Last time: • Lower bound: Learning with mal. noise is hard
• (sad) pos. result for " " " "
• start learning with RCN

Today: Handling RCN.
• Revisit an old PAC learning alg (for mon. conj.) that's not RCN-tolerant, & adapt it to handle RCN
• Motivates new learning model:
STATISTICAL QUERY (SQ) LEARNING model
• Any SQ learning algorithm automatically yields a PAC learning alg that can handle RCN.

Questions?

Recall our old PAC alg for mon. conj. (elim.)

- start w/ $h(x) = x_1 \wedge \dots \wedge x_n$.
- draw set of ex from $EX(c, \mathcal{D})$
- for each pos ex $\underbrace{\text{pos ex}}_{a}$, if $x_i = 0$, remove x_i from $h(x)$.

Noiseless world: get a consistent mon conj, :)

Noise? $EX^{\eta}(c, \mathcal{D})$ may cause alg to fail: we may elim x_i's b/c of noisy examples
↳ that we should Keep.

Mistake to trust indiv. ex. too much.

Fix: look at aggregate statistics of the data.

More precisely:

Define

$$p_i := \Pr_{(x, c(x)) \sim EX(c, \mathcal{D})}[c(x) = 1 \ \& \ x_i = 0]$$

[if x_i in target, $p_i = 0$.]

Intuition: each x_i that isn't in c but is in a hyp h adds $\le p_i$ to error of h.

non conj

Good enough to identify (& eliminate from h) all x_i's s.t. $p_i \ge \varepsilon/n$. If could do this, tot error of h would be

bd on error of each erron. incl. x_i

bd on # of such

$$\le \left(\frac{\varepsilon}{n}\right) \cdot n = \varepsilon.$$

So to PAC learn non conj in presence of RCN, suff to est. each p_i to $\pm \varepsilon/2n$.

Our Goal

If no noise: :) draw m ex from $EX(c, \mathcal{D})$, compute $\hat{p}_i$ = frac of the m ex st $c(x) = 1$ & $x_i = 0$, use $\hat{p}_i$.

CB (add form):

$$\boxed{\begin{array}{l} |\hat{p}_i - p_i| \geq \tau \quad \text{w.p.} \leq \underbrace{2e^{-2\tau^2 m}}_{\text{want} = \delta}. \\ m = O\left(\frac{\ln(2/\delta)}{\tau^2}\right). \end{array}}$$

$\tau = \frac{\varepsilon}{2n}$, want each i to fail w.p. $\leq \frac{\delta}{n}$:

$$\boxed{m \approx O\left(\frac{\ln(2n/\delta) \cdot n^2}{\varepsilon^2}\right)}.$$

There's noise! What to do?

We have $EX^{\eta}(c, \mathcal{D})$.

$$p_i := \Pr_{(x, c(x)) \sim EX(c, \mathcal{D})}[c(x) = 1 \ \&\ x_i = 0]$$

$$= \overbrace{\underbrace{\Pr_{(x, c(x)) \sim EX(c, \mathcal{D})}[c(x) = 1 / x_i = 0]}_{q_i}}^{0 \leq q_i \leq 1} \cdot \overbrace{\Pr_{(x, c(x)) \sim EX(c, \mathcal{D})}[x_i = 0]}^{0 \leq \ \leq 1}$$

Obs: this is $= \text{to} \Pr_{(x, c(x)) \sim EX^{\eta}(c, \mathcal{D})}[x_i = 0]$, so

can estimate this given $EX^{\eta}(c, \mathcal{D})$, b/c "$x_i = 0$" unaffected by noise.

We est. $\Pr_{(x, c(x)) \sim EX(c, \mathcal{D})}[x_i = 0]$ & if it's $\leq \frac{\varepsilon}{4n}$: then $p_i \leq \varepsilon/4n$ & have good est of p_i.

So suppose $\Pr_{(x, c(x))\sim EX(c,\mathcal{D})}[x_i=0]$ is $\geq \frac{\varepsilon}{4n}$.

We know η

(maybe) can est. this given $EX^\eta(c,\mathcal{D})$

Have

$$\bigstar = \Pr_{(x,b)\sim EX^\eta(c,\mathcal{D})}[b=1 \mid x_i=0] = \underbrace{q_i}_{c(x)=1}\cdot\overbrace{(1-\eta)}^{\text{no noise}} + \underbrace{(1-q_i)}_{c(x)=0}\cdot\overbrace{\eta}^{\text{noise}}$$

$$= \eta + q_i(1-2\eta)$$

We can est (?) b/c we have $EX^\eta(c,\mathcal{D})$!

$$\Rightarrow \frac{\bigstar-\eta}{1-2\eta} = q_i$$

Careful: can only use draws from $EX^\eta(c,\mathcal{D})$ that have $x_i=0$ towards est. $\bigstar$

Need, on avg, $\frac{1}{\Pr_{x\sim\mathcal{D}}[x_i=0]}$ many draws from $EX^\eta(c,\mathcal{D})$ to get one sat. $x_i=0$.

If $\Pr_{x\sim\mathcal{D}}[x_i=0]$ very small, ineff; but we saw if it's $<\frac{\varepsilon}{4n}$, no need to est q_i

to get $\frac{\varepsilon}{4n}$-acc est. of p_i.

So we <u>can</u> eff. est. p_i

What happened?

- Reduced noisy learning problem to estimating some (noiseless) prob.'s. But we're in noisy world.
- Decomposed desired prob. into
  - part <u>unaffected</u> by noise $(x_i = 0)$

  \+ - part <u>completely/predictably</u> affected by noise

Handle each + combine.

We'll generalize all this.

New model: all about est. prob's.

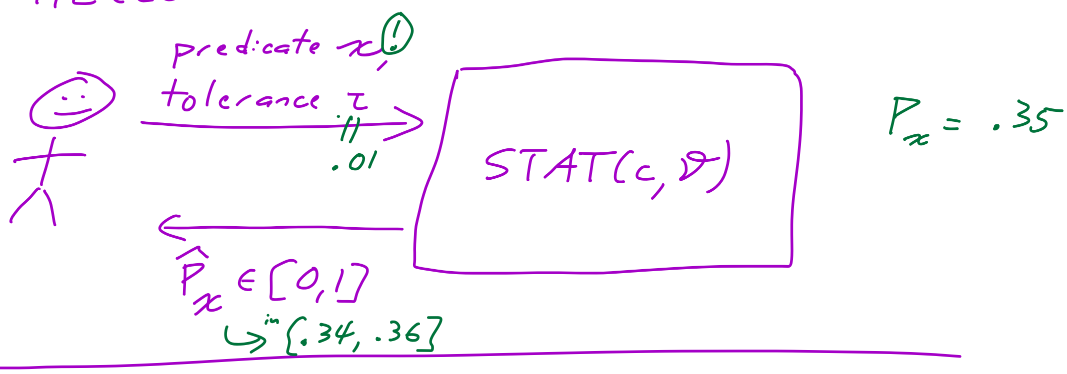
STATISTICAL QUERY MODEL

(For now: forget about noise.)

Alg no longer has $EX(c, \mathcal{D})$ BYE

↳ can only request est. of probabilities

HELLO: new oracle:



A <u>predicate</u> of a lab. ex. (x,y): a statement that's either T or F for (x,y).

ex: " (x,y) has $x_1 = x_2 = 1$, $y = 0$ " !

Formally, $\chi : \overset{x}{X} \times \overset{y}{\{0,1\}} \longrightarrow \overset{T/F}{\{0,1\}}$

Let

$$P_\chi := \Pr_{(x, c(x)) \sim EX(c, \mathcal{D})} [\chi(x, c(x)) = 1]$$

<u>Def</u>: Oracle $STAT(c, \mathcal{D})$ takes χ (pred) + τ (tol, $0 \le \tau \le 1$) as input from learner.

Returns value $\hat{P}_\chi$ s.t.

$$|P_\chi - \hat{P}_\chi| \le \tau.$$

Note: if you had $EX(c, \mathcal{D})$, you can simulate $STAT(c, \mathcal{D})$ w/ failure prob. $\leq \delta$:

- given inputs χ, τ: draw m ex from $EX(c, \mathcal{D})$. Eval. χ on each ex. Return $\hat{P}_\chi = \frac{\#\text{ of the } m \text{ ex that sat. } \chi}{m}$.

Chernoff: taking $m = O\left(\frac{\log 1/\delta}{\tau^2}\right)$, w.p. $\geq 1-\delta$, $|\hat{P}_\chi - P_\chi| \leq \tau$.

More intuition, for "cost" / "efficiency" of SQ learning:

If learner makes a call (χ, τ) to $STAT(c, \mathcal{D})$, learner should be "charged" amount of time that would be required to simulate $STAT(c, \mathcal{D})$ on its own if it had $EX(c, \mathcal{D})$.

So for an SQ alg to be efficient, it better

- not make too many calls to $STAT(c, \mathcal{D})$;
- make each call (χ, τ) with
  - τ not too small, &
  - χ being efficiently computable.

Def: Concept class $\mathcal{C}$ is learnable from statistical queries (SQ-learnable)

if there's a learning alg L s.t.

- for any $c \in \mathcal{C}$,
- for any dist $\mathcal{D}$ over X,
- for any $\varepsilon > 0$,

(Note: no δ param / failure prob, b/c STAT had no prob. of failure.)

if L is given access to $STAT(c, \mathcal{D})$, then L outputs a hyp h s.t. $\Pr_{x \sim \mathcal{D}}[h(x) \neq c(x)] \leq \varepsilon$.

$\mathcal{C}$ is poly-time efficiently SQ learnable by L if

- every query (χ, τ) L makes is s.t. $\chi(x, y)$ can be eval. in $\mathrm{poly}(n, \mathrm{size}(c), \frac{1}{\varepsilon})$ for each x, y, &
  $\tau \geq \frac{1}{\mathrm{poly}(n, \mathrm{size}(c), \frac{1}{\varepsilon})}$
- L runs in time $\mathrm{poly}(n, \frac{1}{\varepsilon}, \mathrm{size}(c))$ where we view each call to $STAT(c, \mathcal{D})$ as unit time.

All above gives:

Thm: Suppose $\mathcal{C}$ is eff. SQ-learnable. Then it's efficiently PAC learnable.

Pf: The SQ alg makes (say) M $(= \mathrm{poly}(\ldots))$ queries to $STAT(c, \mathcal{D})$ with each tol. $\tau \geq \tau_0 = \frac{1}{\mathrm{poly}(\ldots)}$

each x "reasonable"

PAC alg: for each call to STAT, uses $EX(c, \mathcal{D})$ to simulate it s.t. fail prob. of simulation $\left(\Pr\left[|\hat{P}_x - P_x| > \tau\right]\right)$ is $\leq \frac{\delta}{M}$.

This is poly in all relevant params:

n, size(c), $\frac{1}{\varepsilon}$, $\log\frac{1}{\delta}$. ∎

Next time: noise.