Learning theory
Lecture 4

David Sontag
New York University

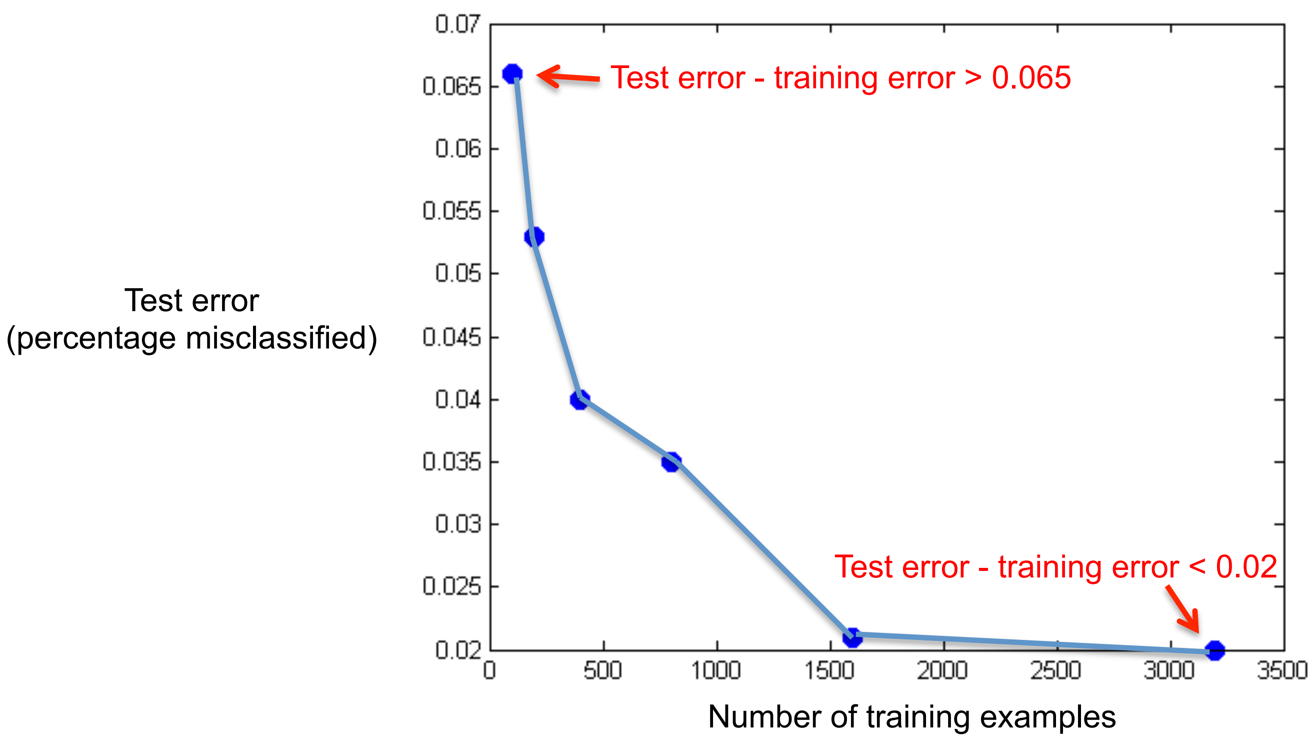
Slides adapted from Carlos Guestrin & Luke Zettlemoyer

What's next…

- We gave several machine learning algorithms:
  - Perceptron
  - Linear support vector machine (SVM)
  - SVM with kernels, e.g. polynomial or Gaussian
- How do we guarantee that the learned classifier will perform well on test data?
- How much training data do we need?

Example: Perceptron applied to spam classification

- In your homework 1, you trained a spam classifier using perceptron
  - **The training error was always zero**
  - With few data points, there was a big gap between training error and test error!



How much training data do you need?

- Depends on what *hypothesis class* the learning algorithm considers

- For example, consider a memorization-based learning algorithm
  - Input: training data S = { ($\mathbf{x}_i$, y_i) }
  - Output: function f($\mathbf{x}$) which, if there exists ($\mathbf{x}_i$, y_i) in S such that $\mathbf{x}=\mathbf{x}_i$, predicts y_i, and otherwise predicts the majority label
  - This learning algorithm will always obtain zero training error
  - But, it will take a ***huge*** amount of training data to obtain small test error (i.e., its generalization performance is horrible)

- Linear classifiers are powerful precisely because of their simplicity
  - Generalization is easy to guarantee

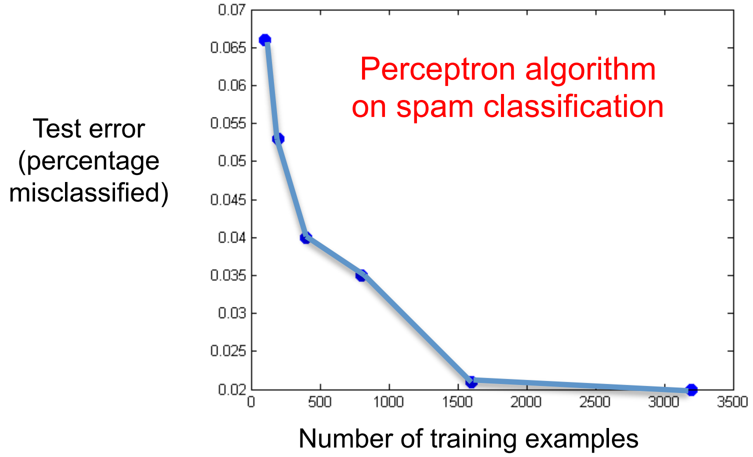
Roadmap of lecture

1. Generalization of finite hypothesis spaces

2. VC-dimension

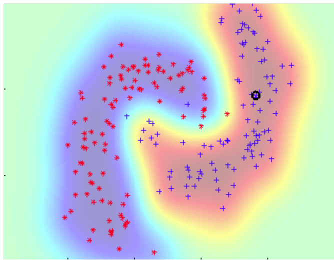
   - Will show that linear classifiers need to see approximately **d** training points, where **d** is the dimension of the feature vectors

   - Explains the good performance we obtained using perceptron!!!! (we had a few thousand features)



3. Margin based generalization

   - Applies to **infinite** dimensional feature vectors (e.g., Gaussian kernel)

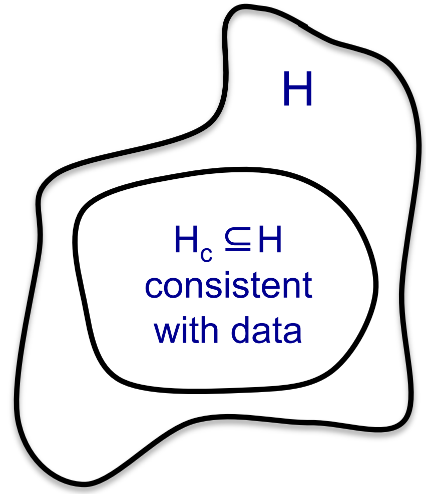


[Figure from Cynthia Rudin]

How big should your validation set be?

- In PS1, you tried many configurations of your algorithms (avg vs. regular perceptron, max # of iterations) and chose the one that had smallest validation error
- Suppose in total you tested **|H|=**40 different classifiers on the validation set of **m** held-out e-mails
- The best classifier obtains 98% accuracy on these **m** e-mails!!!
- But, what is the true classification accuracy?
- How large does **m** need to be so that we can guarantee that the best configuration (measured on validate) is truly good?

A simple setting…

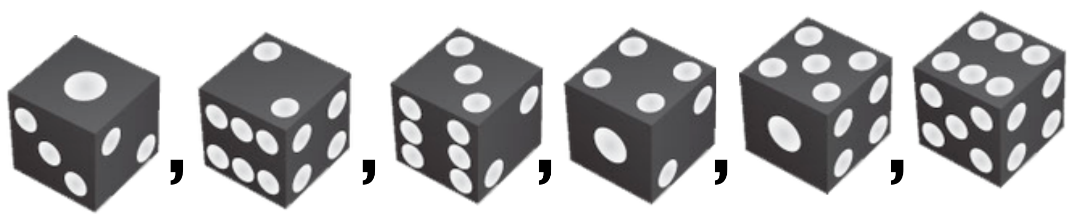


- Classification
  - m data points
  - **Finite** number of possible hypothesis (e.g., 40 spam classifiers)
- A learner finds a hypothesis *h* that is **consistent** with training data
  - Gets zero error in training: $error_{train}(h) = 0$
  - I.e., assume for now that one of the classifiers gets 100% accuracy on the **m** e-mails (we'll handle the 98% case afterward)
- What is the probability that *h* has more than ε **true** error?
  - $error_{true}(h) \geq \varepsilon$

Refresher on probability: outcomes

- An **outcome space** specifies the possible outcomes that we would like to reason about, e.g.

Ω = {  ,  } Coin toss

Ω = {  } Die toss

- We specify a **probability** p(**x**) for each outcome **x** such that

$$p(x) \geq 0, \qquad \sum_{x \in \Omega} p(x) = 1$$

E.g., p() = .6

p() = .4

Refresher on probability: events

- An **event** is a subset of the outcome space, e.g.

E = { , ,  } Even die tosses

O = { , ,  } Odd die tosses

- The **probability** of an event is given by the sum of the probabilities of the outcomes it contains,

$$p(E) = \sum_{x \in E} p(x)$$

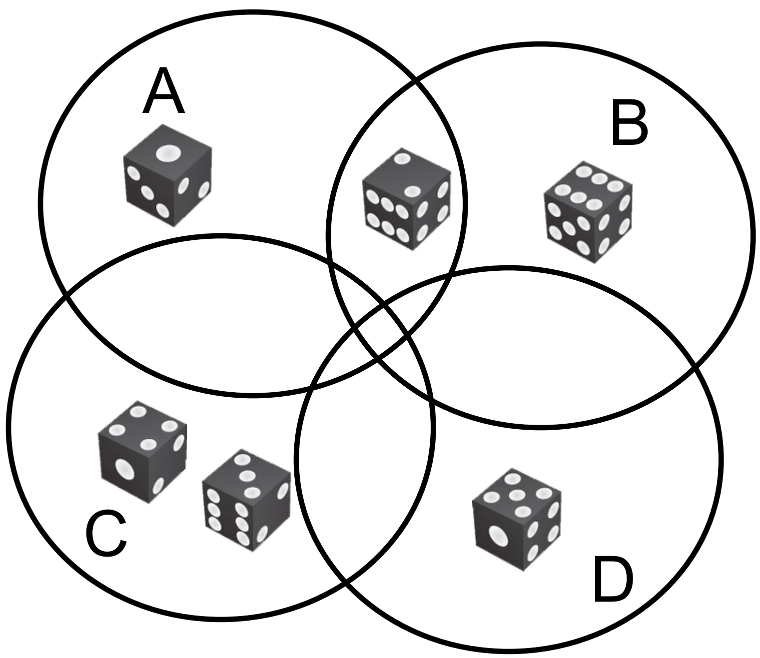
E.g., p(E) = p() + p() + p()

= 1/2, if fair die

Refresher on probability: union bound

- P(A or B or C or D or …)

  ≤ P(A) + P(B) + P(C) + P(D) + …



$$p(A \cup B) = p(A) + p(B) - p(A \cap B)$$
$$\leq p(A) + p(B)$$

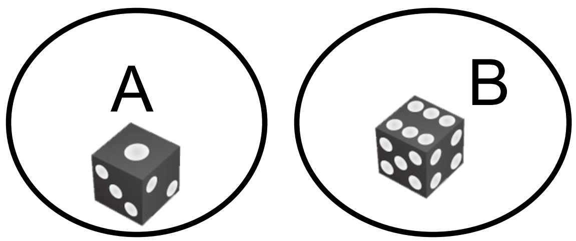
Q: When is this a tight bound?

A: For disjoint events
(i.e., non-overlapping circles)

Refresher on probability: independence

- Two events A and B are **independent** if

$$p(A \cap B) = p(A)p(B)$$



Are these events independent?

No! $p(A \cap B) = 0$

$$p(A)p(B) = \left(\frac{1}{6}\right)^2$$

Refresher on probability: independence

- Two events A and B are **independent** if

$$p(A \cap B) = p(A)p(B)$$

Analogy: outcome space defines all possible sequences of e-mails in training set

- Suppose our outcome space had two different die:

$\Omega = \{$  $\}$ 2 die tosses

$6^2 = 36$ outcomes

and the probability of each outcome is defined as

p() = $a_1\ b_1$ p() = $a_1\ b_2$ $\cdots$

a_1	a_2	a_3	a_4	a_5	a_6
.1	.12	.18	.2	.1	.3

$$\sum_{i=1}^{6} a_i = 1$$

b_1	b_2	b_3	b_4	b_5	b_6
.19	.11	.1	.22	.18	.2

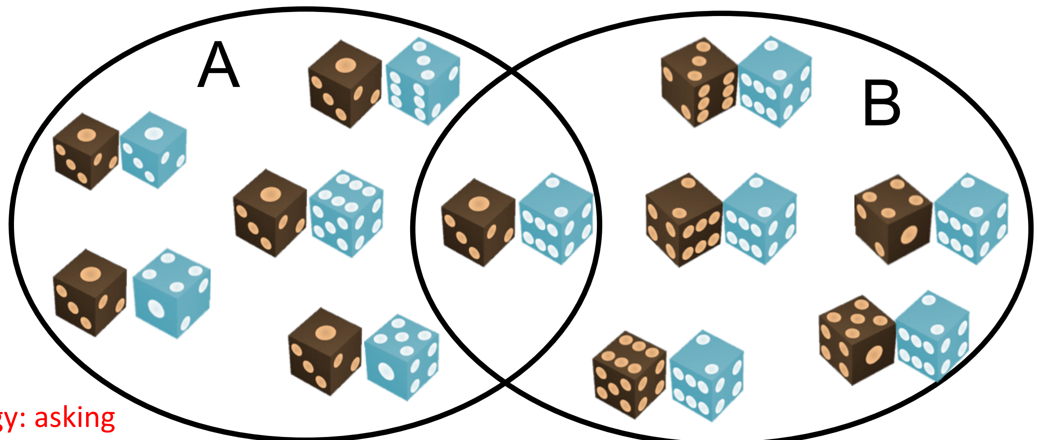
$$\sum_{j=1}^{6} b_j = 1$$

Refresher on probability: independence

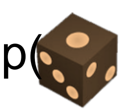
- Two events A and B are **independent** if

$$p(A \cap B) = p(A)p(B)$$

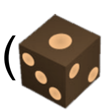
- Are these events independent?



Yes! $p(A \cap B) =$ p()

$p(A)p(B) =$ p() p()

Analogy: asking about first e-mail in training set

p(A) = p()

$$= \sum_{j=1}^{6} a_1 b_j = a_1 \sum_{j=1}^{6} b_j \;= a_1$$

p(B) = p() $= b_2$

Analogy: asking about second e-mail in training set

Refresher of probability: discrete random variables

- A **random variable** X is a mapping $X : \Omega \to D$
  - D is some set (e.g., the integers)
  - Induces a partition of all outcomes Ω
- For some $x \in D$, we say

$$p(X = x) = p(\{\omega \in \Omega : X(\omega) = x\})$$

"probability that variable X assumes state x"

- Notation: $\text{Val}(X)$ = set D of all values assumed by X
  (will interchangeably call these the "values" or "states" of variable X)

$\Omega = \{$  $\}$ 2 die tosses

Refresher of probability: discrete random variables

- $p(X)$ is a distribution: $\sum_{x \in \mathrm{Val}(X)} p(X = x) = 1$
- E.g. X_1 may refer to the value of the first dice, and X_2 to the value of the second dice
- We call two random variables X and Y *identically distributed* if Val(X) = Val(Y) and p(X=s) = p(Y=s) for all s in Val(X)

p() = $a_1\ b_1$ p() = $a_1\ b_2$ …

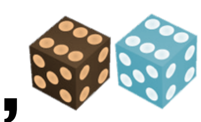
X₁ and X₂ NOT identically distributed

a_1	a_2	a_3	a_4	a_5	a_6
.1	.12	.18	.2	.1	.3

$$\sum_{i=1}^{6} a_i = 1$$

b_1	b_2	b_3	b_4	b_5	b_6
.19	.11	.1	.22	.18	.2

$$\sum_{j=1}^{6} b_j = 1$$

Ω = { , , , …,  }   2 die tosses

Refresher of probability: discrete random variables

- $p(X)$ is a distribution: $\sum_{x \in \mathrm{Val}(X)} p(X = x) = 1$
- E.g. X_1 may refer to the value of the first dice, and X_2 to the value of the second dice
- We call two random variables X and Y *identically distributed* if Val(X) = Val(Y) and p(X=s) = p(Y=s) for all s in Val(X)

p() = $a_1\, a_1$ p() = $a_1\, a_2$ …

X_1 and X_2 identically distributed

a_1	a_2	a_3	a_4	a_5	a_6
.1	.12	.18	.2	.1	.3

$$\sum_{i=1}^{6} a_i = 1$$

$\Omega = \{$  $\}$ 2 die tosses

Refresher of probability: discrete random variables

- X=x is simply an event, so can apply union bound, etc.
- Two random variables **X** and **Y** are **independent** if:

$$p(X = x, Y = y) = p(X = x)p(Y = y) \quad \forall x \in \text{Val}(X), y \in \text{Val}(Y)$$

↑

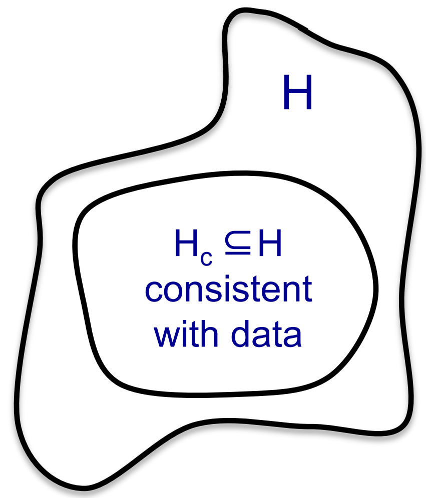
Joint probability. Formally, given by the event $X = x \cap Y = y$

- The **expectation** of **X** is defined as: $E[X] = \sum_{x \in \text{Val}(X)} p(X = x)x$

- If **X** is binary valued, i.e. x is either 0 or 1, then:

$$\begin{aligned} E[X] &= p(X = 0) \cdot 0 + p(X = 1) \cdot 1 \\ &= p(X = 1) \end{aligned}$$

A simple setting…



- Classification
  - m data points
  - **Finite** number of possible hypothesis (e.g., 40 spam classifiers)
- A learner finds a hypothesis *h* that is **consistent** with training data
  - Gets zero error in training: $error_{train}(h) = 0$
  - I.e., assume for now that one of the classifiers gets 100% accuracy on the **m** e-mails (we'll handle the 98% case afterward)
- What is the probability that *h* has more than ε **true** error?
  - $error_{true}(h) \geq \varepsilon$

How likely is a **single** hypothesis to get *m* data points right?

- The probability of a hypothesis *h* incorrectly classifying: $\epsilon_h = \sum_{(\vec{x},y)} p(\vec{x},y) 1[h(\vec{x}) \neq y]$
- Let Z_i^h be a random variable that takes two values: **1 if h correctly classifies** i^th^ data point, and 0 otherwise
- The Z^h variables are **independent** and **identically distributed** (i.i.d.) with

$$\Pr(Z_i^h = 0) = \sum_{(\vec{x},y)} p(\vec{x},y) 1[h(\vec{x}) \neq y] = \epsilon_h$$

- What is the probability that *h* classifies *m* data points correctly?

Pr(h gets m *iid* data points right) $= (1 - \epsilon_h)^m \leq e^{-\epsilon_h m}$

Are we done?

$\Pr(\text{h gets m } \textit{iid} \text{ data points right} \mid \text{error}_{true}(h) \geq \varepsilon) \leq e^{-\varepsilon m}$

- Says "if h gets m data points correct, then with very high probability (i.e. $1-e^{-\varepsilon m}$) it is close to perfect (i.e., will have error $\leq \varepsilon$)"
- This only considers **one** hypothesis!
- Suppose 1 billion classifiers were tried, and each was a *random* function
- For **m** small enough, one of the functions will classify all points correctly – but all have very large true error

How likely is learner to pick a bad hypothesis?

$\Pr(\text{h gets m } \textit{iid} \text{ data points right} \mid error_{true}(h) \geq \varepsilon) \leq e^{-\varepsilon m}$

Suppose there are $|H_c|$ hypotheses consistent with the training data

- How likely is learner to pick a bad one, i.e. with *true* error $\geq \varepsilon$?
- We need a bound that holds for all of them!

$$P(error_{true}(h_1) \geq \varepsilon \text{ OR } error_{true}(h_2) \geq \varepsilon \text{ OR } \ldots \text{ OR } error_{true}(h_{|H_c|}) \geq \varepsilon)$$

$$\leq \sum_k P(error_{true}(h_k) \geq \varepsilon) \quad \leftarrow \text{Union bound}$$

$$\leq \sum_k (1-\varepsilon)^m \quad \leftarrow \text{bound on individual } h_j\text{s}$$

$$\leq |H|(1-\varepsilon)^m \quad \leftarrow |H_c| \leq |H|$$

$$\leq |H|\, e^{-m\varepsilon} \quad \leftarrow (1-\varepsilon) \leq e^{-\varepsilon} \text{ for } 0 \leq \varepsilon \leq 1$$

Generalization error of finite hypothesis spaces [Haussler '88]

We just proved the following result:

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

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

Using a PAC bound

Argument: Since for all *h* we know that

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

… with probability 1-δ the following holds… (either case 1 or case 2)

Typically, 2 use cases:

- 1: Pick ε and δ, compute *m*
- 2: Pick m and δ, compute ε

$$p(\text{error}_{true}(h) \geq \epsilon) \leq |H|e^{-m\epsilon} \leq \delta$$

Says: we are willing to tolerate a δ probability of having ≥ ε error

$$\ln\left(|H|e^{-m\epsilon}\right) \leq \ln \delta$$

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

$\epsilon = \delta = .01, |H| = 40$

Need $m \geq 830$

Case 1

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

Case 2

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

Log dependence on |H|, OK if exponential size (but not doubly)

ε has stronger influence than δ

ε shrinks at rate O(1/m)

Limitations of Haussler '88 bound

- There may be no consistent hypothesis h (where $error_{train}(h)$=0)

- Size of hypothesis space
  - What if |H| is really big?
  - What if it is continuous?

- First Goal: Can we get a bound for a learner with $error_{train}(h)$ in the data set?

Question: What's the expected error of a hypothesis?

- The probability of a hypothesis incorrectly classifying: $\sum_{(\vec{x},y)} p(\vec{x},y) 1[h(\vec{x}) \neq y]$
- Let's now let Z_i^h be a random variable that takes two values, 1 if h correctly classifies i^{th} data point, and 0 otherwise
- The Z variables are **independent** and **identically distributed** (i.i.d.) with

$$\Pr(Z_i^h = 0) = \sum_{(\vec{x},y)} p(\vec{x},y) 1[h(\vec{x}) \neq y]$$

- Estimating the true error probability is like estimating the parameter of a coin!
- **Chernoff bound**: for *m* i.i.d. coin flips, $X_1,\ldots,X_m$, where $X_i \in \{0,1\}$. For $0<\varepsilon<1$:

$$p(X_i = 1) = \theta$$

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

True error probability

Observed fraction of points incorrectly classified

$$E[\frac{1}{m}\sum_{i=1}^m X_i] = \frac{1}{m}\sum_{i=1}^m E[X_i] = \theta$$

(by linearity of expectation)

Generalization bound for |H| hypothesis

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

$$\Pr(\text{error}_{true}(h) - \text{error}_D(h) > \epsilon) \leq |H| e^{-2m\epsilon^2}$$

Why? Same reasoning as before. Use the Union bound over individual Chernoff bounds

PAC bound and Bias-Variance tradeoff

for all h, with probability at least 1-δ:

$$\text{error}_{true}(h) \leq \underbrace{\text{error}_D(h)}_{\text{"bias"}} + \underbrace{\sqrt{\frac{\ln|H| + \ln\frac{1}{\delta}}{2m}}}_{\text{"variance"}}$$

- For large |H|
  - low bias (assuming we can find a good h)
  - high variance (because bound is looser)
- For small |H|
  - high bias (is there a good h?)
  - low variance (tighter bound)

What about continuous hypothesis spaces?

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

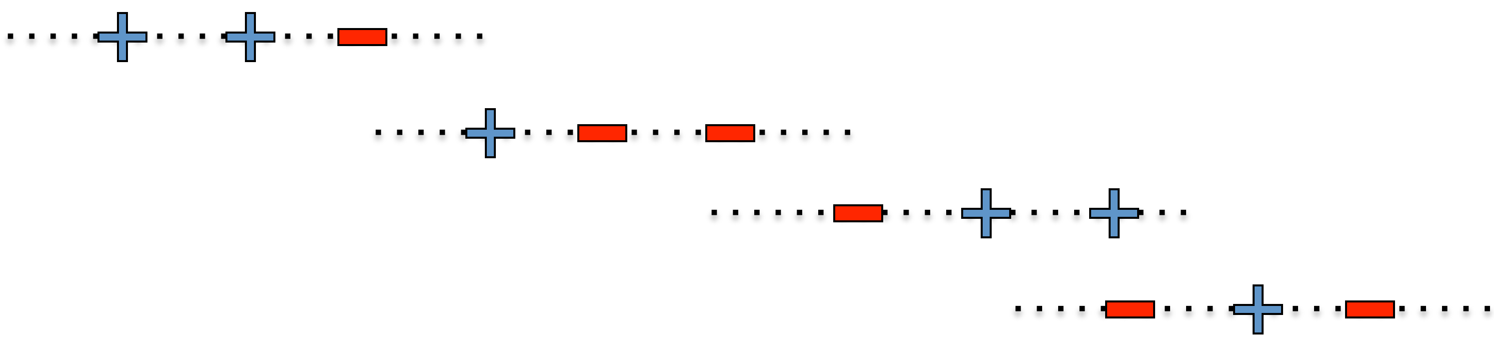
- Continuous hypothesis space:
  - |H| = ∞
  - Infinite variance???

- **Only care about the maximum number of points that can be classified exactly!**

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

2 Points: Yes!!

3 Points: No…



etc (8 total)

Shattering and Vapnik–Chervonenkis Dimension

A **set of points** is *shattered* by a hypothesis space H iff:

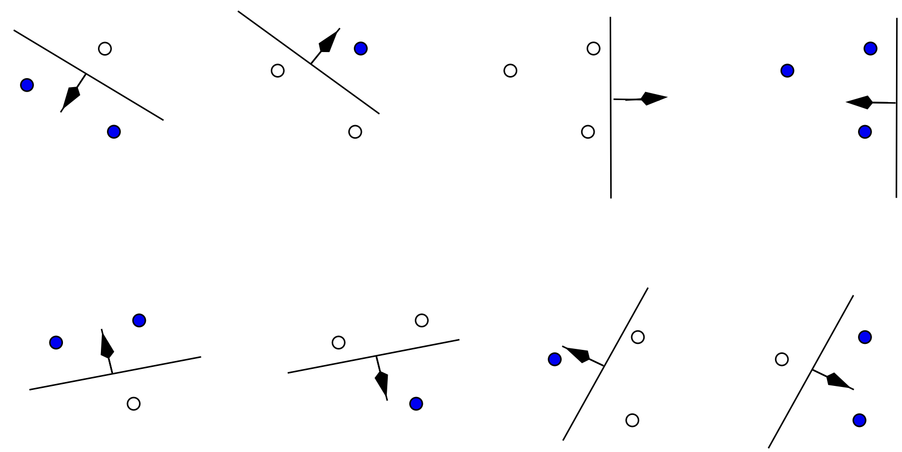
- For all ways of *splitting* the examples into positive and negative subsets
- There exists some *consistent* hypothesis h

The *VC Dimension* of H over input space X

- The size of the *largest* finite subset of X shattered by H

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

3 Points: Yes!!



4 Points: No…

etc.

[Figure from Chris Burges]

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

- A linear classifier $\sum_{j=1..d} w_j x_j + b$ can represent all assignments of possible labels to d+1 points
  - But not d+2!!
  - Thus, VC-dimension of d-dimensional linear classifiers is d+1
  - Bias term b required
  - Rule of Thumb: number of parameters in model often matches max number of points
- Question: Can we get a bound for error as a function of the number of points that can be completely labeled?

PAC bound using VC dimension

- VC dimension: number of training points that can be classified exactly (shattered) by hypothesis space H!!!
  - Measures relevant size of hypothesis space

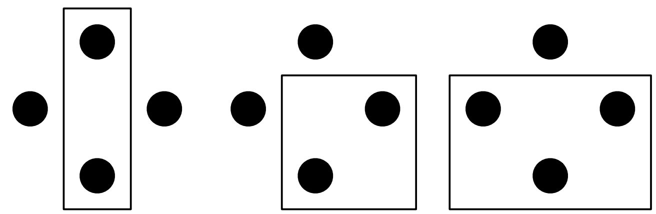
$$\text{error}_{true}(h) \leq \text{error}_{train}(h) + \sqrt{\frac{VC(H)\left(\ln \frac{2m}{VC(H)} + 1\right) + \ln \frac{4}{\delta}}{m}}$$

- Same bias / variance tradeoff as always
  - Now, just a function of VC(H)

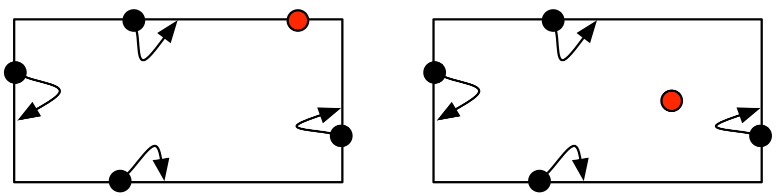
- Note: all of this theory is for **binary** classification
  - Can be generalized to multi-class and also regression

What is the VC-dimension of rectangle classifiers?

- First, show that there are 4 points that *can* be shattered:



- Then, show that no set of 5 points can be shattered:



[Figures from Anand Bhaskar, Ilya Sukhar]

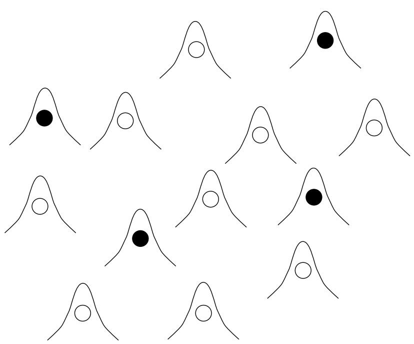
Generalization bounds using VC dimension

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

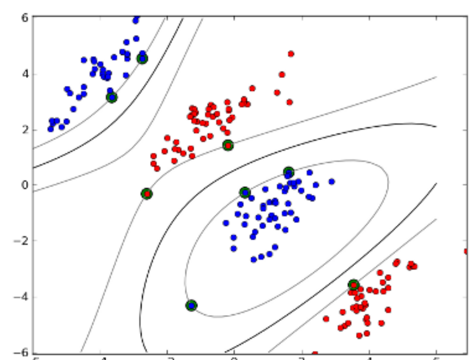
- Linear classifiers:
  - VC(H) = d+1, for *d* features plus constant term *b*
- Classifiers using Gaussian Kernel
  - VC(H) = ∞

$$K(\vec{u}, \vec{v}) = \exp\left(-\frac{||\vec{u} - \vec{v}||_2^2}{2\sigma^2}\right)$$

← Euclidean distance, squared



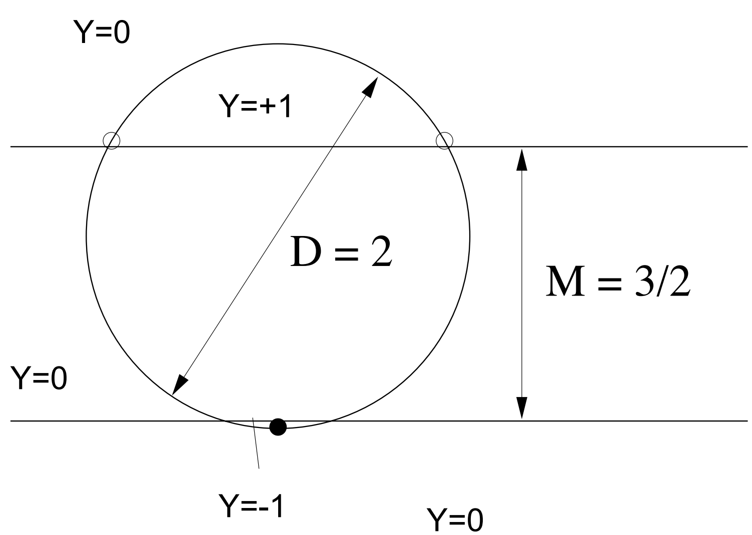
[Figure from Chris Burges]



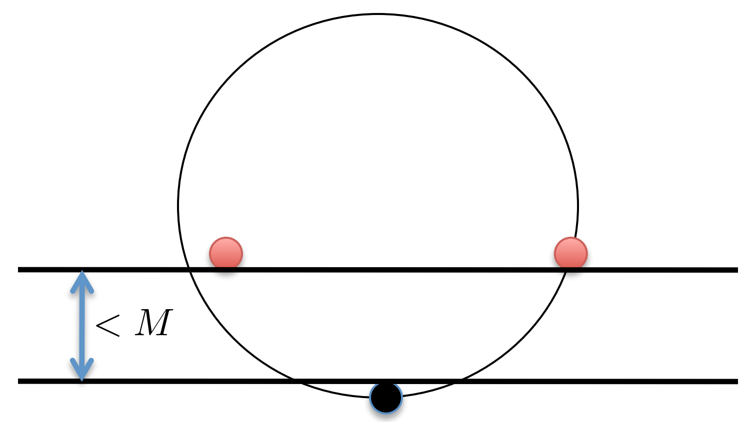
[Figure from mblondel.org]

Gap tolerant classifiers

- Suppose data lies in R^d in a ball of diameter **D**
- Consider a hypothesis class H of linear classifiers that can only classify point sets with margin at least **M**
- What is the largest set of points that H can shatter?



Cannot shatter these points:



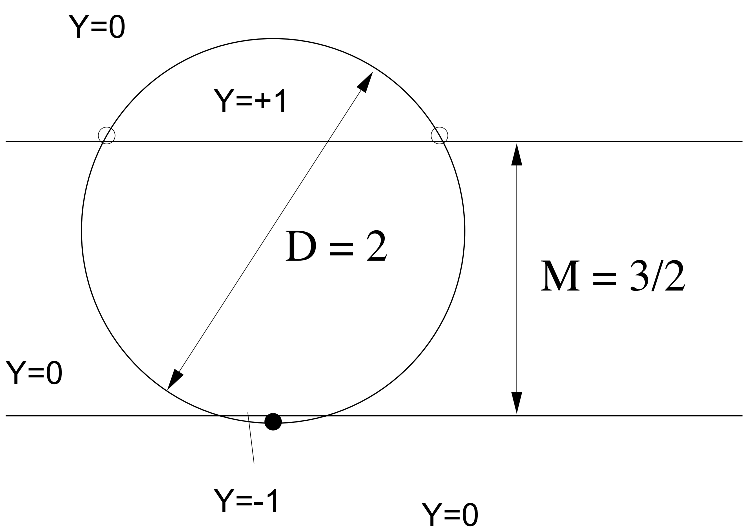
VC dimension = $\min\left(d, \frac{D^2}{M^2}\right)$

$M = 2\gamma = 2\frac{1}{||w||}$ → **SVM attempts to *minimize* ||w||², which *minimizes* VC-dimension!!!**

[Figure from Chris Burges]

Gap tolerant classifiers

- Suppose data lies in R^d in a ball of diameter **D**
- Consider a hypothesis class H of linear classifiers that can only classify point sets with margin at least **M**
- What is the largest set of points that H can shatter?



VC dimension = $\min\left(d, \frac{D^2}{M^2}\right)$

[Figure from Chris Burges]

$$K(\vec{u}, \vec{v}) = \exp\left(-\frac{||\vec{u}-\vec{v}||_2^2}{2\sigma^2}\right)$$

What is R=D/2 for the Gaussian kernel?

$$\begin{aligned} R &= \max_x ||\phi(x)|| \\ &= \max_x \sqrt{\phi(x)\cdot\phi(x)} \\ &= \max_x \sqrt{K(x,x)} \\ &= 1 \quad \text{!!!} \end{aligned}$$

What is $||w||^2$?

$$||w||^2 = \left(\frac{2}{M}\right)^2$$

$$\begin{aligned} ||w||^2 &= ||\sum_i \alpha_i y_i \phi(x_i)||_2^2 \\ &= \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i, x_j) \end{aligned}$$

What you need to know

- Finite hypothesis space
  - Derive results
  - Counting number of hypothesis
- Complexity of the classifier depends on number of points that can be classified exactly
  - Finite case – number of hypotheses considered
  - Infinite case – VC dimension
  - VC dimension of gap tolerant classifiers to justify SVM
- Bias-Variance tradeoff in learning theory