- Markov $\Rightarrow$ **Pr** $\left(\text{\# of instances from } \mathcal{X}\text{' in } S \geq \frac{d-1}{2}\right) \leq \frac{\frac{d-1}{4}}{\frac{d-1}{2}} = 1/2$.
- Define the **bad event**

$$B \equiv S \text{ contains less than } \frac{d-1}{2} \text{ instances from } \mathcal{X}'\,.$$

By the above,

$$\mathbf{Pr}_{S\sim\mathcal{D}^m}(B) = 1 - \mathbf{Pr}_{S\sim\mathcal{D}}\left(\text{\# instances from } \mathcal{X}\text{' in S} \geq \frac{d-1}{2}\right) \geq \frac{1}{2}\,. \qquad (1)$$

Proving the Lower Bound

- h is independent of $\mathcal{X}' \setminus S$
- we pick $c \in \mathcal{C}$ at random

So, h will make a mistake on each instance $x \in \mathcal{X}' \setminus S$ with probability $1/2$.

- Each instance $x \in \mathcal{X}' \setminus S$ contributes to $R'_{\mathcal{D}}(h, c)$ an amount of $\frac{1}{2} \cdot \frac{16\varepsilon}{(d-1)}$.
- When the bad event B occurs, we have $|\mathcal{X}' \setminus S| > \frac{d-1}{2}$.

This implies

$$\mathbf{E}_{c,S}\left[R'_{\mathcal{D}}(h, c) \;\middle|\; B\right] > 4\varepsilon\,. \tag{2}$$

- By (1) and (2) we get a lower bound on $\mathbf{E}_{c,S}\left[R'_{\mathcal{D}}(h, c)\right]$:

$$\mathbf{E}_{c,S}\left[R'_{\mathcal{D}}(h, c)\right] \geq \mathbf{E}_{c,S}\left[R'_{\mathcal{D}}(h, c) \;\middle|\; B\right] \cdot \mathbf{Pr}_S(B) > (4\varepsilon) \cdot (1/2) = 2\varepsilon\,.$$

(We used $\mathbf{E}[Y] = \sum_i \mathbf{E}[Y \mid A_i] \cdot \mathbf{Pr}(A_i)$, where A_i: finite or countable partition of the sample space.)

Proving the Lower Bound

$\mathbf{E}_{c,S}\left[R'_{\mathcal{D}}(h,c)\right] > 2\varepsilon \Longrightarrow (\exists c^\star \in \mathcal{C})\left[\mathbf{E}_S\left[R'_{\mathcal{D}}(h,c^\star)\right] > 2\varepsilon\right]$.

- Take that $c^\star$ as the target concept.
- Show that $\mathcal{A}$ will be prone to produce an h with large risk.

$R'_{\mathcal{D}}(h,c) = \mathbf{Pr}_{x\sim\mathcal{D}}\left(h(x) \neq c(x) \wedge x \in \mathcal{X}'\right) \leq \mathbf{Pr}_{x\sim\mathcal{D}}\left(x \in \mathcal{X}'\right) = 16\varepsilon$. So,

$$\mathbf{E}_S\left[R'_{\mathcal{D}}(h,c) \;\middle|\; R'_{\mathcal{D}}(h,c,>)\,\varepsilon\right] \leq 16\varepsilon\,.$$

Therefore,

$$\begin{aligned}
2\varepsilon &< \mathbf{E}_S\left[R'_{\mathcal{D}}(h,c)\right] \\
&= \mathbf{Pr}_S\left(R'_{\mathcal{D}}(h,c) > \varepsilon\right)\cdot \mathbf{E}_S\left[R'_{\mathcal{D}}(h,c) \;\middle|\; R'_{\mathcal{D}}(h,c) > \varepsilon\right] \\
&\quad +\left(1 - \mathbf{Pr}_S\left(R'_{\mathcal{D}}(h,c) > \varepsilon\right)\right)\cdot \mathbf{E}_S\left[R'_{\mathcal{D}}(h,c) \;\middle|\; R'_{\mathcal{D}}(h,c,\leq)\,\varepsilon\right] \\
&\leq \mathbf{Pr}_S\left(R'_{\mathcal{D}}(h,c) > \varepsilon\right)\cdot(16\varepsilon) + \left(1 - \mathbf{Pr}_S\left(R'_{\mathcal{D}}(h,c) > \varepsilon\right)\right)\cdot(\varepsilon) \\
&= 15\varepsilon\cdot \mathbf{Pr}_S\left(R'_{\mathcal{D}}(h,c) > \varepsilon\right) + \varepsilon\,.
\end{aligned}$$

In other words, $\mathbf{Pr}_S\left(R'_{\mathcal{D}}(h,c) > \varepsilon\right) > \frac{1}{15}$.

Summary of Sample Complexity Bounds – Learning in the Realizable Case

Below are the results that we have seen in class.

Theorem 29 ([3])

Let $\mathcal{H}$ be a finite hypothesis class. Under the realizability assumption, a concept class $\mathcal{C}$ is PAC-learnable by $\mathcal{H}$ with sample complexity

$$m \leq \left\lceil \frac{1}{\varepsilon} \cdot \ln\left(\frac{|\mathcal{H}|}{\delta}\right)\right\rceil .$$

Theorem 30 ([4, 13])

Let $\mathcal{H}$ be a hyp. class with $\text{VC-dim}(\mathcal{H}) = d < \infty$. Under the realizability assumption, a concept class $\mathcal{C}$ is PAC-learnable by $\mathcal{H}$ with sample complexity

- $m \in \mathcal{O}\left(\frac{1}{\varepsilon} \cdot \left(d \ln\left(1/\varepsilon\right) + \ln\left(1/\delta\right)\right)\right)$
- $m \in \Omega\left(\frac{1}{\varepsilon}\left(d + \ln(1/\delta)\right)\right)$.

On the Logarithmic Gap of the Sample Complexity Bounds (Learning in the Realizable Case)

Improved Lower Bound. Auer and Ortner have shown in [1] that

$$m \in \Omega\left(\frac{1}{\varepsilon} \cdot (d \ln(1/\varepsilon) + \ln(1/\delta))\right)$$

examples are necessary when we want to guarantee with probability at least $1-\delta$ that $(\forall h \in \mathcal{H})[\widehat{R}_S(h, c) = 0 \Longrightarrow R_{\mathcal{D}}(h, c) \leq \varepsilon]$.

Improved Upper Bound. On the other hand, Hanneke has shown in [6] that when we do more careful selection of an $h \in \mathcal{H}$ that is not just consistent with the training sample S, then we can in fact improve the upper bound to

$$m \in \mathcal{O}\left(\frac{1}{\varepsilon} \cdot (d + \ln(1/\delta))\right) .$$

Hanneke's algorithm, takes a majority vote on classifiers that have been trained on subsets of the entire training set.

Summary of Sample Complexity Bounds – Agnostic Learning

Recall that we want to satisfy: $\mathbf{Pr}\left(R_{\mathcal{D}}\left(h, c\right) \leq \min_{h^\star \in \mathcal{H}} \left\{R_{\mathcal{D}}\left(h^\star, c\right)\right\} + \varepsilon\right) \geq 1 - \delta$.

Theorem 31 (Agnostic PAC Learning – Finite Hypothesis Space; see, e.g., [9])

Let $\mathcal{H}$ be such that $|\mathcal{H}| < \infty$. Then, $\mathcal{H}$ is agnostic PAC learnable with sample complexity

$$m \in \mathcal{O}\left(\frac{1}{\varepsilon^2} \cdot \ln\left(\frac{|\mathcal{H}|}{\delta}\right)\right)$$

Theorem 32 (Agnostic PAC Learning – Finite VC Dimension; see, e.g., [11])

Let $\mathcal{H}$ be a hypothesis space from a domain $\mathcal{X}$ to $\{0, 1\}^n$, such that $\text{VC-dim}(\mathcal{H}) = d < \infty$. Then, $\mathcal{H}$ is agnostic PAC learnable with sample complexity

$$m \in \Theta\left(\frac{1}{\varepsilon^2}\left(d + \ln(1/\delta)\right)\right)$$

- Note that the bound based on the VC dimension is tight.

References I

[1] Peter Auer and Ronald Ortner. A new PAC bound for intersection-closed concept classes. *Machine Learning*, 66(2-3):151–163, 2007.

[2] Shai Ben-David, Nicolò Cesa-Bianchi, David Haussler, and Philip M. Long. Characterizations of Learnability for Classes of {0, ..., n}-Valued Functions. *Journal of Computer and System Sciences*, 50(1):74–86, 1995.

[3] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Occam's Razor. *Information Processing Letters*, 24(6):377–380, 1987.

[4] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, October 1989.

References II

[5] Jerome S. Bruner, Jacqueline J. Goodnow, and George A. Austin. *A study of thinking.* John Wiley & Sons, New York, NY, USA, 1957.

[6] Steve Hanneke. The optimal sample complexity of PAC learning. *Journal of Machine Learning Research*, 17:38:1–38:15, 2016.

[7] Michael J. Kearns and Umesh V. Vazirani. *An Introduction to Computational Learning Theory.* MIT Press, 1994.

[8] Tom M. Mitchell. *Machine Learning.* McGraw-Hill, New York, 1997.

[9] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning.* Adaptive computation and machine learning. MIT Press, 2012.

[10] B. K. Natarajan. On Learning Sets and Functions. *Machine Learning*, 4:67–97, 1989.

References III

[11] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014.

[12] Leslie G. Valiant. A Theory of the Learnable. *Communications of the ACM (CACM)*, 27(11):1134–1142, 1984.

[13] Vladimir N. Vapnik and Alexey Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971. Original publication appeared in 1968 in Russian in Dokl. Akad. Nauk SSSR, 181 (4): 781. 1968.

Table of Contents

Bounding Euler's Constant

Lemma 33

Let $n \in \mathbb{N}^*$. Then, $\left(1 + \frac{1}{n}\right)^n \le e \le \left(1 + \frac{1}{n}\right)^{n+1}$. *(Back to Φ; Lemma 24)*

Proof.

Let $t \in \left[1, 1 + \frac{1}{n}\right]$. Then, $\frac{1}{1+\frac{1}{n}} \le \frac{1}{t} \le 1$. Hence,

$$\int_1^{1+\frac{1}{n}} \frac{1}{1 + 1/n} dt \le \int_1^{1+\frac{1}{n}} \frac{dt}{t} \le \int_1^{1+\frac{1}{n}} 1 \cdot dt$$

Equivalently, $\frac{1}{1+1/n} \cdot [t]_1^{1+1/n} \le [\ln(t)]_1^{1+1/n} \le [t]_1^{1+1/n}$. In other words,

$$\frac{n}{n+1} \cdot \frac{1}{n} \le \ln\left(1 + \frac{1}{n}\right) \le \frac{1}{n} \tag{3}$$

$\overset{\text{LHS of (3)}}{\Longrightarrow}$ $e^{\frac{1}{n+1}} \le 1 + \frac{1}{n} \Longleftrightarrow e \le (1 + 1/n)^{n+1}$

$\overset{\text{RHS of (3)}}{\Longrightarrow}$ $1 + \frac{1}{n} \le e^{\frac{1}{n}} \Longleftrightarrow (1 + 1/n)^n \le e$ $\square$

Bounding the Inverse of Euler's Constant

In a similar manner, by looking at the interval $\left[1 - \frac{1}{n}, 1\right]$, one can prove the following.

Lemma 34

Let $n \in \mathbb{N}$, such that $n \geq 2$. Then,

$$\left(1 - \frac{1}{n}\right)^{n} \leq \frac{1}{e} \leq \left(1 - \frac{1}{n}\right)^{n-1}$$