Last time: Learning sparse disj., using few examples, with a hypothesis that is a pretty sparse disj, via "greedy set cover heuristic"

- start proper vs. improper learning

Today:
- can efficiently improperly learn 3-term DNF; ☺
- can't " properly " " ", ☹ unless NP=RP
- (maybe) start unit on sample complexity of PAC learning

Questions?

☺ :

To show:
- can efficiently improperly learn 3-term DNF, using 3-CNF hyp.

Last time:
- de Morgan ⇒ any 3-term DNF is ≡ to some 3-CNF.

e.g.

$$\underbrace{(abc)\vee(ef)\vee(gh)}_{\text{3-term DNF}} \equiv \underbrace{(a\vee e\vee g)\wedge(a\vee e\vee h)\wedge(a\vee f\vee g)\wedge\ldots\wedge(c\vee f\vee h).}_{\text{3-CNF}}$$

We'll use: the class $\mathcal{C}$ = all mon. disj. over $\{0,1\}^n$ is PAC learnable with $m = O\left(\frac{1}{\varepsilon}\left(n+\log\frac{1}{\delta}\right)\right)$ ex, in $O(m)$ time.

Thm: The class of 3-term DNF is eff. PAC learnable using $\mathcal{H}$ = 3-CNFs.

Pf: Explicit feat. exp. using length-3 clauses as features
+ mon. conj. learning.

More detail:

There are $\overset{N=}{O(n^3)}$ poss. clauses of length 3:

$$\underbrace{(\overline{x_7} \vee x_9 \vee \overline{x_{21}})}_{y_{994}} \quad \text{feature}$$

Introduce a new var $y_1, \dots, y_N$ for each clause.

3-term DNF

Given an ex. $(x, c(x))$: in $O(n^3)$ time, can transform it to $(y, c'(y))$

y: N bits

c' is an AND of some unknown subset of $y_1, \dots, y_N$.

$\mathcal{D} \rightsquigarrow \mathcal{D}'$ dist over $\{0,1\}^N$; PAC alg works for that dist.

Run mon conj learner on $(y, c'(y))$ pairs; uses $m = O\left(\frac{1}{\varepsilon}\left(\underset{O(n^3)}{\underset{\parallel}{N}} + \log\frac{1}{\delta}\right)\right)$ ex, runtime.

Hyp $h(y) =$ some mon conj over $y_1, \dots, y_N$

some 3 CNF.

Proper 3-term DNF learning: computationally hard!

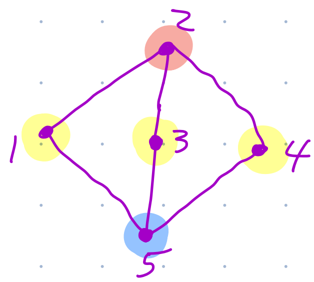
Thm: Suppose there's a poly($n, \frac{1}{\varepsilon}, \frac{1}{\delta}$)-time alg. which PAC learns 3-term DNFs properly. (using 3-term DNF hyp's)
Then there's a poly-time rand. alg. to solve GRAPH 3-COLORABILITY (G3COL).

G3COL:

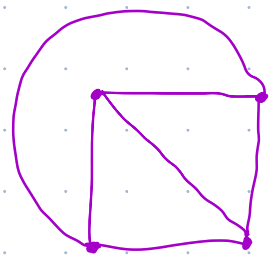
Input: n-node simple undirected graph

Output: Is G 3-vertex-colorable (each vtx R, Y, or B s.t. no edge R—R or Y—Y or B—B)?

Ex:



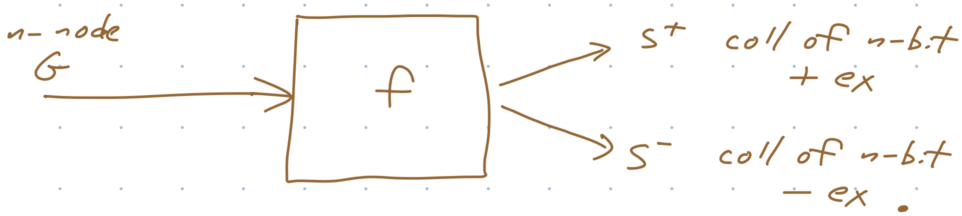
Yes



No

NP-complete: almost surely, no eff. alg.

Pf: **Key:** eff. comput. (deterministic, poly time) mapping f.



Key prop. of f :

$\forall$ input G, G is 3-colorable

$\Updownarrow$ iff

$\exists$ 3-term DNF consistent w/ S^+, S^-.

Suppose we have such an f. (Later.)

Now: use f + assump. that there's a proper 3-term DNF learner to give eff. rand. alg. for G3COL.

Here's the G3COL alg. Its input: G.

The alg: (rand)

1) Feeds G $\xrightarrow{n\text{-node}}$ $\boxed{f}$ + gets S^+, S^-

2) Let $\mathcal{D}$ = unif. dist. over $S^+ \cup S^- = S$
($\frac{1}{|S|}$ prob. on each elt. of S).
Set $\varepsilon = \frac{1}{2|S|}$.
Set $\delta = 0.01$.
Run proper PAC learner with ·. Get 3-term DNF hyp h.

Simulating draws from $\mathcal{D}$ requires randomness

3) Feed each lab ex in S into h; check it gets them all right.
If h cons. w/ S: output "G is 3-colorable"
If h not " " " " "G is not 3-colorable".

The above alg is efficient (poly(n) time):

1) ✓

2) ✓: since f is poly(n) time,

$|S| \le poly(n)$ + $\frac{1}{\varepsilon} \le poly(n)$

$\delta = \frac{1}{100}$.

's $poly(n, \frac{1}{\varepsilon}, \frac{1}{\delta})$ runtime is $poly(n)$

3) $|S| = poly(n)$, h = 3-term DNF: eff to eval h on each ex in S.

This solves G3COL:

- Suppose G is 3-colorable.

  $\Rightarrow \exists$ a 3-term DNF cons. with S.

  So we ran the "on valid data" + so w.p. $\ge \frac{99}{100} = 1-\delta$, it output a 3-term DNF h with error $\le \varepsilon = \frac{1}{2|S|}$.

  Since $\mathcal{D}$ puts weight $\frac{1}{|S|}$ on each pt in S, h must in fact not get any ex in S wrong, so we correctly say "G is 3-colorable" in step 3.

- Suppose G is not 3-colorable.

  By , know no 3-term DNF is cons. w/ S; so h is not cons. w/ S; so we correctly say "G is not 3-colorable" in step 3.

What's f?

Given n-node $G = (V, E)$, where $V = \{1, \ldots, n\}$

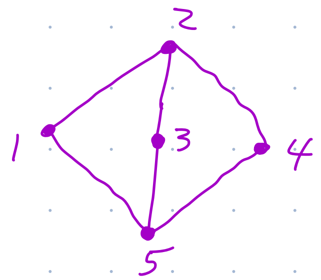
$f(G) = S^+ \cup S^-$

n - bit strings

S^+: n $\overset{+}{\text{ex}}$; i^{th} one has 0 in coord. i,
1 in all other coord.

S^-: one ex in S^- for each edge;
edge $\{i,j\}$ ⟶ neg ex with 0 in i+j,
1 everywhere else.

Ex: $n=5$

$G =$ 

S^+		S^-	
01111		00111	edge {1,2}
10111	vtx 2	01110	
11011		10011	
11101		10101	
11110		11010	
		11100	

__Remains to show:__

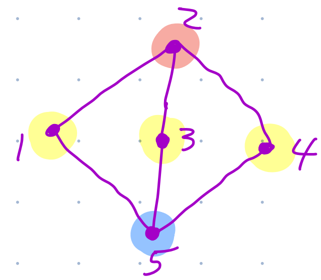
$\forall$ input G, G is 3-colorable

⇕ iff

$\exists$ 3-term DNF consistent w/ S^+, S^-.

__Claim 1:__ If G is 3-colorable, then
$\exists$ 3-term DNF consistent w/ S^+, S^-.

__Pf__: Let R, B, Y be a legit 3-coloring of G.
Let T_R = mon conj. containing each x_i s.t. i __not__ red in
T_B = mon conj. containing each x_i s.t. i __not__ blue
T_Y = mon conj. containing each x_i s.t. i __not__ yellow.



$\Longrightarrow$

$T_R = x_1 x_3 x_4 x_5$

$T_B = x_1 x_2 x_3 x_4$

$T_Y = x_2 x_5$.

- Each ex $\overset{i}{\text{in}}$ S^+ lab. + by $T_R \vee T_B \vee T_Y$:
  - if vtx i is R, ex $\overset{10111}{i}$ (0 only in pos. i) satisfies T_R.

- Each ex in S^- lab. − by $T_R \vee T_B \vee T_Y$:
  - (i,j) ex $11\ldots\overset{i}{0}1\ldots1\overset{j}{0}1\ldots1$: to satisfy T_R, T_R would need to be missing x_i & x_j but that only happens if i,j both R, not so for i,j since valid coloring.

<u>Claim 2:</u> <u>If $\exists$ 3-term DNF consistent w/ S^+, S^-, then G is 3-colorable.</u>

<u>Pf</u>: Let $T_R \vee T_B \vee T_Y$ be a 3-term DNF.

Construct foll. candidate coloring:

- color vtx i R if i^{th} + satisfies T_R
- " " " B " " " " T_B
- " " " Y " " " " T_Y.

Colors every vtx. To show: every edge does <u>not</u> have 2 same-color endpts. (<u>not</u> both R)

Pick any edge i,j: we'll argue)

$i=1, j=2$:

$$\begin{array}{ccccc|c} 0 & 1 & 1 & 1 & 1 & + \\ 1 & 0 & 1 & 1 & 1 & + \\ 0 & 0 & 1 & 1 & 1 & - \end{array}$$

for i,j to both be R: both satisfy T_R. T_R has neither x_1 nor $\bar{x}_1$.

T_R must have neither x_2 nor $\bar{x}_2$

→ neg ex "looks just like" to T_R

So i,j <u>not</u> both R
B
Y. ∎