# Computational Learning Theory

CS 486/686: Introduction to Artificial Intelligence
Fall 2013

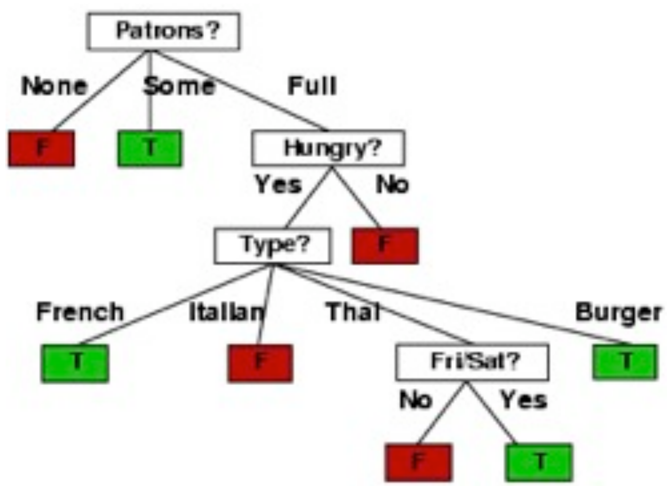
# Overview

- Introduction to Computational Learning Theory
- PAC Learning Theory

Thanks to T Mitchell

# Introduction

- Recall how *inductive learning* works
  - Given a training set of examples of the form (x, f(x)) return a function h (a hypothesis) that approximates f




**Decision Trees**:
Boolean functions

# Computational Learning Theory

- Are there general laws for inductive learning?
- Theory to relate
  - Probability of successful learning
  - Number of training examples
  - Complexity of hypothesis space
  - Accuracy to which f is approximated
  - Manner in which training examples are presented

# Computational Learning Theory

- Sample complexity
  - How many training examples are needed to learn the target function, f?

# Computational Learning Theory

- Sample complexity
  - How many training examples are needed to learn the target function, f?
- Computational complexity
  - How much computation effort is needed to learn the target function, f?

# Computational Learning Theory

- Sample complexity
  - How many training examples are needed to learn the target function, f?
- Computational complexity
  - How much computation effort is needed to learn the target function, f?
- Mistake bound
  - How many training examples will a learner misclassify before learning the target function, f?

# Sample Complexity

- How many examples are sufficient to learn f?
  - If learner proposes instances as queries to a teacher
    - learner proposes x, teacher provides f(x)
  - If the teacher provides training examples
    - teacher provides a sequence of examples of the form (x, f(x))
  - If some random process proposes instances
    - instance x generated randomly, teacher provide f(x)

# Function Approximation

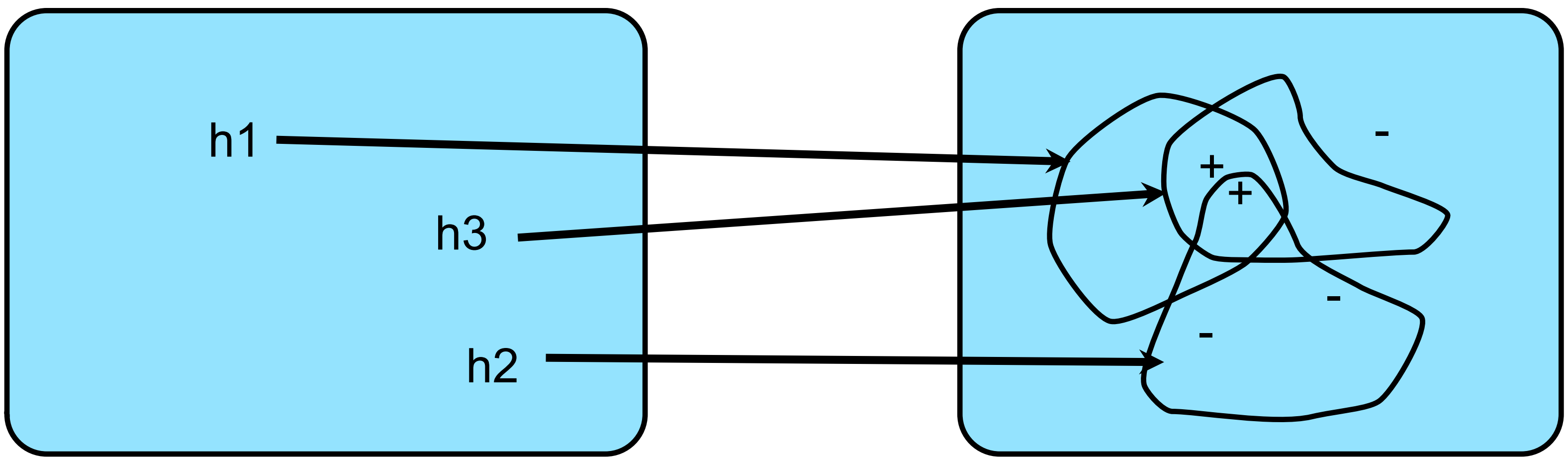
Boolean

$H=\{h:X\text{->}\{0,1\}\}$

$|H|=2^{|X|}=2^{2\text{^}20}$

$X=<x_1,\ldots,x_{20}>$

$|X|=2^{20}$



# Function Approximation

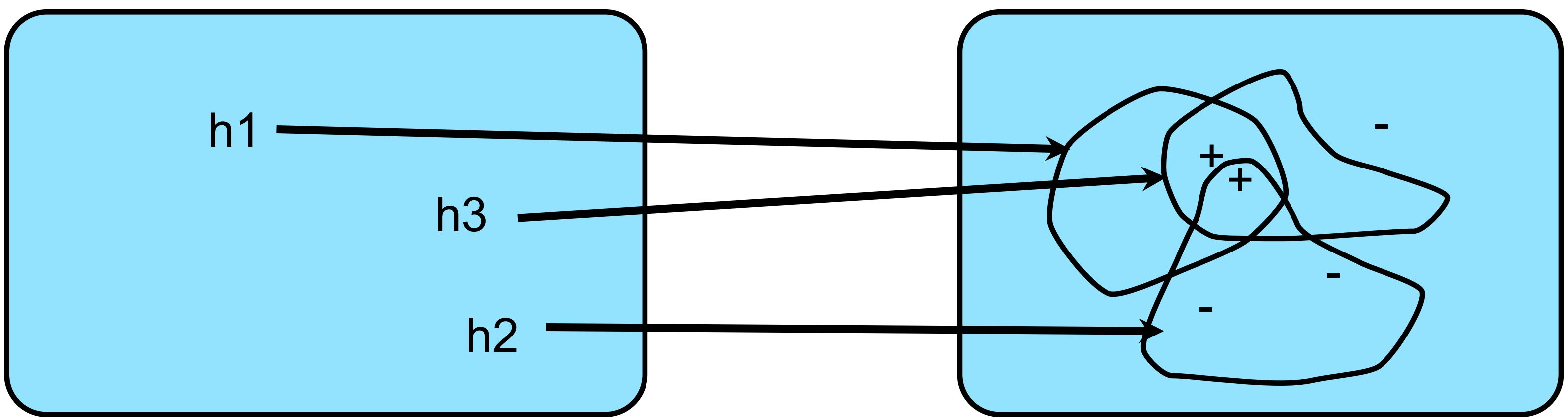
Boolean

$H=\{h:X->\{0,1\}\}$

$|H|=2^{|X|}=2^{2^{20}}$

$X=<x_1,\ldots,x_{20}>$

$|X|=2^{20}$




How many labelled examples are needed in order to determine which of the hypothesis is the correct one?
All $2^{20}$ instances in X must be labelled!
**There is no free lunch!**

# Sample Complexity

- Given
  - set of instances X
  - set of hypothesis H
  - set of possible target concepts F
  - training instances generated by a fixed unknown probability distribution ***D*** over X
- Learner observes sequence, D, of training examples (x,f(x))
  - instances are drawn from distribution ***D***
  - teacher provides f(x) for each instance
- Learner must output a hypothesis h estimating f
  - h is evaluated by its performance on future instances drawn according to ***D***

# True Error of a Hypothesis

- The *true error* ($error_{\boldsymbol{D}}(h)$) of hypothesis h with respect to target function f and distribution ***D*** is the probability that h will misclassify an instance drawn at random according to ***D***
  - $error_{\boldsymbol{D}}(h)=Pr_{x \text{ in } \boldsymbol{D}}[f(x)\neq h(x)]$

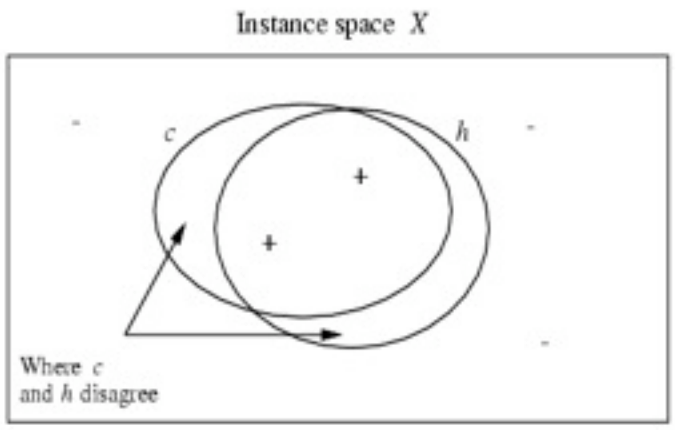



Figure from
*Machine Learning* by T Mitchell
Note our notation is a bit different:
We use f instead of c

# Training Error vs True Error

- Training error of h wrt target function f
  - How often h(x)≠f(x) over training instances D
    - $error_D(h) = Pr_{x \text{ in } D}[f(x) \neq h(x)] = \#(f(x) \neq h(x))/|D|$
    - Note: A consistent h will have $error_D(h)=0$
- True error of h wrt to target function f
  - How often h(x)≠f(x) over future instances drawn at random from D
    - $error_{\boldsymbol{D}}(h) = Pr_{x \text{ in } \boldsymbol{D}}[f(x) \neq h(x)]$

# Version Spaces

- A hypothesis h is consistent with a set of training examples D of target function f if and only if h(x)=f(x).

$$Consistent(h, D) \equiv (\forall (x, f(x)) \in D) h(x) = f(x)$$

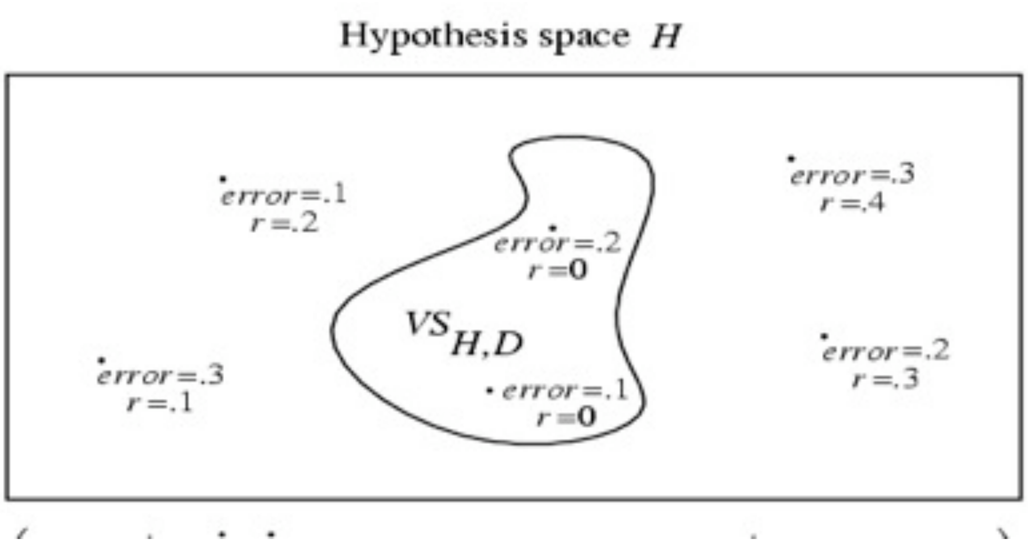
- A version space, $VS_{H,D}$, is the set of all hypothesis from H that are consistent with all training examples in D

$$VS_{H,D} \equiv \{h \in H | Consistent(h, D)\}$$

# ε-Exhausting $VS_{H,D}$

- Version space $VS_{H,D}$ is ε-exhausted wrt f and ***D*** if every h in $VS_{H,D}$ has true error less than ε wrt f and ***D***

$$(\forall h \in VS_{H,D}) error_{\mathcal{D}}(h) < \epsilon$$




r: training error
error: true error

ε-exhausted for ε>0.2

# How many examples will ε-exhaust the VS?

- Theorem[Haussler, 1988]. If
  - H is finite and
  - D is a sequence of m≥1 independent random examples of target function f
- Then for any $0 \leq \varepsilon \leq 1$, the probability that VSH,D is not ε-exhausted (wrt f) is less than $|H|e^{-\varepsilon m}$

# How many examples will ε-exhaust the VS?

Interesting!

This bounds the probability that *any consistent learner* will output a hypothesis h with error(h)≥ε

VSH,D is not ε-exhausted (wrt f) is less than $|H|e^{-\varepsilon m}$

# Proof

# How to interpret this result

$$Pr[(\exists h \in H) st(error_D(h) = 0) \wedge (error_{\mathcal{D}}(h) > \epsilon)] \leq |H|e^{-\epsilon m}$$

- Suppose we want this probability to be at most δ
  - How many training examples are needed?

$$m \geq \frac{1}{\epsilon}(\ln|H| + \ln(1/\delta))$$

  - If $error_D(h)=0$ then with probability at least (1-δ)

$$error_{\mathcal{D}}(h) \leq \frac{1}{m}(\ln|H| + \ln(1/\delta))$$

# Decision Tree Example

# PAC Learning

- Probably Approximately Correct (PAC) Learner
  - **Given** a class C of possible target concepts (f) defined over a set of instances X os length n, and a learner L using hypothesis space H
  - C is **PAC-learnable** by L using H if for all f in C, distributions ***D*** over X, $0<\varepsilon<1/2$ and $0<\delta<1/2$, leaner L will with probability at least $(1-\delta)$ output a hypothesis h such that $error_{\boldsymbol{D}}(h)\leq\varepsilon$ in *time polynomial* in $1/\varepsilon$, $1/\delta$, n, and size(f)

# Agnostic Learners

- So far we have assumed f is in H. What if we do not make this assumption? (Agnostic Learning)

- Goal: Hypothesis h that makes the fewest errors on the training data!

# Agnostic Learning Derivation

- **Hoeffding Bound** (additive Chernoff bound): given a coin with Pr(heads)=θ, after m independent coin flips you observe Pr(heads|m)=$\theta_m$

$$Pr(\theta > \theta_m + \epsilon) \leq e^{-2m\epsilon^2}$$

- In our case, for any single hypothesis

$$Pr(error_{\mathcal{D}}(h) > error_D(h) + \epsilon) \leq e^{-2m\epsilon^2}$$

# Agnostic Learning Derivation

- To assure that the best hypothesis found by L has an error bounded this way

$$Pr[(\exists h \in H)(error_{\mathcal{D}}(h) > error_D(h) + \epsilon)] \leq |H|e^{-2m\epsilon^2}$$

- Sample complexity

$$m \geq \frac{1}{2\epsilon^2}(\ln|H| + \ln(1/\delta))$$

# Infinite Hypothesis Spaces

- Recall $$m \geq \frac{1}{\epsilon}(\ln|H| + \ln(1/\delta))$$
  - What if H ={hlh:X->Y} is infinite? What is the “right” measure of complexity?
  - The largest subset of X for which H can guarantee zero training error (regardless of target function f)
    - Vapnik-Chervonenkis (VC) dimension

# Shattering

- A *dichotomy* of a set S is a partition of S into two disjoint subsets
  - Note: Given S there are $2^{|S|}$ dichotomies

- A set of instances S is *shattered* by hypothesis space H if and only if for every dichotomy of S there exists some hypothesis in H consistent with this dichotomy

# VC Dimension

- The VC(H) dimension of hypothesis space H defined over input space X is the size of the largest finite subset of X shattered by H.
  - If arbitrarily large finite subsets of X can be shattered by H then VC(H)=$\infty$

# VC Examples

- Let X=**R**, H=all open intervals, h:a<x<b (h(x)=1 is a<x<b, 0 otherwise)
- Let S={3.1, 4.5}, |S|=2
  - dichotomies: both are 1, both are 0, one is 1 and the other 0
  - H can shatter S
  - VC(H) is at least 2
- What about S={x,y,z} where x<y<z?

# Sample Complexity

- How many randomly drawn examples suffice to ε-exhaust $VS_{H,D}$ with probability at least (1-δ)?

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

- Lower bound: For any 0<ε<1/8 and 0<δ<0.01, there exists some ***D*** and a f in C such that if L observes fewer than

$$\max\left[\frac{1}{\epsilon}\log(1/\delta), \frac{VC(C)-1}{32\epsilon}\right]$$

- Then with prob. at least δ, L outputs a hypothesis with $\mathrm{error}_{\boldsymbol{D}}(h)>\varepsilon$

# Changing Directions: Observations about No Free Lunch

- I mentioned “No Free Lunch” earlier in todays lecture
- What do we really mean by “No Free Lunch”?

# No Free Lunch Principle

- “No learning is possible without the application of prior domain knowledge”

- For any learning algorithm, there is some distribution that generates data such that when trained over this distribution will produce large error. If $|S|$ is much smaller than $|X|$ then error can be close to 0.5.