Peer Eval for PS2 due today

PS3 due Thu 10/29

Midterm Thu 11/5 in lecture

I'll post some solutions to exercises from lecture notes

Last time: lower bounds on sample complexity for PAC learning

- Reveals role of VC dim in determining whether C is PAC-learnable

Today: Boosting

Boosting

[Strong] PAC learning for $\mathcal{C}$

Alg. A is a PAC learning alg for $\mathcal{C}$ if:

for any $\epsilon \in (0,1)$, $\delta \in (0,1)$,

any dist. P satisfying $\mathcal{C}$-realizability

if A receives an iid sample of size-n from P as input

$n = n\left(\frac{1}{\epsilon}, \frac{1}{\delta}, VCdim(\mathcal{C})\right)$

then w.p. $\geq 1-\delta$,

A returns $\hat{h}: \mathcal{X} \to \{0,1\}$ w/ $err(\hat{h}; P) \leq \epsilon$.

Weak PAC learning for $\mathcal{C}$

Alg. A is a weak PAC learning alg for $\mathcal{C}$ if for some $\epsilon = \epsilon_0 < \frac{1}{2}$

for any ~~$\epsilon \in (0,1)$~~, $\delta \in (0,1)$,

any dist. P satisfying $\mathcal{C}$-realizability

if A receives an iid sample of size-n from P as input

$n = n\left(\frac{1}{\epsilon}, \frac{1}{\delta}, VCdim(\mathcal{C})\right)$

then w.p. $\geq 1-\delta$,

A returns $\hat{h}: \mathcal{X} \to \{0,1\}$ w/ $err(\hat{h}; P) \leq \epsilon$.

Kearns & Valiant 1989

If there's an efficient Weak PAC learner for $\mathcal{C}$ must there exist an efficient Strong PAC learner for $\mathcal{C}$?

Answer: Yes

Schapire 1990

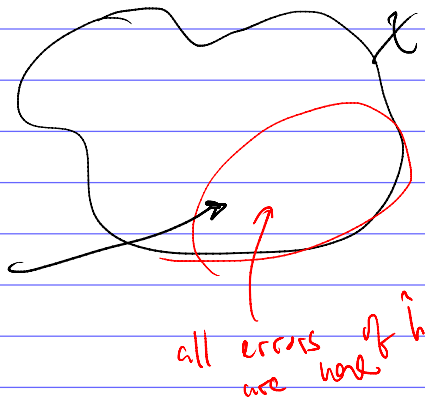
Freund 1990

Freund & Schapire 1997 ← AdaBoost.

Boosting alg: algorithmic "proof" that KSV's question is answered affirmatively

All boosting algorithms take advantage of the requirement that W.L.'s work for <u>all</u> C-realizable distributions.

- Get $S \sim P^n$
  $\hat{h} \leftarrow WL(S)$
- Define a new dist Q that's concentrated here



Get $T \sim Q^n$
$\hat{h}' \leftarrow WL(T)$

Combining $\hat{h}$ & $\hat{h}'$ should be better

Technical issue: how to get sample from Q if you only have access to samples from P?

- Rejection sampling.

↑

This approach is called ~~Boosting by Filtering~~ Boosting for filtering setting

Today: ~~Boosting by sample~~ Boosting for batch setting

Boosting ~~by sample~~ for batch setting

In-sample vs Out-of-sample

In-sample S: How to use WL to find a $\hat{h}$ w/ $\text{err}(\hat{h}; S) = 0$

Out-of-sample: Does $\hat{h}$ then also work well on P? $\text{err}(\hat{h}; P)$

Training data set $S = ((x_1, y_1), \ldots, (x_n, y_n)) \in (\mathcal{X} \times \{-1, 1\})^n$

Weak learner (for "in-sample") is an alg that takes weighted data set (S, p) where $p \in \Delta^{n-1}$ and returns $h: \mathcal{X} \to \{-1, 1\}$ w/

$$\text{err}(h; S, p) = \sum_{i=1}^{n} p_i \mathbb{1}\{h(x_i) \neq y_i\} \leq \frac{1-\gamma}{2}$$

$\gamma > 0$ is the "advantage" over random guessing.

Weak Learning Assumption (WLA): such a WL exists

$$\text{acc}(h; S, p) = \sum_{i=1}^{n} p_i \mathbb{1}\{h(x_i) = y_i\}$$

Alternative characterization of WLA

Assume WL always selects $\hat{h} \in \mathcal{H} = \{h_1, \ldots, h_d\}$

Define $n \times d$ matrix A

$$A_{ij} = \mathbb{1}\{h_j(x_i) = y_i\}$$

WLA: $\forall\, p \in \Delta^{n-1}$ $\exists\, j \in [d]$ s.t.

$$\mathrm{acc}(h_j; S, p) = \sum_{i=1}^{n} p_i \, \mathbb{1}\{h_j(x_i) = y_i\} \geq \frac{1+\gamma}{2}$$

i.e.,

$$p^\top A e_j \geq \frac{1+\gamma}{2}$$

$e_j = j^{th}$ coord. basis vector in $\mathbb{R}^d$.

$$\min_{p \in \Delta^{n-1}} \max_{j \in [d]} p^\top A e_j \geq \frac{1+\gamma}{2}$$

$$\min_{p \in \Delta^{n-1}} \max_{q \in \Delta^{d-1}} p^\top A q \geq \frac{1+\gamma}{2}$$

By U.N.M.M.T.

$$\min_{p} \max_{q} p^\top A q = \max_{q} \min_{p} p^\top A q \geq \frac{1+\gamma}{2}$$

$$\max_{q \in \Delta^{d-1}} \min_{i \in [n]} e_i^\top A q \geq \frac{1+\gamma}{2}$$

$$\exists\, q \in \Delta^{d-1} \quad \forall\, i \in [n] \quad e_i^\top A q \geq \frac{1+\gamma}{2}$$

$$\sum_{j=1}^{d} q_j \boxed{\mathbb{1}\{h_j(x_i) = y_i\}} \geq \frac{1+\gamma}{2}$$

$$\| \quad \frac{1 + y_i h_j(x_i)}{2}$$

$$\sum q_j \frac{1 + y_i h_j(x_i)}{2} \geq \frac{1+\gamma}{2}$$

$$y_i \sum_{j=1}^{d} q_j h_j(x_i) \geq \gamma \qquad \forall\, i \in [n]$$

$$\phi : \mathcal{X} \to \mathbb{R}^d$$

$$\phi(x) = (h_1(x), h_2(x), \ldots, h_d(x))$$

This HLTF

$$x \mapsto \mathrm{sign}\left(q^\top \phi(x)\right)$$

classifies all training examples correctly.