Notes 8: Online to PAC conversion

1. Online to PAC

Theorem 1. *If online algorithm A learns $\mathcal{C}$ with $\leqslant M$ mistakes, then some algorithm PAC-learns $\mathcal{C}$ using*

$$m = \frac{M+1}{\varepsilon} \ln \frac{M}{\delta} \qquad \text{samples}$$

Proof. Can assume A only updates its hypothesis after making a mistake (homework)

PAC Learning Algorithm

> Keep feeding to A independent samples from $\mathrm{EX}(c, \mathcal{D})$
> Until A correctly classifies $\frac{1}{\varepsilon} \ln \frac{M}{\delta}$ samples in a row
> Then output A's current (i.e. last) hypothesis h

A's predictions:

$$\overbrace{\underbrace{✓✓\cdots✓}_{<\frac{1}{\varepsilon}\ln\frac{M}{\delta}}✗}^{h_1}\overbrace{\underbrace{✓✓\cdots✓}_{<\frac{1}{\varepsilon}\ln\frac{M}{\delta}}✗}^{h_2}\text{ (repeat } \leqslant M \text{ times) }\overbrace{\underbrace{✓✓\cdots✓}_{\leqslant\frac{1}{\varepsilon}\ln\frac{M}{\delta}}}^{h_{\text{last}}}$$

$\leqslant M+1$ hypotheses, each applied to $\leqslant \frac{1}{\varepsilon} \ln \frac{M}{\delta}$ samples
#samples used $\leqslant \frac{M+1}{\varepsilon} \ln \frac{M}{\delta}$

We now argue final hypothesis h_{last} has error $\leqslant \varepsilon$ with prob. $\geqslant 1-\delta$

If $\mathrm{err}_{\mathcal{D}}(h_i, c) \geqslant \varepsilon$: $\quad \mathbb{P}\left[h_i \text{ correct } k \stackrel{\text{def}}{=} \frac{1}{\varepsilon} \ln \frac{M}{\delta} \text{ times}\right] \leqslant (1-\varepsilon)^k \leqslant e^{-\varepsilon k} = \dfrac{\delta}{M}$

A uses $\leqslant M+1$ hypotheses $h_1, \ldots, h_{\text{last}}$

$$\mathbb{P}[\text{any of them has error} \geqslant \varepsilon \text{ and correct } k \text{ times}] \leqslant M \cdot \frac{\delta}{M} = \delta$$

Union bound over M (not $M+1$) because if $h_{\text{last}} = h_{M+1}$ then h_{last} has zero error
for otherwise A may make $M+1$ mistakes ∎

If A efficient, so is its PAC version
Implies PAC learning algorithms for
e.g. (sparse) conjuctions/disjunctions, short decision lists, well-seperated LTFs
e.g. monotone disjunctions: Elimination Algorithm makes $\leqslant n$ mistakes
its PAC version uses $O\left(\frac{n}{\varepsilon} \ln\left(\frac{n}{\delta}\right)\right)$ samples

---

2. PAC to Online? No

$X = \text{unit interval} = [0,1]$ $\qquad \mathcal{C} = \text{initial intervals} = \{[0,b] \mid 0 \leqslant b \leqslant 1\}$
where $[0,b] = \{x \in \mathbb{R} \mid 0 \leqslant x \leqslant b\}$
Can be PAC learned with $(1/\varepsilon)\ln(1/\delta)$ samples (same idea as axis-aligned rectangles)

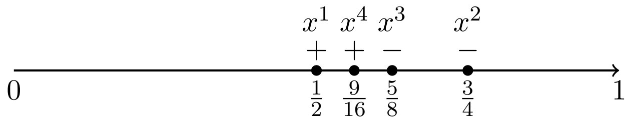
Claim 2. *Any algorithm A for learning closed intervals over $[0,1]$ in the Online model makes an arbitrarily large number of mistakes*

Proof. The adversary below forces A to always err

Adversary

> Initially $I = [0,1]$
> Repeat
> Set $x =$ midpoint of I
> Feed x to A and gets A's prediction
> Label x opposite to A's prediction
> If x's correct label is 0, shrinks I by keeping only its left half, else keep only its right half

e.g. 1st round $x^1 = 1/2$, if A predicts x^1 as 0, then label x^1 as 1, update I as $[1/2, 1]$



All positive samples to the left of all negative samples
Some initial interval correctly classifies all labelled samples so far □

X above is infinite
How about finite X?
Efficient PAC algorithm for $\mathcal{C}$ over finite X implies efficient online algorithm with few mistakes?
Previous example of initial intervals (now over $X = \{1, 2, \ldots, n\}$) has efficient online algorithm
namely Halving algorithm with $\leqslant \log n$ mistakes
In fact Halving algorithm has very efficient implementation in this case (binary search)
Under reasonable cryptographic assumptions, still no PAC-to-online conversion for finite $X = \{0, 1\}^n$