Learning with Classification Noise

The Model

- Same as PAC, but now we see $\langle x, \tilde{y} \rangle$, $x \sim P$ and:

$$\tilde{y} = \begin{cases} c(x) & \text{with prob. } 1-\eta \\ \neg c(x) & \text{with prob. } \eta \end{cases}$$

$c \in \mathcal{H}$ target fn.

$\eta \in [0, 1/2)$ noise rate

- Goal: still want $\varepsilon(h) = P[h(x) \neq c(x)] \leq \varepsilon$

Say $\mathcal{C}$ CN learnable by $\mathcal{H}$ if PAC criteria met by algo using noisy $\tilde{y}$'s.

- What $\mathcal{C}$ are CN-learnable?
- How much harder is CN?
- Are there "general principles" for CN?

Note: Can't use consistency any more...

An Observation

$$\Pr[h(x) \neq \tilde{y}] \triangleq \tilde{\varepsilon}(h)$$

$$= (1-\eta)\,\varepsilon(h) + \eta\,(1-\varepsilon(h))$$

$$= \underbrace{\eta}_{\text{fixed}} + \underbrace{(1-2\eta)\,\varepsilon(h)}_{\text{increasing with } \varepsilon(h)}$$

$\therefore$ ordering by $\check{\varepsilon}(h) \equiv$ ordering by $\varepsilon(h)$

$\therefore$ $\hat{\varepsilon}(h)$ minimization $\equiv$ $\varepsilon(h)$ minimization

<u>Note:</u> $\check{\varepsilon}(h_1) - \check{\varepsilon}(h_2) =$

$$\underbrace{(1-2\eta)}_{\text{shrinks}}\underbrace{(\varepsilon(h_1) - \varepsilon(h_2))}_{\text{need resolution } \varepsilon}$$

$$\boxed{\Rightarrow \text{sample sizes} \geq \frac{1}{(1-2\eta)\varepsilon}}$$

CN learning (monotone) conjunctions

- **Original algo:**
  - start with $h = x_1 x_2 \cdots x_n$
  - $\langle x, + \rangle \Rightarrow$ delete any $x_i = 0$

Won't work on $\tilde{y}$

Algo too brittle/sensitive.

Define $\forall x_i$:

$$p_0(x_i) = P[x_i = 0]$$

$$(\geq)\; p_{01}(x_i) = P[x_i = 0, y = 1]$$

true label $c(x)$

Say x_i is

significant: $p_0(x_i) \geq \varepsilon/8n$

harmful: $p_{01}(x_i) \geq \varepsilon/8n$

Let h contain all x_i that are sig. but not harmful.

$\Pr[h(x)=1, y=0]$: must be some $x_i \in c, x_i \notin h, x_i = 0$

not harmful $\Rightarrow$ not sig. $\Rightarrow$ $p_0(x_i) \leq \varepsilon/8n$

$$\therefore \Pr[h(x)=1, y=0] \leq n\left(\frac{\varepsilon}{8n}\right) = \varepsilon/8$$

$Pr[h(x)=0, y=1]$: must be

some $x_i \notin c$, $x_i \in h$, $x_i = 0$

$$p_{01}(x_i) \le \varepsilon/8n$$

$$\therefore Pr[h(x)=0, y=1] \le n\left(\frac{\varepsilon}{8n}\right) = \frac{\varepsilon}{8}$$

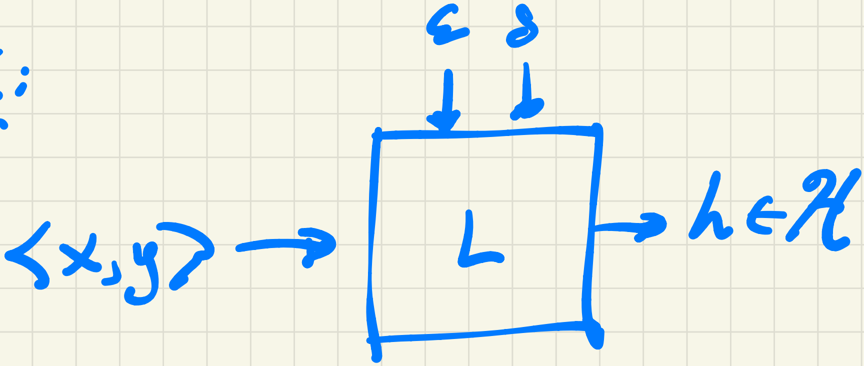
$$\& \ \mathcal{E}(h) \le \varepsilon/8 + \varepsilon/8 = \varepsilon/4.$$

The $p_0(x_i)$ are easy to estimate from $x \sim P$.

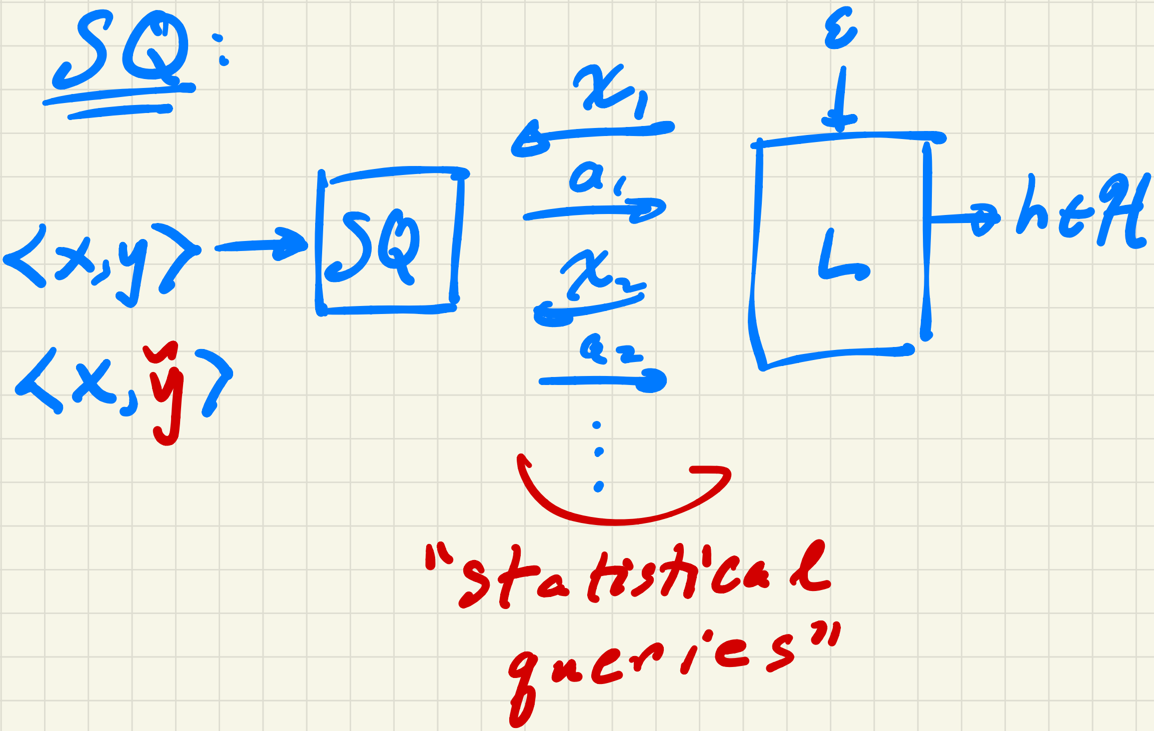
How can we estimate the $p_{01}(x_i)$ from $\langle x, \tilde{y} \rangle$?

The Statistical Query (SQ) Learning Model

PAC:



SQ:



Queries are predicates χ over $\langle x, y \rangle$ (no noise):

$$\chi(x, y) \in \{0, 1\}$$

E.g. $\chi(x, y) = 1 \iff x_5 = 1, \; y = 0$
$(p_{01}(x_5))$

$\chi(x, y) = 1 \iff x_1 \leq 1.7, \; x_2 \geq 2.9, \; y = 1$

Assume χ "reasonable"/computable

View $\chi(x,y)$ as a request for

$$p_\chi \triangleq \Pr[\chi(x,y)=1]$$

But SQ won't answer exactly, but with tolerance τ:

$$\hat{p}_\chi \in p_\chi + [-\tau, \tau]$$

Full query: χ, τ

Get back: $\hat{p}_\chi$

Say $\mathcal{C}$ is SQ-learnable by $\mathcal{H}$

if $\exists$ algo L

$\forall c \in \mathcal{C}, \forall P, \forall \varepsilon > 0$:

- L makes $\mathrm{poly}(1/\varepsilon, n, \mathrm{size}(c))$ queries, each of tolerance $\tau \geq \frac{1}{\mathrm{poly}(\cdot)}$ (e.g. ε^2/n^3)

- L runs in poly time

- L outputs $h \in \mathcal{H}$ s.t. $\varepsilon(h) \leq \varepsilon$.

Theorem

$\mathcal{C}$ SQ learnable by $\mathcal{H}$ $\Rightarrow$
$\mathcal{C}$ PAC learnable by $\mathcal{H}$.

Proof. Let L be SQ algo.
On each query χ, τ of L,
sample m $\langle x, y \rangle$ pairs to
get estimate $\hat{P}_\chi$. If
$m \sim \frac{1}{\tau^2} \log\left(\frac{\ell}{\delta}\right)$ then
w.p. $\geq 1 - \frac{\delta}{\ell}$ $\hat{P}_\chi \in P_\chi + [-\tau, \tau]$.

Now let $\ell = \#$ queries of L
$\leq$ runtime of L.

More Interesting Theorem

$\mathcal{C}$ SQ learnable by $\mathcal{H} \Rightarrow$
$\mathcal{C}$ CN learnable by $\mathcal{H}$.

How can we simulate SQ from $\langle x, \hat{y} \rangle$?

Decomposing P_x

- $P_x = Pr[\chi(x,y)=1]$
- $\hat{P}_x = Pr[\chi(x,\tilde{y})=1]$

& Can estimate directly

Now define disjoint X_1, X_2

$X_1 \cap X_2 = \phi$, $X_1 \cup X_2 = X$:

$$X_1 \triangleq \{x \in X : \chi(x,0) \neq \chi(x,1)\}$$

$$X_2 = X - X_1$$

Can test x for membership

Let $p_1 = P[X_1]$, $p_2 = P[X_2]$

$$p_2 = 1 - p_1$$

Induced distributions:

P_1 over X_1: $P_1[x] = P[x]/p_1$

P_2 over X_2: $P_2[x] = P[x]/p_2$

And finally:

$$p_x^1 = P_1[\chi(x,y) = 1]$$

$$p_x^2 = P_2[\chi(x,y) = 1]$$

Note: The following can be estimated from $\langle x, \tilde{y} \rangle$:

$$\hat{P}_x,\ P_x^2,\ P_1,\ P_2$$

Goal: formula where

P_x = some function of $\hat{P}_x, P_x^2, P_1, P_2, \eta$

$\tilde{y} = y$

$$\tilde{P}_x = (1-\eta) P_x + \eta \Big(p_1 P_1[\chi(x, \neg y) = 1]$$

$\hat{y} = \neg y$

$$+ p_2 P_2[\chi(x, \neg y) = 1] \Big)$$

$$\tilde{P}_x = (1-\eta)\,P_x + \qquad \tilde{y}=y$$

$$\eta\,\big(p_1 P_1[\mathcal{X}(x,{}^{?}y)=1] + p_2 P_2[\mathcal{X}(x,{}^{?}y)=1]\big) \qquad \tilde{y}={}^{?}y$$

$$= p_1 P_x^1 + p_2 P_x^2$$

$\tilde{y}=y$

$$\tilde{P}_x = (1-\eta) P_x + \eta \left(p_1 P_1[\chi(x, \neg y) = 1] + p_2 P_2[\chi(x, \neg y) = 1] \right)$$

$\tilde{y} = \neg y$

$$= P_1[\chi(x,y)=0]$$

$$= 1 - P_x'$$

$\tilde{y}=y$

$$\tilde{P}_x = (1-\eta) P_x +$$

$$\eta \left(p_1 P_1[\chi(x,\neg y)=1] + p_2 P_2[\chi(x,\neg y)=1] \right)$$

$\tilde{y}=\neg y$

$$= P_2[\chi(x,y)=1]$$

$$= P_x^2$$

$$\tilde{P}_x = \overset{\tilde{y}=y}{(1-\eta)}\, P_x +$$

$$\eta\,\big(p_1 P_1[\chi(x,\neg y)=1]$$

$\tilde{y}=\neg y$

$$+\, p_2 P_2[\chi(x,\neg y)=1]\big)$$

$$= (1-\eta)\left(p_1 P_x^1 + p_2 P_x^2\right)$$

$$+\, \eta\left(p_1\left(1-P_x^1\right) + p_2 P_x^2\right)$$

$$\tilde{P}_x = (1-\eta) P_x + \quad \tilde{y}=y$$

$$\eta \left(p_1 P_1[\chi(x,{}^{\neg}y)=1] \right. \quad \tilde{y}={}^{\neg}y$$

$$\left. + p_2 P_2[\chi(x,{}^{\neg}y)=1] \right)$$

$$= (1-\eta)(p_1 P_x^1 + p_2 P_x^2)$$

$$+ \eta (p_1 (1-P_x^1) + p_2 P_x^2)$$

$$= (1-2\eta) p_1 P_x^1$$

$$+ p_2 P_x^2 + \eta p_1$$

Now solve for P_x^1

$$P_x' = \frac{\hat{P}_x - p_2 P_x^2 - \eta \hat{p}_1}{(1-2\eta)p_1}$$

Plug into $P_x = p_1 P_x' + p_2 P_x^2$:

$$P_x = \frac{\hat{P}_x - p_2 P_x^2 - \eta p_1}{1-2\eta} + p_2 P_x^2$$

$$= \frac{\hat{P}_x}{1-2\eta} + \left(1 - \frac{1}{1-2\eta}\right) p_2 P_x^2$$

$$\frac{-\eta}{1-2\eta} p_1$$

Sensitivity Analysis (informal)

If we estimate $\hat{P}_x$ within $\pm\rho$, error in $\dfrac{\hat{P}_x}{1-2\eta}$ is $\pm\dfrac{\rho}{1-2\eta}$

So better make

$$\frac{\rho}{1-2\eta} < \tau, \quad \rho < \tau(1-2\eta)$$

Sensitivity Analysis (informal)

If error in p_2 is $\pm a$,
P_x^2 is $\pm b$, error in
$p_2 P_x^2 \leq a + b + ab$,
make $a, b < \tau(1-2\eta)$

etc.

Guessing η (informal)

- Suppose we are only given an upper bound:

$$\eta \le \hat{\eta} < 1/2$$

- Divide $[0, \hat{\eta}]$ into values $0, \Delta, 2\Delta, \ldots, \frac{\Delta}{\hat{\eta}}$

- Run simulation for each guess $\eta = i \cdot \Delta$
- Verify hypothesis has $\varepsilon(h) \leq \varepsilon$
- One guess will be $\pm \Delta$ of true η
- Make Δ small wrt $\tau, \varepsilon, \frac{1}{1-2\hat{\eta}}$
- Poly. overhead

Say x_i is

significant: $p_0(x_i) \geq \varepsilon/8n$

harmful: $p_{01}(x_i) \geq \varepsilon/8n$

Let h contain all x_i that are sig. but not harmful.

$\Pr[h(x)=1, y=0]$: must be some $\underbrace{x_i \in c}_{\text{not harmful}}, \underbrace{x_i \notin h}_{\text{not sig.}}, \underbrace{x_i = 0}_{p_0(x_i) \leq \varepsilon/8n}$

not harmful $\Rightarrow$ not sig. $\Rightarrow$ $p_0(x_i) \leq \frac{\varepsilon}{8n}$

$\therefore \Pr[h(x)=1, y=0] \leq n\left(\frac{\varepsilon}{8n}\right) = \varepsilon/8$

estimate all these (non-adaptive)

Claim

- "Almost" all $\mathcal{C}$ in PAC are also in SQ (with a different algo), and thus in CN.

- conjunctions, rectangles, decision lists, linear class. etc.

Claim

- Virtually every "practical" ML algo has an SQ variant.

- backprop, dec. tree algos, gradient des., boosting, etc.

So what's hard in SQ?

Parity Functions

- $\mathcal{X} = \{0,1\}^n$
- $\forall$ set $S \subseteq \{1,2,\dots n\}$ define:

$$f_S(x) \triangleq \sum_{i \in S} x_i \mod 2$$
$$= \bigoplus_{i \in S} x_i$$

- So $f_S(x) = 1 \iff \#$ bits $= 1$ in $x|S$ is odd.
- Let PARITY denote the class of all (2^n) f_S.

PARITY is PAC-learnable

- Note: $f_S(x) = \sum_i \alpha_i x_i \mod 2$
  where $\alpha, x \in \{0,1\}^n$ and
  $\alpha_i = 1 \iff i \in S$
- Inner product $\alpha \cdot x$ in $GF^n(2)$
- Given $\langle x_1, y_1 \rangle, \ldots \langle x_m, y_m \rangle$:

$$\begin{bmatrix} - x_1 - \\ - x_2 - \\ \vdots \\ - x_m - \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ \vdots \\ y_m \end{bmatrix}$$

Solve sys. of linear eqns!

Why is PARITY hard for SQ?

- Let P = uniform on $\{0,1\}^n$
- Look at χ's of form:

$$\chi(x,y) = 1 \Leftrightarrow x|T = z \text{ and } y = 1$$

- Draw $S \subseteq \{1,\ldots,n\}$ randomly
- If T is small ($|T| \leq n/2$) then whp over draw of S,

$$S - T \neq \phi$$

and thus

$$\Pr_x[\chi(x, f_S(x)) = 1] = 1/2$$

- "no information" conveyed

Theorem For $\varepsilon = 1/2$, and any polynomial $p(n)$, the number of queries of tolerance $\tau = 1/p(n)$ required to learn PARITY is exponential in n.

So PARITY $\in$ PAC

but PARITY $\notin$ SQ.

Is PARITY $\in$ CN?

Open problem,

but believed hard.

- "Nearby" NP-hardness
- Crypto proposals

Query Complexity in SQ

- target class $\mathcal{C}$, dist P
- view $f \in \mathcal{C}$ as ± 1-valued
- $\forall f_i, f_j \in \mathcal{C}$, inner product:

$$\langle f_i, f_j \rangle_P \triangleq \sum_x P(x) f_i(x) f_j(x)$$
$$= P[f_i = f_j] - P[f_i \neq f_j]$$

- For every $f, f' \in$ PARITY, $f \neq f' \Rightarrow \langle f, f' \rangle_P = 0$ where $P =$ uniform.

- Define

$SQ(\mathcal{C},P) = \max d$ s.t.

$\exists f_1,\dots f_d \in \mathcal{C}$, $\forall 1 \le i \ne j \le d$:

$$\langle f_i, f_j \rangle_P \le 1/d^3$$

- Largest # of "almost orthogonal" fns/vectors in $\mathcal{C}$ w.r.t. P

- $SQ(\text{PARITY}, \text{uniform}) = 2^n$

Theorem Let $SQ(\mathcal{C}, P) = d$.
Then at least $d^{1/3}/2$
queries with $\tau \geq 1/d^{1/3}$
are needed to learn
$\mathcal{C}$ wrt P in SQ model.
(even "weakly")

- d superpoly in $n \Rightarrow$
superpoly # queries

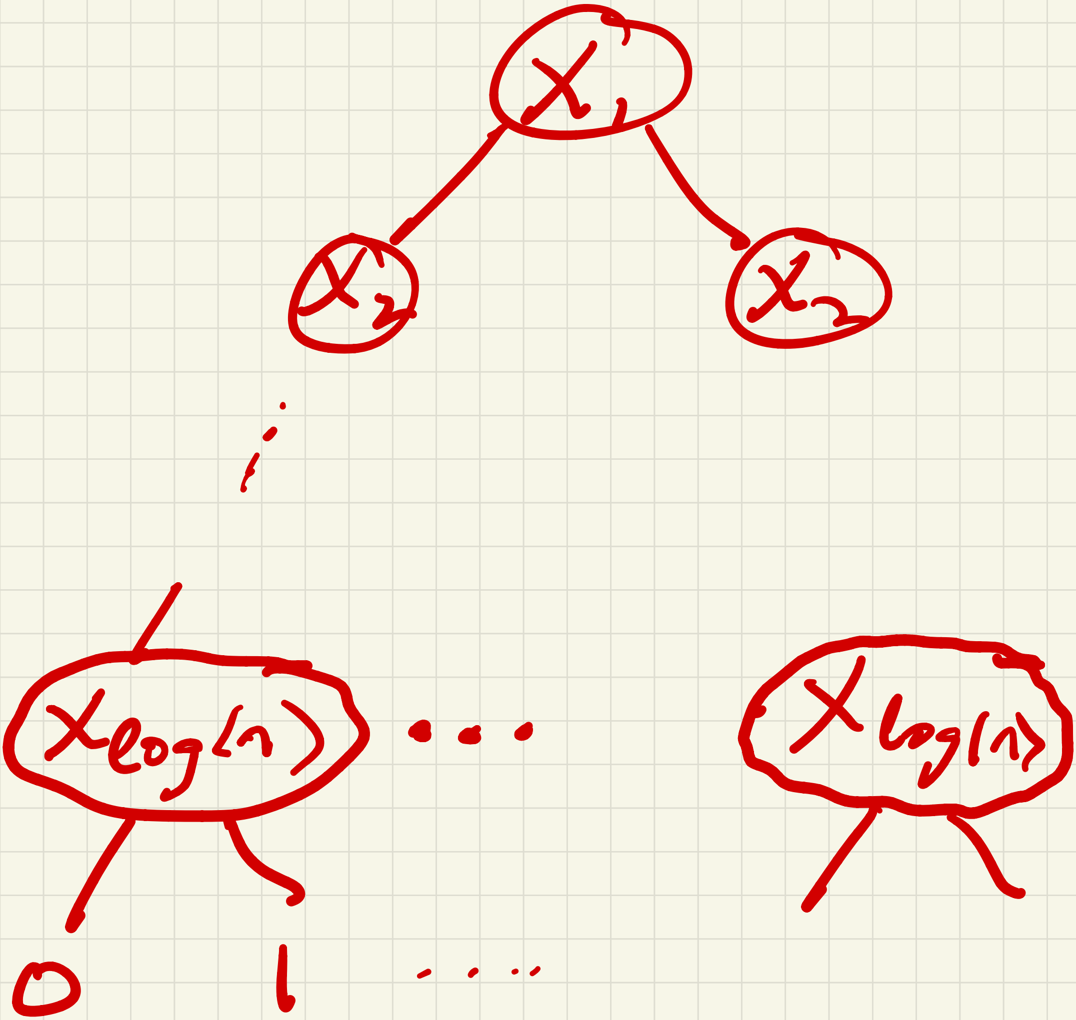
An Application

- Consider "small" parities, i.e. f_S where $|S| \leq \log(n)$
- E.g. parity of first $\log(n)$:

x_1	x_2	$\cdots$		$x_{\log(n)}$	f_S
0	0	0	$\cdots$	0	0
0	0	0	$\cdots$ 0	1	1
0	0	0	$\cdots$ 1	0	1
0	0	0	$\cdots$ 1	1	0
		$\vdots$			

only $2^{\log(n)} = n$ rows

Encode as decision tree:



size = n nodes

Encode as DNF:

$$\left.\begin{array}{l} \neg x_1 \neg x_2 \cdots \neg x_{\log(n)-1} \; x_n \vee \\ \neg x_1 \neg x_2 \cdots x_{\log(n)-1} \neg x_n \vee \\ \vdots \end{array}\right\} \text{all terms where } f_s = 1$$

size $\leq n$ terms

But there are

$$\binom{n}{\log(n)} \sim n^{\log(n)}$$

such small f_s.

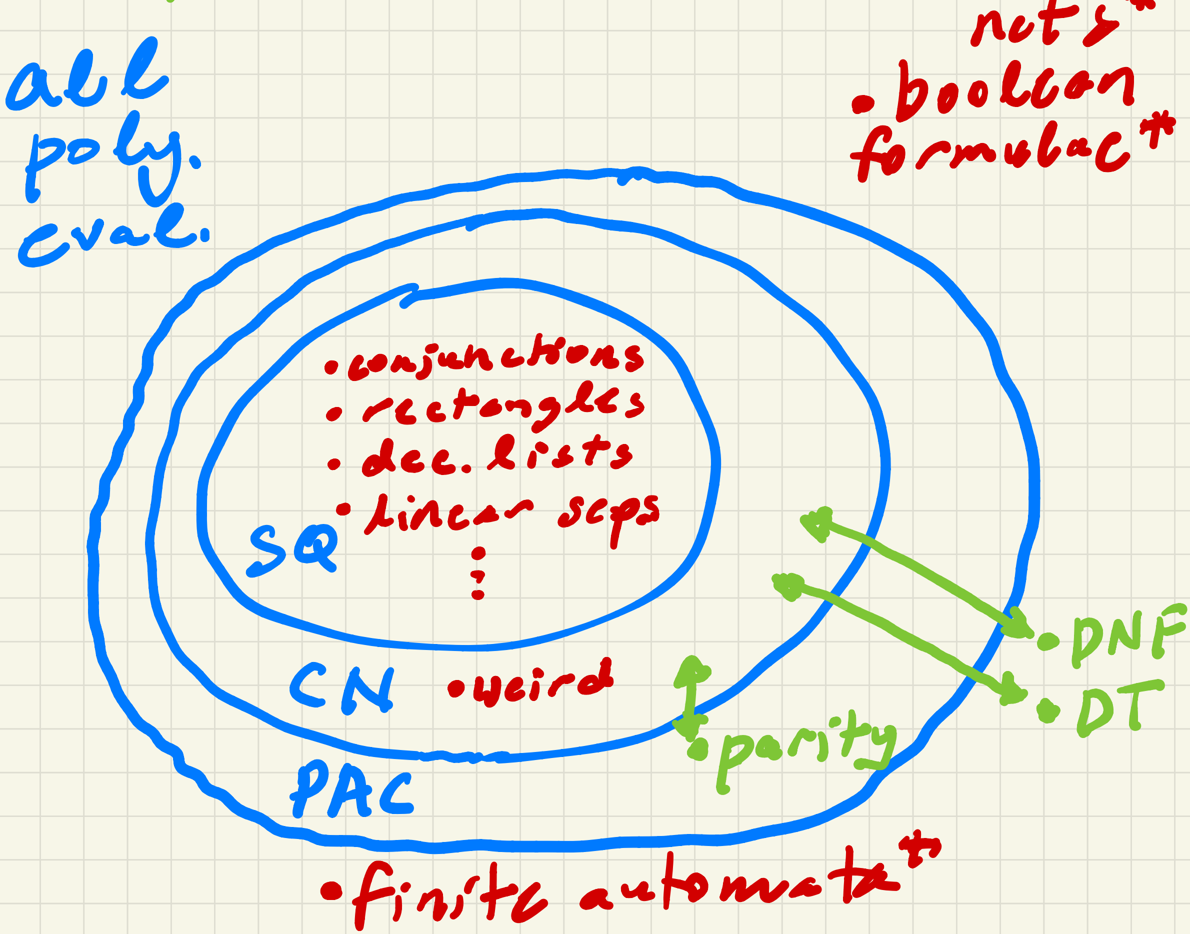
So decision trees & DNF are not learnable in the SQ model.

For informational, not computational, reasons. No assumptions.

Status in PAC: open.

The PAC Solar System

- status known
- (partially) open



$$SQ \subsetneq CN \subseteq PAC \subsetneq_{*} \text{poly. eval}$$

What about the *?

- There are functions that are easy to compute but hard to invert.
- Example: multiplication.

$$f(p,q) = p \cdot q$$

over domain p, q prime,

f^{-1} is factorization.

In fact, there are large families of such functions.

$\forall$ primes p, q define

$$f_{p,q}(x) = x^2 \bmod p \cdot q$$

Inverting for unknown $p, q \equiv$ factoring $p \cdot q$.

Encode as PAC learning neural nets, finite automata, boolean formulae...

Note: Hardness holds even for weak PAC learning, where

$$\varepsilon = 1/2 - 1/\mathrm{poly}(n)$$

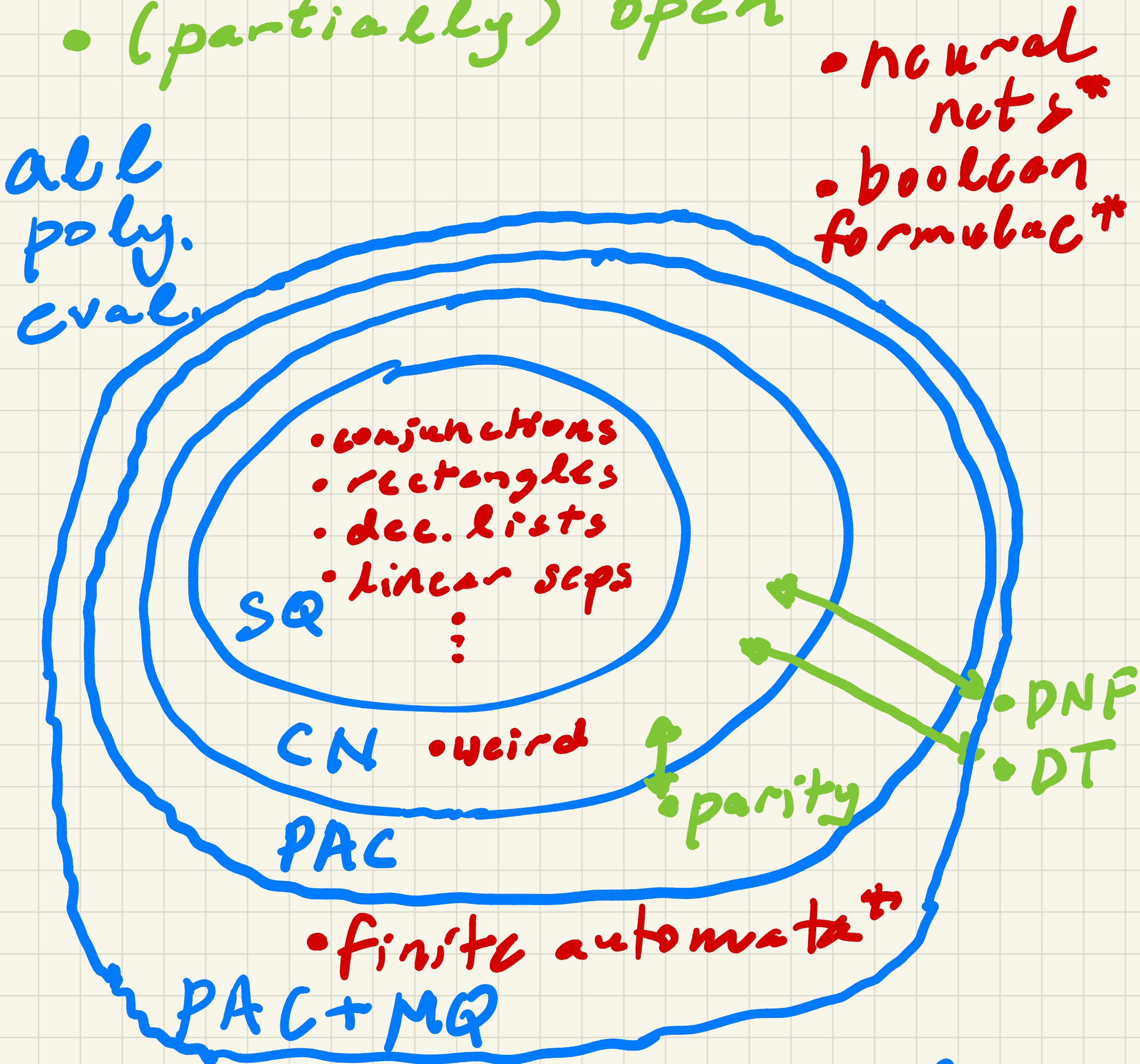
Does weak PAC $\Rightarrow$ PAC?

One final twist...

- Let's make PAC more powerful by adding membership queries
- In addition to $\langle x, y \rangle$, $x \sim P$, algo can get $c(x)$ for any chosen x.
- Goal remains same

The PAC Solar System

- status known
- (partially) open



$SQ \subsetneq CN \leq PAC \subsetneq_{*} \text{poly. eval}$

Next Up:

Boosting and Beyond