Last time:

- Introductions (you, me, topic)
- Admin. overview (web page)
- High level course overview (learning models)
- Some technical material: Key basic notions & terminology

Rocco OH:
Fri 1:30-3:30
5^{th} floor CSB

X = domain, $\mathcal{C}$ = concept class = collection of concepts (concept = subset of X)

- There's a known $\mathcal{C}$, & an unknown target concept $c \in \mathcal{C}$.
- Learner has some source of info. about how c labels examples $x \in X$. Want to find/approximate c.

Today:

- some more examples of $\mathcal{C}$'s (DNFs, LTFs)
- FIRST LEARNING MODEL:

  Online Mistake-Bound Learning (OLMB)

  - Elimination alg. for monotone disjunctions (& more)
  - Decision lists, OLMB alg. for learning them

PS1: coming out on Tues

Readings: see web page

Questions?

Last example:

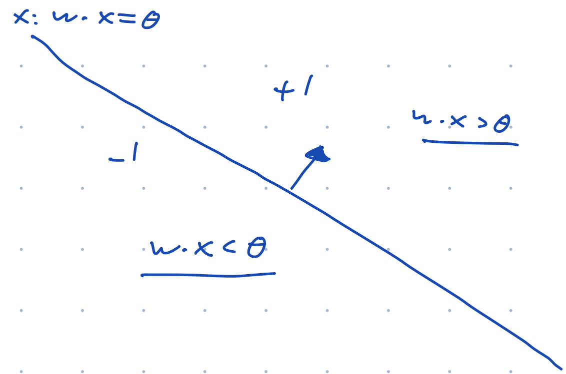
LTFs
Linear Threshold Fn's

- $X = \mathbb{R}^n$ or $\{0,1\}^n$ $\quad \mathcal{C}$ = all halfspaces

A Bool fn $c: \mathbb{R}^n \to \{-1,1\}$ is an LTF if $\exists$ a weight vector $w \in \mathbb{R}^n$, threshold $\theta \in \mathbb{R}$ s.t.

$$c(x) = \text{sign}(w \cdot x - \theta), \qquad \text{sign}(t) = \begin{cases} 1 & \text{if } t \geq 0 \\ -1 & \text{if } t < 0 \end{cases}$$

$\hookrightarrow w_1 x_1 + \cdots + w_n x_n$



Ex: $n=4$, $c(x) = \text{sign}(-100x_1 + 2.7x_2 + 0.03x_3 + 1000x_4 - 7)$

<u>First learning model:</u>

(OLMB)

Online Mistake-Bound Learning

"Learning session" consists of a seq. of <u>trials</u>.

Learner always has a hypothesis <u>$h: X \to \{0,1\}$</u>

Know $\mathcal{C}$, don't know target concept $c \in \mathcal{C}$.

Each trial:

- Learner given <u>unlabeled</u> $x \in X$.
- Learner outputs $h(x) \in \{0,1\}$.
- Learner then given true <u>$c(x) \in \{0,1\}$</u>.

If $h(x) \neq c(x)$, learner is charged a mistake.

Learner can update h before next trial. (this is the learning alg, along w/ h_{init}.)

Performance on a seq. of trials: # mistakes made.

Def: A learning alg. A has mistake bd M for $\mathcal{C}$ if, for any seq. of ex. from X, any target $c \in \mathcal{C}$, A makes $\leq M$ mistakes. (of any length!)

- No noise
- worst case;
- no data distrib.

Obs #1: • If X finite $(X = \{x^1, x^2, \ldots, x^N\})$, can achieve mist. bd. $M \leq N$. (memorization.)

• If $\mathcal{C}$ is finite $(\mathcal{C} = \{c^1, \ldots, c^L\})$

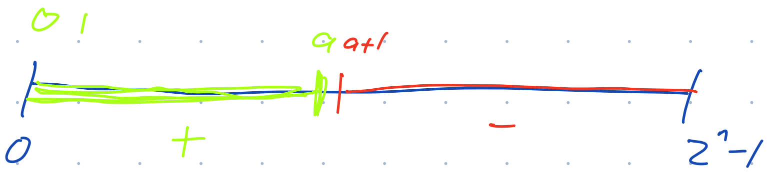
$M \leq L-1$. (try each concept in order)

Can often do better:

Ex: $X = \{0, 1, 2, \ldots, 2^n - 1\}$

$\mathcal{C}$ = class of initial intervals

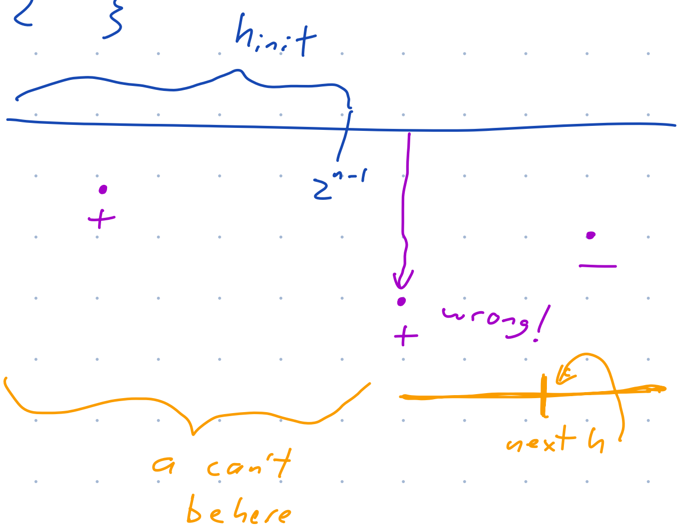
$\mathcal{C} = \{ \{0, 1, \ldots, a\} : a \le 2^n - 1\}$



$|X| = 2^n, \quad |\mathcal{C}| = 2^n$

Better alg.: binary seach. Mist. bd. $M \le n$

$h_{init} = \{0, 1, \ldots, 2^{n-1}\}$



Each mistake lets us halve the interval containing a, so after $\log_2(2^n)$ mistakes, know a exactly, + no more mistakes.

Ex: $X = [0,1]$ (cont. interval):

$\mathcal{C}$ = init. intervals: no finite mist. bound.

$c = [0, 0.237185620014\cdots]$

Some OLMB Algorithms

(1) Learning Mon. Disjunctions

$X = \{0,1\}^n$, $\mathcal{C}$ = all mon. disj.

$c(x) = x_5 \vee x_8 \vee x_9 \vee x_{10}$

"Elim. alg."

- $h_{init} = x_1 \vee x_2 \vee \cdots \vee x_n$ (all n vars)
- On false pos mistake ($h(x)=1$ but $c(x)=0$): remove from h all x_i's s.t. $x_i = 1$ in that example (elimination)
- On false neg mistake ($h(x)=0$, $c(x)=1$): FAIL.

Ex: $n=5$, $h_{init}(x) = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5$

Get $x = 10010$: $h_{init}(x) = 1$

Told $c(x)=0$: new $h(x) = x_2 \vee x_3 \vee x_5$.

<u>Claim A:</u> No var that's in c is ever removed from h.

<u>Pf</u>: x_i only rem. from h if $x_i = 1$, $c(x) = 0$;
but means x_i can't be in c. ■

<u>Claim B:</u> Alg only makes false pos on c = any mon. disj. (don't FAIL)

<u>Pf</u> By Claim A, h always includes all c's var's.
So if $c(x) = 1$, also $h(x) = 1$. ■

<u>Thm:</u> <u>Elim alg. has mistake bd $\le n$ for $\mathcal{C}$ = mon disj.</u>

<u>Pf</u>: By B, don't fail.
By A, (vars in h) $\supseteq$ (vars in c) always.
Alg starts w/ n vars in h;
each mistake removes ≥ 1 var from h,
so $\le n$ mistakes.

<u>Notes:</u>

- n = "size";

  $\left.\begin{array}{ll} \text{poly}(n) & \text{time/trial} + \\ \quad " & \text{mistake bd} \end{array}\right\}$ 

- Noise would been fatal....

- Can extend to learn arb. disj. (non-mon):

run it over $2n$ "meta-variables" $x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n$

$n=2$:

$$h_{init}(x_1, x_2) = x_1 \vee x_2 \vee \bar{x}_1 \vee \bar{x}_2$$

must be $h_{init} = 1$, $c = 0$

mist. bd. $= n+1$ (first mistake: cross off n of $2n$ literals)

- Can extend to learn (mon) conjunctions.

negate each $c(x)$, + learn the disj.

$$c(x) = x_1 \wedge \bar{x}_2 \wedge x_3$$

$$\Downarrow$$

$$\overline{c(x)} = \bar{x}_1 \vee x_2 \vee \bar{x}_3$$

---

(2) Learning Decision Lists

b_i = bit 0/1

ℓ_i = literal

(1-decision list) (b/c each rule involves just 1 literal)

A decision list is a fn $\{0,1\}^n \to \{0,1\}$ of the form

"if ℓ_1 then output b_1
else if ℓ_2 " " b_2
⋮
else if ℓ_r " " b_r
else output b_{r+1} "

$r = \#$

Ordered list of rules

Ex:

$$r=4$$

$$\begin{tikzcd}
x_7 \arrow[r, "F", "0"'] \arrow[d, "T", "1"'] & \overline{x_3} \arrow[r, "F"] \arrow[d, "T"] & x_4 \arrow[r] \arrow[d, "T"] & x_6 \arrow[r] \arrow[d] & 0 \\
1 & 0 & 0 & 1 &
\end{tikzcd}$$

"if ℓ then output b" : a **rule**.

$$\underbrace{4}_{\substack{\pm \text{ literal} \\ \pm \text{ bit}}}\overset{\text{var. choice}}{n} + 2 \quad \overset{\text{poss.}}{\text{rules}}$$

Length-r DL: $\approx (4n)^r$ poss.

Never should have (redundant!)

$$\cdots\; x_7 \rightarrow \cdots\; \overline{x_7}$$
$$\quad\downarrow \qquad\qquad \downarrow$$

Think of every DL over $\{0,1\}$ as having length $\leq n$.

Every ~~conj~~ disj can be written as a DL....

$x_1 \wedge \overline{x_2} \wedge x_3$ is $\equiv$ to

$$\begin{tikzcd}
\overline{x_1} \arrow[r] \arrow[d] & x_2 \arrow[r] \arrow[d] & \overline{x_3} \arrow[r] \arrow[d] & 1 \\
0 & 0 & 0 &
\end{tikzcd}$$

Next time: an efficient OLMB alg. for DLs...