Statistical Learning Theory

Why learning doesn’t always work

- Unrealizability
  - f may not be in H or not easily represented in H
- Variance
  - There may be many ways to represent f
  - depends on the specific training set
- Noise/stochasticity
  - Elements that cannot be predicted: Missing attributes or stochastic process
- Complexity
  - Finding f may be intractable

Regularization

- Forcing solutions to be simple
  - Add penalty for complex models
  - E.g. accuracy + size of tree
  - Number of samples in Thin-KNN
  - Sum of weights or number of nonzero weights (number of connections) in NN
- **Minimum Description Length (MDL)**

Example: Smart Investing

Task: Pick stock analyst based on past performance.

Experiment:

- Have analyst predict “next day up/down” for 10 days.
- Pick analyst that makes the fewest errors.

Situation 1:

- 1 stock analyst {A1}, A1 makes 5 errors

Situation 2:

- 3 stock analysts {A1,B1,B2}, B2 best with 1 error

Situation 3:

- 1003 stock analysts {A1,B1,B2,C1,…,C1000}, C543 best with 0 errors

Which analysts are you most confident in: A1, B2, or C543?

Outline

Questions in Statistical Learning Theory:

- How good is the learned rule after ***n*** examples?
- How many examples do I need before the learned rule is accurate?
- What can be learned and what cannot?
- Is there a universally best learning algorithm?

In particular, we will address:

What is the true error of h if we only know the training error of h?

- Finite hypothesis spaces and zero training error
- (Finite hypothesis spaces and non-zero training error)

Game: Randomized 20-Questions

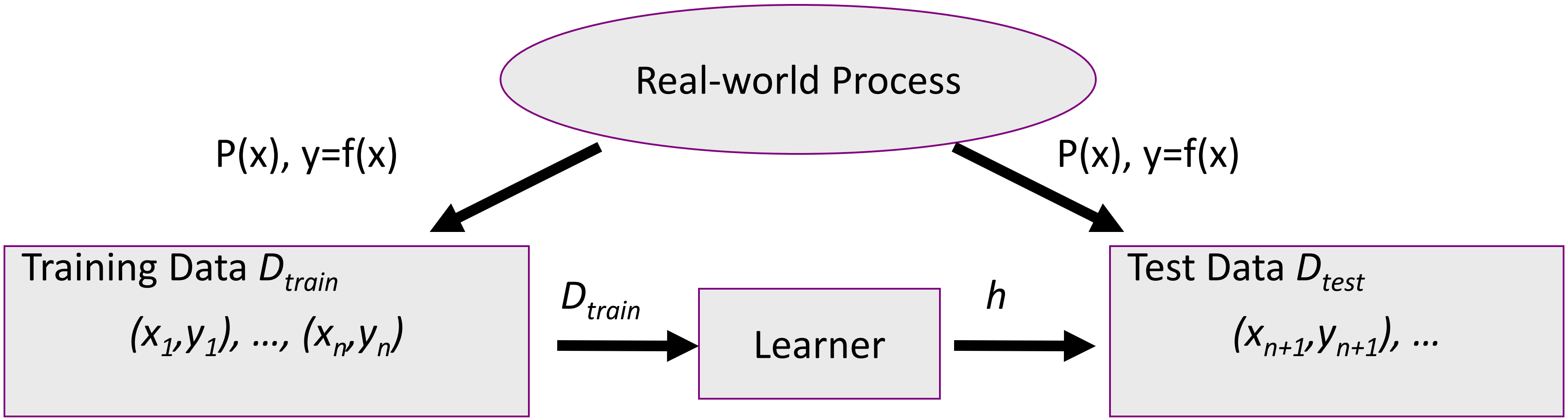
Game: 20-Questions

- I think of object f
- For i = 1 to 20
  - You get to ask 20 yes/no questions about f and I have to answer truthfully
- You make a guess h
- You win, if $f=h$

Game: Randomized 20-Questions

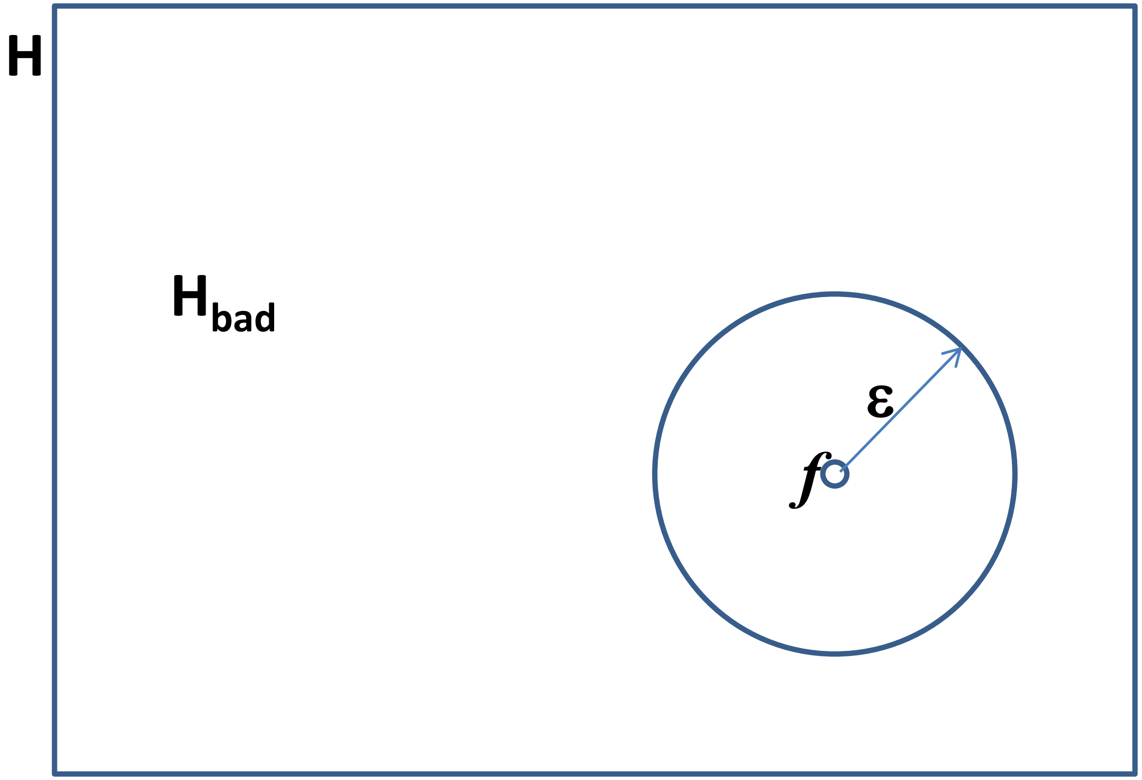
- I pick function $f \in H$, where f$: X \rightarrow \{-1,+1\}$
- For i = 1 to 20
  - World delivers instances $x \in X$ with probability $P(x)$ and I have to tell you $f(x)$
- You form hypothesis $h \in H$ trying to guess my $f \in H$
- You win if $f(x)=h(x)$ with probability at least $1-\varepsilon$ for x drawn according to $P(x)$.

Inductive Learning Model



- Probably Approximately Correct (PAC) Learning Model:
  - Take any function *f* from *H*
  - Draw *n* Training examples D_{train} from P(x), label as $y=f(x)$
  - Run learning algorithm on D_{train} to produce *h* from *H*
  - Gather Test Examples D_{test} from P(x)
  - Apply *h* to D_{test} and measure fraction (probability) of h(x)≠f(x)
  - How likely is it that error probability is less than some threshold ε (for any f from H)?

What are the chances of a wrong hypothesis making correct predictions?



Useful Formulas

- Binomial Distribution: The probability of observing *x* heads in a sample of *n* independent coin tosses, where in each toss the probability of heads is *p*, is

$$P(X = x|p, n) = \frac{n!}{r!(n-x)!}p^x(1-p)^{n-x}$$

- Union Bound:

$$P(X_1 = x_1 \vee X_2 = x_2 \vee ... \vee X_n = x_n) \leq \sum_{i=1}^{n} P(X_i = x_i)$$

- Unnamed:

$$(1-\epsilon) \leq e^{-\epsilon}$$

Chances of getting it wrong

- Chances that $h_b \in H_{bad}$ is consistent with N examples
  - ErrorRate(h_b)>ε so chances it agrees with an example is $\leq (1-\varepsilon)$
  - Chances it agrees with N examples $\leq (1-\varepsilon)^N$
  - P(H_{bad} contains a consistent hypothesis) = $|H_{bad}|\,(1-\varepsilon)^N \leq |H|\,(1-\varepsilon)^N$
  - We want to reduce this below some probability δ so $|H|\,(1-\varepsilon)^N \leq \delta$
  - Given $(1-\varepsilon)\leq e^{-\varepsilon}$ we get

$$N \geq \frac{1}{\varepsilon}\left(\ln\frac{1}{\delta} + \ln|H|\right)$$

Size of hypothesis space |H|

- **How many possible Boolean functions are there on *n* binary attributes?**
- A = n
- B = 2^n
- C = 2^{2n}
- D = 2^{2^n}
- E = $2^{2^{2^n}}$

Size of hypothesis space |H|

x1	x2	x3	Function
1	1	1	y_0
1	1	0	y_1
1	0	1	y_2
1	0	0	Y_3
0	1	1	Y_4
0	1	0	Y_5
0	0	1	Y_6
0	0	0	Y_7

N=3 → |H|=256 N=10→|H|=1.8×10^{308}

All Boolean functions

- If $|H| = 2^{2^n}$ then

$$N \geq \frac{1}{\varepsilon}\left(\ln \frac{1}{\delta} + O(2^n) \right)$$

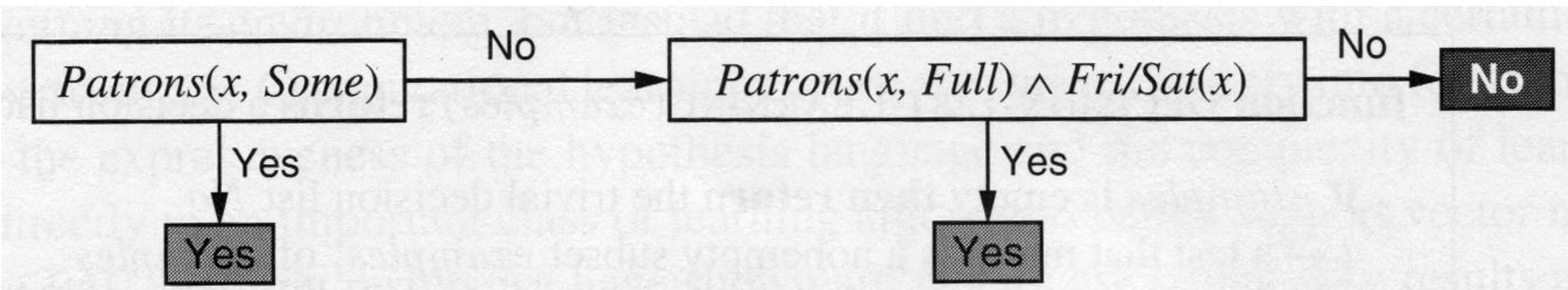
- So we need to see the entire space to determine the function reliably

Approach

- Look for simplest hypothesis
- Limit the size of |H| by only looking at simple (limited) subspace

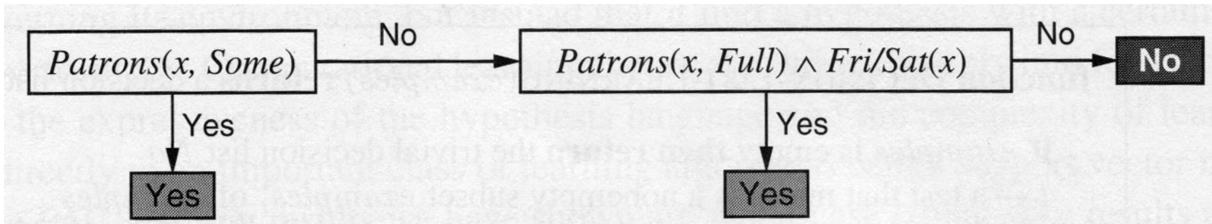
Example: Decision lists

- List of tests, executed serially



- *k*-DL: Each test contains at most *k* literals
- Includes as a subset *k*-DT
  - All decision trees of depth at most *k*

Example: Decision lists



- Number of possible tests of size ***k*** from ***n*** attributes is

$$C(n,k)=\sum_{i=0}^{k}\binom{2n}{i}=O\left(n^{k}\right)$$

- Total size of hypothesis space |H| is
  - Each test can yield *Yes, No,* or be *Absent*
  - Tests can be ordered in any sequence

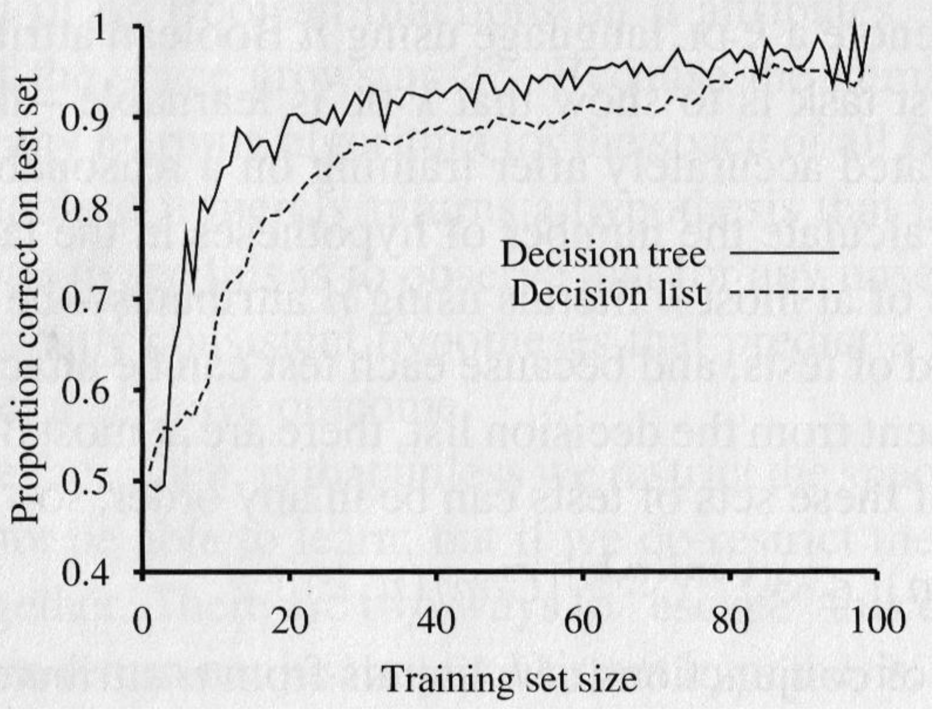
$$\left|kDL(n)\right|=3^{C(n,k)}C(n,k)! \quad \rightarrow \quad \left|kDL(n)\right|=2^{O(n^{k}\log_2 n^{k})}$$

- Therefore number of training samples is reasonable for small k

$$N\geq\frac{1}{\varepsilon}\left(\ln\frac{1}{\delta}+O(n^{k}\log_{2}n^{k})\right)$$

Example: Decision lists

- Search for simple (small k) tests that classify large portion of data
- Add test to list, remove classified datapoints
- Repeat with remaining data




Free!

Inductive bias

- The inductive bias of a learning algorithm is the set of assumptions that the learner uses to predict outputs given inputs that it has not encountered (Mitchell, 1980)
  - No Free Lunch (Mitchell, Wolpert,…)
  - Bias-free learning is futile*

*Wolpert and Macready have proved that there are free lunches in [coevolutionary] optimization

Generalization Error Bound: Finite H, Zero Training Error

- Model and Learning Algorithm
  - Learning Algorithm *A* with a finite hypothesis space *H*
  - Sample of *n* labeled examples D_{train} drawn according to *P(x)*
  - Target function f $\in H$
    ➔ At least one $h \in H$ has zero training error $Err_{D_{train}}(h)$
  - Learning Algorithm *A* returns zero training error hypothesis $\hat{h}$
- **What is the probability δ that the true prediction error of $\hat{h}$ is larger than ε?**

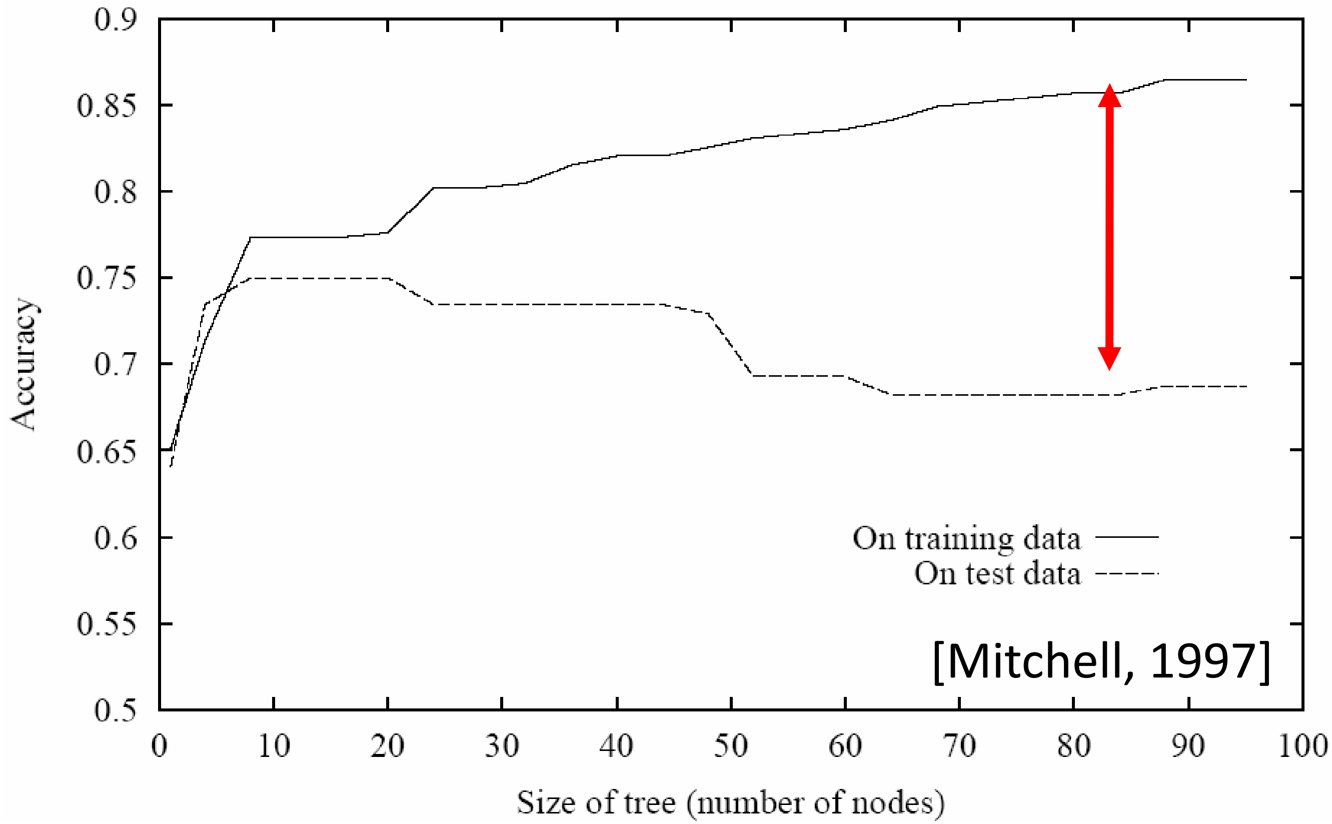
$$P(Err_P(\widehat{h}) \geq \epsilon) \leq |H| e^{-\epsilon n}$$

Generalization Error Bound: Finite H, Non-Zero Training Error

- Model and Learning Algorithm
  - Sample of n labeled examples D_{train}
  - Unknown (random) fraction of examples in D_{train} is mislabeled (noise)
  - Learning Algorithm A with a finite hypothesis space H
  - A returns hypothesis $\hat{h}=A(S)$ with lowest training error
- **What is the probability δ that the prediction error of $\hat{h}$ exceeds the fraction of training errors by more than ε?**

$$P\left(\left|Err_{D_{train}}(h_{\mathcal{A}(D_{train})}) - Err_P(h_{\mathcal{A}(D_{train})})\right| \geq \epsilon\right) \leq 2|H|e^{-2\,\epsilon^2\,n}$$

Overfitting vs. Underfitting

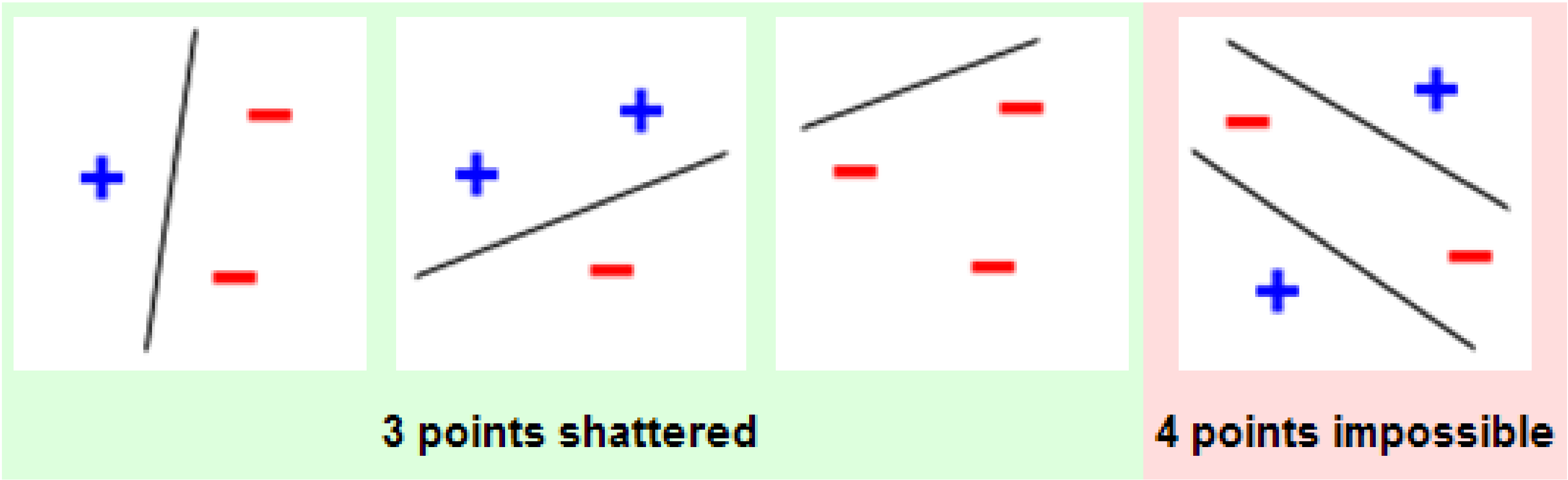


With probability at least *(1-δ)*:

$$Err_P(h_{\mathcal{A}(D_{train})}) \leq Err_{D_{train}}(h_{\mathcal{A}(D_{train})}) + \sqrt{\frac{1}{2n}(\ln(2|H|) - \ln(\delta))}$$

VC-Dimension

- The capacity of a hypothesis space H
  - The maximum number of points with arbitrary labelings that could be separated (“shattered”)
  - VC dimension of linear classifiers is 3



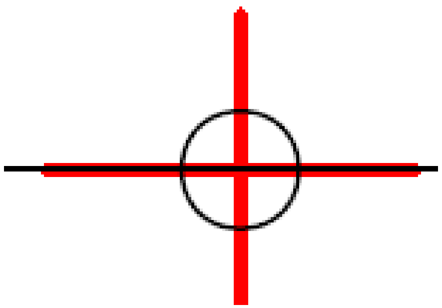
Representational power

- Machine f can shatter a set of points x_1, x_2 .. X_r if and only if…
  - For every possible training set of the form (x_1,y_1) , (x_2,y_2) ,… (x_r ,y_r)…There exists some value of a that gets zero training error.

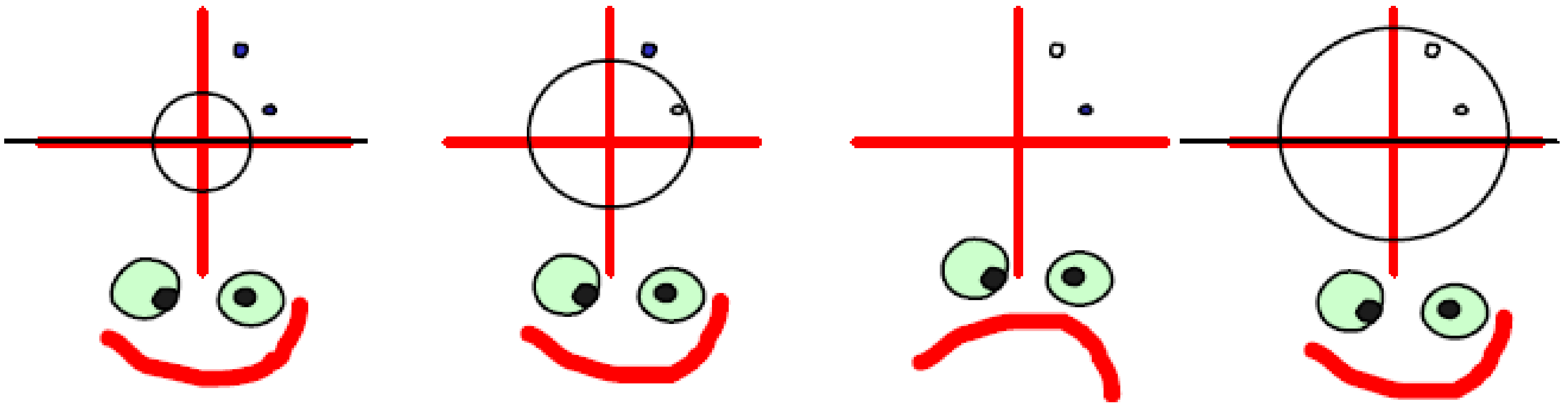
Representational power

- What is the VC dimension of the hypothesis space of all circles centered at the origin?

h = f(x,b) = sign(x.x-b)



A=1 B=2 C=3 D=4 E=Whatever



$f(x,b) = \mathrm{sign}(x.x-b)$

From Andrew Moore

Generalization Error Bound: Infinite H, Non-Zero Training Error

- Model and Learning Algorithm
  - Sample of n labeled examples D_{train}
  - Learning Algorithm A with a hypothesis space H with $VCDim(H)=d$
  - A returns hypothesis $\hat{h}=A(S)$ with lowest training error
- Given hypothesis space H with $VCDim(H)$ equal to d and a training sample D_{train} of size n, with probability at least $(1-\delta)$ it holds that

$$Err_P(h_{\mathcal{A}(D_{train})}) \leq Err_{D_{train}}(h_{\mathcal{A}(D_{train})}) + \sqrt{\frac{d\left(\ln\frac{2n}{d}+1\right)-\ln\frac{\delta}{4}}{n}}$$

This slide is not relevant for exam.