Last time: Randomized WMA : $\mathbb{E}[\#\text{mist.}] \le \dfrac{m \ln 1/\beta + \ln N}{1-\beta}$

Started Probably Approximately Correct (PAC) Learning

- motivation
- basic definition
- started learning $\mathcal{C}$ = intervals of $\mathbb{R}$

Today:

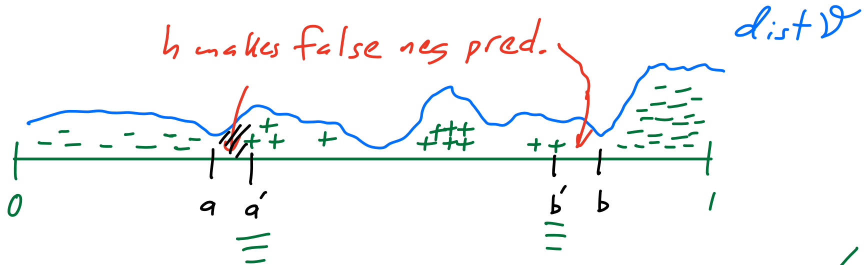
- finish
- OLMB $\rightarrow$ PAC conversion (get all our OLMB results "for free" in PAC setting ☺)
- Revisit PAC learning def:
  - "size" of concepts
  - efficiency of evaluating h
- "Chernoff bounds" + applic. to hypothesis testing
  (tail bounds on sums of indep. bounded random vars.)

Questions?

Recall "learning intervals" problem: $X = [0,1]$, $\mathcal{C} =$ all $[a,b]$

Alg:

data pic.



Alg: $\rightarrow$ a' = smallest pos. ex. in m samples

b' = largest " " " " "

unknown!

$[a,b]$ = target c

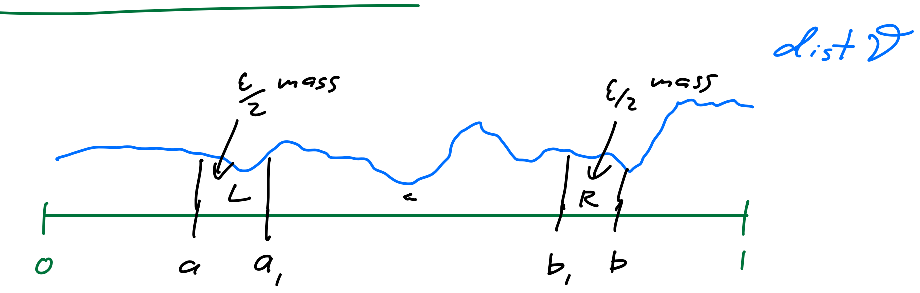
Alg outputs hyp $h = [a', b']$ (proper)

$\geq 1-\delta$

Note: only errors of h are false negative: $h=0$, $c=1$

To show: whp h doesn't miss more than ε amount of $\mathcal{D}$'s probability mass.

Analysis:



$[a,b]$ = target c

Define a_1 = value s.t. $\Pr_{x\sim\mathcal{D}}[x\in[a,a_1]] = \varepsilon/2$, &

Define b_1 = value s.t. $\Pr_{x\sim\mathcal{D}}[x\in[b_1,b]] = \varepsilon/2$.

$L=[a,a_1]$

$R=[b_1,b]$.

Obs:

If our sample of lab. ex. contains a pt in L, &
" " " " R,

then $err_{\mathcal{D}}[h,c] \leq \overset{\text{from left}}{\frac{\varepsilon}{2}} + \overset{\text{from right}}{\varepsilon/2} = \varepsilon$:)

So want to u.b. $\Pr[\text{miss } L]$, $\Pr[\text{miss } R]$.

Since a single $x\sim\mathcal{D}$ misses L w.p. $1-\varepsilon/2$,
have $\Pr[m \text{ i.i.d. ex. all miss } L] = (1-\varepsilon/2)^m$.
(independence!)

Likewise $\Pr[m \text{ i.i.d. ex. all miss } R] = \underline{(1-\varepsilon/2)^m}$.

By union bd ($\Pr[A \text{ or } B] \le \Pr[A] + \Pr[B]$),

suff. to have $\boxed{\left(1-\frac{\varepsilon}{2}\right)^m \le \frac{\delta}{2}}$: gives

$\Pr[m \text{ ex all miss } L \text{ or all miss } R] \le \delta$.

So m must sat. $\left(1-\frac{\varepsilon}{2}\right)^m \le \frac{\delta}{2}$:

recall $\boxed{1-x \le e^{-x}}$; so $\boxed{(1-x)^{\frac{1}{x}} \le e^{-1}}$

$m = \frac{2}{\varepsilon}\ln(2/\delta)$ suff, b/c

$$\underbrace{\boxed{\left(1-\frac{\varepsilon}{2}\right)^{\frac{2}{\varepsilon}}}}_{\le \frac{1}{e}}{}^{\cdot \ln(2/\delta)} = \left(e^{-1}\right)^{\ln(2/\delta)} = \frac{\delta}{2}.$$

δ = confidence; $1-\delta = \Pr[\text{hyp } h \text{ is "good"}]$

↳ "guards against unlikely samples"

ε = error; even "good" h is allowed to err on $\le \varepsilon$ frac. of $\mathcal{D}$

PAC vs OLMB Learning

Sps A is an OLMB alg. for $\mathcal{C}$ with mist. bound M.
Can we get a PAC learner from A? Y!

Thm: Let . Then there's a PAC learner A' for $\mathcal{C}$ with sample complexity m, where

$$m \le M + \frac{M+1}{\varepsilon} \cdot \ln\left(\frac{M+1}{\delta}\right) \checkmark$$

Pf: WLOG, have that A takes it easy (only changes h when makes mistake).
PAC alg A' is:

A'
- run A on seq of fresh i.i.d. draws from $EX(c, \mathcal{D})$.
- keep track of # ex. current h got right; if h ever goes for

$$\frac{1}{\varepsilon} \cdot \ln\left(\frac{M+1}{\delta}\right)$$

ex w/o a mistake, stop + output h.

Claim: sample cxity of A' is as claimed.

$\checkmark$ = hyp was right

$\times$ = hyp was wrong

$$\overline{\underbrace{\checkmark\,\checkmark\ldots\checkmark}_{\leq \frac{1}{\varepsilon}\ln\frac{M+1}{\delta}}}\ \times\ \overline{\underbrace{\checkmark\,\checkmark\ldots\checkmark}_{\leq \frac{1}{\varepsilon}\ln\frac{M+1}{\delta}}}\ \times\ \overline{\checkmark\ldots\checkmark}\ \times\ \overline{\checkmark\ldots\ldots\checkmark}$$

$\leq M$ x's in sequence

M x's

$M+1$ blocks of $\checkmark$'s, each block $\leq \frac{1}{\varepsilon}\ln\frac{M+1}{\delta}$

Why is A' PAC?

"h bad" means $\underline{err_{\mathcal{D}}[h,c] > \varepsilon}$.

Sps <u>curr. hyp h is bad.</u>

$\Pr[h$ gets k ex right in a row$] <^{\text{indep!}} (1-\varepsilon)^k$.

So $\Pr[h$ " $k = \frac{1}{\varepsilon}\ln\left(\frac{M+1}{\delta}\right)$ " " " "$] <$

$$(1-\varepsilon)^{\frac{1}{\varepsilon}\ln\left(\frac{M+1}{\delta}\right)} \leq e^{-\ln\left(\frac{M+1}{\delta}\right)} = \frac{\delta}{M+1}.$$

Since <u>A takes it easy</u>, only ever uses $\leq M+1$ diff h's. Hence At most $M+1$ bad h's ever used by A'.]
M

So UB $\Rightarrow \Pr[$<u>any</u> bad hyp is output by A']

$$\leq \frac{\delta}{M+1}\cdot(M+1) = \delta.$$

So A' is PAC.

Note: if A comput. eff, so is A'.

So we can PAC learn conj, disj, DL, r-out-of-k thr. fns, etc.

Ex: plug in elim alg for mon disj:

$$m(\varepsilon, \delta) = n + \frac{n+1}{\varepsilon} \ln\left(\frac{n+1}{\delta}\right)$$ sample cxity

$$O\left(\frac{n}{\varepsilon} \cdot \ln\left(\frac{n}{\delta}\right)\right)$$

$\mathcal{C}$ is OLMB learnable

$\Downarrow$

$\mathcal{C}$ is PAC learnable.

? No: intervals.

Revisit PAC learning def; 2 more issues.

(1) "size" / "complexity" of target concept.

For some $\mathcal{C}$'s, natural notion of size(c) $\qquad c \in \mathcal{C}$.

Refined notion of eff. PAC learning: if you're learning $\mathcal{C}$ + target concept $c \in \mathcal{C}$ has $\text{size}(c) = s$, we measure time, sample cxity of learner also in terms of s:

$X = \{0,1\}^n$, eff PAC learner runs in time $\text{poly}(n, \frac{1}{\varepsilon}, \frac{1}{\delta}, \underline{\underline{\text{size}(c)}})$

Ex: DNF formulas

$X = \{0,1\}^n$: any $f: \{0,1\}^n \to \{0,1\}$ is an s-term DNF for large enough s.

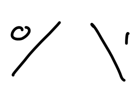
DNFSIZE(f) = min # terms in any DNF for f.

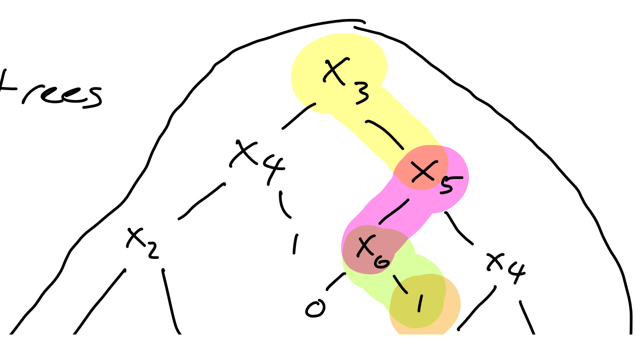
$x_1 x_2 \vee x_3 x_4$: 2-term DNF

$x_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_3 x_4$: same function, 3 terms

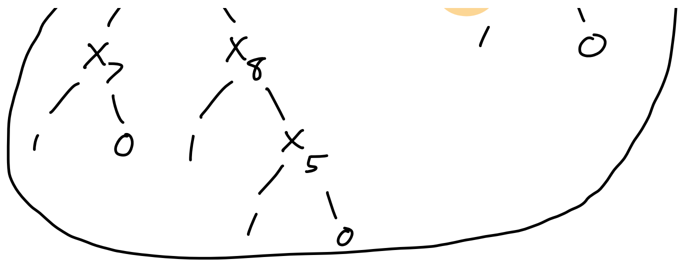
DNFSIZE(f) = 2.

EX: decision trees





$f(10110110) = 1$



DTSIZE(f) = min # leaves in any DT for f

Issue #2: hypotheses.

Learning alg. outputs a representation of a function $h : X \to \{0,1\}$, i.e. a program.

If a learning alg. outputs a program which takes a very long time to run, it's useless.

A hyp. class $\mathcal{H}$ (class of programs!) over $X = \{0,1\}^n$ or $\mathbb{R}^n$ is polynomially evaluatable if every $h \in \mathcal{H}$ can be run on any $x \in X$ in poly(n) time.

An "efficient PAC learning alg A":

runtime of A is polynomial, + for any h that A might output, runtime of h is polynomial, i.e.

Any "efficient PAC learning alg A" must use a polynomially evaluatable $\mathcal{H}$.

Next time: Chernoff bds,

hyp. testing

generic way to PAC learn.
