Lecture 14:

PAC learnability

Uniform convergence

Recall:

Probably Approximately Correct (PAC)

X instance space — set of all instances

(input space / domain)

$c: X \to \{+1, -1\}$ concept — a function to label elements

C concept class — a collection of labeling functions

c^* target concept — $c^* \in C$ and label all instances correctly

D target distribution — distribution over instances

sample / training data set $\left\{ \begin{array}{l} \langle x_1, c^*(x_1) \rangle \\ \langle x_2, c^*(x_2) \rangle \\ \vdots \\ \langle x_n, c^*(x_n) \rangle \end{array} \right.$

Recall

PAC learning (Probably Approximately Correct)

Suppose that we have a concept class C over X. We say that C is PAC learnable if there exists an algorithm A s.t:

$\forall\ c \in C$, $\forall\ D$ over X, $\forall\ \varepsilon, \delta \in (0, 0.5]$

A receives ε, δ, and samples $\langle x_1, c(x_1) \rangle$ $\ldots, \langle x_n, c(x_n) \rangle$ where x_i's are iid samples from D.

Then, w. p. $\geq 1-\delta$, A outputs $\hat{c}$ $\underset{\text{proper}}{[\in C]}$ s.t.

$$\text{err}(\hat{c}) \leq \varepsilon.$$

The probability is taken over the randomness in the samples and any internal coin flips of A.

other notation

true error:

$$\mathrm{err}(c) = \Pr_{(x,y)\sim D}\left[c(x) \neq y \right]$$

training error:

$$\widehat{\mathrm{err}}(c) = \frac{\#\text{ samples in } T \text{ s.t. } c(x_i) \neq y_i}{|T|}$$

fraction of samples in the training set that c is mis-labeled.

Example 2 Boolean conjunctions :

$X = \{0,1\}^n$ literals $\begin{cases} x_i \\ \bar{x}_i \end{cases}$

Conjunction = $\begin{cases} \text{literal} \\ \text{literal} \wedge \text{conjunction} \end{cases}$

$\wedge \longrightarrow$ logical and

concept : a conjunction

example : $h(x) = x_1 \wedge \bar{x}_2$ $\quad h((1,0,1)) = 1$

$x = (x_1, \dots, x_n)$ $\quad h((0,0,1)) = 0$

$\mathcal{H}$: the set of all conjunction function

Goal: PAC learning of $\mathcal{H}$

Suppose we have samples of the form $\langle x, h^*(x) \rangle$ from a distribution D

$\hookrightarrow$ realizable

Algorithm:

- start with $h = x_1 \wedge \bar{x}_1 \wedge x_2 \wedge \bar{x}_2 \wedge \ldots \wedge x_n \wedge \sim x_n$
- Try $m = ?$ examples
    - ignore negative example.
    - for positive example, remove inconsistent terms.
- Output h

Deleting an inconsistant literal

$h = \boxed{x_2} \wedge x_3 \wedge \bar{x}_4$

sample: $\langle (1, \boxed{0}, 1, 0), 1 \rangle$ → x_2 is inconsistant

↓

we delete x_2 from h

↓

new $h = x_3 \wedge \bar{x}_4$

Our goal is to analyze the performance of the algorithm.

First, we start by the error of the output hypothesis $\hat{h}$.

Initially, h contains all literals. we only remove inconsistent literals. So, we never removes literals in h^* from h. That is, $\hat{h}$ contains all the literals in h^*. This fact implies if $h^*(x) = 0$, $\hat{h}(x)$ must be zero too

$\Rightarrow$ Hence $\hat{h}$ always labels x correctly if $h^*(x) = 0$

Now consider the rest of the domain elements x such that $h^*(x) = 1$

If $\hat{h}$ makes a mistake (i.e. $\hat{h}(x) = 0$), there must be a literal in $\hat{h}$, z that is inconsistent:

$$\text{true error of } \hat{h} = err(\hat{h})$$

$$= \Pr_{x \sim D}[\hat{h}(x) \neq h^*(x)]$$

$$= \Pr_{x \sim D}\left[\begin{array}{l}\exists \text{ a literal } z \in \hat{h} \text{ such that } z \\ x_{|z} = 0 \text{ but } h^*(x) = 1\end{array}\right]$$

$$\leq \sum_{z \in \hat{h}} \underbrace{\Pr_{x \sim D}[x_{|z} = 0 \text{ but } h^*(x) = 1]}_{\text{call this } p(z)}$$

by the union bound

$$= \sum_{z \in \hat{h}} p(z) \quad *$$

We call a literal bad iff $p(z)$ is at most $\frac{\varepsilon}{2n}$

$$\text{bad } z \iff p(z) > \frac{\varepsilon}{2n}$$

Using * it is easy to see it no bad literal survives in $\hat{h}$ then

$$\text{err}(\hat{h}) \leq \sum_{z \in \hat{h}} p(z) \leq 2n \cdot \frac{\varepsilon}{2n} \leq \varepsilon$$

$\Rightarrow$ hence the error of $\hat{h}$ is good

Now, let's focus on the probability of $err(\hat{h}) > \varepsilon$

$$\Pr_T[\text{outputting an inaccurate } \hat{h}]$$

training set

$$= \Pr_T[err(\hat{h}) > \varepsilon]$$

$$\leq \Pr[\exists \text{ a bad literal } z \text{ in } \hat{h}]$$

$$\leq 2n \cdot \Pr\left[\begin{array}{l}\text{a bad literal survives} \\ \text{all the } m \text{ samples (not} \\ \text{been deleted)}\end{array}\right]$$

$$\leq 2n \cdot (1 - p(z))^m$$

It is not hard to see that with probability $p(z)$ we delete z at every round.

$$\leq 2n \left(1 - \frac{\varepsilon}{2n}\right)^m$$

$$\leq 2n\, e^{-\frac{\varepsilon m}{2n}} \overset{?}{\leq} \delta$$

by setting $m = \frac{2n}{\varepsilon} \log(2n/\delta)$

Hence our algorithm with prob. $1-\delta$ output $\hat{h}$ that has low error. $(\mathrm{err}(\hat{h}) \leq \varepsilon)$

$\Rightarrow$ we PAC-learned $\mathcal{H}$:)

ERM

In both example we picked concepts $\hat{R}$ and $\hat{h}$ that were consistent with the samples in the training set

What we did is called :

ERM : Empirical Risk Minimization

comes from samples   error

ERM algorithm: it finds a concept $\hat{h}$ such that $\hat{err}(\hat{h}) = 0$

+ Uniform convergence (UC)

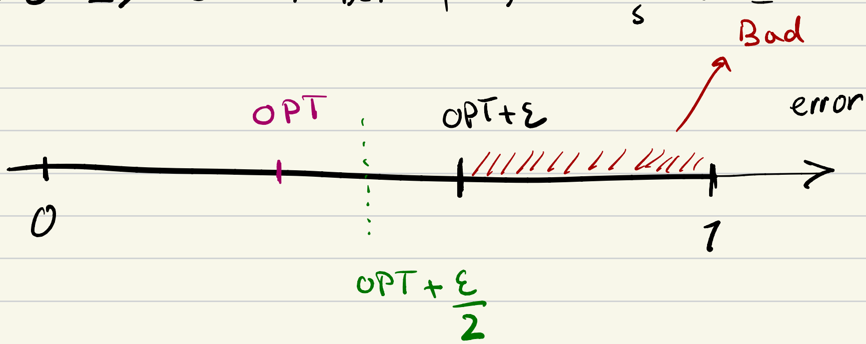
Class C has the uniform convergence property if $\forall\, \varepsilon, \delta \in (0,1)$, dist D $\exists\, m$ (as a function of $\varepsilon, \delta, \mathcal{H}$, but not D since we don't know D). s.t. for a training set of size m:

$$\Pr_{T\sim D^m}\left[\forall c\in C: \left|\hat{err}_T(c) - err(c)\right| \le \varepsilon\right] \ge 1-\delta$$

$\boxed{\text{Uniform convergence implies agnostic PAC learnability via EMR.}}$

$UC \Rightarrow \forall c \in C_B \quad \hat{err}_S(c) > OPT + \varepsilon/2$

$UC \Rightarrow c^* =$ the best option) $\quad \hat{err}_S(c^*) \le OPT + \varepsilon$



There are two types of error in the agnostic setting:

$$\mathrm{err}(\hat{c}) < \underbrace{\min_{c \in C} \mathrm{err}(c)}_{\varepsilon_{app} := \text{approximation error}} + \underbrace{\varepsilon}_{\varepsilon_{est} := \text{estimation error}}$$

↓

depends only to the choice of the class C

- Is C rich enough to capture how data is labeled?

C	ε_{app}	ε_{est}
larger		
more complex		
↑	↓	↑