Computational Learning Theory

15-486/782: Artificial Neural Networks
David S. Touretzky

Fall 2006

What is Computational Learning Theory?

Theoretical analysis of machine learning algorithms:

- How many examples required to learn something?
  - This is called the **sample complexity**.
- How much time required to learn it?
  - This is called the **computational complexity**.

Concept Learning

Let X denote an instance space.

A **concept** is a collection of points in X.

Example: X = {A, B, C, D, E}

Concepts: c_1 = {A, C}
c_2 = {A, B, D, E}

There are 2^5 = 32 distinct concepts over the space X.

Concept Classes

Let X = R^2 : points in the plane.

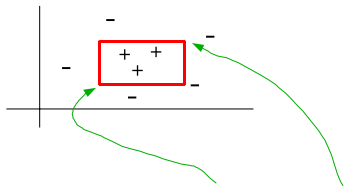
X is infinite.

How can we define concepts over X?

Concept class C : set of concepts defined according to some rule.

Lots of rules are possible...

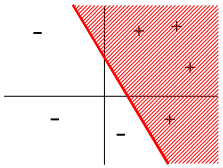
Concept Class: "Axis-Aligned Rectangles"



Four parameters: (x_l, y_l) *and* (x_h, y_h)

Perfect fit requires infinite training data.

Concept Class: "Linear Half-Planes"



$$w_0 + w_1 x_1 + w_2 x_2 > 0$$

Perfect fit requires infinite training data.

Searching Infinite Concept Classes

Let $c \in C$ be the concept we're trying to learn.

How can we find c when C is infinite?

We can't. But...

We can come close.

The more training data, the closer we should come.

How close can we get?

PAC Learning (Valiant, 1984)

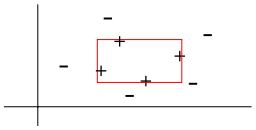
P*robably*	$p > 1-\delta$, *where* $\delta < 1/2$
A*pproximately*	*fractional error* $< \epsilon < 1$
C*orrect*	*for points drawn from D*

A PAC learner will likely (with prob. $1-\delta$*) come pretty close (within a fraction* ϵ*) to the correct concept, given training data drawn from the distribution D.*

How many samples m are required?

Learning Strategy for Axis-Aligned Rectangles

Pick the tightest rectangle that encloses all the positive instances.



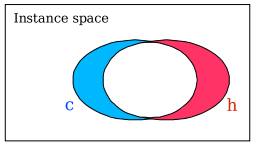
(Algorithm avoids false positives. False negatives revise h.)

How many samples m are required to meet the PAC bound?

$$m \geq \frac{4}{\epsilon}\ln\frac{4}{\delta}$$

Proof to follow.

Size of Symmetric Difference Between Concepts Gives Error Measure



c = *target concept*

h = *hypothesis concept*

$$error(h) = \Pr_{x \in D}[c(x) \neq h(x)]$$

Depends on D

Oracle $EX(c, D)$ *generates points* $\langle x, c(x)\rangle$ *drawn from D, where* $c(x) = 0$ *or* 1.

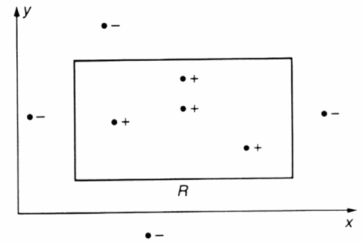
Require that: $error(h) < \epsilon$ *with probability* $1-\delta$

Ideal PAC Learner

- The number of calls to EX(c,D) is small.
- The amount of computation to revise h with each new example is small.
- For the output h, error(h) is small.

Axis-Aligned Rectangles

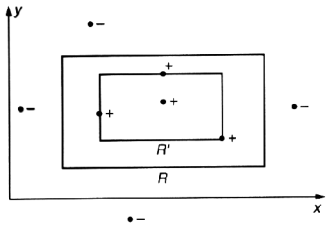
Let R denote the target concept to be learned.



Axis-Aligned Rectangles: Hypothesis R'

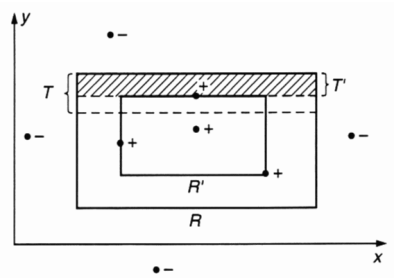
Let R' denote the current hypothesis: the tightest rectangle that encloses all positive samples.

Due to this choice of learning algorithm, R' $\subset$ R.

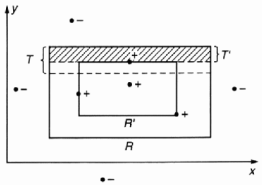


Error Strip T

Define T to have weight exactly $\epsilon/4$.



Constraints on T



T' has weight $> \epsilon/4$ *iff* $T' \supset T$
$T' \supset T$ *iff no point in T appears in the sample S.*

What is the probability that m independent draws from D all miss T?

$$(1-\epsilon/4)^m$$

Same analysis holds for the other three strips.
So prob. that ***any*** *strip has weight* $> \epsilon/4$ *is:*

at most $4(1-\epsilon/4)^m$

Deriving m

Choose m such that $4(1-\epsilon/4)^m \leq \delta$.

$$(1-\epsilon/4)^m \leq \delta/4 \qquad \textit{divide by 4}$$

Note: $(1-x) \leq e^{-x}$, *so...*

$$e^{-\epsilon m/4} \leq \delta/4 \qquad \textit{by substitution}$$

$$-\epsilon m/4 \leq \ln(\delta/4) \qquad \textit{take the log}$$

$$m \geq \frac{4}{\epsilon}\ln\frac{4}{\delta} \qquad \textit{rearrange}$$

Definition of the PAC Model

Let C be a concept class over X.

C is ***PAC Learnable*** *if there is an algorithm L such that:*

for every $c \in C$
for every distribution D on X
for all $\epsilon, \delta \in [0, 1/2]$

if L has access to EX(c,D) and ϵ *and* δ*, then:*

with probability $\geq 1-\delta$*,*
L outputs a hypothesis $h \in C$
such that $error(h) \leq \epsilon$*.*

Computational Complexity

Concept class C is ***efficiently PAC-learnable*** *if*
L runs in time polynomial in $1/\epsilon$*,* $1/\delta$*, and size(c).*

ϵ *is the error parameter*
δ *is the confidence parameter*

Assume each call to EX(c,D) takes one unit of time.

Concept class C = axis-aligned rectangles is efficiently PAC-learnable.

Families of Problems

C_n = *concepts over n variables.*

points in $\mathbb{R}^n$
variable assignments in $\{0, 1\}^n$

Family $C = \cup_{n \geq 1} C_n$

C is efficiently PAC-learnable if L runs in time polynomial in n*,* $1/\epsilon$*, and* $1/\delta$*.*

Learning Boolean Conjunctions

Given a set of variables $x_1, x_2, \ldots x_n$
Instance space $X = \{0, 1\}^n$ *of possible variable assignments.*

Concept class C of conjunctions of literals. Examples:

$x_1 \wedge \bar{x}_3 \wedge x_4$
$x_2 \wedge x_5 \wedge \bar{x}_5$ *(empty)*

This class is efficiently PAC-learnable.

Proof?

Learning Boolean Conjunctions

Initial hypothesis:

$h = x_1 \wedge \bar{x}_1 \wedge x_2 \wedge \bar{x}_2 \wedge \ldots \wedge x_n \wedge \bar{x}_n$

Algorithm:

Generate examples with EX(c,D).
Ignore negative examples.
For positive example ***a****:*
if $a_i = 0$, *delete* x_i *from* h
if $a_i = 1$, *delete* $\bar{x}_i$ *from* h

To meet desired confidence level, we need:

$$m \geq \frac{2n}{\epsilon}\left[\ln(2n) + \ln(\frac{1}{\delta})\right]$$

Proof...

Proof of Sample Complexity Bound for Boolean Conjunctions

$h(x)$ *contains at most* $2n$ *terms.*
Note that $Terms(h) \supseteq Terms(c)$.
Error only occur when

$$h(x)=0 \quad but \quad c(x)=1$$

Let z be a term in h but not in c.
Define $p(z) = Pr_{a \in D}[c(a)=1$ *and z is 0 in* $a]$

Every error in h is caused by at least one literal z.

So $error(h) \leq \sum_{z \in h} p(z)$

Proof (cont.)

Call a literal 'bad' if $p(z) \geq \epsilon/2n$.

Note: retaining the non-bad literals of h, even if not in c, cannot violate the error bound ϵ.

If h has no bad literals, then:

$$\begin{aligned} error(h) &\leq \sum_{z \in h} p(z) \\ &\leq 2n \cdot (\epsilon/2n) \\ &= \epsilon \end{aligned}$$

So the error constraint will be satisfied.

Proof (cont.)

For any bad literal z, a call to EX(c,D) will delete it with probability $> \epsilon/2n$.

Prob. of z remaining in h after m calls to EX(c,D) is

$$\leq (1-\epsilon/2n)^m.$$

Prob. that h has ***some*** *bad literal remaining after m calls to EX(c,D) is*

$$\leq 2n \cdot (1-\epsilon/2n)^m.$$

Proof (cont.)

To meet confidence bound, we need:

$$\begin{aligned} 2n\cdot(1-\epsilon/2n)^m &\leqslant \delta \\ (1-\epsilon/2n)^m &\leqslant \delta/2n && \textit{divide by 2n} \end{aligned}$$

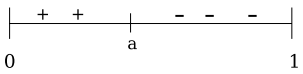
using $(1-x) \leqslant \exp(-x)$*, we get...*

$$\begin{aligned} \exp(-\epsilon\, m/2n) &\leqslant \delta/2n \\ -\epsilon\, m/2n &\leqslant \ln(\delta/2n) && \textit{take log} \\ \epsilon\, m/2n &\geqslant \ln(2n) + \ln(1/\delta) && \textit{negate} \\ m &\geqslant \frac{2n}{\epsilon}\left[\ln(2n)+\ln(1/\delta)\right] \end{aligned}$$

PAC Bounds for Continuous Spaces

PAC bound: if target $c \in C$*, sufficient to see*

$$m \geqslant \frac{1}{\epsilon}\left[\ln(|C|)+\ln(\frac{1}{\delta})\right]$$



Not a good bound for 'initial subintervals'.
Problem: $|C|$ *is infinite.*

But not that many 'really different' subintervals.
Intuitively, we should be measuring degrees of freedom.

PAC Bounds (cont.)

Define C[m] = maximum number of ways to split m points using concepts in class C.

Only C[m] "different" concepts in C w.r.t. m examples.

Theorem: if target $c \in C$*, then*

$$m \geqslant \left[\log_2(2C[2m]) + \log_2(\frac{1}{\delta})\right]$$

So "complexity" of class C has to do with the growth rate of C[m].

Examples of C[m]

What is C[m] for initial subintervals?

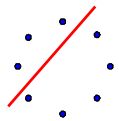
$m+1$

What is C[m] for intevals [a,b] ?

$m(m+1)/2 + 1$

What is C[m] for linear separators in the plane?

$m(m-1) + 2$



Examples of C[m]

What is C[m] for axis-parallel boxes?

$$\Theta(m^4)$$

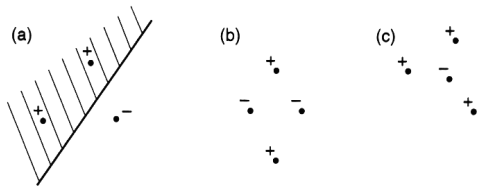
Can think of $\frac{\log C[m]}{\log m}$ *as the effective number of degrees of freedom.*

"Shattering" a Concept Class

Definition: a set of points S is **shattered** by a concept class C if there are concepts in C that split S in all of the $2^{|S|}$ possible ways.

In other words, all ways of classifying points in S are expressible in C.

Example: any 3 non-colinear points can be shattered by linear threshold functions in 2D. No set of 4 points can.



Vapnik-Chervonenkis Dimension

Definition: the **VC-dimension** of a concept class C is the size of the largest set of points that can be shattered by C.

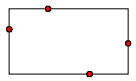
If VCdim(C) = d, that means there exists some set of d points that can be shattered, but there is no set of d+1 points that can be shattered.

Example: VCdim(linear threshold functions in 2D) is 3.

Examples of VC Dimension

C = "intervals of the real line" VCdim = 2

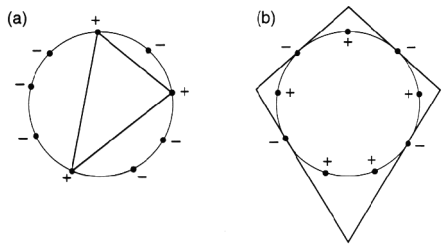
C = "axis-parallel boxes in 2D" VCdim = 4



Convex Polygons in the Plane

For convex d-gons in the plane, the VC dimension is 2d + 1.

Construction for (a) fewer positive labels, (b) fewer negative labels.



More VC Dimension Examples

C = "monotone disjunctions of n features"

$x_1 \vee x_3 \vee x_7$ VC dim = n

C = "all functions on n features"

VC dim = 2^n

VC Dimension and Complexity

Theorem: $C[m] = O(m^{VCdim(C)})$

Theorem: *If target* $c \in C$, *then*

$$m = O\left(\frac{1}{\epsilon}\left[VCdim(C)\log(1/\epsilon) + \log(1/\delta)\right]\right)$$

Why is VC Dimension Important?

- Measures the "richness" or "power" of a representation for describing concepts.
- Tells us something about the difficulty of learning concepts in that space.
- Universal measure: applies to neural nets, decision trees, Boolean formulas, etc.

VC Dimension of Perceptrons

Theorem: for $n \geq 1$, let P_n be the simple real perceptron with n inputs. Then:

$$VCdim(P_n) = n + 1$$

Proof:

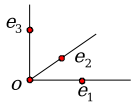
Use Radon's theorem to show that $n+2$ points in $\mathbb{R}^n$ cannot be shattered.

Show by construction that n+1 points can be shattered.

Shattering n+1 Inputs

Consider points in $\mathbb{R}^n$.
Let e_i be the point with coordinate i=1, rest zero.
Let o be the origin. Let $T = \{o, e_1, e_2, \ldots, e_n\}$

Let T_1 be some concept that classfies points in T.
If $e_i \in T_1$ set weight $w_i = +1$, else set $w_i = -1$
If $o \in T_1$ set threshold $\theta = -1/2$, else set $\theta = +1/2$.
The resulting perceptron recognizes concept T_1.



VC Dimension of Feedforward Nets

Theorem due to Cover (1968), Baum &Haussler (1989):

Let Q be an arbitrary feedforward neural net with w weights that consists of linear threshold gates.

Then VCdim(Q) = O(w log w).

Feedforward Networks (cont.)

Theorem due to Karpinski & MacIntyre (1995):

Let Q be a feedforward network with a linear threshold unit as output unit, and the remaining N units having the standard sigmoid activation function.

If Q has w variable weights and thresholds, then

$$VCdim(Q) \leq (wN)^2 + 11wN \log_2 (18wN^2)$$

Piecewise Polynomial Activation Functions

Consider an arbitrary feedforward neural network containing w weights, whose units employ piecewise polynomial activation functions.

Goldberg & Jerrum (1995): if depth is unbounded, VC-dimension grows as $O(w^2)$.

Bartlett et al. (1998): if depth is bounded, then VC-dimension grows as O(w log w).