

Machine Learning (CS 567)

Fall 2008

Time: T-Th 5:00pm - 6:20pm

Location: GFS 118

Instructor: Sofus A. Macskassy (macskass@usc.edu)

Office: SAL 216

Office hours: by appointment

Teaching assistant: Cheol Han (cheolhan@usc.edu)

Office: SAL 229

Office hours: M 2-3pm, W 11-12

Class web page:

<http://www-scf.usc.edu/~csci567/index.html>

Learning Theory Outline

- PAC Learning
- Other COLT models
- VC Dimensions

Binary Classification: The golden goal

- Fundamental Question: Predict Error Rates
 - Given:
 - The instance space X
 - e.g., Possible days, each described by the attributes *Sky, AirTemp, Humidity, Wind, Water, Forecast*
 - A target function (or concept) $f: X \rightarrow \{0,1\}$
 - E.g., $f: \textit{EnjoySport} \rightarrow \{0,1\}$
 - The space H of hypotheses
 - E.g., conjunctions of literals: $\langle ?, \textit{Cold}, \textit{High}, ?, ?, ? \rangle$
 - A set of training examples S (containing positive and negative examples of the target function)
 - $\langle \mathbf{x}_1, f(\mathbf{x}_1) \rangle, \dots, \langle \mathbf{x}_m, f(\mathbf{x}_m) \rangle$
 - Find:
 - A hypothesis $h \in H$ such that $h(\mathbf{x}) = f(\mathbf{x}) \forall \mathbf{x} \in S$

Sample Complexity

How many training examples are sufficient to learn the target concept?

1. If learner proposes instances as queries to teacher (*active learning*)
 - Learner proposes \mathbf{x} , teacher provides $f(\mathbf{x})$
2. If teacher (who knows f) provides training examples
 - Teacher provides example sequence $\langle \mathbf{x}, f(\mathbf{x}) \rangle$
3. If some random process (e.g., nature) proposes instances (*standard case in supervised learning*)
 - \mathbf{x} generated randomly, teacher provides $f(\mathbf{x})$
4. If examples are given by an opponent (who knows f) (*on-line learning, mistake-bound model*)
 - (we won't cover this here)

Sample Complexity: 1

Learner proposes instance \mathbf{x} , teacher provides $f(\mathbf{x})$
(assume f is known to be in learner's hypothesis space H)

Optimal query strategy: play 20 questions

- Pick instance \mathbf{x} such that half of hypotheses in S classify \mathbf{x} positive, half classify \mathbf{x} negative
- If this is always possible, $\log_2 |H|$ queries suffice to learn f
- When it's not possible, need more

Sample Complexity: 2

Teacher (who knows f) provides training examples
(assume f is in learner's hypothesis space H)

Optimal teaching strategy: depends on H used by learner

Consider the case $H =$ conjunctions of up to n Boolean literals and their negations

e.g., $(AirTemp = Warm) \wedge (Wind = Strong)$,
where $AirTemp, Wind, \dots$ each have 2 possible values.

- if n possible Boolean attributes in H , $n+1$ examples suffice. Why?

Sample Complexity: 2

Teacher (who knows f) provides training examples
(assume f is in learner's hypothesis space H)

- if n possible Boolean attributes in H , $n+1$ examples suffice. Why?

Consider the case $f = n$ Boolean literals.

1. Show the learner the one true instance
→ There are many hypotheses that are consistent with this. For example, a hypothesis with $x_1=T$ is consistent.
2. Show an example, with each Boolean literal missing, where the example is false.

Sample Complexity: 3

Given:

- set of instances X
- set of hypotheses H
- set of possible target concepts F
- training instances generated by a fixed, unknown probability distribution D over X

Sample Complexity: 3

Learner observes a sequence D of training examples of form $\langle \mathbf{x}, f(\mathbf{x}) \rangle$, for some target concept $f \in F$

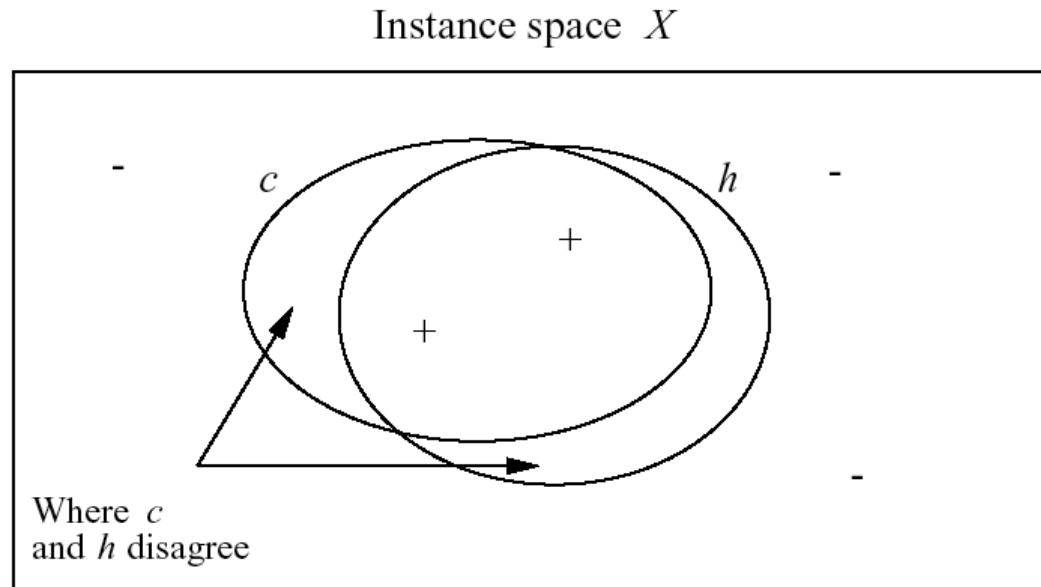
- instances \mathbf{x} are drawn from distribution D
- teacher provides target values $f(\mathbf{x})$

Learner must output a hypothesis h estimating f

- h is evaluated by its performance on subsequent instances drawn from D

Note: randomly drawn instances, noise-free classifications

True Error of a Hypothesis



Definition: The **true error** (denoted $error_D(h)$) of hypothesis h with respect to target concept c and distribution D is the probability that h will misclassify an instance drawn at random via D .

$$error_D(h) \equiv \Pr_{\mathbf{x} \in D}[c(\mathbf{x}) \neq h(\mathbf{x})]$$

Two Notions of Error

Training error of hypothesis h with respect to target concept f :

- How often $h(\mathbf{x}) \neq f(\mathbf{x})$ over the training instances

True error of hypothesis h with respect to target concept f :

- How often $h(\mathbf{x}) \neq f(\mathbf{x})$ over future, unseen instances (but drawn according to D)

Questions:

- Can we bound the true error of a hypothesis given only its training error?
- How many examples are needed for a good approximation?

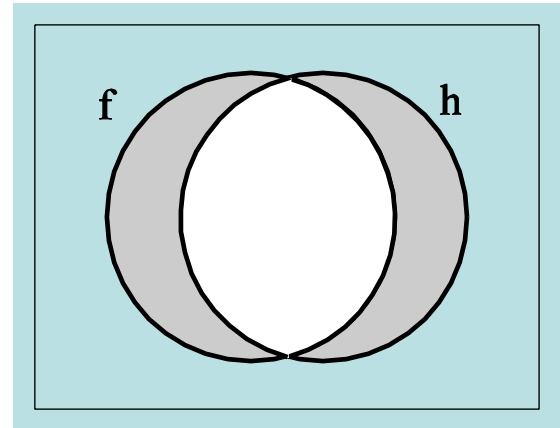
Approximate Concept Learning

- Requiring a learner to acquire the right concept is too strict
- Instead, we will allow the learner to produce a good approximation to the actual concept
- For any instance space, there is a non-uniform likelihood of seeing different instances
- We assume that there is a fixed probability distribution D on the space of instances X
- The learner is trained and tested on examples whose inputs are drawn independently and randomly, according to D .

General Assumptions (Noise-Free Case)

- Assumption: Examples are generated according to a probability distribution $D(\mathbf{x})$ and labeled according to an unknown function $f: \mathcal{X} \rightarrow \mathcal{Y}$
- Learning Algorithm: The learning algorithm is given a set of m examples, and it outputs a hypothesis $h \in H$ that is consistent with those examples (i.e., correctly classifies all of them).
- Goal: h should have a low error rate ϵ on new examples drawn from the same distribution D .

$$\text{error}(h, f) = P_D[f(x) \neq h(x)]$$



Probably-Approximately Correct (PAC) Learning

- We allow our algorithms to fail with probability δ
- Imagine drawing a sample of m examples, running the learning algorithm, and obtaining h .
Sometimes, the sample will be unrepresentative, so we only want to insist that $1 - \delta$ of the time, the hypothesis will have error less than ϵ . For example, we might want to obtain a 99% accurate hypothesis 90% of the time.
- Let $P_D^m(S)$ be the probability of drawing data set S of m examples according to D .

$$P_D^m[\text{error}(f, h) > \epsilon] < \delta$$

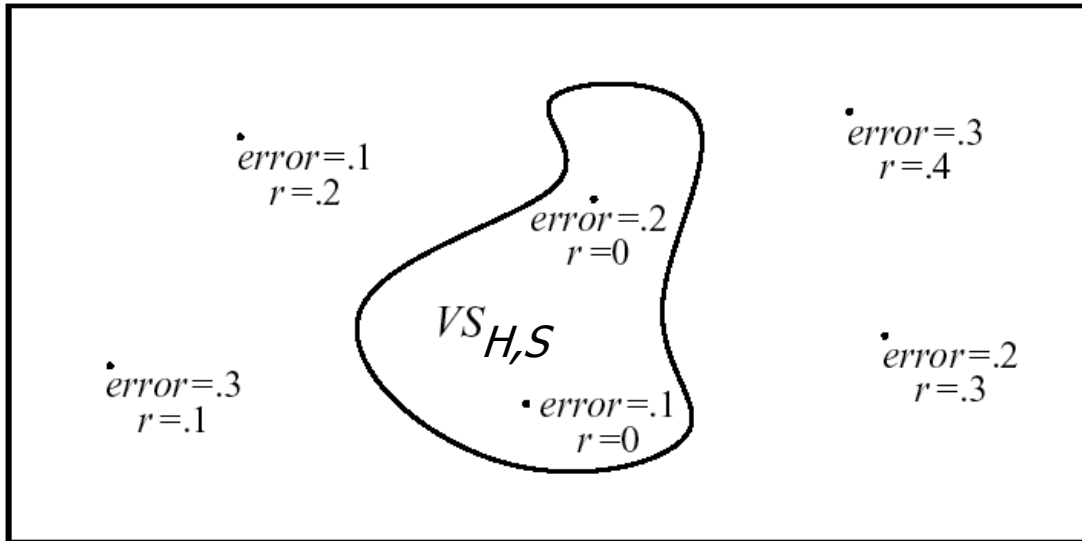
Finite Hypotheses

A Word on Version Spaces

$V_{H,S}$ is the set of hypotheses in the hypothesis space H that are consistent with the training examples S .

ε -Exhausting the Version Space

Hypothesis space H



(r = training error,
 $error$ = true error)

Definition: The version space $VS_{H,S}$ is said to be ε -**exhausted** with respect to f and S , if every hypothesis $h \in VS_{H,S}$ has error less than ε with respect to f and S .

$$(\forall h \text{ in } VS_{H,S}) \text{ error}_D(h) < \varepsilon$$

Case 1: Finite Hypothesis Space

- Assume H is finite
- Consider $h_1 \in H$ such that $error(h_1, f) > \varepsilon$. What is the probability that it will correctly classify m training examples?
- If we draw one training example, (\mathbf{x}_1, y_1) , what is the probability that h_1 classifies it correctly?

$$P [h_1(\mathbf{x}_1) = y_1] \leq (1 - \varepsilon)$$

- What is the probability that h_1 will be right m times?

$$P_D^m [h_1(\mathbf{x}_i) = y_i] \leq (1 - \varepsilon)^m$$

Finite Hypothesis Spaces (2)

- Now consider a second hypothesis h_2 that is also ε -bad. What is the probability that either h_1 or h_2 will survive the m training examples?

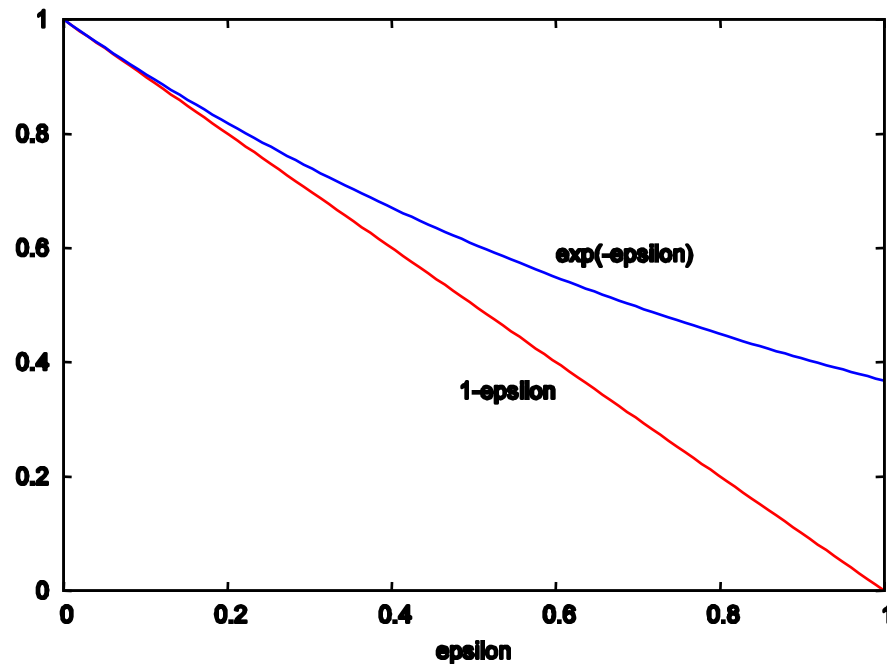
$$\begin{aligned} P_D^m [h_1 \vee h_2 \text{ survives}] &= P_D^m [h_1 \text{ survives}] + \\ &\quad P_D^m [h_2 \text{ survives}] - \\ &\quad P_D^m [(h_1 \wedge h_2) \text{ survives}] \\ &\leq P_D^m [h_1 \text{ survives}] + \\ &\quad P_D^m [h_2 \text{ survives}] \\ &\leq 2(1 - \varepsilon)^m \end{aligned}$$

- So if there are k ε -bad hypotheses, the probability that any one of them will survive is $\leq k(1 - \varepsilon)^m$
- Since $k < |\mathcal{H}|$, this is $\leq |\mathcal{H}|(1 - \varepsilon)^m$

Finite Hypothesis Spaces (3)

- Fact: When $0 \leq \varepsilon \leq 1$, $(1 - \varepsilon) \leq e^{-\varepsilon}$
therefore

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



Blumer Bound

(Blumer, Ehrenfeucht, Haussler, Warmuth)

- Lemma. For a finite hypothesis space H , given a set of m training examples drawn independently according to D , the probability that there exists a hypothesis $h \in H$ with true error greater than ε consistent with the training examples is less than $|H|e^{-\varepsilon m}$.
- We want to ensure that this probability is less than δ .

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

- This will be true when

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

Finite Hypothesis Space Bound

- Corollary: If $h \in H$ is consistent with all m examples drawn according to D , then the error rate ε on new data points can be estimated as

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

Examples

- Boolean conjunctions over n features.

$|H| = 3^n$, since each feature can appear as x_j , $\neg x_j$, or be missing.

$$\varepsilon = \frac{1}{m} \left(\ln 3^n + \ln \frac{1}{\delta} \right) = \frac{1}{m} \left(n \ln 3 + \ln \frac{1}{\delta} \right)$$

- k -DNF formulas:

$$(x_1 \wedge x_3) \vee (x_2 \wedge \neg x_4) \vee (x_1 \wedge x_4)$$

There are at most $(2n)^k$ disjunctions, so $|H| \leq 2^{(2n)^k}$

- for fixed k , this gives

$$\log_2 |H| = (2n)^k$$

- which is polynomial in n : $\varepsilon = \frac{1}{m} O \left(n^k + \ln \frac{1}{\delta} \right)$

Example: Finding m for *EnjoySport*

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

If H is as given in *EnjoySport* and $|H| = 729$, and

$$m \geq \frac{1}{\varepsilon} \left(\ln 729 + \ln \frac{1}{\delta} \right)$$

... if want to assure that with probability 95%, VS contains only hypotheses with $error_D(h) \leq .1$, then it is sufficient to have m examples, where

$$\begin{aligned} m &\geq \frac{1}{.1} \left(\ln 729 + \ln \frac{1}{0.05} \right) \\ &= 10 \cdot (\ln 729 + \ln 20) = 10 \cdot (6.59 + 3.00) \\ &= 95.9 \end{aligned}$$

PAC Learning

Let F be a concept (target function) class defined over a set of instances X in which each instance has length n . An algorithm L , using hypothesis class H is a **PAC learning algorithm** for F if:

- For any concept $f \in F$
 - For any probability distribution D over X
 - For any parameters $0 < \epsilon < 0.5$ and $0 < \delta < 0.5$
- the learner L will, with probability at least $(1-\delta)$, output a hypothesis with true error at most ϵ .

A class of concepts F is **PAC-learnable** if there exists a PAC learning algorithm for F .

PAC in action

Machine	Example Hypothesis	$ H $ (n features)	m required to PAC-learn																																																																																					
And-positive-literals	$X_3 \wedge X_7 \wedge X_8$	2^n	$\frac{1}{\epsilon} \left(n(\ln 2) + \ln \frac{1}{\delta} \right)$																																																																																					
And-literals	$X_3 \wedge \neg X_7$	3^n	$\frac{1}{\epsilon} \left(n(\ln 3) + \ln \frac{1}{\delta} \right)$																																																																																					
Lookup Table	<table border="1"> <thead> <tr> <th>X1</th> <th>X2</th> <th>X3</th> <th>X4</th> <th>Y</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	X1	X2	X3	X4	Y	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	1	1	0	0	1	0	0	1	0	1	0	1	0	0	1	1	0	0	0	1	1	1	1	1	0	0	0	0	1	0	0	1	0	1	0	1	0	0	1	0	1	1	1	1	1	0	0	0	1	1	0	1	0	1	1	1	0	0	1	1	1	1	0	2^{2^n}	$\frac{1}{\epsilon} \left(2^n (\ln 2) + \ln \frac{1}{\delta} \right)$
X1	X2	X3	X4	Y																																																																																				
0	0	0	0	0																																																																																				
0	0	0	1	1																																																																																				
0	0	1	0	1																																																																																				
0	0	1	1	0																																																																																				
0	1	0	0	1																																																																																				
0	1	0	1	0																																																																																				
0	1	1	0	0																																																																																				
0	1	1	1	1																																																																																				
1	0	0	0	0																																																																																				
1	0	0	1	0																																																																																				
1	0	1	0	0																																																																																				
1	0	1	1	1																																																																																				
1	1	0	0	0																																																																																				
1	1	0	1	0																																																																																				
1	1	1	0	0																																																																																				
1	1	1	1	0																																																																																				
And-lits or And-lits	$(X_2 \wedge \neg X_7 \wedge X_8)$	$(3^n)^2 = 3^{2n}$	$\frac{1}{\epsilon} \left(2n(\ln 3) + \ln \frac{1}{\delta} \right)$																																																																																					

Empirical Risk Minimization

- Suppose we are given a hypothesis class H
- We have a magical learning machine that can sift through H and output the hypothesis with the smallest training error, h_{emp}
- This process is called **empirical risk minimization**
- Is this a good idea?
- What can we say about the error of the other hypotheses in H ?

First tool: The union bound

- Let $E_1 \dots E_k$ be k different events (not necessarily independent).
- Then:

$$P(E_1 \cup \dots \cup E_k) \leq P(E_1) + \dots + P(E_k)$$

Second Tool: Hoeffding (Chernoff) bounds

Let $Z_1 \dots Z_m$ be m independent identically distributed (iid) binary variables, drawn from a bernoulli distribution:

$$P(Z_i=1) = \phi \text{ and } P(Z_i=0) = 1 - \phi$$

Let $\hat{\phi}$ be the mean of these variables:

$$\hat{\phi} = \frac{1}{m} \sum_{i=1}^m Z_i$$

Let ϵ be a fixed error tolerance parameter. Then:

$$P(|\phi - \hat{\phi}| > \epsilon) \leq 2e^{-2\epsilon^2 m}$$

In other words, if you have many examples, the empirical mean is a good estimator of the true probability.

Finite Hypothesis Space

- Suppose we are considering a finite hypothesis class $H = \{h_1, \dots, h_k\}$ (e.g., conjunctions, decision trees, etc.)
- Take an arbitrary hypothesis $h_i \in H$
- Suppose we sample data according to our distribution and let $Z_j = 1$ iff $h_i(\mathbf{x}_j) \neq y_j$
- So $e(h_i)$ (the true error of h_i) is the expected value of Z_j
- Let $\hat{e}(h_i) = \frac{1}{m} \sum_{j=1}^m Z_j$ (this is the empirical training error of h_i on the data set we have)
- Using the Hoeffding bound, we have:

$$P(|e(h_i) - \hat{e}(h_i)| > \epsilon) \leq 2e^{-2\epsilon^2 m}$$

- So, if we have lots of data, the training error of a hypothesis h_i will be close to its true error with high probability.

What about all hypotheses?

- We showed that the empirical error is “close” to the true error for one hypothesis
- Let E_i denote the event $|e(h_i) - \hat{e}(h_i)| > \varepsilon$
- Can we guarantee this is true for all hypotheses?

$$\begin{aligned} P(\exists h_i \in H, |e(h_i) - \hat{e}(h_i)| > \varepsilon) &= P(E_1 \cup \dots \cup E_{|H|}) \\ &\leq \sum_{i=1}^{|H|} P(E_i) \quad (\text{union bound}) \\ &\leq \sum_{i=1}^{|H|} 2e^{-2\varepsilon^2 m} \quad (\text{shown before}) \\ &= 2|H|e^{-2\varepsilon^2 m} \end{aligned}$$

A uniform convergence bound

- We showed that:

$$P(\exists h_i \in H, |e(h_i) - \hat{e}(h_i)| > \varepsilon) \leq 2|H|e^{-2\varepsilon^2 m}$$

- So we have:

$$1 - P(\exists h_i \in H, |e(h_i) - \hat{e}(h_i)| > \varepsilon) \geq 1 - 2|H|e^{-2\varepsilon^2 m}$$

or, in other words:

$$P(\forall h_i \in H, |e(h_i) - \hat{e}(h_i)| < \varepsilon) \geq 1 - 2|H|e^{-2\varepsilon^2 m}$$

- This is called a **uniform convergence** result because the bound holds for all hypotheses
- What is this good for?

Sample Complexity

- Suppose we want to guarantee that with probability at least $1-\delta$, the sample (training) error is within ε of the true error.
- From our bound, we can set $\delta \geq 2|H|e^{-2\varepsilon^2 m}$
- Solving for m , we get that the number of samples should be:

$$m \geq \frac{1}{2\varepsilon^2} \ln \frac{2|H|}{\delta}$$

- So the number of samples needed is logarithmic in the size of the hypothesis space.

Example: Conjunctions of Boolean Literals

- Let H be the space of all pure conjunctive formulae over n Boolean attributes.
- Recall, $|H|=3^n$ (why?)
- From the previous result, we get:

$$m \geq \frac{1}{2\varepsilon^2} \ln \frac{2|H|}{\delta} = n \frac{1}{2\varepsilon^2} \ln \frac{6}{\delta}$$

- This is linear in n !

Another application: Bounding the true error

$$P(\forall h_i \in H, |e(h_i) - \hat{e}(h_i)| < \varepsilon) \geq 1 - 2|H|e^{-2\varepsilon^2 m} = 1 - \delta$$

- Suppose we hold m and δ fixed, and we solve for ε . Then we get:

$$|e(h_i) - \hat{e}(h_i)| \leq \sqrt{\frac{1}{2m} \ln \frac{2|H|}{\delta}}$$

inside the probability term.

Can we now prove anything about the generalization power of the empirical risk minimization algorithm?

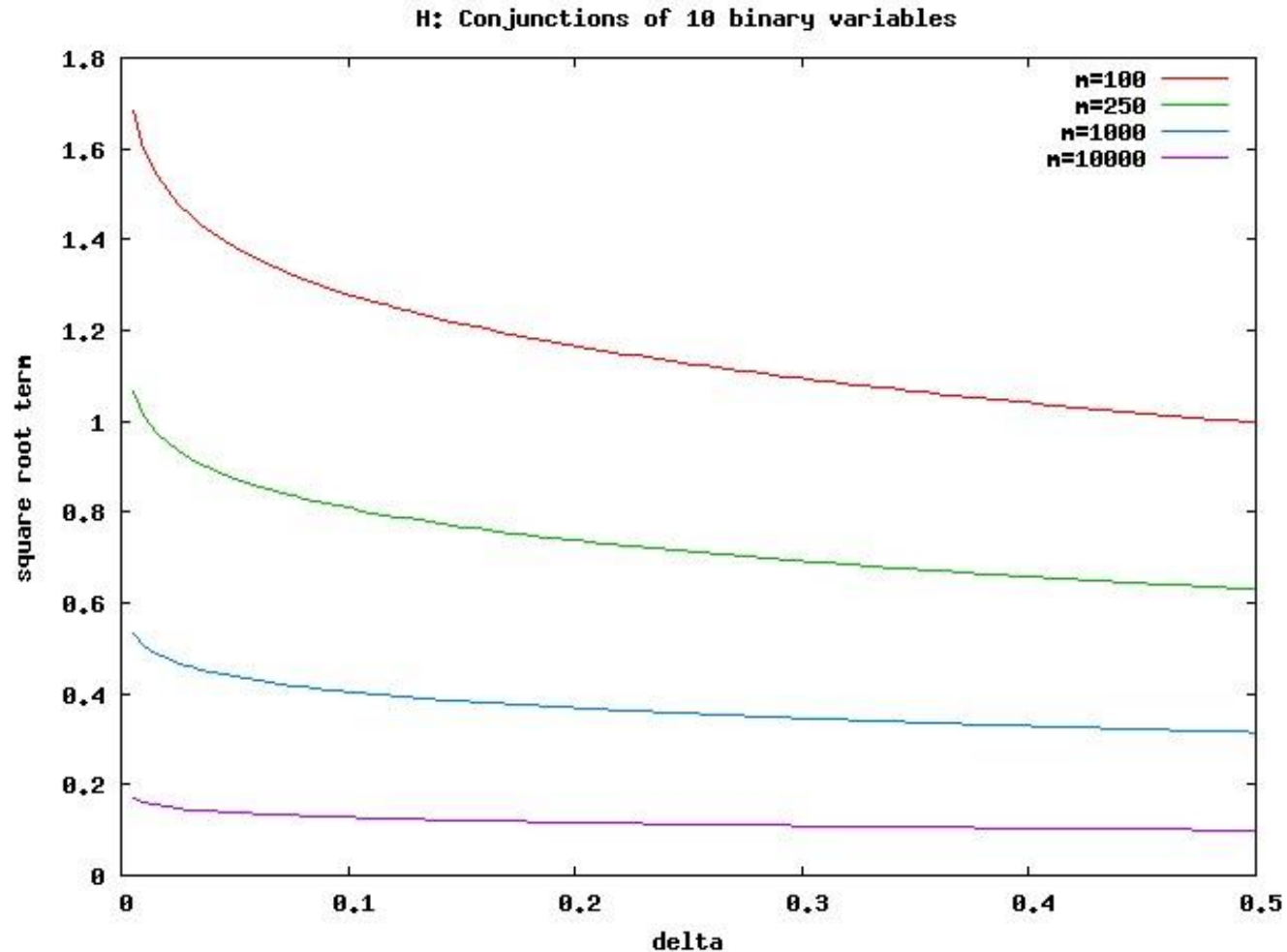
Empirical Risk Minimization

- Let h^* be the best hypothesis in our class (in terms of true error).
- Based on our uniform convergence assumption, we can bound the true error of h_{emp} as follows:

$$\begin{aligned} e(h_{\text{emp}}) &\leq \hat{e}(h_{\text{emp}}) + \varepsilon \\ &\leq \hat{e}(h^*) + \varepsilon \\ &\leq e(h^*) + 2\varepsilon \\ &\leq e(h^*) + 2\sqrt{\frac{1}{2m} \ln \frac{2|H|}{\delta}} \end{aligned}$$

- This bounds how much worse h_{emp} is with respect to the best hypothesis we can hope for!

Empirical Risk Minimization



Infinite Hypothesis Spaces

- Most of our classifiers (LTUs, neural networks, SVMs) have continuous parameters and therefore, have infinite hypothesis spaces
- Despite their infinite size, they have limited expressive power, so we should be able to prove something

Shattering a Set

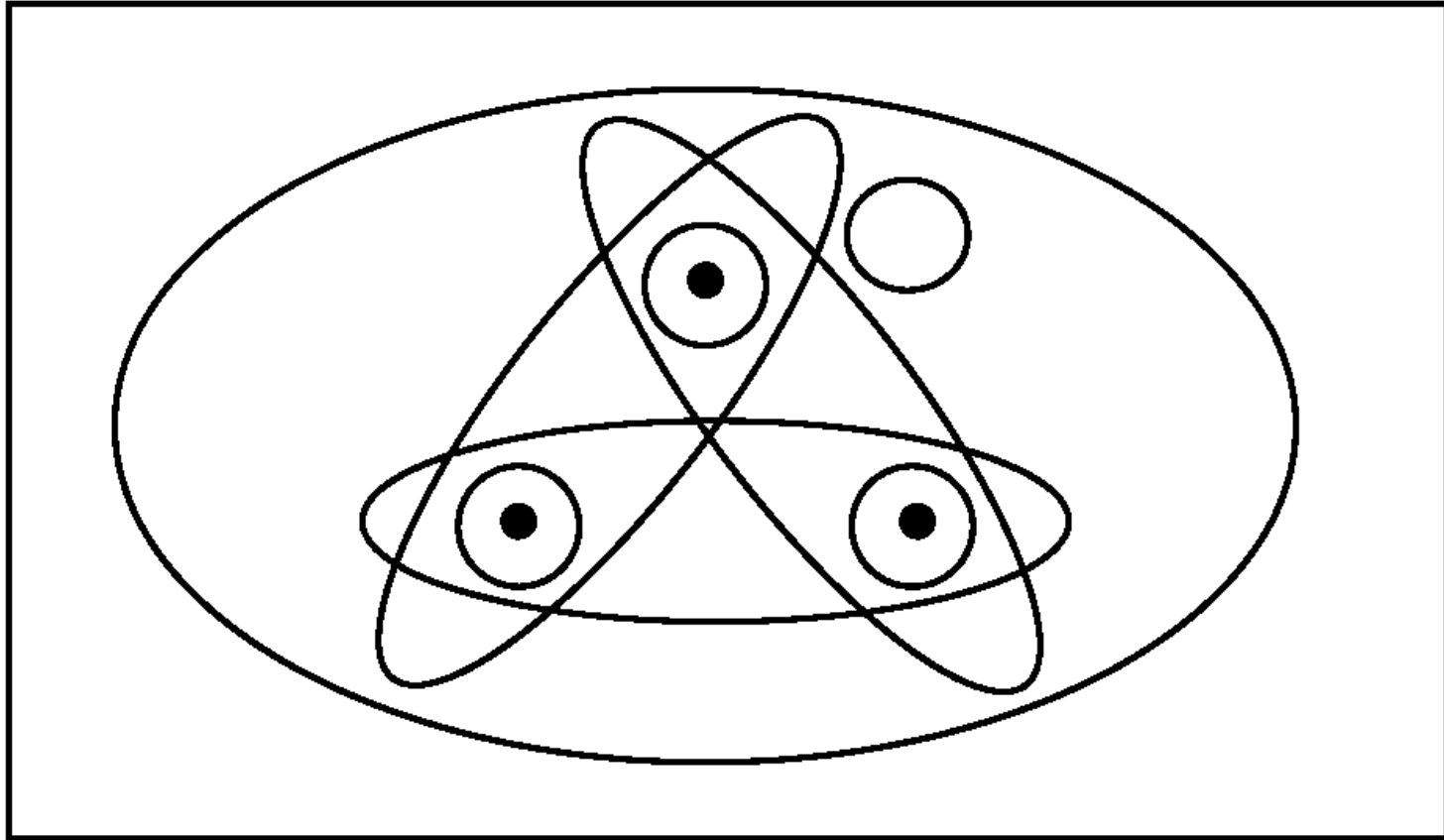
Definition: a **dichotomy** of a set S is a partition of S into two disjoint subsets.

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.

In other words: The instances can be classified in every possible way.

Three Instances Shattered

Instance space X



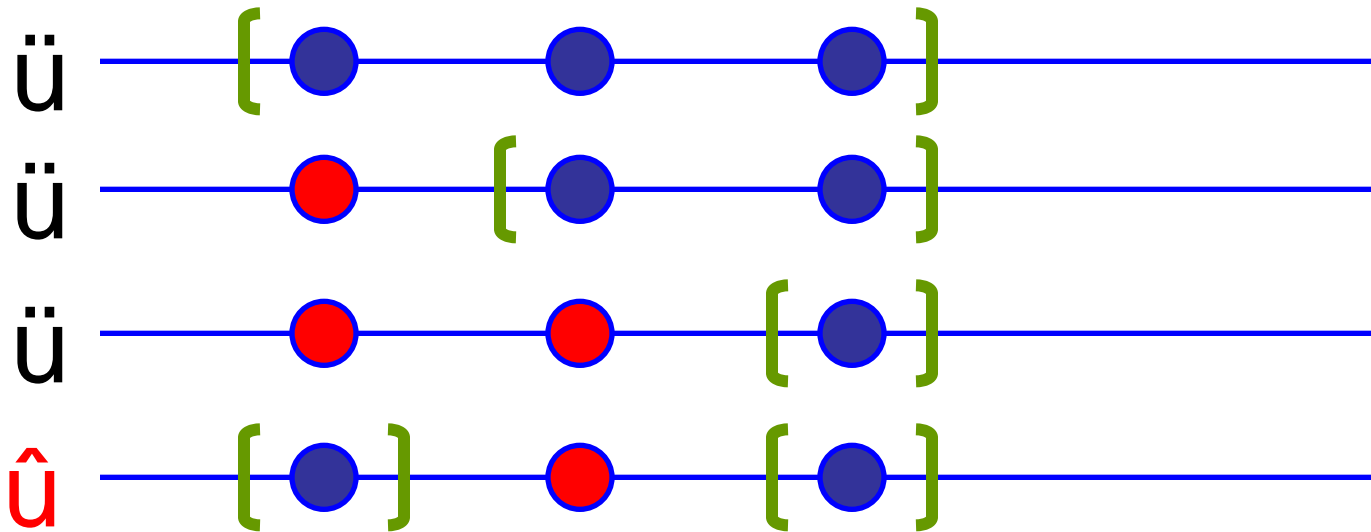
The VC Dimension

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 (but finite) sets of X can be shattered by H , then $VC(H) \equiv \infty$.

- For finite H , $VC(H) \leq \log_2 |H|$

Example: Shattering an interval

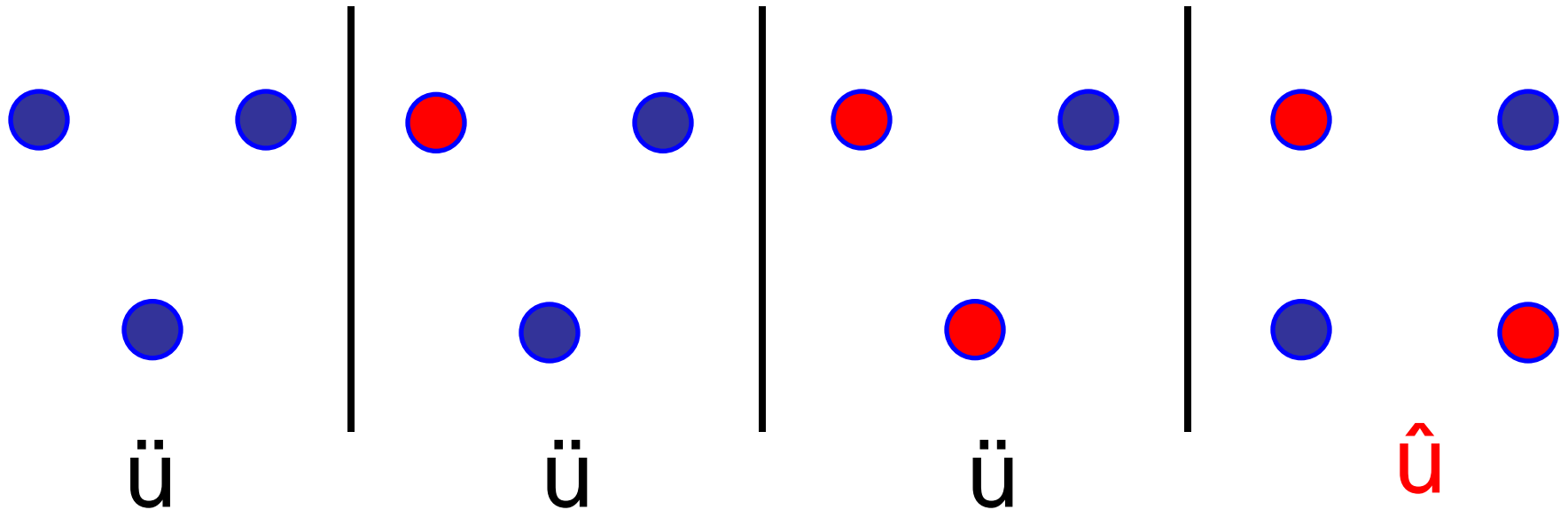
- Let H be the set of intervals on the real line such that $h(\mathbf{x})=1$ iff \mathbf{x} is in the interval.
- How many points can be shattered by H ?



- 2 points. It cannot shatter 3. $VC(H)=2$

Example: Shattering a linear separator

- Let H be the set of linear separators in the 2-D plane.
- How many points can be shattered by H ?



- Can shatter 3, but not 4 points. $VC(H)=3$

Example: Shattering a linear separator

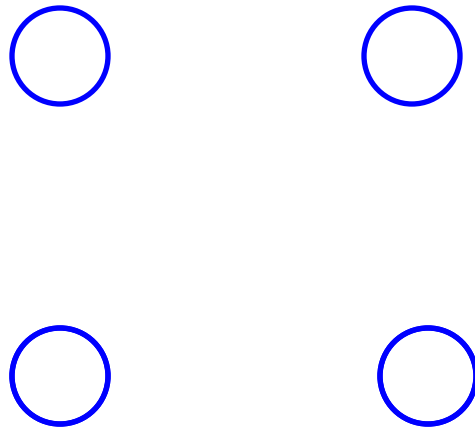
- We cannot separate any set of 4 points (XOR).
- In general, the VC(LTU) in n -dimensional space is $n+1$.
- A good heuristic is that the VC-dimension is equal to the number of tunable parameters in the model (unless the parameters are redundant)

Example: Shattering a circle

- Let H be the set of circles in 2-D such that $h(\mathbf{x})=1$ iff \mathbf{x} is inside the circle.
- How many points can be shattered by H ?

Example: Shattering a circle

- Let H be the set of circles in 2-D such that $h(\mathbf{x})=1$ iff \mathbf{x} is inside the circle.
- How many points can be shattered by H ?



- $VC(H)=3$

Error Bound for Consistent Hypotheses

- The following bound is analogous to the Blumer bound. If h is an hypothesis that makes no error on a training set of size m , and h is drawn from a hypothesis space H with VC-dimension d , then with probability $1-\delta$, h will have an error rate less than ε if

$$m \geq \frac{1}{\varepsilon} \left(4 \log_2 \left(\frac{2}{\delta} \right) + 8d \log_2 \left(\frac{13}{\varepsilon} \right) \right)$$

Error Bound for Inconsistent Hypotheses

- Theorem. Suppose H has VC-dimension d and a learning algorithm finds $h \in H$ with error rate ε_T on a training set of size m . Then with probability $1 - \delta$, the error rate ε on new data points is

$$\varepsilon \leq 2\varepsilon_T + \frac{4}{m} \left(d \log \frac{2em}{d} + \log \frac{4}{\delta} \right)$$

- Empirical Risk Minimization Principle
 - If you have a fixed hypothesis space H , then your learning algorithm should minimize ε_T : the error on the training data. (ε_T is also called the “empirical risk”)

Remarks on VC dimension

- The previous bound is tight up to log factors. In other words, for hypothesis classes with large VC dimension, we can show that there exists some data distribution which will produce a bad approximation.
- For many reasonable hypothesis classes (e.g., linear approximators) the VC dimension is linear in the number of “parameters” of the hypothesis. This shows that to learn “well”, we need a number of examples that is linear in the VC dimension (so linear in the number of parameters, in this case).
- An important property:
 - if $H_1 \subseteq H_2$ then $VC(H_1) \leq VC(H_2)$

Variable-Sized Hypothesis Spaces

- A fixed hypothesis space may not work well for two reasons
 - Underfitting: Every hypothesis in H has high ε_T . We would like to consider a larger hypothesis space H' so we can reduce ε_T
 - Overfitting: Many hypotheses in H have $\varepsilon_T = 0$. We would like to consider a smaller hypothesis space H' so we can reduce d .
- Suppose we have a nested series of hypothesis spaces:

$$H_1 \subseteq H_2 \subseteq \dots \subseteq H_k \subseteq \dots$$

with corresponding VC dimensions and errors

$$d_1 \leq d_2 \leq \dots \leq d_k \leq \dots$$

$$\varepsilon_T^1 \geq \varepsilon_T^2 \geq \dots \geq \varepsilon_T^k \geq \dots$$



















Structural Risk Minimization Principle (Vapnik)

- Choose the hypothesis space H_k that minimizes the combined error bound

$$\varepsilon \leq 2\varepsilon_T + \frac{4}{m} \left(d \log \frac{2em}{d} + \log \frac{4}{\delta} \right)$$

Structural Risk Minimization

$$\varepsilon \leq 2\varepsilon_T^k + \frac{4}{m} \left(d_k \log \frac{2em}{d_k} + \log \frac{4}{\delta} \right)$$

i	H_i	ε_T	VC-Conf	Probable upper bound on ε	Choice
1	H_1				
2	H_2				
3	H_3				<input type="checkbox"/>
4	H_4				
5	H_5				
6	H_6				

Using VC-dimensionality

That's what VC-dimensionality is about

People have worked hard to find VC-dimension for..

- Decision Trees
- Perceptrons
- Neural Nets
- Decision Lists
- Support Vector Machines
- And many many more











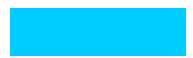
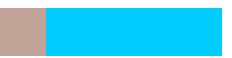






All with the goals of

1. Understanding which learning machines are more or less powerful under which circumstances
2. Using Structural Risk Minimization to choose the best learning machine

Alternatives to VC-dim-based model selection













- What could we do instead of the scheme below?

$$\varepsilon \leq 2\varepsilon_T^k + \frac{4}{m} \left(d_k \log \frac{2em}{d_k} + \log \frac{4}{\delta} \right)$$

i	H_i	ε_T	VC-Conf	Probable upper bound on ε	Choice
1	H_1				
2	H_2				
3	H_3				<input type="checkbox"/>
4	H_4				
5	H_5				
6	H_6				

Alternatives to VC-dim-based model selection

- What could we do instead of the scheme below?
 1. Cross-validation

i	H_i	ε_T	10-FOLD-CV-ERR	Choice
1	H_1			
2	H_2			
3	H_3			<input type="checkbox"/>
4	H_4			
5	H_5			
6	H_6			



















Alternatives to VC-dim-based model selection

- What could we do instead of the score test?
 1. Cross-validation
 2. AIC (Akaike Information Criterion)

As the amount of data goes to infinity, AIC promises* to select the model that'll have the best likelihood for future data

*Subject to about a million caveats

$$\text{AICSCORE} = LL(\text{Data} | \text{MLE params}) - (\# \text{ parameters})$$

i	H_i	LOGLIKE(ϵ_T)	#parameters	AIC	Choice
1	H_1				
2	H_2				
3	H_3				
4	H_4				<input type="checkbox"/>
5	H_5				
6	H_6				



















Alternatives to VC-dim-based model selection

As the amount of data goes to infinity, BIC promises* to select the model that the data was generated from. More conservative than AIC.

*Another million caveats

- What could we do instead of the score test?
 1. Cross-validation
 2. AIC (Akaike Information Criterion)
 3. BIC (Bayesian Information Criterion)

$$\text{BICSCORE} = LL(\text{Data} | \text{MLE params}) - \frac{\# \text{ params}}{2} \log N$$

i	H_i	LOGLIKE(ϵ_T)	#parameters	BIC	Choice
1	H_1				
2	H_2				
3	H_3				<input type="checkbox"/>
4	H_4				
5	H_5				
6	H_6				

Which model selection method is best?

1. (CV) Cross-validation
 2. AIC (Akaike Information Criterion)
 3. BIC (Bayesian Information Criterion)
 4. (SRMVC) Structural Risk Minimize with VC-dimension
- AIC, BIC and SRMVC have the advantage that you only need the training error.
 - CV error might have more variance
 - SRMVC is wildly conservative
 - Asymptotically AIC and Leave-one-out CV should be the same
 - Asymptotically BIC and a carefully chosen k-fold should be the same
 - BIC is what you want if you want the best structure instead of the best predictor (e.g. for clustering or Bayes Net structure finding)
 - Many alternatives to the above including proper Bayesian approaches.
 - It's an emotional issue.

Extra Comments

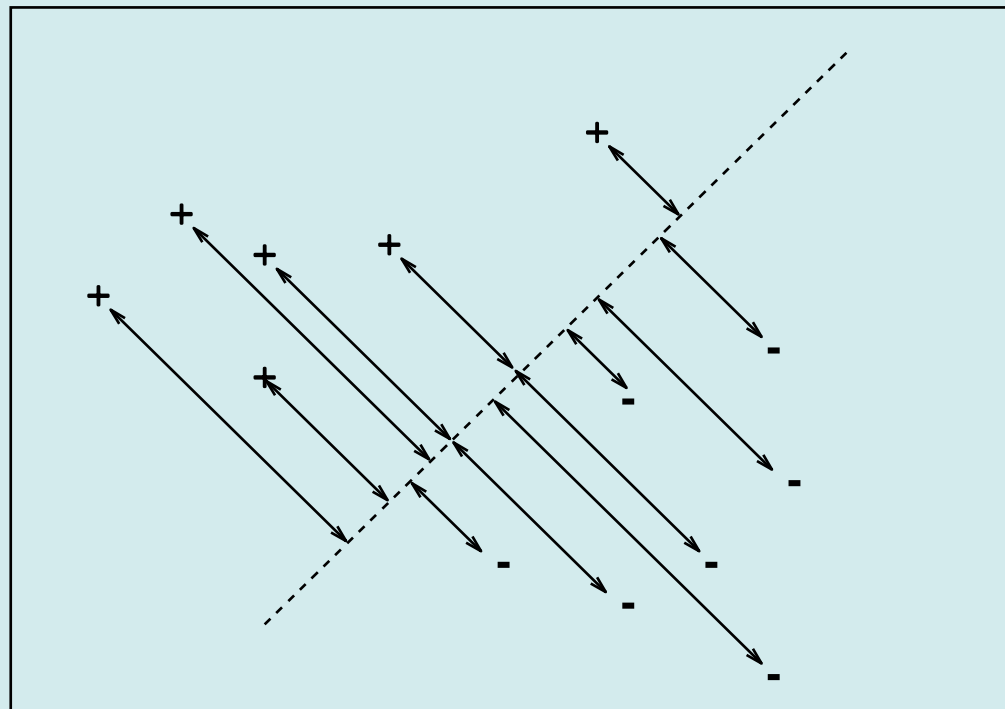
- Beware: that second “VC-confidence” term is usually very very conservative (at least hundreds of times larger than the empirical overfitting effect).
- An excellent tutorial on VC-dimension and Support Vector Machines:
C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):955-974, 1998.
<http://citeseer.ist.psu.edu/burges98tutorial.html>

Data-Dependent Bounds

- So far, our bounds on ε have depended only on ε_T and quantities that could be computed prior to training
- The resulting bounds are “worst case”, because they must hold for all but $1 - \delta$ of the possible training sets.
- Data-dependent bounds measure other properties of the fit of h to the data. Suppose S is not a worst-case training set. Then we may be able to obtain a tighter error bound

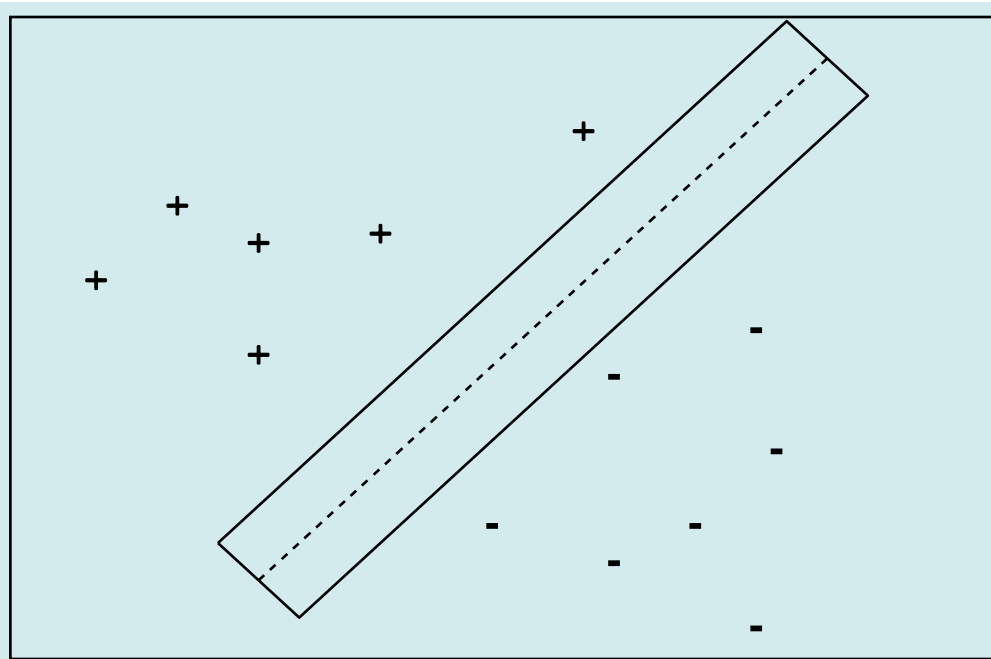
Margin Bounds

- Suppose $g(\mathbf{x})$ is a real-valued function that will be thresholded at 0 to give $h(\mathbf{x})$: $h(\mathbf{x}) = \text{sgn}(g(\mathbf{x}))$.
- The functional margin γ of g on training example $\mathcal{R}\mathbf{x}, y^\top$ is $\gamma = yg(\mathbf{x})$.
- The margin with respect to the whole training set is defined as the minimum margin over the entire set: $\gamma(g, S) = \min_i y_i g(\mathbf{x}_i)$



Margin Bounds: Key Intuition

- Consider the space of real-valued functions G that will be thresholded at 0 to give H .
- This space has some VC dimension d .
- But now, suppose that we consider “thickening” each $g \in G$ by requiring that it correctly classify every point with a margin of at least γ .
- The VC dimension of these “fat” separators will be much less than d . It is called the fat shattering dimension: $\text{fat}_G(\gamma)$



Noise-Free Margin Bound

- Suppose a learning algorithm finds a $g \in G$ with margin $\gamma = \gamma(g, S)$ for a training set S of size m . Then with probability $1 - \delta$, the error rate on new points will be

$$\varepsilon \leq \frac{2}{m} \left(d \log \frac{2em}{d\gamma} \cdot \log \frac{32m}{\gamma^2} + \log \frac{4}{\delta} \right)$$

- where $d = \text{fat}_G(\gamma/8)$ is the fat shattering dimension of G with margin $\gamma/8$.
- We can see that the fat shattering dimension is behaving much as the VC dimension did in our error bounds

Fat Shattering using Linear Separators

- Let D be a probability distribution such that all points \mathbf{x} drawn according to D satisfy the condition $\|\mathbf{x}\| \leq R$, so all points \mathbf{x} lie within a sphere of radius R .
- Consider the functions defined by a unit weight vector:
$$G = \{g \mid g = \mathbf{w} \cdot \mathbf{x} \text{ and } \|\mathbf{w}\| = 1\}$$
- Then the fat shattering dimension of G is

$$\text{fat}_G(\gamma) = \left(\frac{R}{\gamma} \right)^2$$

Noise-Free Margin Bound for Linear Separators

- By plugging this in, we find that the error rate of a linear classifier with unit weight vector and with margin γ on the training data (lying in a sphere of radius R) is

$$\varepsilon \leq \frac{2}{m} \left(\frac{64R^2}{\gamma^2} \log \frac{em\gamma}{8R^2} \cdot \log \frac{32m}{\gamma^2} + \log \frac{4}{\delta} \right)$$

- Ignoring all of the log terms, this says we should try to minimize

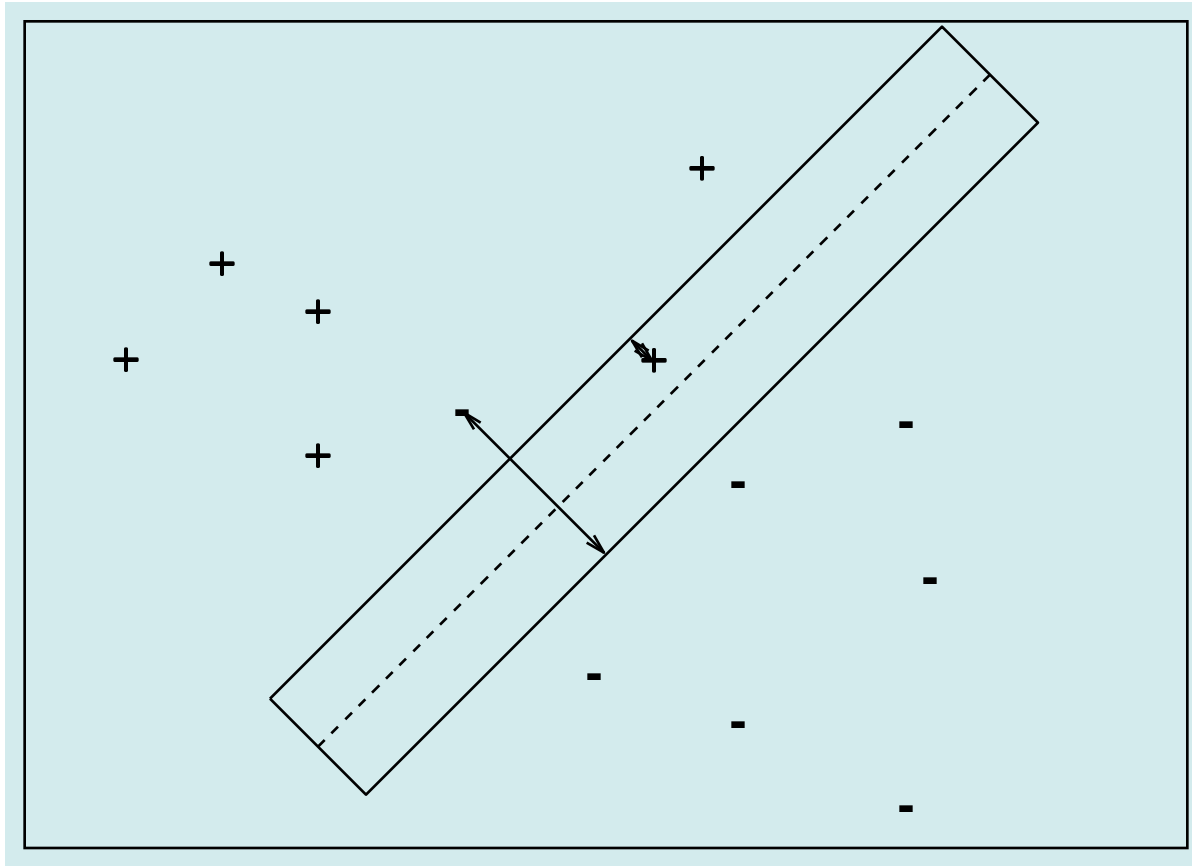
$$\frac{R^2}{m\gamma^2}$$

- R and m are fixed by the training set, so we should try to find a g that maximizes γ . This is the theoretical rationale for finding a maximum margin classifier.

Margin Bounds for Inconsistent Classifiers (soft margin classification)

- We can extend the margin analysis to the case when the data are not linearly separable (i.e., when a linear classifier is not consistent with the data). We will do this by measuring the margin on each training example
- Define $\xi_i = \max\{0, \gamma - y_i g(\mathbf{x}_i)\}$
 ξ_i is called the margin slack variable for example $\mathcal{R}\mathbf{x}_i, y_i^\top$
- Note that $\xi_i > \gamma$ implies that \mathbf{x}_i is misclassified by g .
- Define $\xi = (\xi_1, \dots, \xi_m)$ to be the margin slack vector for the classifier g on training set S

Soft Margin Classification (2)



$$\xi_i = \max\{0, \gamma - y_i g(\mathbf{x}_i)\}$$

Soft Margin Classification (3)

- Theorem. With probability $1 - \delta$, a linear separator with unit weight vector and margin γ on training data lying in a sphere of radius R will have an error rate on new data points bounded by

$$\varepsilon \leq \frac{C}{m} \left(\frac{R^2 + \|\xi\|^2}{\gamma^2} \log^2 m + \log \frac{1}{\delta} \right)$$

- for some constant C .
- This result tells us that we should
 - maximize γ
 - minimize $\|\xi\|^2$
 - but it doesn't tell us how to tradeoff among these two (because C may vary depending on γ and ξ)
- This gives us the full support vector machine which we covered earlier.

Statistical Learning Theory: Summary

- There is a 3-way tradeoff between ε , m , and the complexity of the hypothesis space H .
- The complexity of H can be measured by the VC dimension
- For a fixed hypothesis space, we should try to minimize training set error (empirical risk minimization)
- For a variable-sized hypothesis space, we should be willing to accept some training set errors in order to reduce the VC dimension of H_k (structural risk minimization)
- Margin theory shows that by changing γ , we continuously change the effective VC dimension of the hypothesis space. Large γ means small effective VC dimension (fat shattering dimension)
- Soft margin theory tells us that we should be willing to accept an increase in $\|\xi\|^2$ in order to get an increase in γ .
- We will be able to implement structural risk minimization within a single optimizer by having a dual objective function that tries to maximize γ while minimizing $\|\xi\|^2$