Last time: • introductions (me, you, topic)
• administrative overview (web page)
• high level overview of course → learning models
• some technical material: key basic notions + terminology

Learner knows $X, \mathcal{C}$; goal is to learn unknown target concept $c \in \mathcal{C}$

X = instance space $x \in X$
$c: X \to \{0,1\}$ concept $c \in \mathcal{C}$
$\mathcal{C}$ = concept class = set of concepts

Today: • a few more examples of $\mathcal{C}$'s (DNFs, LTFs)
• FIRST LEARNING MODEL:
Online Mistake-Bound Learning

specific algs. for specific $\mathcal{C}$
• Elimination alg. for monotone disjunctions, extensions
• Decision lists, OCMB alg. for learning them

Admin: • PS 1 out on Wed

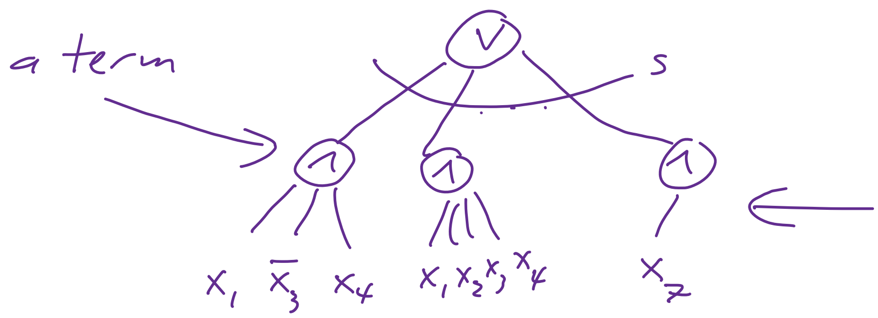
Readings: web page

Questions?

Some more $\mathcal{C}$'s of interest:

$X = \{0,1\}^n$, $\mathcal{C}$ = all s-term (s-clause) DNF (Disjunctive Normal Form; CNF: Conjunctive) formulas

s-term DNF: s-way disj. (OR; AND) of conj (AND; ORs) of literals.

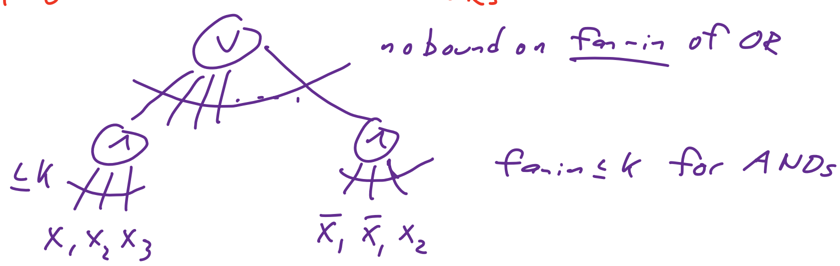


K- CNFs

$X = \{0,1\}^n$, $\mathcal{C}$ = K- DNFs

K-DNF : OR of K-literal ANDs.

K- CNF   AND   ORs



---

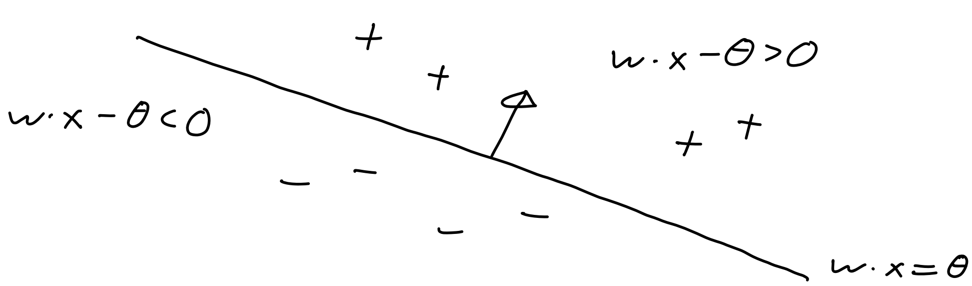
halfspace (LTF)

$X = \mathbb{R}^n$, $\mathcal{C}$ = linear threshold functions

A function $c: \mathbb{R}^n \to \{-1, 1\}$ is an LTF if
if there are $w_1, \dots, w_n$, $\theta \in \mathbb{R}$ s.t.

$$\text{sign}(t) = \begin{cases} 1 & t \ge 0 \\ -1 & t < 0 \end{cases}$$

$$c(x) = \text{sign}(w \cdot x - \theta)$$



$\mathbb{1}[w \cdot x \ge \theta]$ $\{0,1\}$
indication function

ex:

$$c(x) = \mathrm{sign}(100x_1 - 7x_2 + 8x_3 - 16)$$

Can also consider LTFs over $X = \{0,1\}^n$.

First learning model:

ONLINE MISTAKE-BOUND LEARNING MODEL

A learning session consist of a seq. of trials.
Throughout, learner maintains some hypothesis
$h: X \to \{0,1\}$.

Learning an unknown $c \in \mathcal{C}$ goes like this.
Each trial:

- Learner is given an unlabeled $x \in X$.
- Learner outputs $h(x) \in \{0,1\}$
- (*) Learner is given $c(x) \in \{0,1\}$.
  If $c(x) \neq h(x)$, learner is charged a mistake.

(*) Learner can update h before next trial.

this update rule, + initial hyp, constitute the learning alg.

Perf. measure: worst-case tot. # mistakes.

Def: OL Learning alg A has mistake bound M (could be arbitrarily long!) for $\mathcal{C}$ if for any sequence of examples from X, any $c \in \mathcal{C}$, A makes $\leq M$ mistakes when learning c.

Observe:

- If X finite, for any $\mathcal{C}$ over X, can achieve m.b. $\leq |X|$. (memorization)

- If $\mathcal{C}$ finite, can achieve m.b. $|\mathcal{C}| - 1$ (try each concept till it causes a mistake).

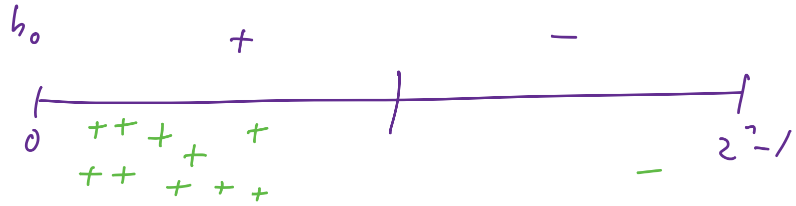
Let's do better:

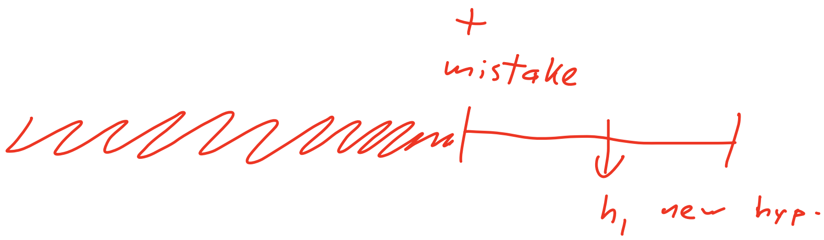
EX $X = \{0, 1, \ldots, 2^n - 1\}$

$\mathcal{C}$ = initial intervals $\quad c = \{0, 1, 2, \ldots, a\}$

Alg. w/ m.b. n: binary search

h_0 init. hyp: $\{0, \ldots, 2^{n-1}\}$ (midpoint of "uncertainty region")





halve uncertainty at each mistake $\Rightarrow$ $\leq n$ mistakes.

$X = [0,1]$ continuous, $\mathcal{C}$ = initial intervals

$c = [0, 0.371428556001\ldots]$

(no finite mistake bound...)

Learning monotone disjunctions

$X = \{0,1\}^n$, $\mathcal{C}$ = mon. disj. $\qquad x_1 \vee x_5 \vee x_6 \vee x_7$

Algorithm ("elimination alg."):

- Initial hyp is $h(x) = x_1 \vee \ldots \vee x_n$
- Get $z \in \{0,1\}^n$ as example, predict $h(z)$, given $c(z)$
- If $h(z) = 1$, $c(z) = 0$ (false positive): remove x_i from h for all i s.t. $z_i = 1$.
- If $h(z) = 0$, $c(z) = 1$, stop + FAIL (won't happen!)
- If $h(z) = c(z)$, keep h the same.

Ex $\qquad n = 5 \qquad h_0 = x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5$

Get $z = 01001$

$h(z) = 1$. Suppose $c(z) = 0$.

Remove x_2, x_5: new $h(x)$ is $x_1 \vee x_3 \vee x_4$.

MB n.

Claim 1: if x_i is in c, x_i never removed from h.

Pf: only way x_i removed is if ex z has $c(z) = 0$, $z_i = 1$; can't happen if x_i is in c. ∎

Claim 2: alg only makes false pos. mistakes (won't FAIL).

Pf: by prev claim, h always includes all vars in c. So if $c(z) = 1$ b/c of $z_i = 1$, z_i also causes $h(z) = 1$. So no false neg. ∎

Thm: Elim alg. has mistake bound n for $\mathcal{C}$ = mon disj.

Pf: Doesn't FAIL.

Vars in h $\supseteq$ Vars in c.

Every mistake removes ≥ 1 var from h.

h initially n variables, never has < 0 vars;

So # mistakes can't be greater than n.

Note: • n = "size" of input $x \in \{0,1\}^n$; can describe any $c \in \mathcal{C}$ with n bits.

- Alg is efficient: n mistakes, $O(n)$ time per trial

- Noise? OLMB Model: no noise.

  This alg would be very susceptible to noise: elimination is irrevocable.

- Extension to alg for $\mathcal{C}$ = all disj:

  initial $h = \cancel{x_1} \vee \bar{x}_1 \vee x_2 \vee \cancel{\bar{x}_2} \vee \cdots \vee x_n \vee \bar{x}_n$

  First mistake z 10100: $c(z)=0$, $h(z)=1$:

  updated $h = \bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee x_4 \vee x_5$

  After 1 mistake: learning mon. disj. "in disguise"

  $n+1$ for .

- Easy ext. to $\mathcal{C}$ = mon, conj, general conj.
  (think about it...)

Decision Lists

A 1-decision list (1-DL) is a concept

$c: \{0,1\}^n \to \{0,1\}$ which is an ordered

list of if-then-else rules: one literal per rule

literal

if ℓ_1 true then output b_1

else " ℓ_2 2n poss " " " b_2 2 poss

⋮

" ℓ_r " " " b_r

else output b_{r+1}. 2 poss.

Ex:

$$\begin{tikzcd}
x_7 \arrow[r, "=0"] \arrow[d, "=1"] & \bar{x}_3 \arrow[r, "=0"] \arrow[d, "=1"] & \bar{x}_5 \arrow[r, "=0"] \arrow[d, "=1"] & x_2 \arrow[r, "=0"] \arrow[d, "=1"] & 1 \\
0 & 0 & 1 & 0 &
\end{tikzcd}$$

$r = 4$ rules in this one

- Would never need to repeat lits or vars:

$$\cdots \to x_3 \to \cdots \quad \begin{matrix}\bar{x}_3 \\ x_3\end{matrix}$$
$$\downarrow$$

- Any conj is expressible as a DL:

$$x_2 \wedge \overline{x_3} \wedge x_4 \equiv \begin{tikzcd} \overline{x_2} \arrow[r] \arrow[d] & x_3 \arrow[r] \arrow[d] & \overline{x_4} \arrow[r] \arrow[d] & 1 \\ 0 & 0 & 0 & \end{tikzcd}$$

Also any disj.

$$\mathcal{C}_{conj} \subseteq \mathcal{C}_{DL}$$

Note: # of 1-DL of length r

is $\approx 2^{r+1} \cdot (2n)^r \approx (4n)^r$.

Total # of possible rules, including the two "default rules", is $4n+2$

Next time: $O(nr)$-mistake bound alg. for class of 1-DL's of length r. Based on reordering rules in careful way.