More details:
General: http://www.learning-with-kernels.org/
Example of more complex bounds:
http://www.research.ibm.com/people/t/tzhang/papers/jmlr02_cover.ps.gz

PAC-learning, VC Dimension and Margin-based Bounds

Machine Learning – 10701/15781

Carlos Guestrin

Carnegie Mellon University

February 28th, 2005

Review: Generalization error in finite hypothesis spaces [Haussler '88]

- ***Theorem***: Hypothesis space *H* finite, dataset *D* with *m* i.i.d. samples, $0 < \varepsilon < 1$: for any learned hypothesis *h* that is consistent on the training data:

$$P(\text{error}_{\mathcal{X}}(h) > \epsilon) \leq |H|e^{-m\epsilon}$$

Consistent with D $\Rightarrow$ $\text{Error}_D(h) = 0$

$\not\Rightarrow$ zero errors in test set

$\text{error}_{\mathcal{X}}(h) \rightarrow$ expected error $x \in \mathcal{X}$

Even if *h* makes zero errors in training data, may make errors in test

Using a PAC bound

- Typically, 2 use cases:
  - 1: Pick ε and δ, give you *m*
  - 2: Pick m and δ, give you ε

$$P(\text{error}_{\mathcal{X}}(h) > \epsilon) \leq |H|e^{-m\epsilon}$$

1

$P(\text{error}_{\mathcal{X}}(h) > \varepsilon) \leq \delta$

$|H|e^{-m\varepsilon} \leq \delta$, log on both sides

$\ln|H| - m\varepsilon \leq \ln \delta$

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

2

$\varepsilon \geq \frac{\ln|H| + \ln 1/\delta}{m}$

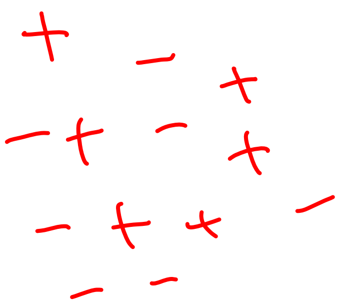
$\text{error}_{\mathcal{X}}(h) \leq \frac{\ln|H| + \ln 1/\delta}{m}$

with prob. at least $1-\delta$

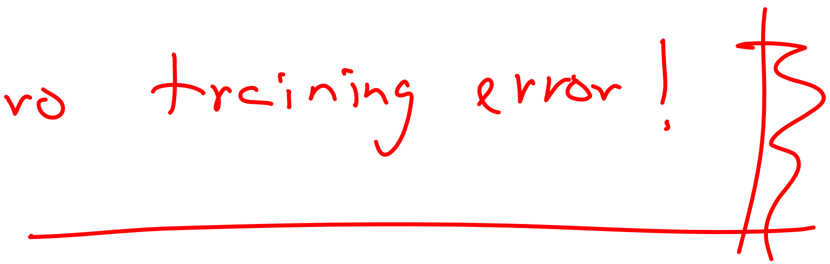
Limitations of Haussler '88 bound

- Consistent classifier

$$P(\text{error}_{\mathcal{X}}(h) > \epsilon) \leq |H| e^{-m\epsilon}$$



Zero training error!



- Size of hypothesis space

$|H|$

what if it's too large

continuous

What if our classifier does not have zero error on the training data?

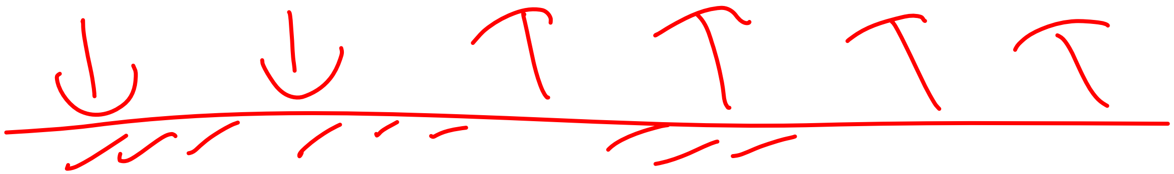
- A learner with zero training errors may make mistakes in test set
- A learner with $error_D(h)$ in training set, may make even more mistakes in test set

$error_X(h)$ relates $error_D(h)$?

Simpler question: What's the expected error of a hypothesis?

- The error of a hypothesis is like estimating the parameter of a coin!

$\theta_H = \frac{2}{6}$



- Chernoff bound: for *m* i.d.d. coin flips, $x_1,\ldots,x_m$, where $x_i \in \{0,1\}$. For $0<\varepsilon<1$:

$$\boxed{P\left(\theta - \frac{1}{m}\sum_i x_i > \epsilon\right) \leq e^{-2m\epsilon^2}}$$

Using Chernoff bound to estimate error of a single hypothesis

$$\boxed{P\left(\theta - \frac{1}{m}\sum_i x_i > \epsilon\right) \leq e^{-2m\epsilon^2}}$$

Given hypothesis h, how well will it do on test data?

$error_{\mathcal{X}}(h) \equiv \theta$

$error_D(h) \equiv \frac{1}{m}\sum_i x_i$

$$\underline{P(error_{\mathcal{X}}(h) - error_D(h) > \varepsilon) \leq e^{-2m\varepsilon^2}}$$

But we are comparing many hypothesis: **Union bound**

$P(\text{error}_{\mathcal{X}}(h_1) - \text{error}_D(h_1) > \varepsilon) \le \delta$

$P(\text{error}_{\mathcal{X}}(h_2) - \text{error}_D(h_2) > \varepsilon) \le \delta$

δ δ

h_1 mistakes

h_2 mistakes

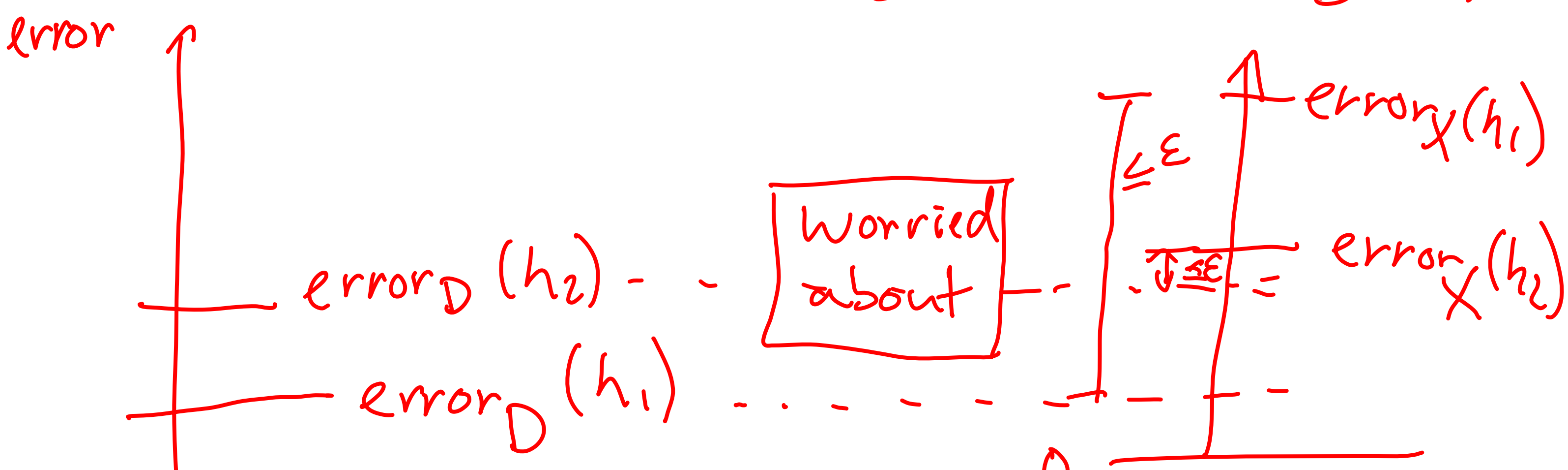
$P(h_1 \,\&\, h_2 \text{ OK}) \ge 1 - 2\delta$

For each hypothesis h_i:

$$P\left(\text{error}_{\mathcal{X}}(h_i) - \text{error}_D(h_i) > \epsilon\right) \le e^{-2m\epsilon^2}$$

What if I am comparing two hypothesis, h_1 and h_2?

Choose h_1, because $\text{error}_D(h_1) \le \text{error}_D(h_2)$



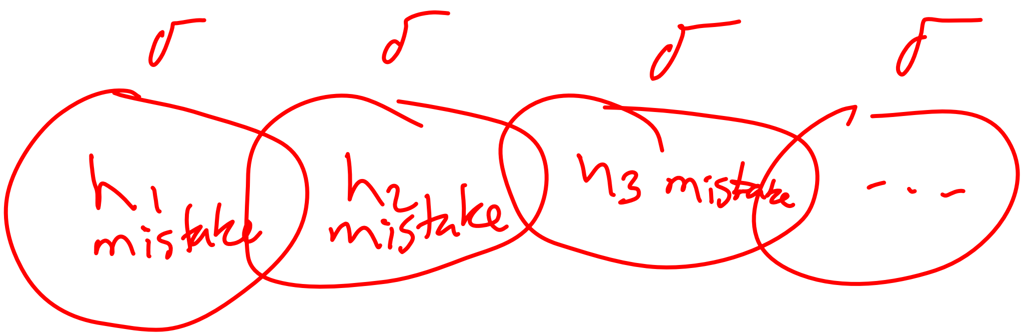
$\text{error}_{\mathcal{X}}(h_1) - \text{error}_D(h_1) \le \varepsilon$ & $\text{error}_{\mathcal{X}}(h_2) - \text{error}_D(h_2) \le \varepsilon$

Generalization bound for |H| hypothesis

- ***Theorem***: Hypothesis space *H* finite, dataset *D* with *m* i.i.d. samples, 0 < ε < 1 : for any learned hypothesis *h*:

test data training data

$$P\left(\text{error}_{\mathcal{X}}(h) - \text{error}_D(h) > \epsilon\right) \le |H|e^{-2m\epsilon^2}$$



Prob. mistake $\le |H|\delta$

PAC bound and Bias-Variance tradeoff

$$P\left(\text{error}_{\mathcal{X}}(h) - \text{error}_D(h) > \epsilon\right) \le |H| e^{-2m\epsilon^2}$$

or, after moving some terms around, with probability at least 1-δ:

$$\text{error}_{\mathcal{X}}(h) \le \overset{\text{bias}}{\text{error}_D(h)} + \underbrace{\overset{\text{Variance}}{\sqrt{\frac{\ln|H| + \ln\frac{1}{\delta}}{2m}}}}_{\varepsilon}$$

testset mistakes

training mistakes

if H is big: ↓ ↑

if H is small: ↑ ↓

- **Important: PAC bound holds for all *h*, but doesn't guarantee that algorithm finds best *h*!!!**

What about the size of the hypothesis space?

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

- How large is the hypothesis space?

|H| is large $\Rightarrow$ need many training examples

Boolean formulas with *n* binary features

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

bad!

look up table

x_1	...	x_n	y
1	1	1	⋮
1	1	0	
⋮	⋮	⋮	

$\ln|H| = O(2^n)$

$|H| = 2^{2^n}$

pretty good...

conjuctions:

$\langle 1, 0, ?, ?, 1, \dots \rangle$

$|H| = 3^n$

$\ln|H| = O(n)$

look up table for k conjunction for n-k

$|H| = 2^{2^k} \cdot 3^{n-k}$

$\ln|H| = O(2^k + (n-k))$

grow fast with k

Number of decision trees of depth k

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

Recursive solution

Given *n* attributes

H_k = Number of decision trees of depth k !

$H_0 = 2$

H_{k+1} = (#choices of root attribute) * (n)

(# possible left subtrees) * (H_k)

(# possible right subtrees) (H_k)

$= n * H_k * H_k$

$\ln|H| = O(2^k \cdot \log n)$

grow fast with depth

Write $L_k = \log_2 H_k$

$L_0 = 1$

$L_{k+1} = \log_2 n + 2L_k$!

So $L_k = (2^k-1)(1+\log_2 n) + 1$

PAC bound for decision trees of depth k

$$m \geq \frac{\ln 2}{2\epsilon^2}\left((2^k - 1)(1 + \log_2 n) + 1 + \ln\frac{1}{\delta}\right)$$

DT of depth k → 2^k leaves

grow exp. in k

- Bad!!!
  - Number of points is exponential in depth!

learning algorithm only gets here if there is enough data

- But, for *m* data points, decision tree can't get too big…

only reach m leaves

Number of leaves never more than number data points

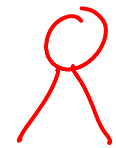
Number of decision trees with k leaves

plug in here $m \geq \frac{1}{2\epsilon^2}\left(\ln|H| + \ln\frac{1}{\delta}\right)$

H_k = Number of decision trees with k leaves

H_0 =2

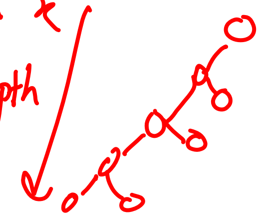
$$H_{k+1} = n\sum_{i=1}^{k} H_i H_{k+1-i}$$



H_2 $\quad H_{k+1-2}$

Depth k

depth m



m leaves

for all my data

Loose bound:

$$H_k \leq n^{k-1}(k+1)^{2k-1}$$

$\ln|H| = O(n k^2)$

a lot better

Reminder:

$$|\text{DTs depth } k| = 2*(2n)^{2^k-1}$$

$\ln|H| = O(2^k n)$

PAC bound for decision trees with k leaves – Bias-Variance revisited

$$H_k = n^{k-1}(k+1)^{2k-1}$$

$$\text{error}_{\mathcal{X}}(h) \leq \text{error}_D(h) + \sqrt{\frac{\ln|H| + \ln\frac{1}{\delta}}{2m}}$$

$$\boxed{\text{error}_{\mathcal{X}}(h) \leq \text{error}_D(h) + \sqrt{\frac{(k-1)\ln n + (2k-1)\ln(k+1) + \ln\frac{1}{\delta}}{2m}}}$$

suppose ~~n=m~~ $k=m$	$\emptyset$		↑ really big
if $k=\alpha m$, $\alpha<1$	>0		↓ smaller

What did we learn from decision trees?

- Bias-Variance tradeoff formalized

$$\text{error}_{\mathcal{X}}(h) \leq \text{error}_D(h) + \sqrt{\frac{(k-1)\ln n + (2k-1)\ln(k+1) + \ln\frac{1}{\delta}}{2m}}$$

- Moral of the story:

  Complexity of learning not measured in terms of size hypothesis space, but in maximum *number of points* that allows consistent classification
  - Complexity *m* – no bias, lots of variance
  - Lower than *m* – some bias, less variance

What about continuous hypothesis spaces?

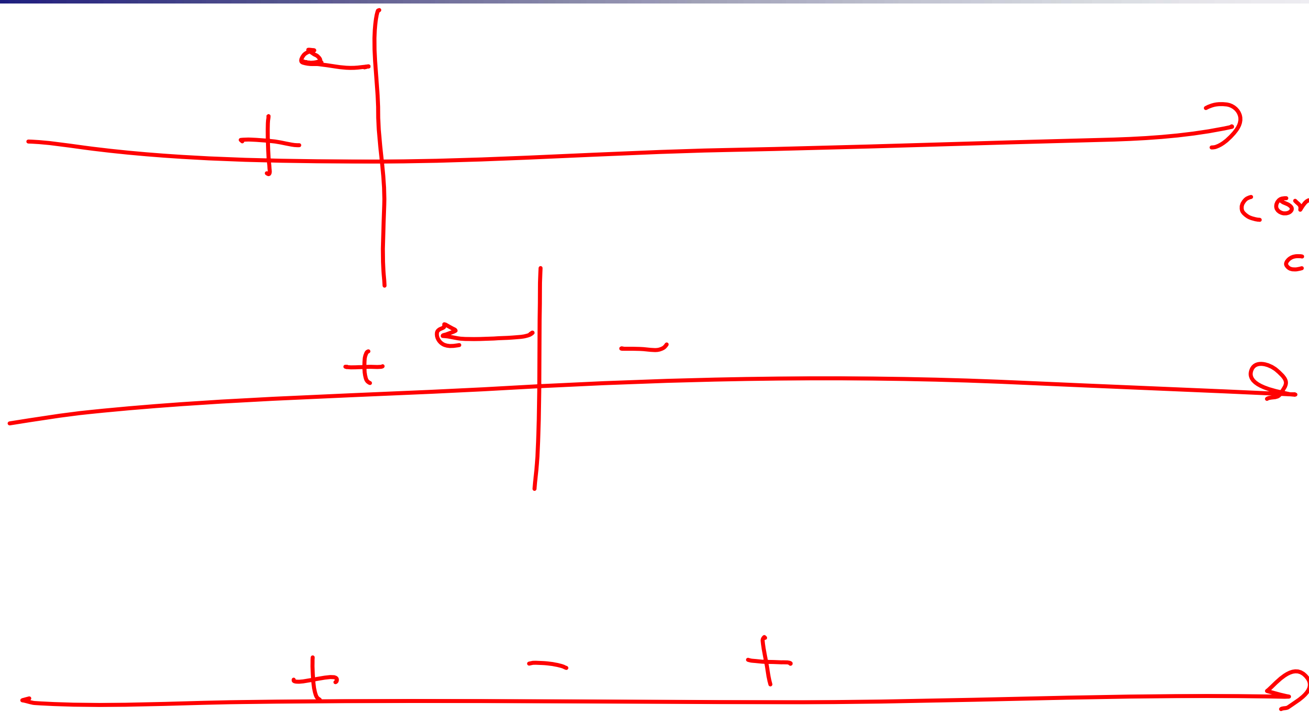
bias Variance

$$\text{error}_{\mathcal{X}}(h) \leq \text{error}_D(h) + \sqrt{\frac{\ln |H| + \ln \frac{1}{\delta}}{2m}}$$

- Continuous hypothesis space:
  - |H| = ∞
  - Infinite variance???

- **As with decision trees, only care about the maximum number of points that can be classified exactly!**

How many points can a linear boundary classify exactly? (1-D)



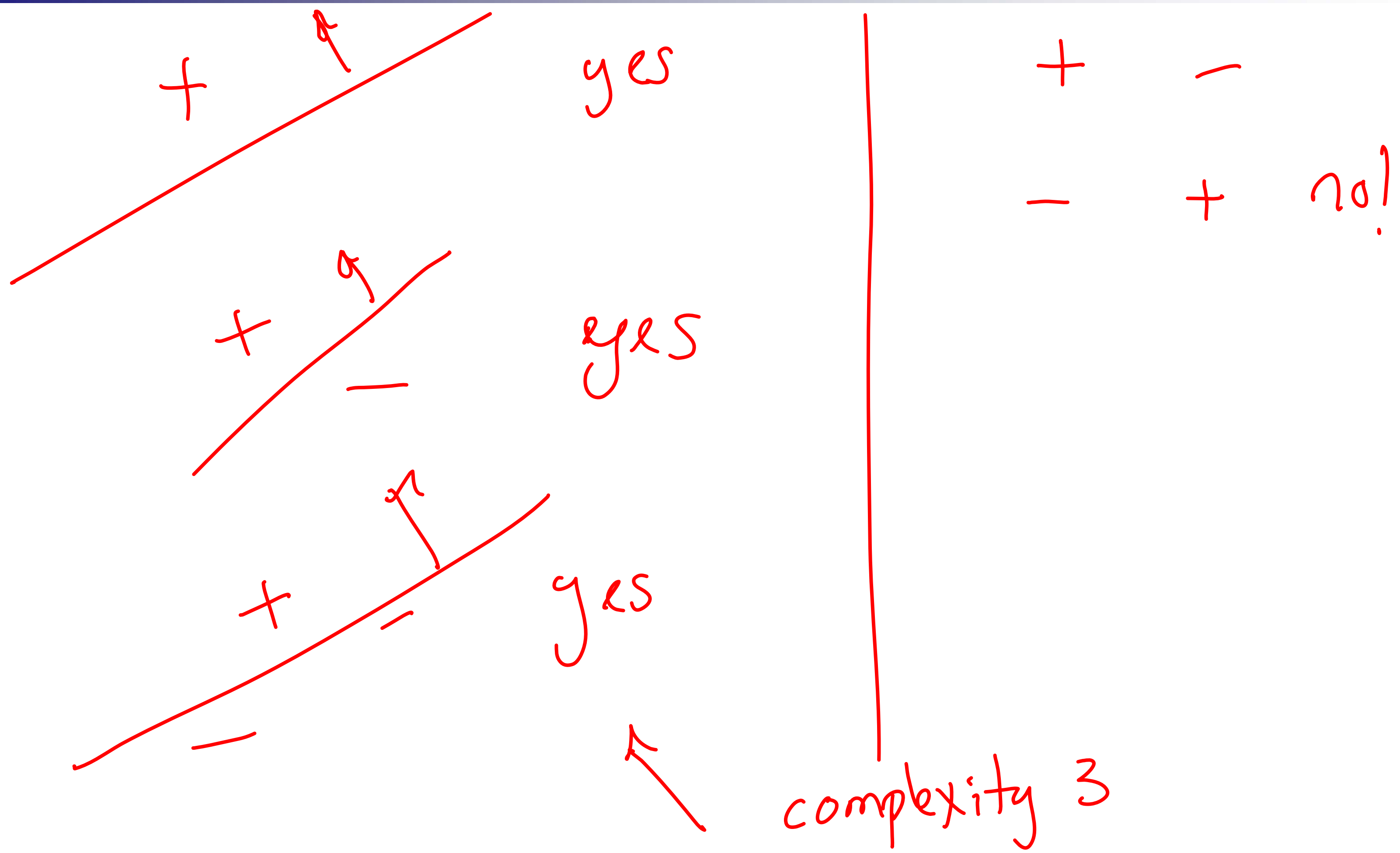
yes

complexity of linear classifier is 2

yes

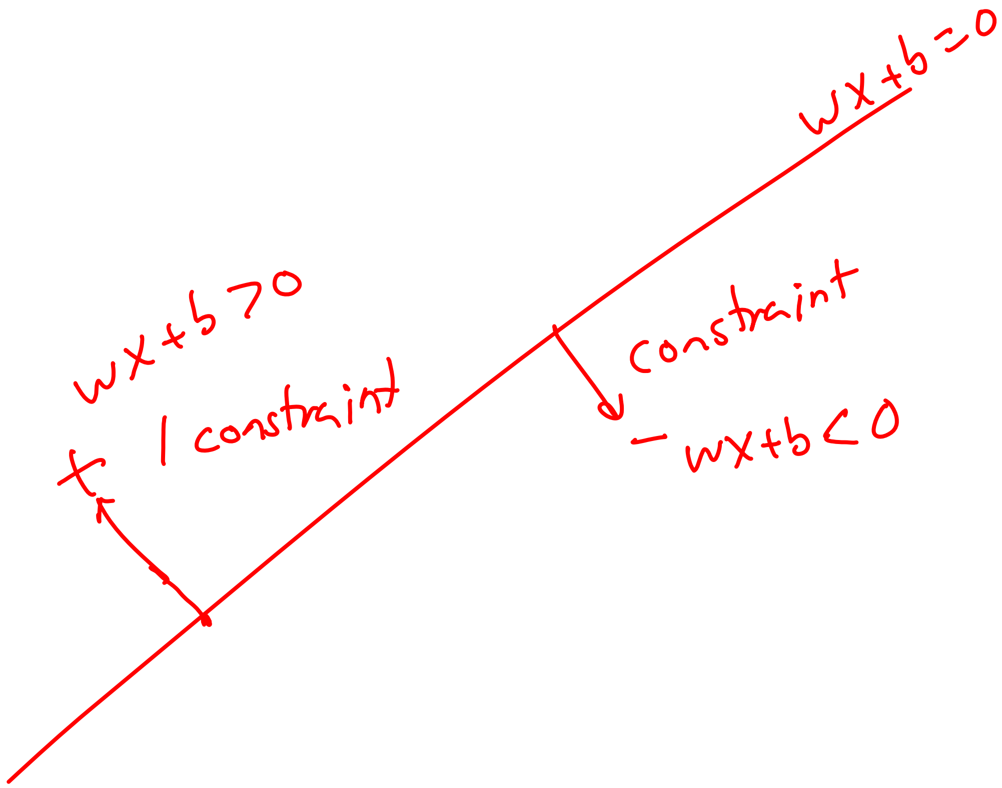
no!

How many points can a linear boundary classify exactly? (2-D)



How many points can a linear boundary classify exactly? (d-D)

$d+1$



$d+1$ variables

need $d+1$ constraints

$\Rightarrow$ $d+1$ points

PAC bound using VC dimension

- **Number of training points that can be classified exactly is VC dimension!!!**
  - **Measures relevant size of hypothesis space, as with decision trees with k leaves**

test error

$$\text{error}_{\mathcal{X}}(h) \leq \text{error}_D(h) + \sqrt{\frac{VC(H)\left(\ln \frac{2m}{VC(H)} + 1\right) + \ln \frac{4}{\delta}}{m}}$$

"bias"

"variance"

Game: picking "right" VC(H)

big VC(H) ↓ ↑

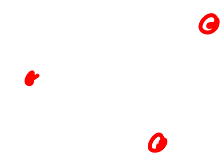
small VC(H) ↑ ↓

Shattering a set of points

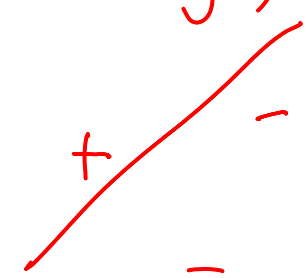
Definition: a **dichotomy** of a set S is a partition of S into two disjoint subsets.

Definition: 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.

you get pick
data



adversary, + and -:



question; classify exactly

VC dimension

Definition: 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 of X can be shattered by H, then $VC(H) \equiv \infty$.

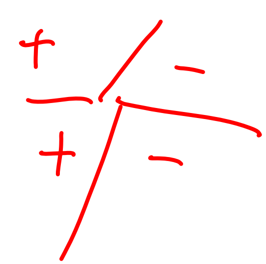
largest set that I can pick

Examples of VC dimension

$$\text{error}_{\mathcal{X}}(h) \leq \text{error}_D(h) + \sqrt{\frac{VC(H)\left(\ln \frac{2m}{VC(H)} + 1\right) + \ln \frac{4}{\delta}}{m}}$$

- Linear classifiers:
  - VC(H) = d+1, for *d* features plus constant term *b*
- Neural networks
  - VC(H) = #parameters
  - Local minima means NNs will probably not find best parameters
- 1-Nearest neighbor?

$VC(1\text{-NN}) = \infty!$



PAC bound for SVMs

- SVMs use a linear classifier
  - For *d* features, VC(H) = d+1:

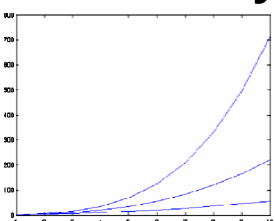
$$\text{error}_{\mathcal{X}}(h) \leq \text{error}_{D}(h) + \sqrt{\frac{(d+1)\left(\ln \frac{2m}{d+1} + 1\right) + \ln \frac{4}{\delta}}{m}}$$

VC dimension and SVMs: Problems!!!

Doesn't take margin into account

$$\text{error}_{\mathcal{X}}(h) \leq \text{error}_D(h) + \sqrt{\frac{(d+1)\left(\ln \frac{2m}{d+1} + 1\right) + \ln \frac{4}{\delta}}{m}}$$

- What about kernels?
  - Polynomials: num. features grows really fast = Bad bound

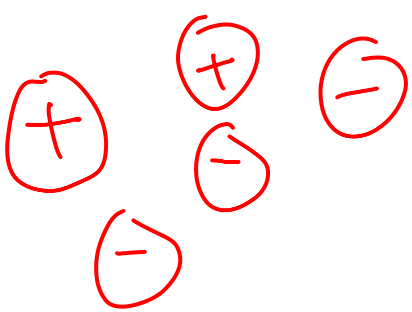


$$\text{num. terms} = \binom{p+n-1}{p} = \frac{(p+n-1)!}{p!(n-1)!}$$

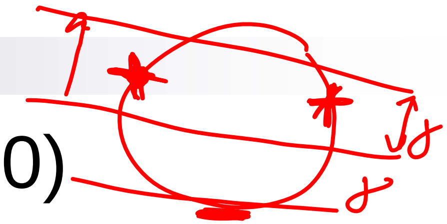
n – input features

p – degree of polynomial

  - Gaussian kernels can classify any set of points exactly

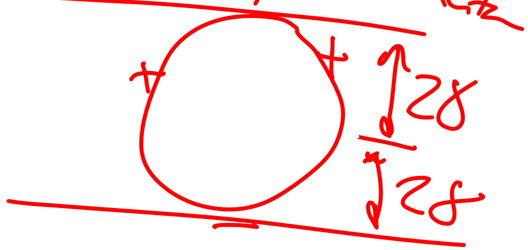


Margin-based VC dimension



- H: Class of linear classifiers: $\mathbf{w}.\Phi(\mathbf{x})$ (b=0)
  - Canonical form: $\min_j |\mathbf{w}.\Phi(\mathbf{x}_j)| = 1$
- VC(H) = R^2 $\mathbf{w}.\mathbf{w}$
  - Doesn't depend on number of features!!!
  - $R^2 = \max_j \Phi(\mathbf{x}_j).\Phi(\mathbf{x}_j)$ – magnitude of data
  - R^2 is bounded even for Gaussian kernels $\rightarrow$ bounded VC dimension

can't separate data



- Large margin, low $\mathbf{w}.\mathbf{w}$, low VC dimension – Very cool!

Applying margin VC to SVMs?

$$\text{error}_{\mathcal{X}}(h) \leq \text{error}_D(h) + \sqrt{\frac{VC(H)\left(\ln \frac{2m}{VC(H)} + 1\right) + \ln \frac{4}{\delta}}{m}}$$

- $VC(H) = R^2\ \mathbf{w}.\mathbf{w}$
  - $R^2 = \max_j \Phi(\mathbf{x}_j).\Phi(\mathbf{x}_j)$ – magnitude of data, doesn't depend on choice of **w**
- SVMs minimize **w.w**

- **SVMs minimize VC dimension to get best bound?**
- **Not quite right:** ☹
  - **Bound assumes VC dimension chosen before looking at data**
  - **Would require union bound over infinite number of possible VC dimensions…**
  - **But, it can be fixed!**

Structural risk minimization theorem

bias

$$\boxed{\text{error}_{\mathcal{X}}(h) \leq \text{error}_{D}^{\gamma}(h) + C\sqrt{\frac{\frac{R^2}{\gamma^2}\ln m + \ln\frac{1}{\delta}}{m}}}$$

as $\gamma \uparrow$

$\downarrow$ variance goes down

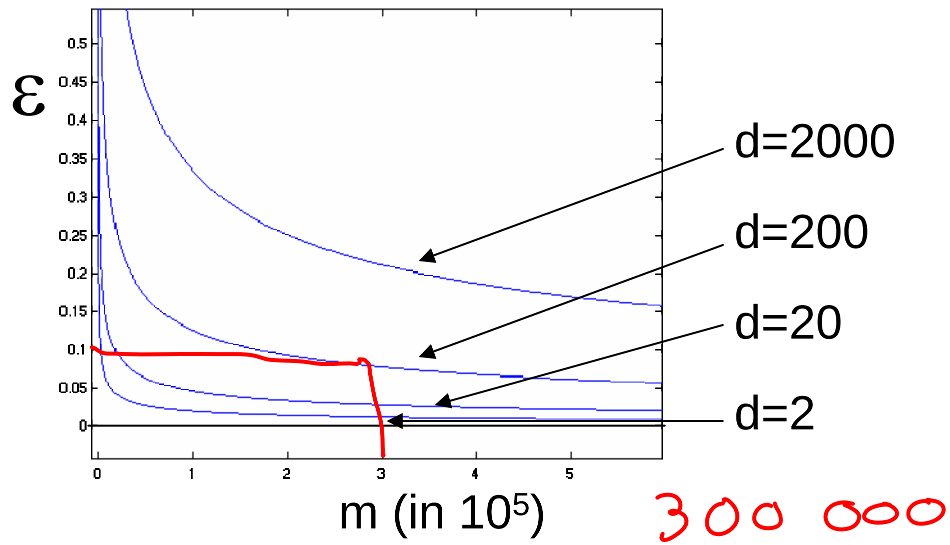
$$\text{error}_{D}^{\gamma}(h) = \text{num. points with margin} < \gamma \quad \Uparrow$$

more training errors

- For a family of hyperplanes with margin $\gamma > 0$
  - **w.w** ≤ 1
- SVMs maximize margin γ + hinge loss
  - Optimize tradeoff training error (bias) versus margin γ (variance)

Reality check – Bounds are loose

$$\text{error}_{\mathcal{X}}(h) \leq \text{error}_D(h) + \underbrace{\sqrt{\frac{(d+1)\left(\ln\frac{2m}{d+1}+1\right)+\ln\frac{4}{\delta}}{m}}}_{\epsilon}$$



- Bound can be very loose, why should you care?
  - There are tighter, albeit more complicated, bounds
  - Bounds gives us formal guarantees that empirical studies can't provide
  - Bounds give us intuition about complexity of problems and convergence rate of algorithms

What you need to know

- Finite hypothesis space
  - Derive results
  - Counting number of hypothesis
  - Mistakes on Training data
- Complexity of the classifier depends on number of points that can be classified exactly
  - Finite case – decision trees
  - Infinite case – VC dimension
- Bias-Variance tradeoff in learning theory
- Margin-based bound for SVM
- Remember: will your algorithm find best classifier?