



# 10-301/10-601 Introduction to Machine Learning

Machine Learning Department  
School of Computer Science  
Carnegie Mellon University

# PAC Learning

Slides from Matt Gormley

Lecture 14

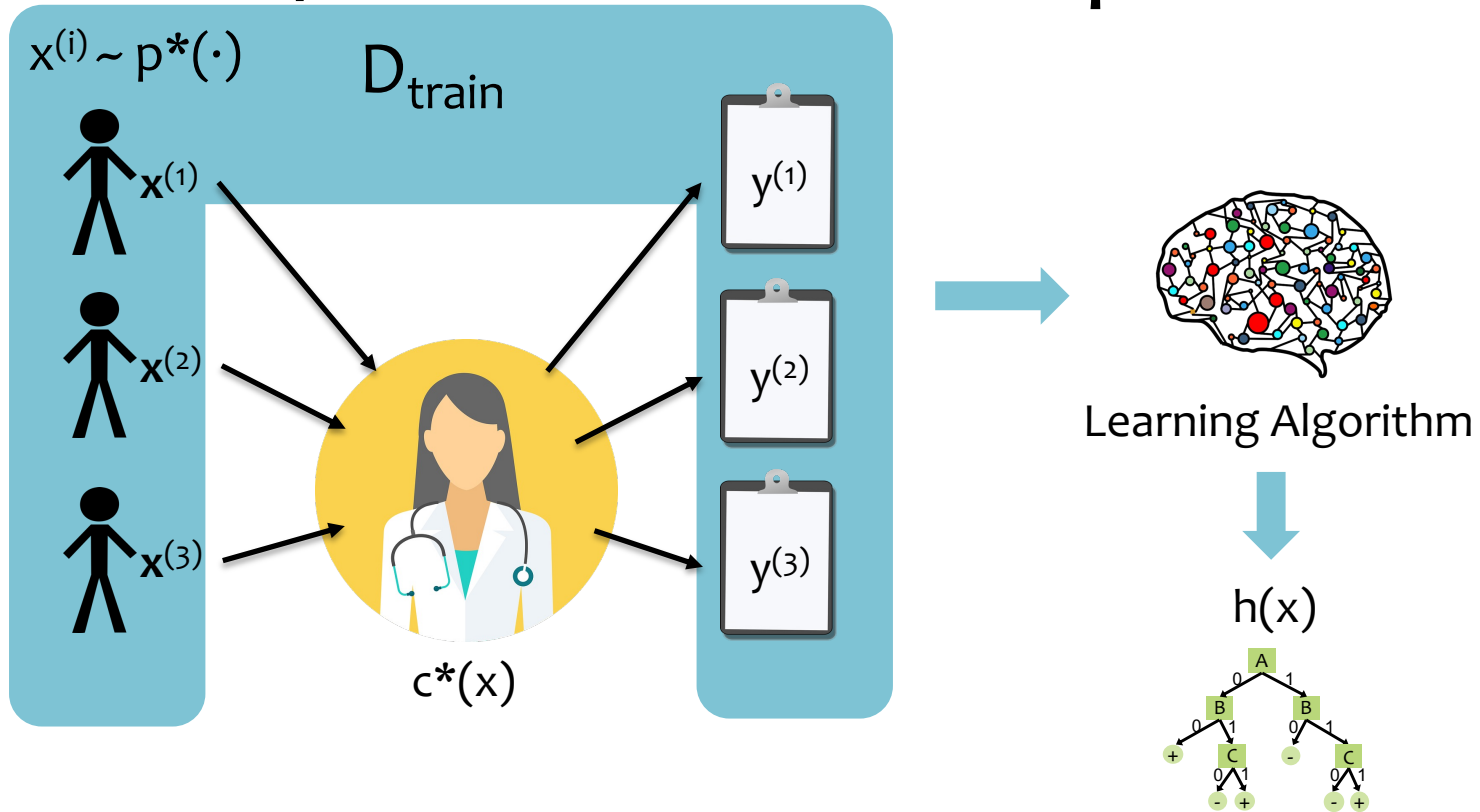
Mar. 11, 2024

# **LEARNING THEORY**

# 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



# 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  
 $D = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$   
where  $\mathbf{x}^{(i)} \sim p^*(\cdot)$  and  $y^{(i)} = c^*(\mathbf{x}^{(i)})$   
(history of patients and their diagnoses)

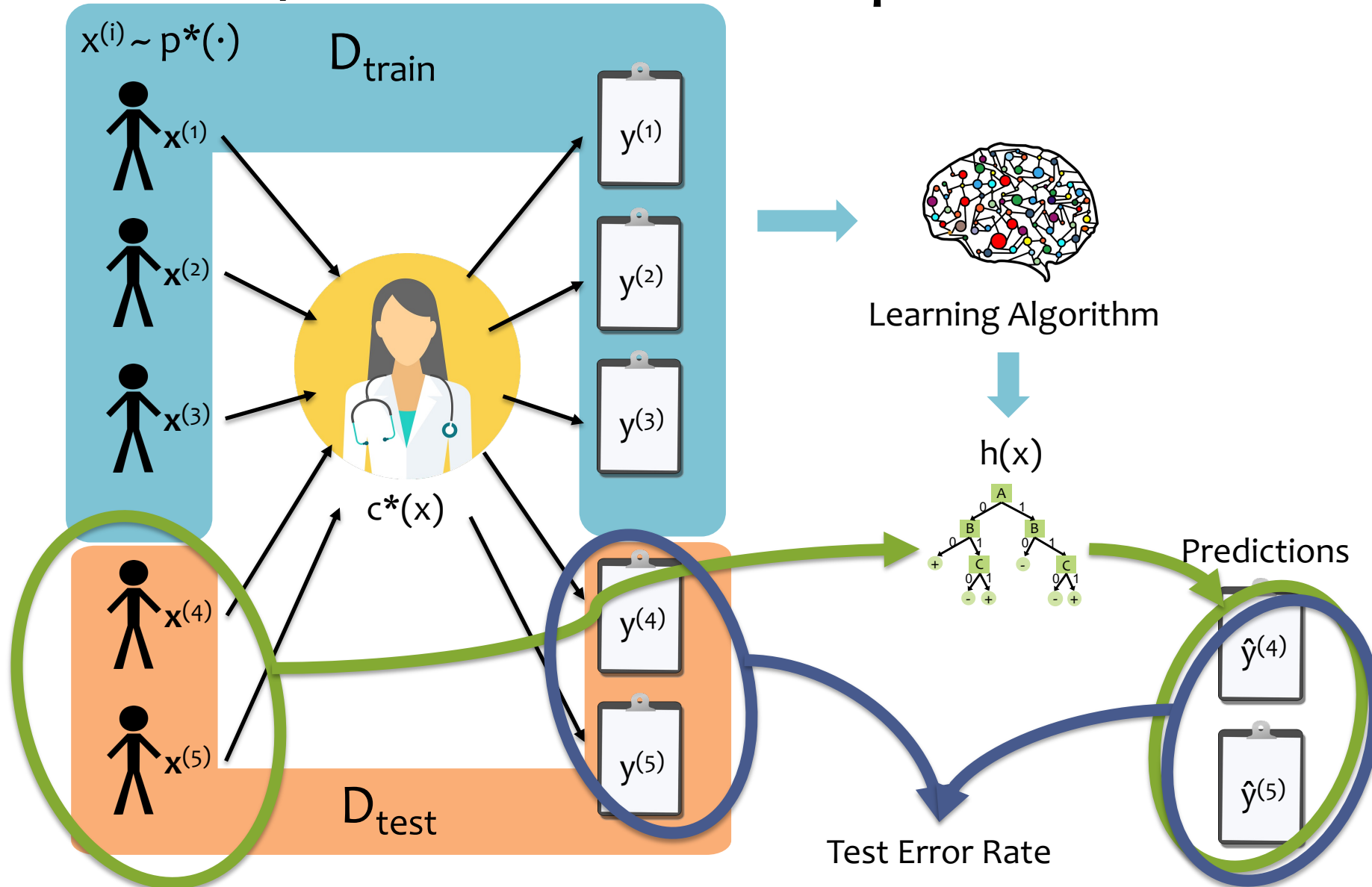
- **Learner produces** a hypothesis function,  $\hat{y} = h(\mathbf{x})$ , that best approximates unknown target function  $y = c^*(\mathbf{x})$  on the training data

# 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**)

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

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  $p^*$

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

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

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

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

$$\hat{h} = \underset{h}{\operatorname{argmin}} \hat{R}(h) \quad (3)$$

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:

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

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

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

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

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

# Three Hypotheses of Interest

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

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

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

**Answer:**

# **PAC LEARNING**

# PAC Learning

- Q: Can we bound  $R(h)$  in terms of  $\hat{R}(h)$ ?

- A: Yes!

- **PAC** stands for

Probably

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

**PAC Criterion**

**Sample Complexity**

**Consistent Learner**

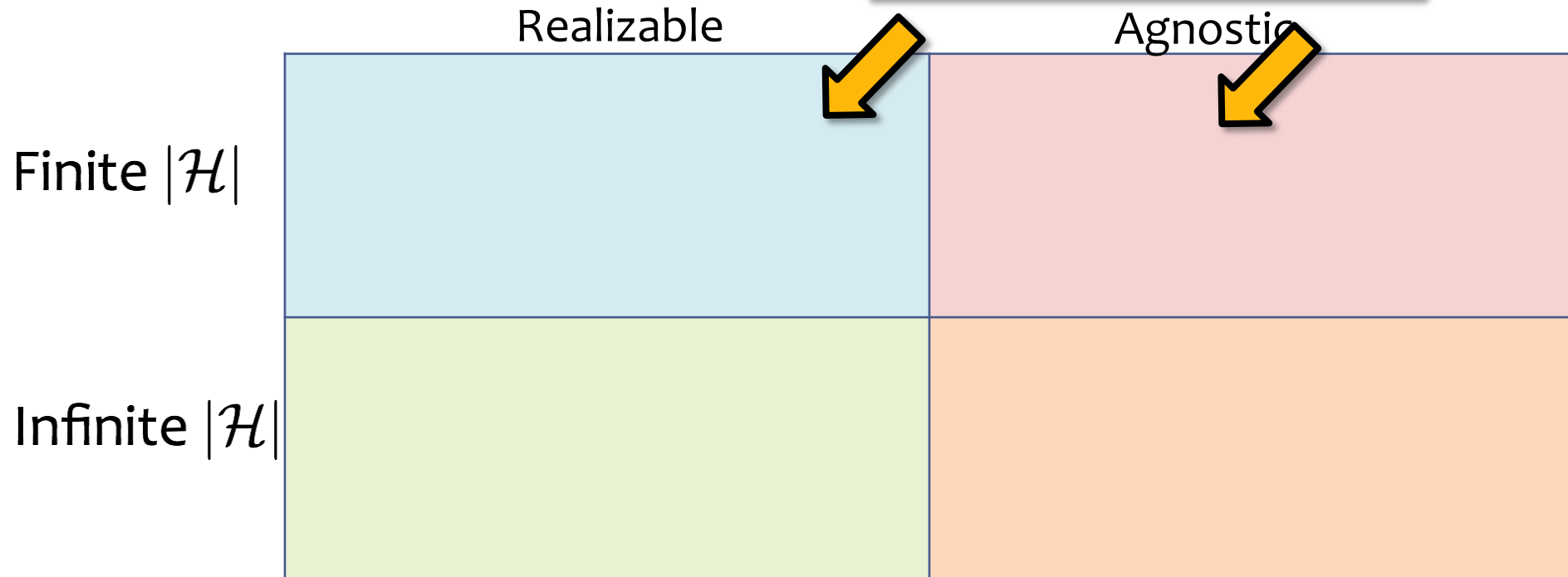
# **SAMPLE COMPLEXITY RESULTS**

# 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...





# Probably Approximately Correct (PAC) Learning

**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} $	<b>Thm. 1</b> $N \geq \frac{1}{\epsilon} [\log( \mathcal{H} ) + \log(\frac{1}{\delta})]$ 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:

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

Example hypotheses:

$$h(\mathbf{x}) = x_1 (1-x_3) x_5$$

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

If  $M = 10$ ,  $\epsilon = 0.1$ ,  $\delta = 0.01$ , how many examples suffice according to Theorem 1?

## Answer:

- A.  $10^{*(2*\ln(10)+\ln(100))} \approx 92$
- B.  $10^{*(3*\ln(10)+\ln(100))} \approx 116$
- C.  $10^{*(10*\ln(2)+\ln(100))} \approx 116$
- D.  $10^{*(10*\ln(3)+\ln(100))} \approx 156$
- E.  $100^{*(2*\ln(10)+\ln(10))} \approx 691$
- F.  $100^{*(3*\ln(10)+\ln(10))} \approx 922$
- G.  $100^{*(10*\ln(2)+\ln(10))} \approx 924$
- H.  $100^{*(10*\ln(3)+\ln(10))} \approx 1329$

**Thm. 1**  $N \geq \frac{1}{\epsilon} [\log(|\mathcal{H}|) + \log(\frac{1}{\delta})]$  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} $	<b>Thm. 1</b> $N \geq \frac{1}{\epsilon} [\log( \mathcal{H} ) + \log(\frac{1}{\delta})]$ 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!

# Proof of Theorem 1

# Proof of Theorem 1

# 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...**

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

1. Bound is **inversely linear in epsilon** (e.g. halving the error requires double the examples)
2. Bound is **only logarithmic in  $|\mathcal{H}|$**  (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  $|\mathcal{H}|$**  (i.e. same as Realizable case)



Realizable



Agnostic

Finite  $|\mathcal{H}|$

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

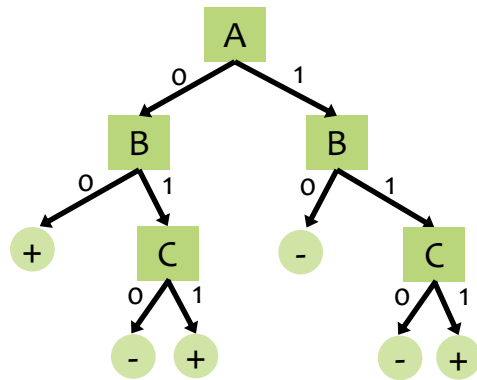
**Thm. 2**  $N \geq \frac{1}{2\epsilon^2} [\log(|\mathcal{H}|) + \log(\frac{2}{\delta})]$  labeled examples are sufficient so that with probability  $(1 - \delta)$  for all  $h \in \mathcal{H}$  we have that  $|R(h) - \hat{R}(h)| \leq \epsilon$ .

Infinite  $|\mathcal{H}|$

# Finite vs. Infinite $|H|$

## Finite $|H|$

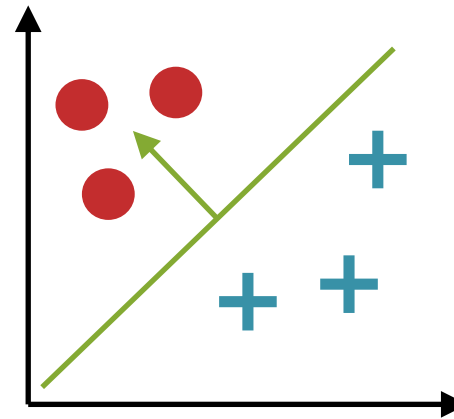
- *Example:*  $H$  = the set of all decision trees of depth  $D$  over binary feature vectors of length  $M$



- *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

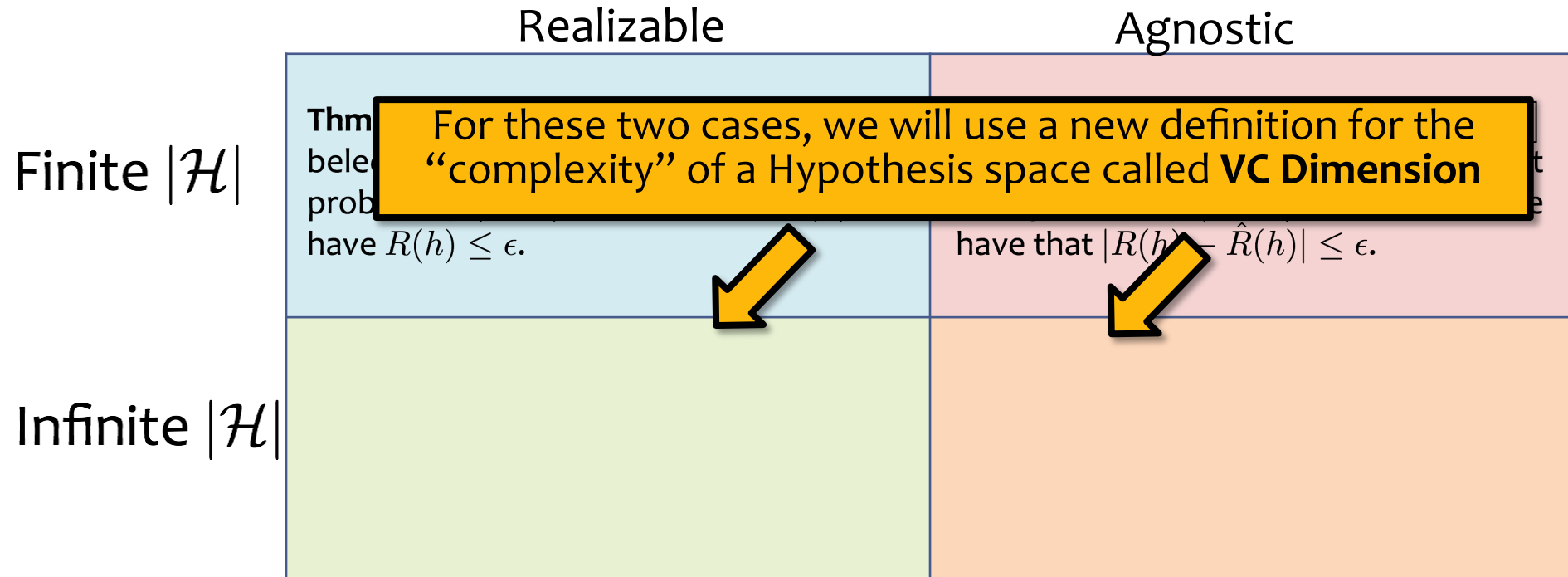


- *Example:*  $H$  = the set of all neural networks with 1-hidden layer with length  $M$  inputs

# 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...**



# 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} $	<p><b>Thm. 1</b> <math>N \geq \frac{1}{\epsilon} [\log( \mathcal{H} ) + \log(\frac{1}{\delta})]</math> labeled examples are sufficient so that with probability <math>(1 - \delta)</math> all <math>h \in \mathcal{H}</math> with <math>\hat{R}(h) = 0</math> have <math>R(h) \leq \epsilon</math>.</p>	<p><b>Thm. 2</b> <math>N \geq \frac{1}{2\epsilon^2} [\log( \mathcal{H} ) + \log(\frac{2}{\delta})]</math> labeled examples are sufficient so that with probability <math>(1 - \delta)</math> for all <math>h \in \mathcal{H}</math> we have that <math> R(h) - \hat{R}(h)  \leq \epsilon</math>.</p>
Infinite $ \mathcal{H} $	<p><b>Thm. 3</b> <math>N = O(\frac{1}{\epsilon} [\text{VC}(\mathcal{H}) \log(\frac{1}{\epsilon}) + \log(\frac{1}{\delta})])</math> labeled examples are sufficient so that with probability <math>(1 - \delta)</math> all <math>h \in \mathcal{H}</math> with <math>\hat{R}(h) = 0</math> have <math>R(h) \leq \epsilon</math>.</p>	<p><b>Thm. 4</b> <math>N = O(\frac{1}{\epsilon^2} [\text{VC}(\mathcal{H}) + \log(\frac{1}{\delta})])</math> labeled examples are sufficient so that with probability <math>(1 - \delta)</math> for all <math>h \in \mathcal{H}</math> we have that <math> R(h) - \hat{R}(h)  \leq \epsilon</math>.</p>