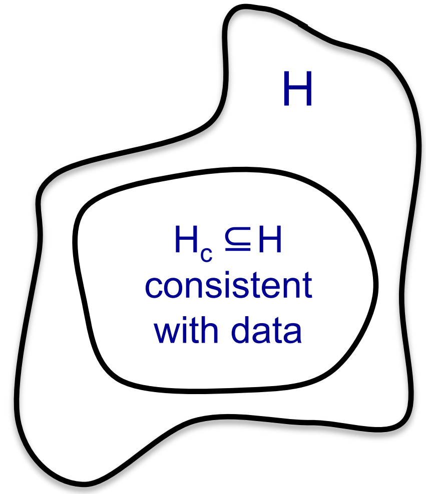
Learning theory
Lecture 9

David Sontag
New York University

Slides adapted from Carlos Guestrin & Luke Zettlemoyer

A simple setting…



- Classification
  - m data points
  - **Finite** number of possible hypothesis (e.g., 20,000 face recognition 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 the winner gets 100% accuracy on the **m** labeled images (we'll handle the 98% case afterward)
- What is the probability that *h* has more than ε **true** error?
  - $error_{true}(h) \geq \varepsilon$

Using a PAC bound

Typically, 2 use cases:

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

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)

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

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

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

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

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

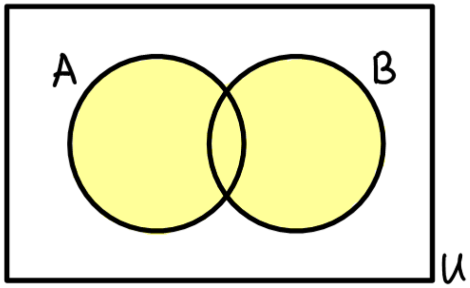
- There may be no consistent hypothesis h (where $error_{train}(h)=0$)

- 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 training set?

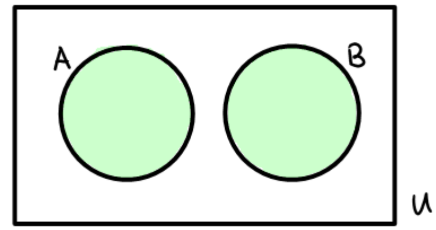
Introduction to probability (continued)

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$



Mutually Exclusive

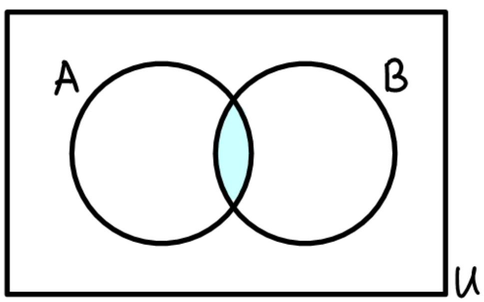
$$P(A \cap B) = 0$$
$$P(A \cup B) = P(A) + P(B)$$



U = outcome space
A,B events

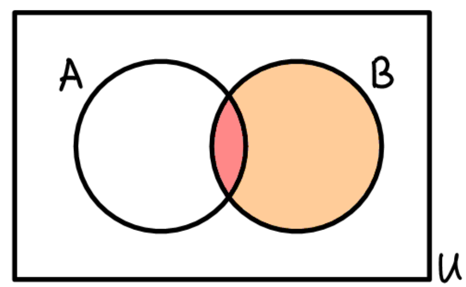
Independence

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



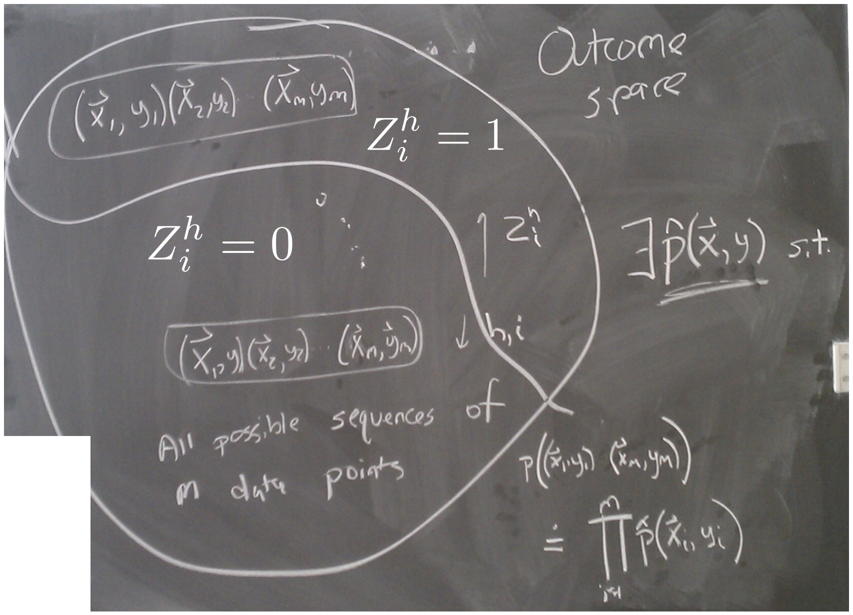
Conditional Probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$



[Figures from http://ibscrewed4maths.blogspot.com/]

Introduction to probability (continued)



Z_i^h = Event that *h* correctly classifies the *i*'th data point

$$= \{(\vec{x}_1, y_1) \dots (\vec{x}_m, y_m) : h(\vec{x}_i) = y_i\}$$

$$p(Z_i^h) = \sum_{(\vec{x}_1,y_1)\dots(\vec{x}_m,y_m)\in Z_i^h} p((\vec{x}_1, y_1)\dots(\vec{x}_m, y_m))$$

$$= \sum_{(\vec{x}_1,y_1)\dots(\vec{x}_m,y_m)} \left(\prod_{j=1}^{m} \hat{p}(\vec{x}_j, y_j)\right) 1[h(\vec{x}_i) = y_i]$$

$$= \sum_{\vec{x}_i, y_i} \hat{p}(\vec{x}_i, y_i) 1[h(\vec{x}_i) = y_i]$$

$$= \sum_{\vec{x}, y} \hat{p}(\vec{x}, y) 1[h(\vec{x}) = y]$$

A **random variable** X is a partition of the outcome space

- Each disjoint set of outcomes is given a label

$$\Pr(Z_i^h = 1) = \Pr(\{(\vec{x}_1, y_1)\dots(\vec{x}_m, y_m) : h(\vec{x}_i) = y_i\})$$
$$\Pr(Z_i^h = 0) = \Pr(\{(\vec{x}_1, y_1)\dots(\vec{x}_m, y_m) : h(\vec{x}_i) \neq y_i\})$$

← **Discrete** random variable

"Probability that variable X assumes state x"

Introduction to probability (continued)

Notation: Val(X) = set D of all values assumed by variable X

p(X) specifies a distribution: $p(X = x) \geq 0 \;\; \forall x \in \text{Val}(X)$

$$\sum_{x \in \text{Val}(X)} p(X = x) = 1$$

X=x is simply an event, so can apply union bound, conditioning, etc.

Two random variables **X** and **Y** are **independent** if:

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

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

For example, $E[Z_i^h] = \sum_{z \in \{0,1\}} p(Z_i^h = z)z \;\; = p(Z_i^h = 1)$

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

- The probability of a hypothesis incorrectly classifying: $\sum_{(\vec{x},y)} \hat{p}(\vec{x},y)1[h(\vec{x}) \neq y]$

- We showed that the Z_i^h random variables are **independent** and **identically distributed** (i.i.d.) with $\Pr(Z_i^h = 0) = \sum_{(\vec{x},y)} \hat{p}(\vec{x},y)1[h(\vec{x}) \neq y]$

- 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*:

$$P\left(\text{error}_{true}(h) - \text{error}_{train}(h) > \epsilon\right) \leq |H|e^{-2m\epsilon^2}$$

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:

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

- 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)

PAC bound: How much data?

$$P\left(\text{error}_{true}(h) - \text{error}_{train}(h) > \epsilon\right) \leq |H|e^{-2m\epsilon^2}$$

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

- Given δ,ε how big should m be?

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

Returning to our example…

- Suppose Facebook holds a competition for the best face recognition classifier (+1 if image contains a face, -1 if it doesn't)
- All recent worldwide graduates of machine learning and computer vision classes decide to compete
- Facebook gets back 20,000 face recognition algorithms
- They evaluate all 20,000 algorithms on **m** labeled images (not previously shown to the competitors) and chooses a winner
- The winner obtains 98% accuracy on these **m** images!!!
- Facebook already has a face recognition algorithm that is known to be 95% accurate
  - Should they deploy the winner's algorithm instead?
  - Can't risk doing worse… would be a public relations disaster!

[Fictional example]

Returning to our example…

|H|=20,000 competitors

$$\text{error}_{true}(\text{facebook}) = .05$$

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

=.02 error on the m images

m = 100 images

Suppose δ=0.01 and m=100: $.02 + \sqrt{\frac{\ln(20,000) + \ln(100)}{200}} \approx .29$

Suppose δ=0.01 and m=10,000: $.02 + \sqrt{\frac{\ln(20,000) + \ln(100)}{20,000}} \approx .047$

So, with only ~100 test images, confidence interval too large! Do not deploy!

But, if the competitor's error is still .02 on m>10,000 images, then we can say that it is truly better with probability at least 99/100

What about continuous hypothesis spaces?

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

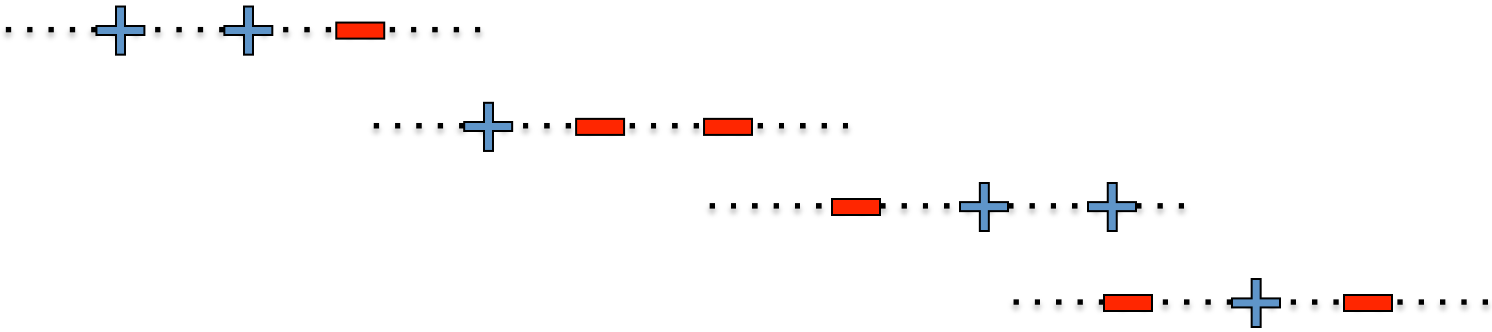
- Continuous hypothesis space:
  - $|H| = \infty$
  - Infinite variance???

- **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:

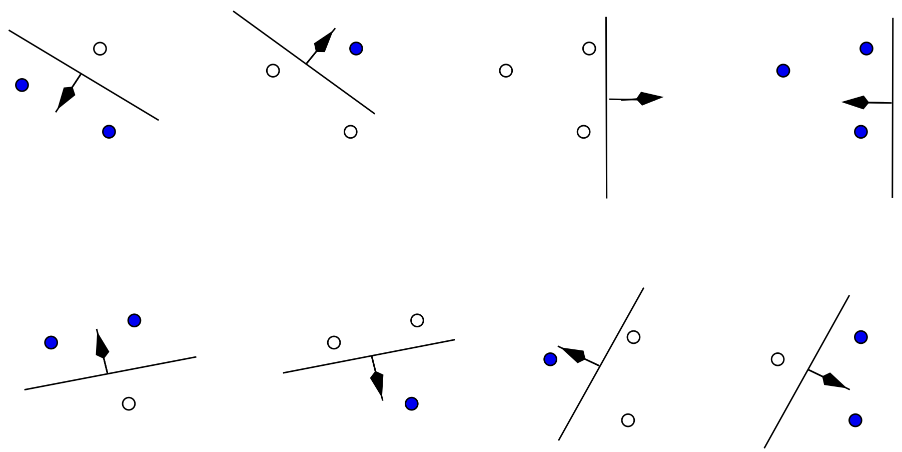
- For all ways of *splitting* the examples into positive and negative subsets
- There exists some *consistent* hypothesis h

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 $w_0+\sum_{j=1..d}w_jx_j$ 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 w_0 required
  - Rule of Thumb: number of parameters in model often matches max number of points

- Question: Can we get a bound for error in 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

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

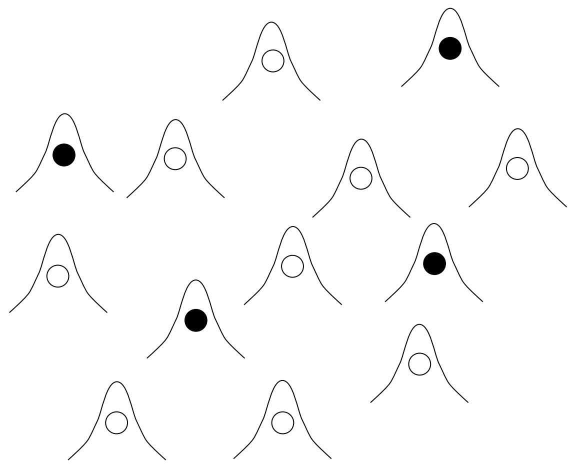
- Same bias / variance tradeoff as always
  - Now, just a function of VC(H)

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

Examples of 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*
- SVM with Gaussian Kernel
  - VC(H) = ∞



[Figure from Chris Burges]

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 – number of hypotheses considered
  - Infinite case – VC dimension
- Bias-Variance tradeoff in learning theory