10-301/10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

PAC Learning

Matt Gormley
Lecture 15
Mar. 13, 2023

Q&A

Q: Ummm Matt, what happened to you? You seem… shorter

A: I’m Henry; don’t worry Matt will be right back

Q: Okay, so why are you here?

A: To recruit summer 10-301/601 TAs!

Apply at: https://forms.gle/S9ksw7G9dp5LB1Hj9

Deadline: Monday, April 10th

Note: you must be in Pittsburgh over the summer to be considered!

Questions? Email me at hchai2@andrew.cmu.edu

Q&A

Q: What is “bias”?

A: That depends. The word “bias” shows up all over machine learning! Watch out…

1. The additive term in a linear model (i.e. b in $w^T x + b$)
2. Inductive bias is the principle by which a learning algorithm generalizes to unseen examples
3. Bias of a model in a societal sense may refer to racial, socio-economic, gender biases that exist in the predictions of your model
4. The difference between the expected predictions of your model and the ground truth (as in “bias-variance tradeoff”)

Reminders

- Homework 5: Neural Networks
  - Out: Sun, Feb 26
  - Due: Fri, Mar 17 at 11:59pm
- Peer Tutoring

LEARNING THEORY

PAC(-MAN) Learning

For some hypothesis $h \in \mathcal{H}$:

1. True Error

$$R(h)$$

2. Training Error

$$\hat{R}(h)$$

Question 2:
What is the expected number of PAC-MAN levels Matt will complete before a **Game-Over**?

A. 1-10

B. 11-20

C. 21-30

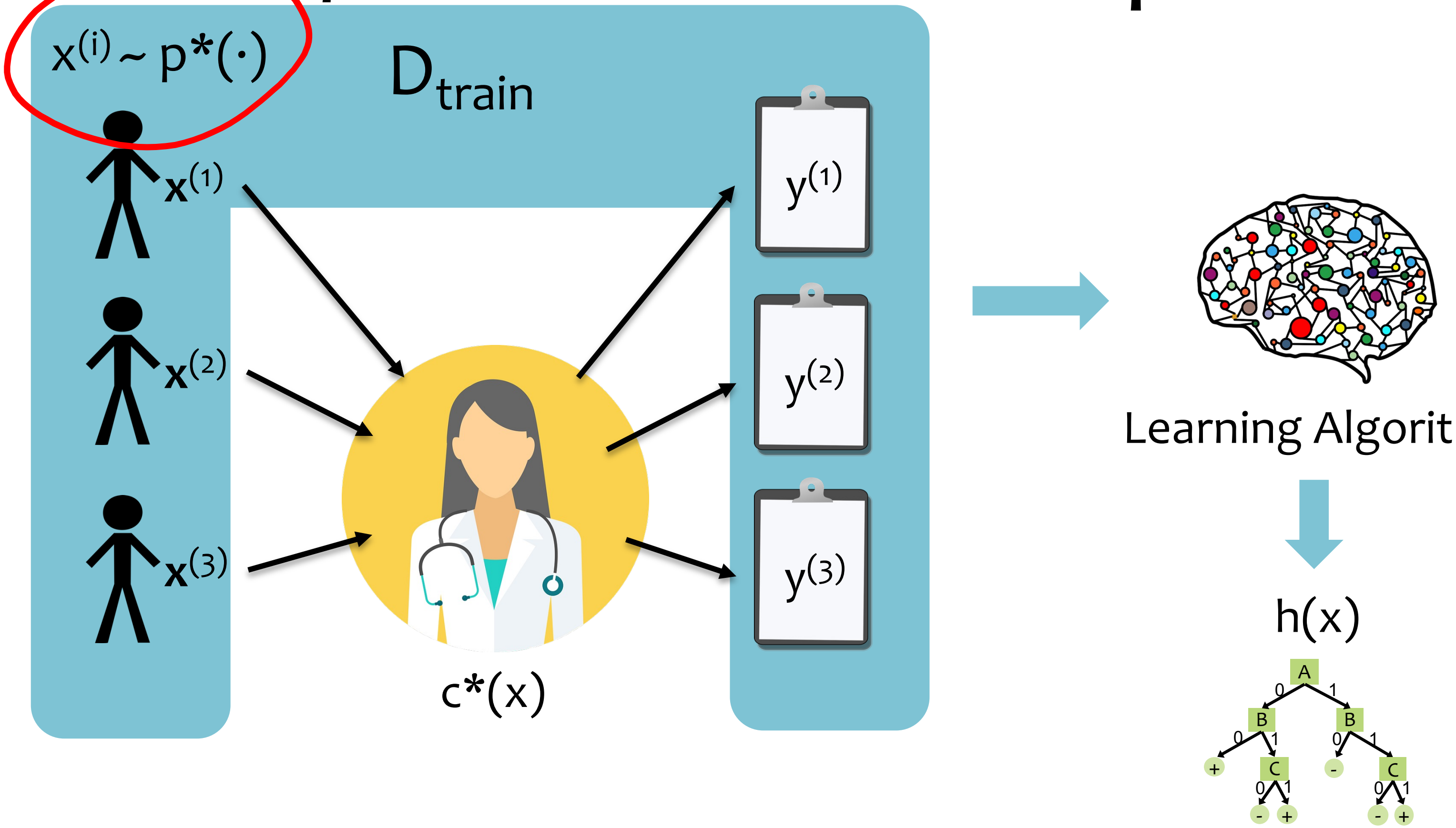
Questions for today (and next lecture)

1. Given a classifier with **zero training error**, what can we say about **true error** (aka. generalization error)?
   (Sample Complexity, Realizable Case)

2. Given a classifier with **low training error**, what can we say about **true error** (aka. generalization error)?
   (Sample Complexity, Agnostic Case)

3. Is there a **theoretical justification for regularization** to avoid overfitting?
   (Structural Risk Minimization)

PAC/SLT Model for Supervised ML



$x^{(i)} \sim p^*(\cdot)$

D_{train}

$\mathbf{x}^{(1)}$

$\mathbf{x}^{(2)}$

$\mathbf{x}^{(3)}$

$c^*(x)$

$y^{(1)}$

$y^{(2)}$

$y^{(3)}$

Learning Algorithm

$h(x)$

PAC/SLT Model for Supervised ML

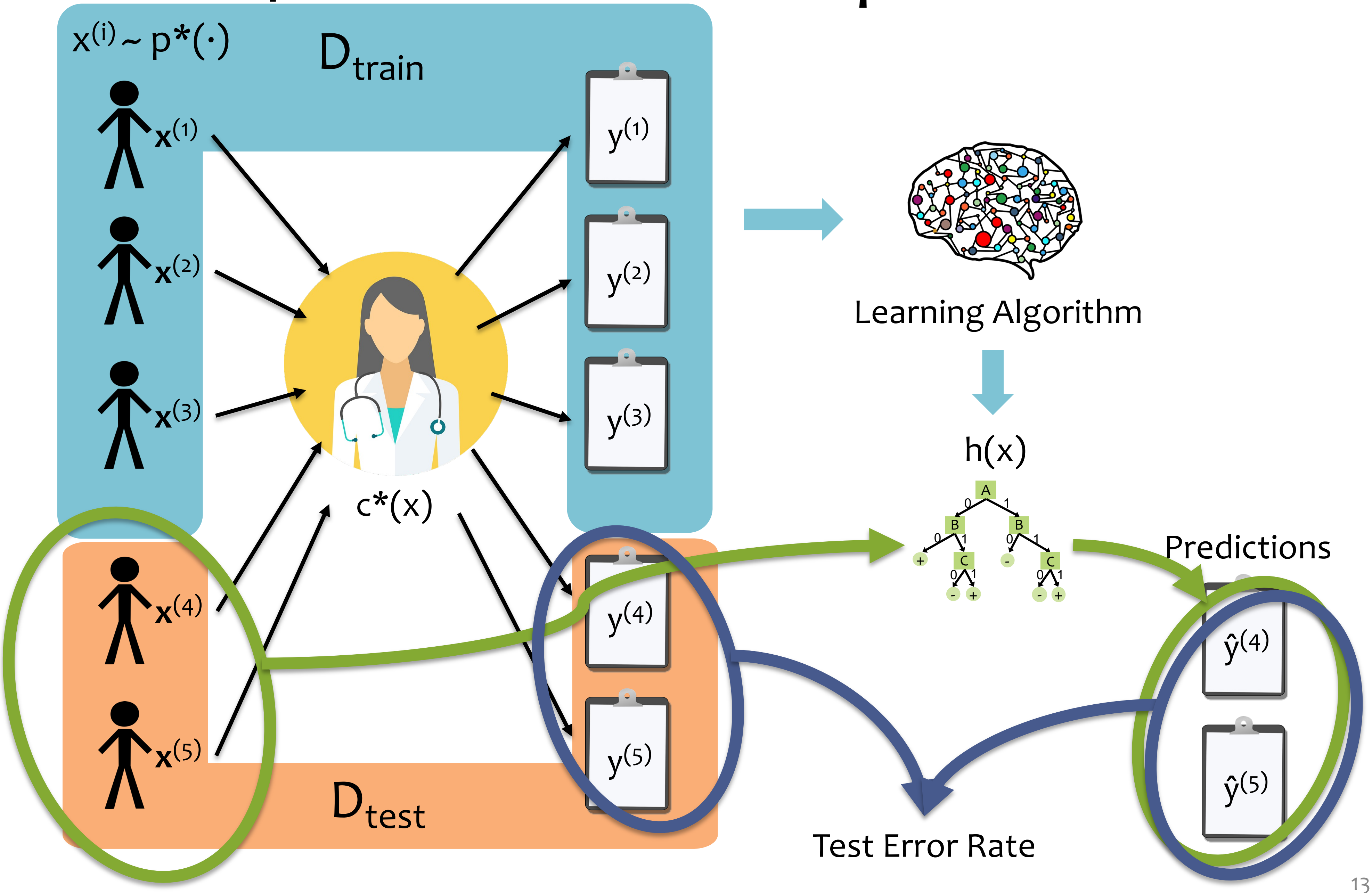
- **Problem Setting**
  - Set of possible inputs, $\mathbf{x} \in \mathcal{X}$ (all possible patients)
  - Set of possible outputs, $y \in \mathcal{Y}$ (all possible diagnoses)
  - Distribution over instances, $p^*(\cdot)$
  - Exists an unknown target function, $c^* : \mathcal{X} \rightarrow \mathcal{Y}$ (the doctor's brain)
  - Set, $\mathcal{H}$, of candidate hypothesis functions, $h : \mathcal{X} \rightarrow \mathcal{Y}$ (all possible decision trees)
- **Learner is given** N training examples $D = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})\}$ where $x^{(i)} \sim p^*(\cdot)$ and $\underline{y^{(i)}} = \boxed{c^*}(\mathbf{x}^{(i)})$ (history of patients and their diagnoses)
- **Learner produces** a hypothesis function, $\underline{\hat{y}} = \boxed{h}(x)$, that best approximates unknown target function $y = c^*(x)$ on the training data

IMPORTANT NOTE

In our discussion of PAC Learning, we are only concerned with the problem of **binary** classification

There are other theoretical frameworks (including PAC) that handle other learning settings, but this provides us with a representative one.

PAC/SLT Model for Supervised ML



Two Types of Error

1. True Error (aka. **expected risk**)

$$R(h) = P_{\underline{\mathbf{x} \sim p^*(\mathbf{x})}}(c^*(\mathbf{x}) \neq h(\mathbf{x}))$$

This quantity is always **unknown**

2. Train Error (aka. **empirical risk**)

$$\hat{R}(h) = P_{\mathbf{x} \sim \mathcal{S}}(c^*(\mathbf{x}) \neq h(\mathbf{x}))$$

$$= \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(c^*(\mathbf{x}^{(i)}) \neq h(\mathbf{x}^{(i)}))$$

$$= \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(y^{(i)} \neq h(\mathbf{x}^{(i)}))$$

We can **measure** this on the training data

where $\mathcal{S} = \{\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)})\}_{i=1}^{N}$ is the training data set, and $\mathbf{x} \sim \mathcal{S}$ denotes that $\mathbf{x}$ is sampled from the empirical distribution.

PAC / SLT Model

We've also referred to this as the "Function Approximation View"

1. Generate instances from *unknown* distribution p^*

$$\mathbf{x}^{(i)} \sim \underline{p^*}(\mathbf{x}),\ \forall i \tag{1}$$

2. Oracle labels each instance with *unknown* function c^*

$$y^{(i)} = \underline{c^*}(\mathbf{x}^{(i)}),\ \forall i \tag{2}$$

3. Learning algorithm chooses hypothesis $h \in \mathcal{H}$ with low(est) training error, $\hat{R}(h)$

$$\boxed{\hat{h} = \underset{h}{\text{argmin}}\ \underline{\hat{R}(h)}} \tag{3}$$

4. Goal: Choose an h with low generalization error $R(h)$

Three Hypotheses of Interest

The **true function** c^* is the one we are trying to learn and that labeled the training data:

$$y^{(i)} = c^*(\mathbf{x}^{(i)}),\ \forall i \tag{1}$$

The **expected risk minimizer** has lowest true error:

R.W

$$h^* = \operatorname*{argmin}_{h \in \mathcal{H}} R(h) \tag{2}$$

The **empirical risk minimizer** has lowest training error:

$$\hat{h} = \operatorname*{argmin}_{h \in \mathcal{H}} \hat{R}(h) \tag{3}$$

Three Hypotheses of Interest

$$y^{(i)} = c^*(\mathbf{x}^{(i)}), \forall i$$

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \, R(h)$$

85%

Q1 : A=True B=toxic C=False

Question: *True or False*: h* and c* are always equal.

Answer:

Three Hypotheses of Interest

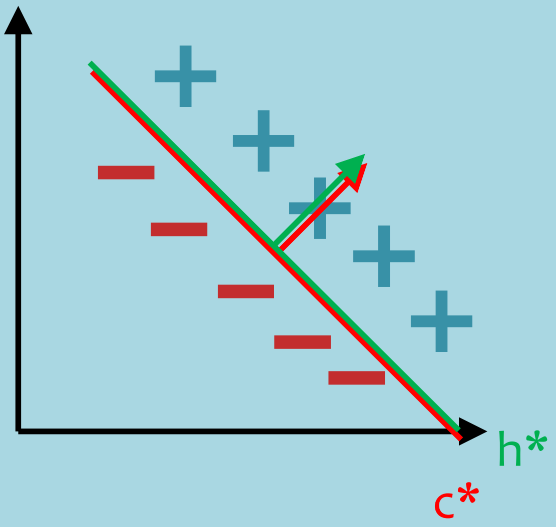
$$y^{(i)} = c^*(\mathbf{x}^{(i)}), \ \forall i$$

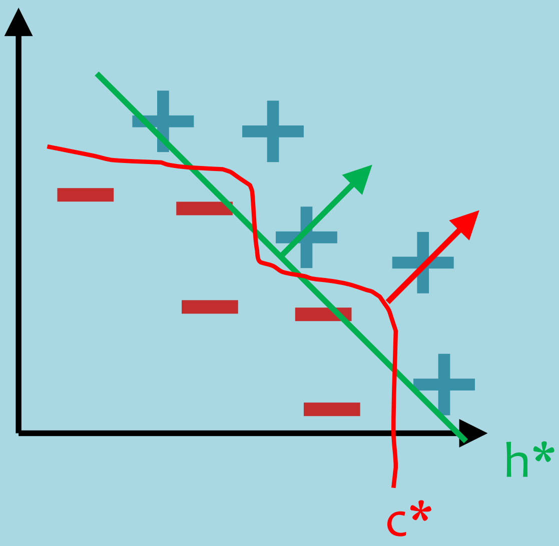
$$h^* = \underset{h \in \mathcal{H}}{\text{argmin}}\, R(h)$$

Question: *True or False:* h* and c* are always equal.

Answer:

$\mathcal{H}$ = linear separators





PAC LEARNING

PAC Learning

- Q: Can we bound R(h) in terms of R̂(h)?
- A: Yes!

- **PAC** stands for **P**robably

  Approximately

  Correct

A **PAC Learner** yields a hypothesis $h \in \mathcal{H}$ which is…

approximately correct $R(h) \approx 0$

with high probability $\Pr(R(h) \approx 0) \approx 1$

Probably Approximately Correct (PAC) Learning

Whiteboard:

- PAC Criterion
- Sample Complexity
- Consistent Learner

SAMPLE COMPLEXITY RESULTS

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

We'll start with the finite case…

	$c^* \in \mathcal{H}$ Realizable	$c^* \notin \mathcal{H}$ or $c^* \in \mathcal{H}$ Agnostic
Finite $\lvert\mathcal{H}\rvert$ $\lvert\mathcal{H}\rvert < +\infty$		
Infinite $\lvert\mathcal{H}\rvert$ $\lvert\mathcal{H}\rvert = +\infty$		

Probably Approximately Correct (PAC) Learning

Whiteboard:

- Theorem 1: Realizable Case, Finite |H|

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

	Realizable	Agnostic
Finite $\|\mathcal{H}\|$	**Thm. 1** $N \geq \frac{1}{\epsilon}\left[\log(\|\mathcal{H}\|) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.	
Infinite $\|\mathcal{H}\|$		

Example: Conjunctions

Question: Q2

Suppose H = class of conjunctions over **x** in $\{0,1\}^M$

$\vec{x} = [0, 1, 1, 1, 0]$

Example hypotheses:

→ $h(\mathbf{x}) = \underline{x_1}\ (\underline{1-x_3})\ x_5$ $= x_1 \wedge \neg x_3 \wedge x_5$

$h(\mathbf{x}) = x_1\ (1-x_2)\ x_4\ (1-x_5)$

$= x_1 \wedge \neg x_2 \wedge x_4 \wedge \neg x_5$

If M = 10, ε = 0.1, δ = 0.01, how many examples suffice according to Theorem 1?

Answer:

A. 10*(2*ln(10)+ln(100)) ≈ 92

B. 10*(3*ln(10)+ln(100)) ≈ 116

C. 10*(10*ln(2)+ln(100)) ≈ 116 44%

[✓ D.] 10*(10*ln(3)+ln(100)) ≈ 156 33%

E. 100*(2*ln(10)+ln(10)) ≈ 691

F. 100*(3*ln(10)+ln(10)) ≈ 922

G. 100*(10*ln(2)+ln(10)) ≈ 924

H. 100*(10*ln(3)+ln(10)) ≈ 1329

I = toxic

$|\mathcal{H}| = 3^{10}$

$= 3^M$

Thm. 1 $N \geq \frac{1}{\epsilon}\left[\log(|\mathcal{H}|) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

	Realizable	Agnostic
Finite $\|\mathcal{H}\|$	**Thm. 1** $N \geq \frac{1}{\epsilon}\left[\log(\|\mathcal{H}\|) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.	
Infinite $\|\mathcal{H}\|$		

Background: Contrapositive

- *Definition:* The **contrapositive** of the statement
  $$A \Rightarrow B$$
  is the statement
  $$\neg B \Rightarrow \neg A$$
  and the two are logically equivalent (i.e. they share all the same truth values in a truth table!)

- *Proof by contrapositive:*
  If you want to prove $A \Rightarrow B$, instead prove $\neg B \Rightarrow \neg A$ and then conclude that $A \Rightarrow B$

- *Caution:* sometimes negating a statement is easier said than done, just be careful!

Probably Approximately Correct (PAC) Learning

Whiteboard:

- Proof of Theorem 1

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

$c^* \in \mathcal{H}$

$c^* \notin \mathcal{H}$ or $c^* \in \mathcal{H}$

	Realizable	Agnostic
Finite $\lvert\mathcal{H}\rvert$	**Thm. 1** $N \geq \frac{1}{\epsilon}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.	**Thm. 2** $N \geq \frac{1}{2\epsilon^2}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{2}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\lvert R(h) - \hat{R}(h)\rvert \leq \epsilon$.
Infinite $\lvert\mathcal{H}\rvert$		

1. Bound is **inversely linear in epsilon** (e.g. halving the error requires double the examples)
2. Bound is **only logarithmic in |H|** (e.g. quadrupling the hypothesis space only requires double the examples)

1. Bound is **inversely quadratic in epsilon** (e.g. halving the error requires 4x the examples)
2. Bound is **only logarithmic in |H|** (i.e. same as Realizable case)

	Realizable	Agnostic
Finite $\|\mathcal{H}\|$	**Thm. 1** $N \geq \frac{1}{\epsilon}\left[\log(\|\mathcal{H}\|) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.	**Thm. 2** $N \geq \frac{1}{2\epsilon^2}\left[\log(\|\mathcal{H}\|) + \log(\frac{2}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\|R(h) - \hat{R}(h)\| \leq \epsilon$.
Infinite $\|\mathcal{H}\|$		

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

	Realizable	Agnostic
Finite $\lvert\mathcal{H}\rvert$	**Thm** belec prob have $R(h) \leq \epsilon$.	] t e have that $\lvert R(h) - \hat{R}(h)\rvert \leq \epsilon$.
Infinite $\lvert\mathcal{H}\rvert$		

For these two cases, we will use a new definition for the "complexity" of a Hypothesis space called **VC Dimension**

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

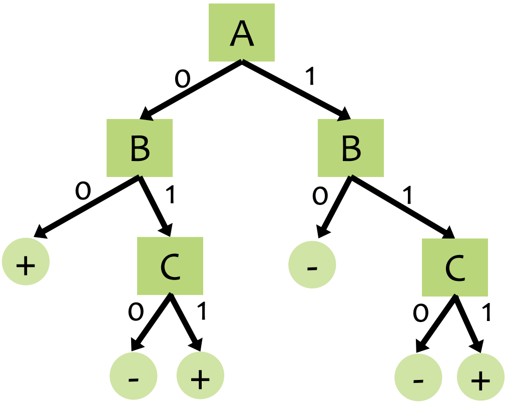
	Realizable	Agnostic
Finite $\lvert\mathcal{H}\rvert$	**Thm. 1** $N \geq \frac{1}{\epsilon}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.	**Thm. 2** $N \geq \frac{1}{2\epsilon^2}\left[\log(\lvert\mathcal{H}\rvert) + \log(\frac{2}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\lvert R(h) - \hat{R}(h)\rvert \leq \epsilon$.
Infinite $\lvert\mathcal{H}\rvert$	**Thm. 3** $N = O(\frac{1}{\epsilon}\left[\text{VC}(\mathcal{H}) \log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta})\right])$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $\hat{R}(h) = 0$ have $R(h) \leq \epsilon$.	**Thm. 4** $N = O(\frac{1}{\epsilon^2}\left[\text{VC}(\mathcal{H}) + \log(\frac{1}{\delta})\right])$ labeled examples are sufficient so that with probability $(1-\delta)$ for all $h \in \mathcal{H}$ we have that $\lvert R(h) - \hat{R}(h)\rvert \leq \epsilon$.

VC-DIMENSION

Finite vs. Infinite |H|

Finite |H|

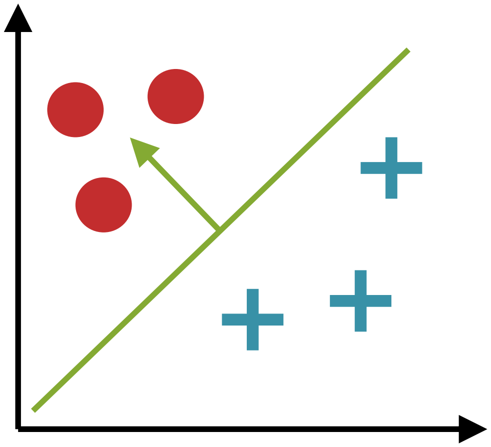
- *Example*: H = the set of all decision trees of depth D over binary feature vectors of length M



- *Example*: H = the set of all conjunctions over binary feature vectors of length M

Infinite |H|

- *Example*: H = the set of all linear decision boundaries in M dimensions



- *Example*: H = the set of all neural networks with 1-hidden layer with length M inputs

Labelings & Shattering

Def: A hypothesis h applied to some dataset S generates a **labeling** of S.

$S = \{x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}\}$

$L = \{c^*(x^{(1)}), \ldots, c^*(x^{(4)})\}$
$= \{+, -, +, +\}$

Def: Let $\mathcal{H}[S]$ be the set of all (distinct) labelings of S generated by hypotheses $h \in \mathcal{H}$.

$\mathcal{H}$ **shatters** S if $|\mathcal{H}[S]| = 2^{|S|}$

Equivalently, the hypotheses in $\mathcal{H}$ can generate every possible labeling of S.