# Hypothesis Testing and Computational Learning Theory

EECS 349 Machine Learning

With slides from Bryan Pardo, Tom Mitchell

# Overview

- Hypothesis Testing: How do we know our learners are "good" ?
  - What does performance on test data imply/guarantee about future performance?
- Computational Learning Theory: Are there general laws that govern learning?
  - ***Sample Complexity:*** How many training examples are needed to learn a successful hypothesis?
  - ***Computational Complexity:*** How much computational effort is needed to learn a successful hypothesis?

# Some terms

$X$ is the set of all possible instances

$C$ is the set of all possible concepts $c$

where $c: X \to \{0,1\}$

$H$ is the set of hypotheses considered

by a learner, $H \subseteq C$

$L$ is the learner

$D$ is a probability distribution over $X$

that generates observed instances

# Definition

- The ***true error*** of hypothesis *h*, with respect to the target concept *c* and observation distribution *D* is the probability that *h* will misclassify an instance drawn according to *D*

$$error_D \equiv \underset{x \in D}{P}[c(x) \neq h(x)]$$

- In a perfect world, we'd like the true error to be 0

# Definition

- The ***sample*** ***error*** of hypothesis *h*, with respect to the target concept *c* and sample *S* is the proportion of S that that *h* misclassifies:

$$error_S(h) = 1/|S| \sum_{x \in S} \delta(c(x), h(x))$$

where $\delta(c(x), h(x))$ = 0 if c(x) = h(x),
1 otherwise

# Problems Estimating Error

1. *Bias:* If $S$ is training set, $error_S(h)$ is optimistically biased

$$bias \equiv E[error_S(h)] - error_{\mathcal{D}}(h)$$

   For unbiased estimate, $h$ and $S$ must be chosen independently

2. *Variance:* Even with unbiased $S$, $error_S(h)$ may still *vary* from $error_{\mathcal{D}}(h)$

# Example on Independent Test Set

Hypothesis $h$ misclassifies 12 of the 40 examples in $S$

$$error_S(h) = \frac{12}{40} = .30$$

What is $error_D(h)$?

# Estimators

Experiment:

1. choose sample $S$ of size $n$ according to distribution $\mathcal{D}$

2. measure $error_S(h)$

$error_S(h)$ is a random variable (i.e., result of an experiment)

$error_S(h)$ is an unbiased *estimator* for $error_{\mathcal{D}}(h)$

Given observed $error_S(h)$ what can we conclude about $error_{\mathcal{D}}(h)$?

# Confidence Intervals

If

- $S$ contains $n$ examples, drawn independently of $h$ and each other
- $n \geq 30$ and $n$**error*$_S$(*h*), $n$*(1-*error*$_S$(*h*)) each > 5

Then

- With approximately 95% probability, $error_{\mathcal{D}}(h)$ lies in interval

$$error_S(h) \pm 1.96\sqrt{\frac{error_S(h)(1 - error_S(h))}{n}}$$

# Confidence Intervals

- Under same conditions…

- With approximately N% probability, $error_{\mathcal{D}}(h)$ lies in interval

$$error_S(h) \pm z_N \sqrt{\frac{error_S(h)(1 - error_S(h))}{n}}$$

where

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $N\%$: | 50% | 68% | 80% | 90% | 95% | 98% | 99% |
| $z_N$: | 0.67 | 1.00 | 1.28 | 1.64 | 1.96 | 2.33 | 2.58 |

# Life Skills

- "Convincing demonstration" that certain enhancements improve performance?

- Use online Fisher Exact or Chi Square tests to evaluate hypotheses, e.g:
  - http://www.socscistatistics.com/tests/chisquare2/Default2.aspx

# Overview

- Hypothesis Testing: How do we know our learners are "good" ?
  - What does performance on test data imply/guarantee about future performance?

- Computational Learning Theory: Are there general laws that govern learning?
  - ***Sample Complexity:*** How many training examples are needed to learn a successful hypothesis?

  - ***Computational Complexity:*** How much computational effort is needed to learn a successful hypothesis?

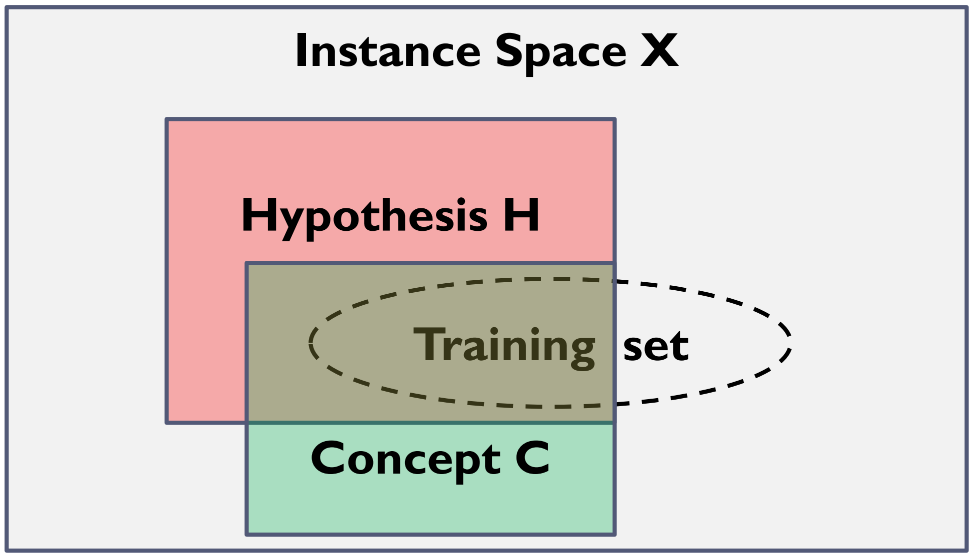
# Computational Learning Theory

- Are there general laws that govern learning?
  - ***No Free Lunch Theorem:*** The expected accuracy of *any* learning algorithm across all concepts is 50%.

- But can we still say something positive?
  - Yes.
  - *Probably Approximately Correct* (PAC) learning

# The world isn't perfect

- If  we can't provide every instance for training, a consistent hypothesis may have error on unobserved instances.




- How many training examples do we need to bound the likelihood of error to a reasonable level?

When is our hypothesis Probably Approximately Correct (PAC)?

# Definitions

- A hypothesis is **consistent** if it has zero error on training examples

- The **version space** ($VS_{H,T}$) is the set of all hypotheses consistent on training set T in our **hypothesis space** H
  - (reminder: hypothesis space is the set of concepts we're considering, e.g. depth-2 decision trees)

# Definition: ε-exhausted

**IN ENGLISH:**

The set of hypotheses consistent with the training data ***T*** is ***ε***-exhausted if, when you test them on the actual distribution of instances, all consistent hypotheses have error below ε

**IN MATH:**

$$VS_{H,T} \text{ is } \varepsilon\text{-exhausted for concept } c$$

$$\text{and sample distribution } D, \text{ if....}$$

$$\forall h \in VS_{H,T}, error_D(h) < \varepsilon$$

# A Theorem

If hypothesis space $H$ is finite, & training set $T$ contains $m$ independent randomly drawn examples of concept $c$

THEN, for any $0 \leq \varepsilon \leq 1$...

$$P(VS_{H,T} \text{ is NOT } \varepsilon\text{-exhausted}) \leq |H|e^{-\varepsilon m}$$

# Proof of Theorem

If hypothesis $h$ has true error $\varepsilon$, the probability of it getting a single random exampe right is :

$$P(h \text{ got 1 example right}) = 1\text{-}\varepsilon$$

Ergo the probability of $h$ getting $m$ examples right is :

$$P(h \text{ got } m \text{ examples right}) = (1\text{-}\varepsilon)^m$$

# Proof of Theorem

If there are $k$ hypotheses in $H$ with error at least $\varepsilon$, call the probability at least one of those $k$ hypotheses got $m$ instances right $P$(at least one bad $h$ looks good).

This prob. is BOUNDED by $k(1\text{-}\varepsilon)^m$

$$P(\text{at least one bad } h \text{ looks good}) \leq k(1\text{-}\varepsilon)^m$$

**"Union" bound**

# Proof of Theorem (continued)

Since $k \leq |H|$, it follows that $k(1\text{-}\varepsilon)^m \leq |\mathrm{H}|(1\text{-}\varepsilon)^m$

If $0 \leq \varepsilon \leq 1$, then $(1-\varepsilon) \leq e^{-\varepsilon}$

Therefore...

$P(\text{at least one bad } h \text{ looks good}) \leq k(1\text{-}\varepsilon)^m \leq |\mathrm{H}|(1\text{-}\varepsilon)^m \leq |\mathrm{H}|e^{-\varepsilon m}$

Proof complete!

We now have a bound on the likelihood that a hypothsesis consistent with the training data will have error $\geq \varepsilon$

# Using the theorem

Let's rearrange to see how many training examples we need to set a bound $\delta$ on the likelihood our true error is $\varepsilon$.

$$|\mathrm{H}|e^{-\varepsilon m} \le \delta$$

$$\ln\left(|\mathrm{H}|e^{-\varepsilon m}\right) \le \ln(\delta)$$

$$\ln\left(|\mathrm{H}|\right) + \ln\left(e^{-\varepsilon m}\right) \le \ln(\delta)$$

$$\ln\left(|\mathrm{H}|\right) - \varepsilon m \le \ln(\delta)$$

$$\ln\left(|\mathrm{H}|\right) - \ln(\delta) \le \varepsilon m$$

$$\frac{1}{\varepsilon}\left(\ln\left(|\mathrm{H}|\right) - \ln(\delta)\right) \le m$$

$$\frac{1}{\varepsilon}\left(\ln\left(|\mathrm{H}|\right) + \ln\left(\frac{1}{\delta}\right)\right) \le m$$

# Probably Approximately Correct (PAC)

$$\frac{1}{\varepsilon}\left(\ln(|H|) - \ln(\delta)\right) \leq m$$

The worst error we'll tolerate

hypothesis space size

The likelihood a hypothesis consistent with the training data will have error ε

number of training examples

# Using the bound

$$\frac{1}{\varepsilon}\left(\ln\left(|H|\right)-\ln\left(\delta\right)\right)\leq m$$

Plug in $\varepsilon$, $\delta$, and ***H*** to get a number of training examples ***m*** that will "guarantee" your learner will generate a hypothesis that is Probably Approximately Correct.

**NOTE: This assumes that the concept is actually IN *H, that H is finite, and that your training set is drawn using distribution D***

# Think/Pair/Share

Average accuracy of any learner across all concepts is 50%, but also:

$$\frac{1}{\varepsilon}\left(\ln\left(|\mathrm{H}|\right)-\ln(\delta)\right)\leq m$$

How can both be true?

Think

Start

End

# Think/Pair/Share

Average accuracy of any learner across all concepts is 50%, but also:

$$\frac{1}{\varepsilon}\left(\ln\left(\left|\mathrm{H}\right|\right)-\ln\left(\delta\right)\right)\leq m$$

How can both be true?

Pair

Start

End

# Think/Pair/Share

Average accuracy of any learner across all concepts is 50%, but also:

$$\frac{1}{\varepsilon}\left(\ln\left(|H|\right)-\ln\left(\delta\right)\right)\leq m$$

How can both be true?

Share

# Problems with PAC

- The PAC Learning framework has 2 disadvantages:

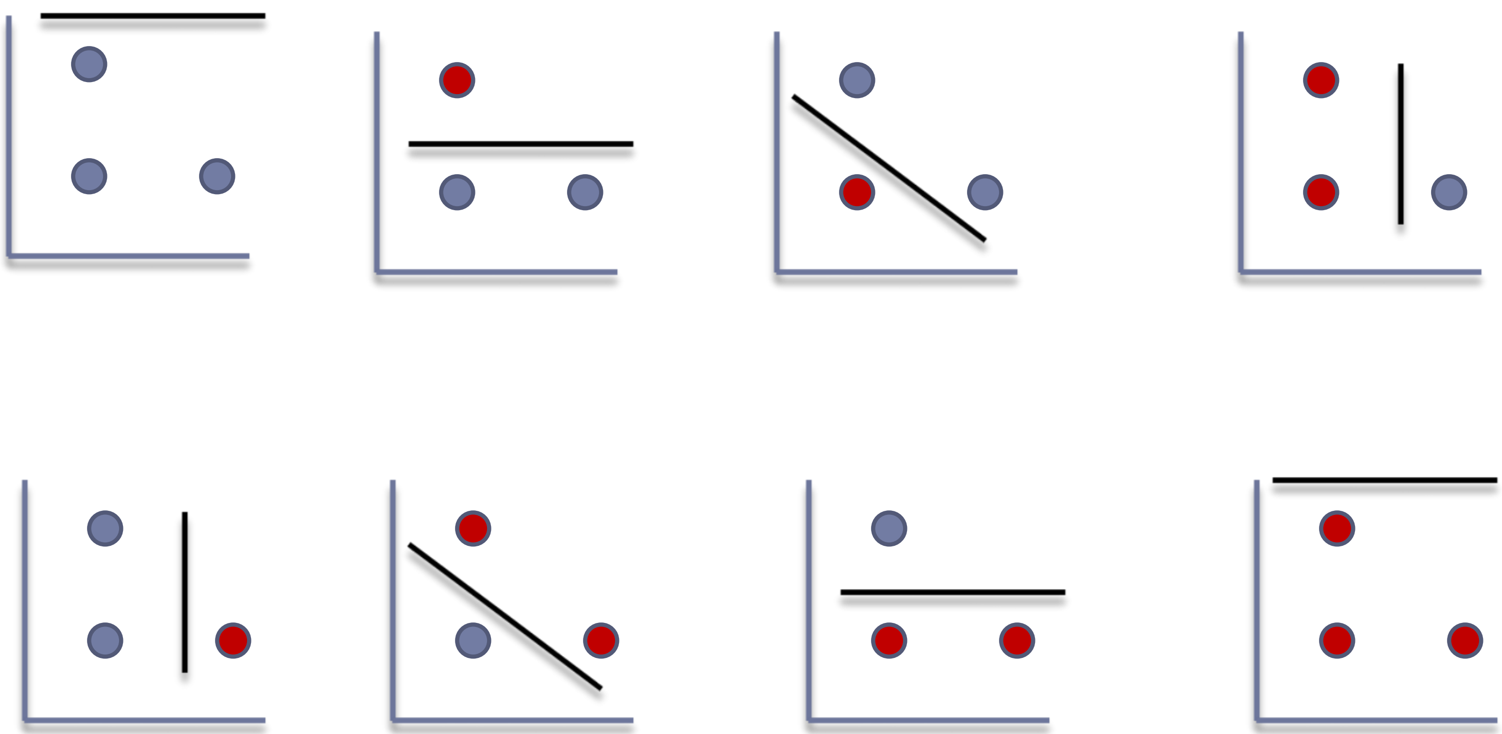
  1) It can lead to weak bounds

  2)Sample Complexity bound cannot be established for infinite hypothesis spaces

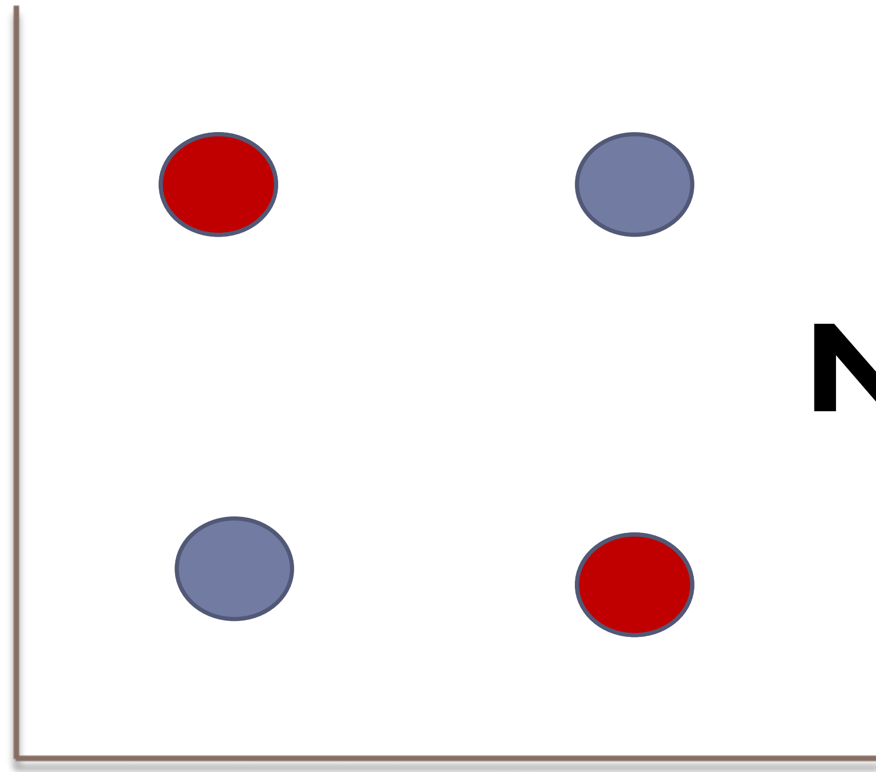
- We introduce the VC dimension for dealing with these problems

# Shattering

**Def:** A set of instances ***S*** is **shattered** by hypothesis set ***H*** iff for every possible concept ***c*** on ***S*** there exists a hypothesis ***h*** in ***H*** that is consistent with that concept.



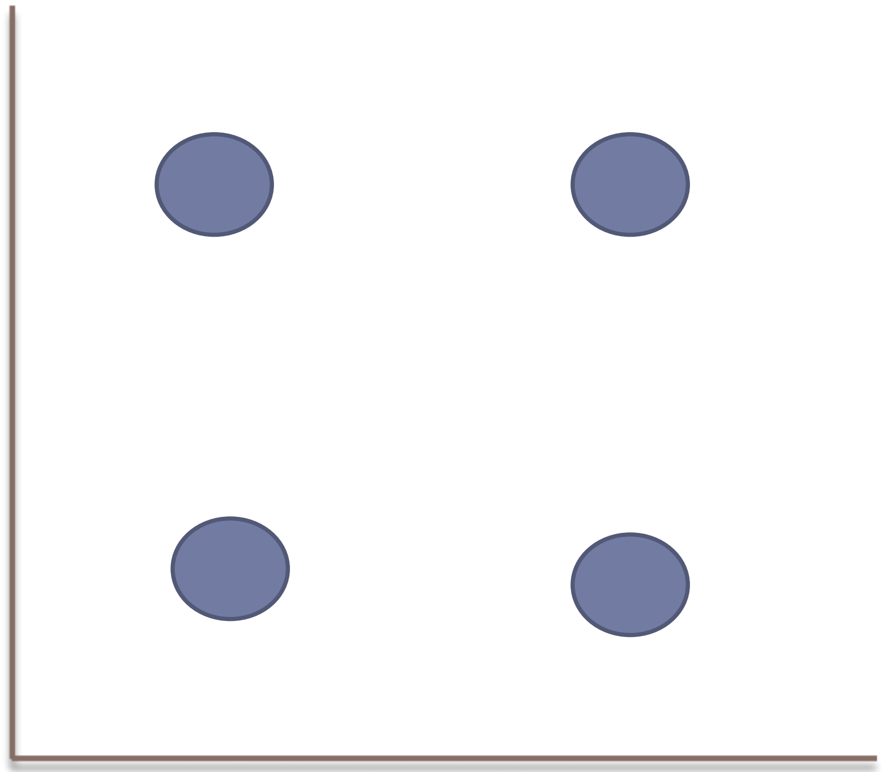
# Can a linear separator shatter this?



**NO!**

The ability of H to shatter a set of instances is a measure of its capacity to represent target concepts defined over those instances

# Can a quadratic separator shatter this?



# Vapnik-Chervonenkis Dimension

**Def:** The **Vapnik-Chervonenkis dimension**, ***VC(H)*** of hypothesis space ***H*** defined over instance space ***X*** is the size of the largest finite subset of ***X*** shattered by ***H***. If arbitrarily large finite sets can be shattered by ***H***, then ***VC(H)*** is infinite.

# How many training examples needed?

- Lower bound on ***m*** using ***VC(H)***

$$m \geq \frac{1}{\varepsilon}(4\log_2(2/\delta) + 8VC(H)\log_2(13/\varepsilon))$$

# Infinite VC dimension?

# Think/Pair/Share

What kind of classifier (that we've talked about) has infinite VC dimension?

|Think |

Start End

# Think/Pair/Share

What kind of classifier (that we've talked about) has infinite VC dimension?

Pair

Start

End

# Think/Pair/Share

What kind of classifier (that we've talked about) has infinite VC dimension?

Share