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

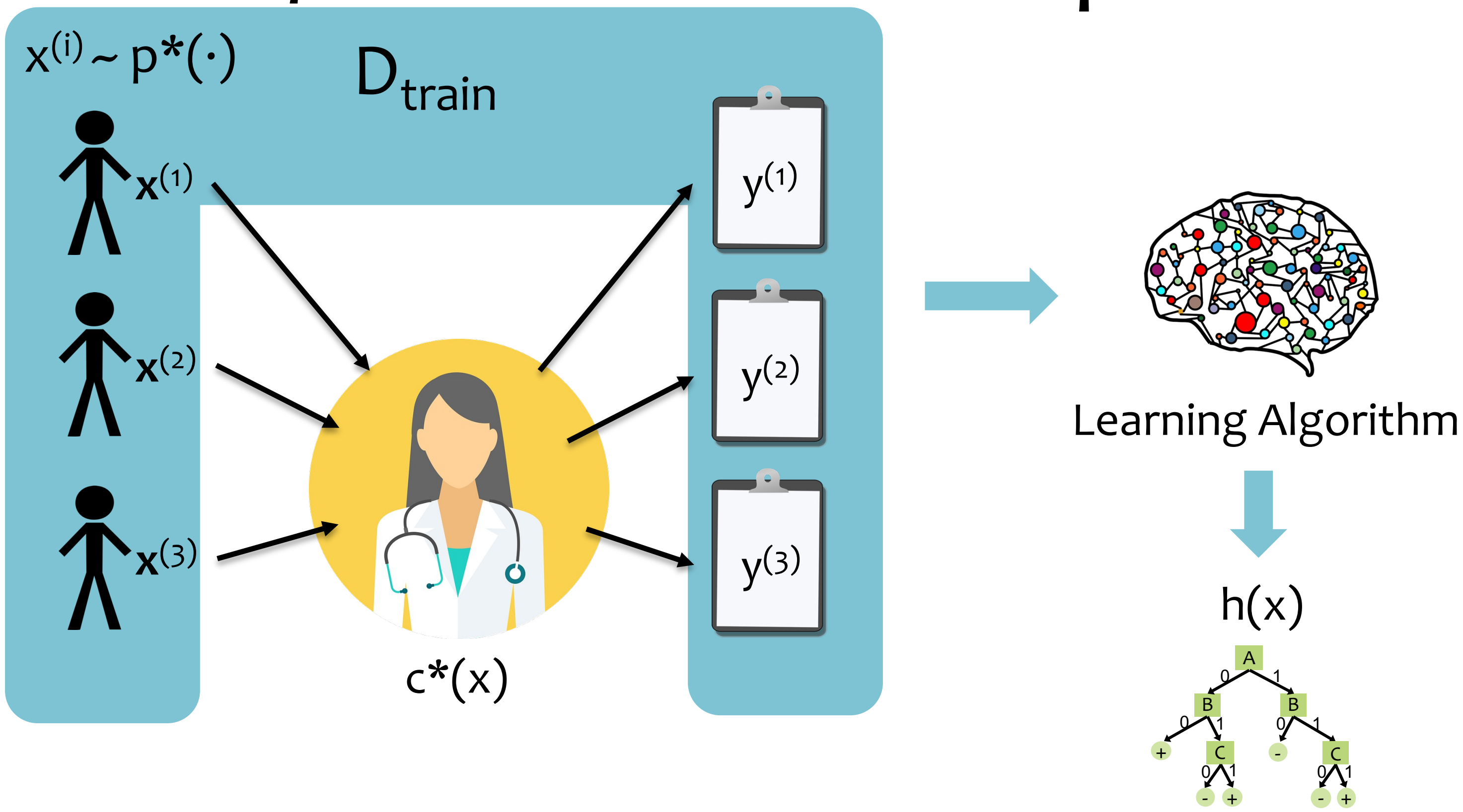
Questions for today (and next lecture)

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

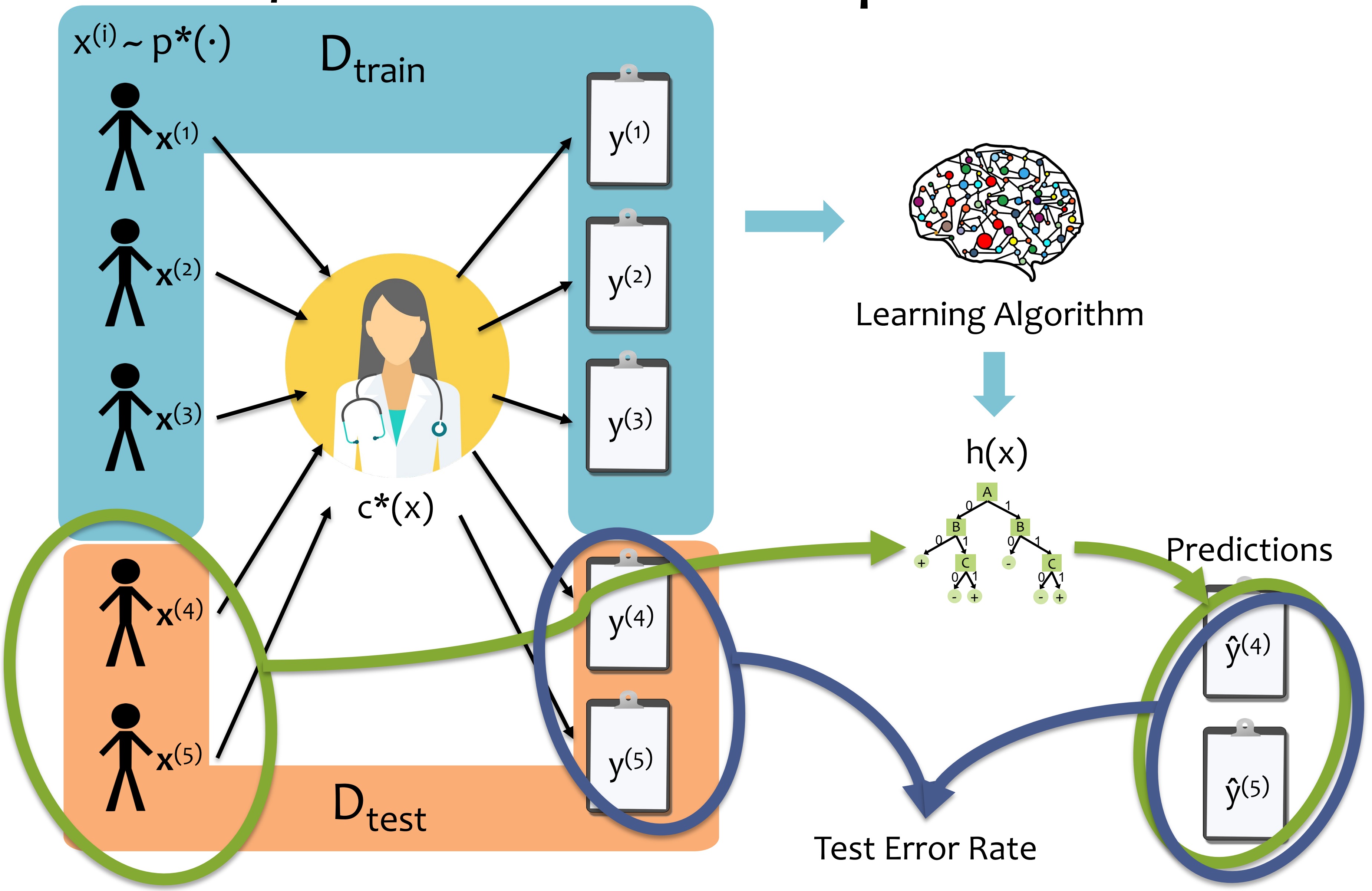
- **Problem Setting**
  - Set of possible inputs, $\mathbf{x} \in \mathcal{X}$ (all possible patients)
  - Set of possible outputs, $y \in \mathcal{Y}$ (all possible diagnoses)
  - Distribution over instances, $p^*(\cdot)$
  - Exists an unknown target function, $c^* : \mathcal{X} \rightarrow \mathcal{Y}$
    (the doctor's brain) labeling
  - 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 $x^{(i)} \sim p^*(\cdot)$ and $y^{(i)} = c^*(\mathbf{x}^{(i)})$
  (history of patients and their diagnoses)
- **Learner produces** a hypothesis function, $\hat{y} = h(x)$, 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**)

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

This quantity is always **unknown**

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

$$\begin{aligned}\hat{R}(h) &= P_{\mathbf{x} \sim \mathcal{S}}(c^*(\mathbf{x}) \neq h(\mathbf{x})) \\ &= \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(\underbrace{c^*(\mathbf{x}^{(i)})}_{y^{(i)}} \neq h(\mathbf{x}^{(i)})) \\ &= \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}(y^{(i)} \neq h(\mathbf{x}^{(i)}))\end{aligned}$$

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

where $\mathcal{S} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)})\}_{i=1}^{N}$ 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^*

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

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

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

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

$$\hat{h} = \underbrace{\underset{h \in \mathcal{H}}{\text{argmin}}\, \hat{R}(h)} \tag{3}$$

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:

best-in-class

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

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

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

A = toxic
B = true
C = false

Three Hypotheses of Interest

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

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

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

Answer:

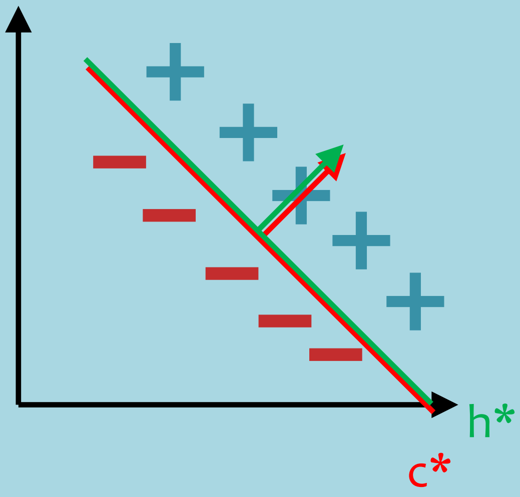
Three Hypotheses of Interest

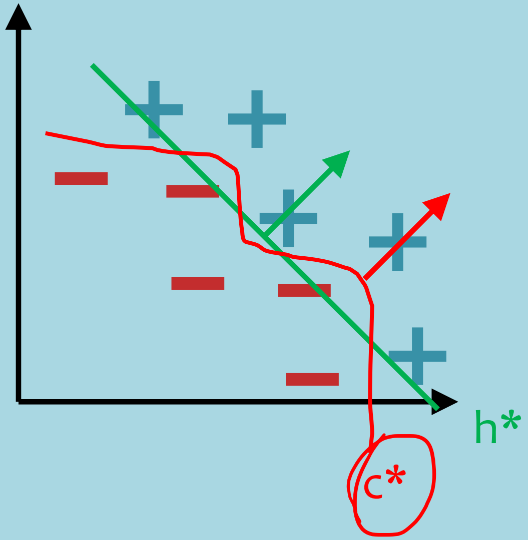
$$y^{(i)} = c^*(\mathbf{x}^{(i)}), \; \forall i$$

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

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

Answer:





PAC LEARNING

PAC Learning

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

- **PAC** stands for

Probably → confidence

Approximately → estimation

Correct

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

approximately correct $\quad R(h) \approx \cancel{0}$ R(h*) ←

with high probability $\quad \Pr(R(h) \approx \cancel{0}) \approx 1$

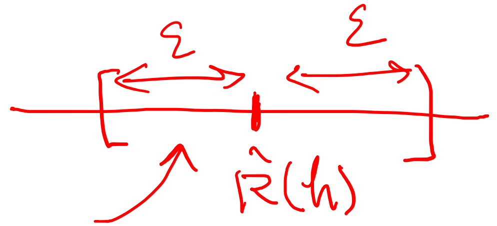
R(h*)

Probably Approximately Correct (PAC) Learning

ε, δ small numbers ≈ 0

PAC Criterion

$\forall h \in \mathcal{H} \quad Pr\left(|R(h) - \hat{R}(h)| < \varepsilon\right) \geq 1-\delta$



$R(h)$

$\hat{R}(h)$ is defined w.r.t. D

and D is random sample from P^*

Sample Complexity

is the min number of training examples, $N(\varepsilon, \delta)$ s.t. the PAC criterion is satisfied for ε, δ.

Consistent Learner

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

A hypothesis $h \in \mathcal{H}$ is consistent with training data D if $\hat{R}_D(h) = 0$

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).

best-in-class h^*

Four Cases we care about…

We'll start with the finite case…

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

c^* is not necessarily in $\mathcal{H}$

	Realizable	Agnostic
Finite $\|\mathcal{H}\|$		
Infinite $\|\mathcal{H}\|$		

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 $\|\mathcal{H}\|$	**Thm. 1** $\boxed{N \geq \frac{1}{\epsilon}\left[\log(\|\mathcal{H}\|) + \log(\frac{1}{\delta})\right]}$ 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: $x_1, \ldots, x_M \in \{0,1\}$

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

Example hypotheses:

$x_1 \wedge \sim x_3 \wedge x_5$ ← $h(\mathbf{x}) = x_1 (1-x_3) x_5$ $= 1$ iff $x_1 = 1$, $x_3 = 0$, $x_5 = 1$

$x_1 \wedge \sim x_2 \wedge x_4 \wedge \sim x_5$ $h_2(\mathbf{x}) = x_1 (1-x_2) x_4 (1-x_5)$ $x_1 = 1$, $x_2 = 0$, $x_4 = 1$, $x_5 = 0$

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

Answer:

A. $10*(2*\ln(10)+\ln(100)) \approx 92$
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. $100*(10*\ln(3)+\ln(10)) \approx 1329$ → Toxic

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 $R(h) \leq \epsilon$.

$x_1, x_2, \ldots, x_{10}$

x_1

$\sim x_1$

$-$

$|\mathcal{H}| = 3^{10}$ $\ln(3^{10}) = 10 \ln 3$

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…

A

if $\boxed{N \geq \dots}$

then PAC-criteria

$f(\epsilon, \delta, |\mathcal{H}|)$

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

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

- Assume we have $k > 0$ bad hypotheses in $\mathcal{H}$ where a bad model h_i is consistent $(\hat{R}(h_i) = 0)$ but $R(h_i) > \varepsilon$.

- Pick bad hypothesis h_i. The prob that h_i is consistent with $(x^{(1)}, y^{(1)})$ is $\leq (1-\varepsilon)$

— " — — " — $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)})\} \leq (1-\varepsilon)^2$

⋮

— " — D $\leq (1-\varepsilon)^N$

- Prob that at least one bad h is consistent with D $\leq k(1-\varepsilon)^N \leq |\mathcal{H}|(1-\varepsilon)^N$

Union bound: $P(A \cup B) \leq P(A) + P(B)$

$\hookrightarrow = P(A) + P(B) - \cancel{P(A \cap B)}$

Proof of Theorem 1

* Prob of a bad hypothesis looking good empirically $\leq |\mathcal{H}|(1-\varepsilon)^N$

$$\leq \underbrace{|\mathcal{H}| \exp(-\varepsilon N)}$$

Known fact: $\forall x \,;\, (1-x) \leq \exp(-x)$

$$|\mathcal{H}| \exp(-\varepsilon N) \leq \delta$$

$$\Leftrightarrow \frac{|\mathcal{H}|}{\delta} \leq \exp(\varepsilon N)$$

$$\Leftrightarrow \log(|\mathcal{H}|) + \log\left(\frac{1}{\delta}\right) \leq \varepsilon N$$

$$\Leftrightarrow \underline{\left(\frac{1}{\varepsilon}\right)\left[\log(|\mathcal{H}|) + \log\left(\frac{1}{\delta}\right)\right]} \leq \underline{N}$$

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 $\|\mathcal{H}\|$	**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 $\|\mathcal{H}\|$		

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)

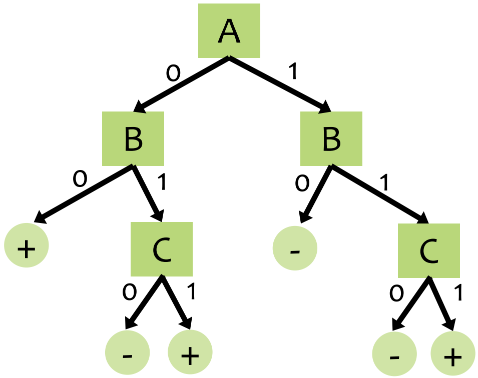
1. Bound is **inversely quadratic in epsilon** (e.g. halving the error requires 4x the examples)
2. Bound is **only logarithmic in |H|** (i.e. same as Realizable case)

	Realizable	Agnostic
Finite $\lvert\mathcal{H}\rvert$	**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 $\lvert\mathcal{H}\rvert$		

Finite vs. Infinite |H|

Finite |H|

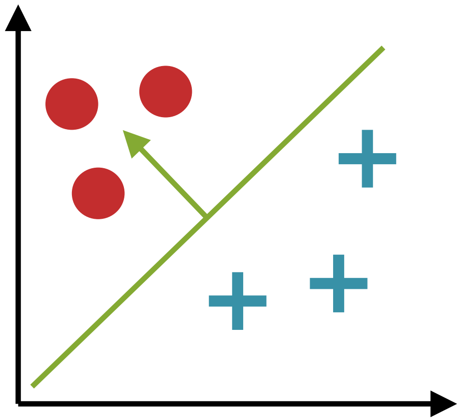
- *Example*: H = the set of all decision trees of depth D over binary feature vectors of length M



- *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…

	Realizable	Agnostic
Finite $\lvert\mathcal{H}\rvert$	**Thm** bele… 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 $\lvert R(h) - \hat{R}(h)\rvert \leq \epsilon$.