PAC-learning, VC Dimension, cont.

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

October 24th, 2007

A simple setting…

- Classification
  - m data points
  - **Finite** number of possible hypothesis (e.g., dec. trees of depth d)
- A learner finds a hypothesis h that is **consistent** with training data
  - Gets zero error in training – $\text{error}_{train}(h) = 0$
- What is the probability that h has more than ε true error?
  - $\text{error}_{true}(h) \geq \varepsilon$

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

- Hypothesis *h* that is **consistent** with training data → got *m* i.i.d. points right
  - h "bad" if it gets all this data right, but has high true error
- Prob. *h* with $error_{true}(h) \geq \varepsilon$ gets one data point right

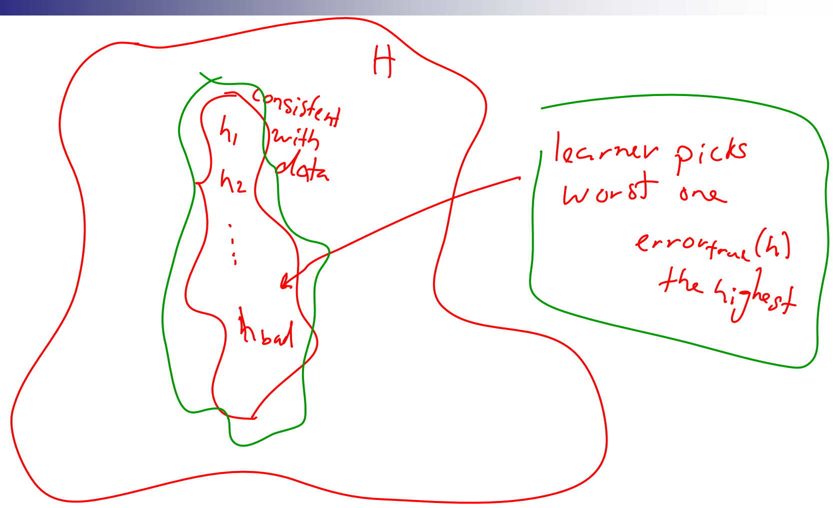
$P(h_{bad} \text{ gets one point right}) \leq 1-\varepsilon$

- Prob. *h* with $error_{true}(h) \geq \varepsilon$ gets *m* data points right

$P(h_{bad} \text{ gets } m \text{ iid points right}) \leq (1-\varepsilon)^m$

exponentially small (as m increases)

But there are many possible hypothesis that are consistent with training data



How likely is learner to pick a bad hypothesis

- Prob. h with $\text{error}_{\text{true}}(h) \geq \varepsilon$ gets m data points right

  $P(h_{bad} \text{ consistent with data}) \leq (1-\varepsilon)^m$

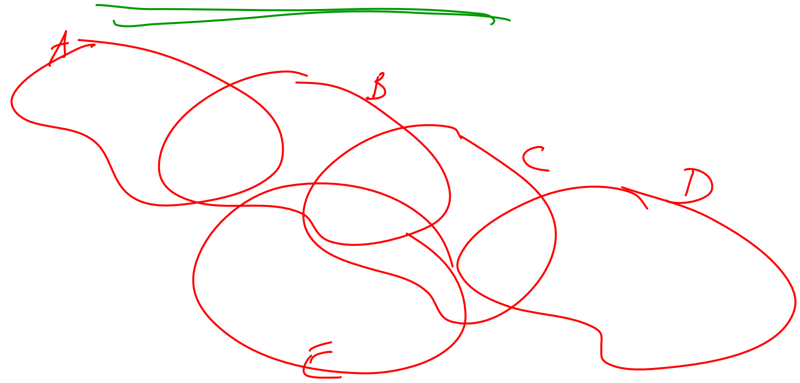
- There are k hypothesis consistent with data
  - How likely is learner to pick a bad one?

$P(\exists h \text{ that is bad and consistent with data})$

$= P(h_1 \text{ bad consistent} \vee h_2 \text{ bad consistent} \vee \ldots \vee h_k \text{ bad \& consistent})$

Union bound

- P(A or B or C or D or …) $\leq P(A) + P(B) + P(C) + \ldots$



How likely is learner to pick a bad hypothesis

- Prob. h with $\text{error}_{true}(h) \geq \varepsilon$ gets m data points right

  $P(h_{bad}, \text{consistent}) \leq (1-\varepsilon)^m$

- There are k hypothesis consistent with data
  - How likely is learner to pick a bad one?

$P(\exists \text{ bad } h \text{ consistent with data}) \leq k\,(1-\varepsilon)^m$

$\leq |H|\,(1-\varepsilon)^m$

$\leq |H|\,e^{-m\varepsilon}$

$(1-\varepsilon)^m \leq e^{-m\varepsilon}$

what's k?

$k \leq |H|$

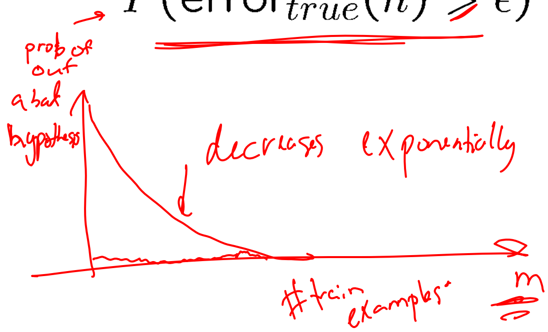
↑ # of hypothesis

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:

  (you give me ε)

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



Using a PAC bound

- Typically, 2 use cases: $P(\text{error}_{true}(h) > \epsilon) \leq |H|e^{-m\epsilon}$
  - 1: Pick ε and δ, give you *m*
  - 2: Pick m and δ, give you ε

pick m, δ, equation holds ∀ε: $\varepsilon \geq \frac{1}{m}\left(\ln|H| + \ln\frac{1}{\delta}\right)$

⇓ with prob at least 1−δ

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

1. ε = 0.1, δ = 0.05

$|H| e^{-m\varepsilon} \leq \delta$

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

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

1/ε: small ε requires many more points

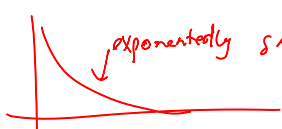
|H|: may be big, only depends ln|H|

ln 1/δ: not hard to achieve low δ

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

- ***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:

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



exponentially small with m

result about h consistent with train data; no guarantee your alg. will find such h.

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

Limitations of Haussler '88 bound

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

- Consistent classifier
- Size of hypothesis space

ln|H|

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
- What about a learner with $error_{train}(h)$ in training set?

no longer assume $error_{train}(h) = 0$

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

- The error of a hypothesis 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 $0<\varepsilon<1$:

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

Using Chernoff bound to estimate error of a single hypothesis

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

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

$e_{true} = error_{true}$

For each hypothesis h_i:

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

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

better? h_1 or h_2

$error_{train}(h_1) \overset{?}{>} error_{train}(h_2)$, want $error_{true}(h_1) \overset{?}{>} error_{true}(h_2)$

$P(e_{true}(h_1) - e_{train}(h_1) > \epsilon \ \vee \ e_{true}(h_2) - e_{train}(h_2) > \epsilon)$

$\leq P(e_{true}(h_1) - e_{train}(h_1) > \epsilon) + P(e_{true}(h_2) - e_{train}(h_2) > \epsilon)$

$\leq 2e^{-2m\epsilon^2}$

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:

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

holds for: $\epsilon \geq \sqrt{\frac{\ln|H| + \ln\frac{1}{\delta}}{2m}}$

with prob. $1-\delta$:

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

PAC bound and Bias-Variance tradeoff

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

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

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

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

What about the size of the hypothesis space?

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

- How large is the hypothesis space?

Boolean formulas with *n* binary features

n binary attributes

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

H: any boolean formula

x_1 x_2 ... x_n	y
0 0 0 0 0 0	0 1
0 0 0 0 0 1	0 0
⋮	1 1
	1 0
	0 0
	0 0
1 1 1 1 1	1 1

2^n rows

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

$\ln|H| = 2^n \ln 2$

really really big

need every row to learn

H: all conjunctions of literals

$x_1 \wedge x_2 \wedge x_7$

$x_3 \wedge x_8 \wedge \neg x_{12}$

⋮

for each x_i: 3 choices
- not include
- include positive
- include negated

$|H| = 3^n$

$\ln|H| = n \ln 3$

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) *

(# possible left subtrees) *

(# possible right subtrees)

$H_{k+1} = n * H_k * H_k$

Write $L_k = \log_2 H_k$

$L_0 = 1$

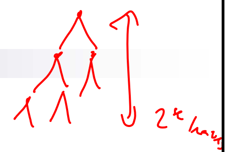
$L_{k+1} = \log_2 n + 2L_k$

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

$\ln|H| \leq (2^k - 1)(1 + \log_2 n) + 1$

PAC bound for decision trees of depth k

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



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

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

no more than m leaves

Number of leaves never more than number data points

Number of decision trees with k leaves

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

H_k = Number of decision trees with k leaves

H_0 =2

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

Loose bound:

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

really big

Reminder:

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

really really big

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

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

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

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

Announcements

- Midterm:
  - Thursday Oct. 25th, Thursday 5-6:20pm, MM A14
    - All content up to, and including SVMs and Kernels
      - Not learning theory
    - any book, class notes, your printouts of class materials that are on the class website, including my annotated slides and relevant readings, and Andrew Moore's tutorials. You cannot use materials brought by other students.
    - Calculators are not necessary.
    - No laptops, PDAs or cellphones.

What did we learn from decision trees?

- Bias-Variance tradeoff formalized

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

- 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

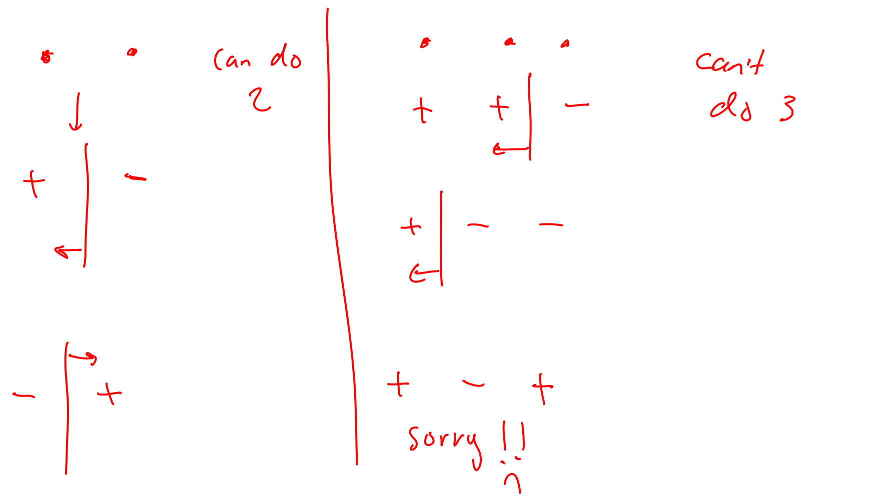
What about continuous hypothesis spaces?

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

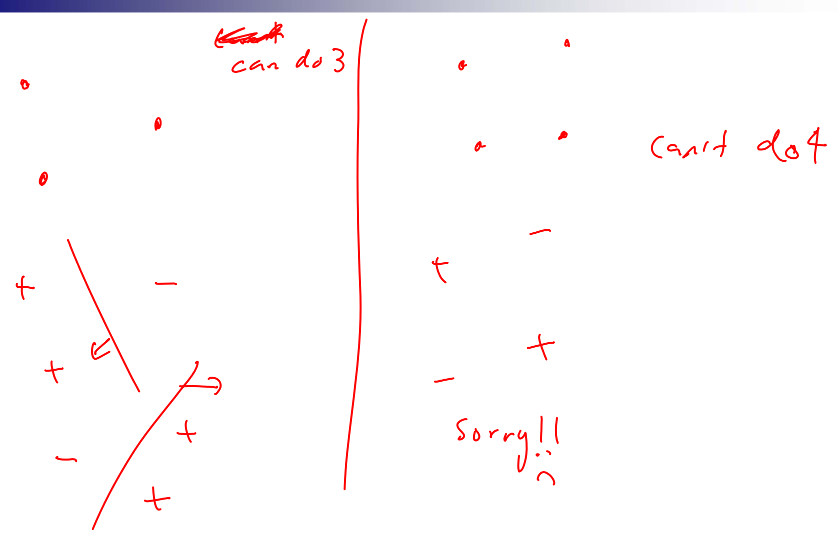
- Continuous hypothesis space:
  - $|H| = \infty$
  - Infinite variance???

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

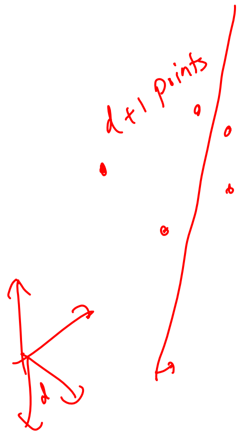
How many points can a linear boundary classify exactly? (1-D)



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



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



can do d+1 points

how many parameters in a linear classifier in d-dimensions?

$$w_0 + \sum_{i=1}^{d} w_i x_i$$

d+1

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**

e.g., linear classifiers

may be continuous

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

only depend on VC(H)
not on |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.

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 $VC(H) \equiv \infty$.

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**
  - **Bound for infinite dimension hypothesis spaces:**

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

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*
- Neural networks
  - VC(H) = #parameters
  - Local minima means NNs will probably not find best parameters
- 1-Nearest neighbor?

Another VC dim. example - What can we shatter?

- What's the VC dim. of decision stumps in 2d?

Another VC dim. example - What can't we shatter?

- What's the VC dim. of decision stumps in 2d?

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
- Remember: will your algorithm find best classifier?

 37

Big Picture

Machine Learning – 10701/15781
Carlos Guestrin
Carnegie Mellon University

October 24^{th}, 2007

 38

What you have learned thus far

- Learning is function approximation
- Point estimation
- Regression
- Naïve Bayes
- Logistic regression
- Bias-Variance tradeoff
- Neural nets
- Decision trees
- Cross validation
- Boosting
- Instance-based learning
- SVMs
- Kernel trick
- PAC learning
- VC dimension
- Margin bounds
- Mistake bounds

Review material in terms of…

- Types of learning problems
- Hypothesis spaces
  what they can represent
- Loss functions
- Optimization algorithms

BIG PICTURE

(a few points of comparison)

learning task / loss function

DE	density estimation
Cl	Classification
Reg	Regression
LL	Log-loss/MLE
Mrg	Margin-based
RMS	Squared error

Naïve Bayes
DE, LL

Boosting
Cl, exp-loss

SVM regression
Reg, Mrg

Logistic regression LL
DE, LL

CL

SVMs
Cl, Mrg
mrg hinge

kernel regression
Reg, RMS

Instance-based Learning
DE,Cl,Reg

Neural Nets
DE,Cl,Reg,RMS

Decision trees
DE,Cl,Reg

linear regression
Reg, RMS

This is a very incomplete view!!!