REVIEW

PAC LEARNING

- find $h \in \mathcal{H}$ that makes no mistakes

$$Pr[err_{true}(h) > \varepsilon] \leq |\mathcal{H}| e^{-m\varepsilon}$$

$$err_{true}(h) \leq \frac{\ln|\mathcal{H}| + \ln(1/\delta)}{m}$$

INCONSISTENT HYPOTHESES

- find $h \in \mathcal{H}$ that makes few mistakes

$$Pr[err_{true}(h) > err_{train}(h) + \varepsilon] \leq |\mathcal{H}| e^{-2m\varepsilon^2}$$

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

- $err_{true}(h)$: WANT SMALL
- $err_{train}(h)$: "BIAS"
- $\sqrt{\frac{\ln|\mathcal{H}| + \ln(1/\delta)}{2m}}$: "VARIANCE"

CODING-BASED BOUNDS

- $\ln|\mathcal{H}|$ can be replaced by $\log_2 e \cdot size(h)$ ← IN BITS

CODING FIXED IN ADVANCE

OCCAM'S RAZOR: SIMPLER HYPOTHESES THAT EXPLAIN THE DATA HAVE BETTER GENERALIZATION ERROR

WHY DO WE CARE ABOUT BOUNDS ?

dec. trees:

$$err_{true}(h) \leq err_{train}(h) + \sqrt{\frac{2k + (k-1)\log_2 n + \log_2(1/\delta)}{2\log_2 2 \cdot m}}$$

- CAN OPTIMIZE THEM DIRECTLY!

  $$\min \{ error + penalty \}$$

  UNLIKE REGULARIZATION, NO TUNING

  PROBLEM: bounds are too loose

  INSIGHT: we can still use them to derive a functional form of regularization

- BOUNDS TELL US HOW DIFFERENT PARAMETERS AFFECT OVERFITTING (e.g. # leaves vs # attributes in a decision tree)

CONTINUOUS HYPOTHESIS SPACES

(e.g. hypotheses = half-spaces)

- an alternative complexity measure: VC dimension

INSIGHT:

- the number or encoding of hypotheses shouldn't matter, what matters: WHAT HYPOTHESES DO WITH MY DATA!

DICHOTOMY of a set of instances S:
a partition into two disjoint subsets
(positively labeled & negatively labeled)

$$Pr[err_{true}(h) > err_{train}(h) + \varepsilon]$$

$$\leq \left| \begin{array}{l} \text{worst-case \# dichotomies} \\ \text{induced by } \mathcal{H} \text{ on } m \text{ points} \end{array} \right| \cdot 8e^{-m\varepsilon^2/32}$$

BEFORE: THIS = $|\mathcal{H}|$

BEFORE: THIS = $e^{-2m\varepsilon^2}$

BOUND DUE TO VAPNIK-CHERVONENKIS 1971, improved since then (only constants improved)

EXAMPLES

- one-dimensional half-spaces

for m points: 2m DICHOTOMIES

- decision stumps over n continuous attributes

given m points, we get at most 2m DICHOTOMIES considering splits on a single attribute (as in one-dimensional half-spaces)

across all attributes: #DICHOTOMIES $\leq 2mn$

- two-dimensional half-spaces ?

:) NEED A NEW IDEA!

Def A set of instances S is <u>SHATTERED</u> by $\mathcal{H}$ if for every dichotomy there exists $h \in \mathcal{H}$ consistent with that dichotomy.

Def The VAPNIK-CHERVONENKIS DIMENSION of $\mathcal{H}$, written $VC(\mathcal{H})$, is the largest number of instances shattered by $\mathcal{H}$. If arbitrarily large sets can be shattered, then $VC(\mathcal{H}) = \infty$.

EXAMPLES

- 1D half-spaces:

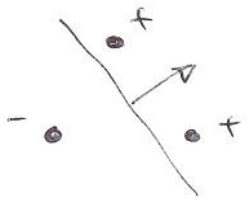
CAN SHATTER 2

↳ $VC(\mathcal{H}) = 2$

CANNOT SHATTER 3

[this is only intuition; proof requires that NO THREE POINTS CAN BE SHATTERED]

- 2D half-spaces



CAN DO 3

CANNOT DO 4

[again, proof requires that NO FOUR POINTS CAN BE SHATTERED]

$VC(\mathcal{H}) = 3$

EXAMPLE

- d-DIMENSIONAL HALFSPACES

CAN SHATTER d+1 POINTS
- I get to pick the points and show a hypothesis for each dichotomy
- INTUITION: I have sufficiently many degrees of freedom for d+1 points

CANNOT SHATTER d+2 POINTS
- no matter which d+2 points I choose an adversary can come up with a labeling I cannot represent
- INTUITION: I don't have sufficiently many degrees of freedom for d+2 points

$$VC(\mathcal{H}) = d+1$$

WHY IS VC DIMENSION USEFUL?
- Can be used to bound # dichotomies:

$$\#\text{dichotomies} \leq O(m^{VC(\mathcal{H})})$$

THUS MAX #DICHOTOMIES OF m POINTS (for a fixed $\mathcal{H}$)

$$= \begin{cases} 2^m \\ \text{or } \leq O(m^k) \end{cases}$$

I.E.: EITHER ALL POSSIBLE, OR BOUNDED BY A POLYNOMIAL.

AN EXAMPLE VC BOUND:

$$err_{true}(h) \leq err_{train}(h) + \sqrt{\frac{\overbrace{VC(H)\ln\left(\frac{2m}{VC(H)}\right)}^{\text{BEFORE: } \ln|H|} + \ln\frac{4}{\delta}}{m}}$$

LINEAR CLASSIFIER IN 2-dimensions	larger	smaller
LIN. CLASSIFIER IN 1,000 dimensions	small	larger

ONLINE LEARNING

- see an example x_t
- predict a label $\hat{y}_t$
- observe true label y_t

(loop forever)

NO STATISTICAL ASSUMPTIONS

GOAL: minimize #mistakes

[NO NOTION OF TRAINING ERROR AND TRUE ERROR]

=

LEARNING from a set of labeled examples with statistical assumptions

⇙

BATCH LEARNING

=

LEARNING W/ EXPERT ADVICE

EXAMPLE: want to predict if Dow Jones will go up or down on a given day

APPROACH: follow N experts and combine their predictions to calculate your own

CAN WE APPROACH THE PERFORMANCE OF THE BEST EXPERT?

SIMPLE CASE:

LEARNING WITH A PERFECT EXPERT

[Halving Algorithm]

For each round $t = 1, 2, \ldots, T$

- take a majority vote among experts that made no mistakes so far
- predict $\hat{y}_t$ = result of vote
- observe y_t

HOW MANY MISTAKES DO WE MAKE?

- best case: no mistakes
- worst case

W = # surviving experts

initially: $W = N$

after one mistake: $W \leq \frac{1}{2} N$

k mistakes: $W \leq \left(\frac{1}{2}\right)^k N$

since W cannot shrink below one:

$$\#\text{mistakes} \leq \log_2 N$$

↖ MISTAKE BOUND

AVERAGE #MISTAKES:

$$\frac{\#\text{mistakes}}{T} \leq \frac{\log_2 N}{T}$$

ANALOGY WITH PAC LEARNING

- experts can be anything
  (e.g. different learning algorithms
  or different hypotheses)

If experts ≙ hypotheses

then perfect expert ≙ concept $c \in \mathcal{H}$ [CONSISTENT SETTING]

$$N = |\mathcal{H}|$$

MISTAKE BOUND FOR ONLINE LEARNING:

$$\frac{\#\text{mistakes}}{T} \leq \frac{\log_2 |\mathcal{H}|}{T}$$

- T: # examples seen in T steps
- mistakes: average error

CORRESPONDING PAC BOUND FOR BATCH LEARNING from $\boxed{m = T}$ i.i.d. examples

$$err_{true}(h) \leq \frac{\ln|\mathcal{H}| + \ln(1/\delta)}{T}$$

h: ANY HYPOTHESIS CONSISTENT WITH DATA