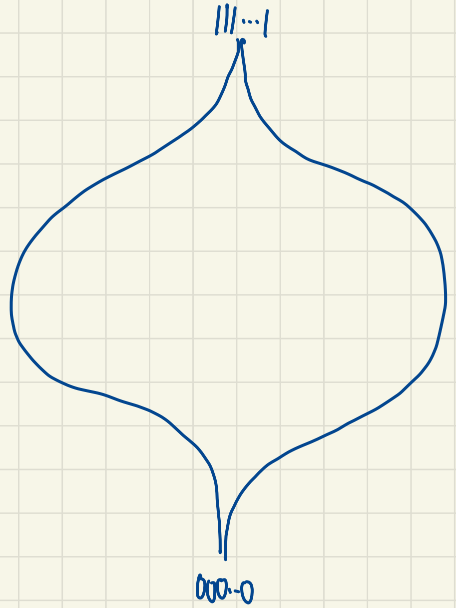
Lecture 21

- weak learning of monotone fctns

- begin: distribution-free weak learning
  $\Rightarrow$ strong learning

Boolean Cube

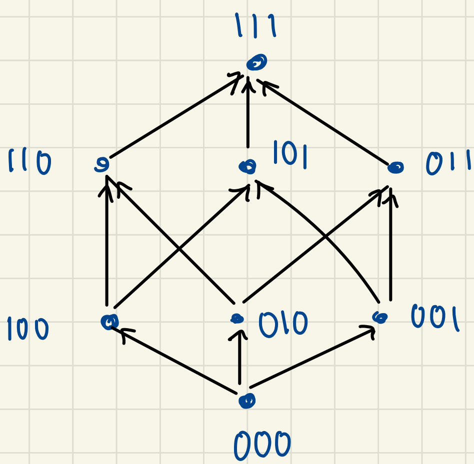


hypercube

level k:
nodes labeled by
k 1's + n-k 0's

nodes on level k:
$\binom{n}{k}$

edges:
$x \to y$
if flip one 0 in x to a 1 to get y



example for $n=3$

nodes: 2^n
edges: $\frac{n \cdot 2^n}{2}$

Monotone Functions

def. partial order $\leq$: $x \leq y$ iff $\forall i \;\; x_i \leq y_i$

monotone fctn f: $x \leq y \Rightarrow f(x) \leq f(y)$

Are there fast learning algorithms for the class of monotone functions?

Occam's razor:

poly$(\log |C|)$ samples suffice

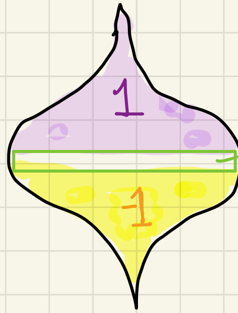
$\uparrow$ Class of monotone fctns

$\geq 2^{2^n/n\sqrt{n}}$ monotone fctns

So only gives exponential bound

Why so many monotone fctns?

Consider "slice" fctns:



set middle row in all possible ways w/o violating monotonicity

$2^{\binom{n}{\frac{n}{2}}}$ options

all are monotone!

Note: on uniform dist, easy to learn slice fctns.
i.e. Output "Majority"
$\Rightarrow$ Occam is "weak" on this class/distribution

H.W.: $2^{O(\sqrt{n})}$ random samples suffice for unif dist

Today: what if you compromise on error?

Can get very slight "win"

All monotone fctns have weak agreement with <u>some</u> dictator fctn.

<u>Thm</u> $\forall f$ monotone, $\exists g \in \{\pm 1, x_1, x_2, \ldots x_n\} \equiv S'$

s.t. $\Pr_x [f(x) = g(x)] \geq \frac{1}{2} + \Omega(\frac{1}{n})$

uniform distribution

slightly better than random guessing

(can get $\frac{1}{2} + \Omega(\frac{1}{\sqrt{n}})$ if add majority)

<u>note</u> Slice fctns have weak agreement with all dictators on uniform dist

$\Rightarrow$ learning algorithm:

estimate agreement of f with all members of S'

output best

<u>Pf.</u>

Case 1: $f(x)$ has weak agreement with $+1$ or -1 ✓

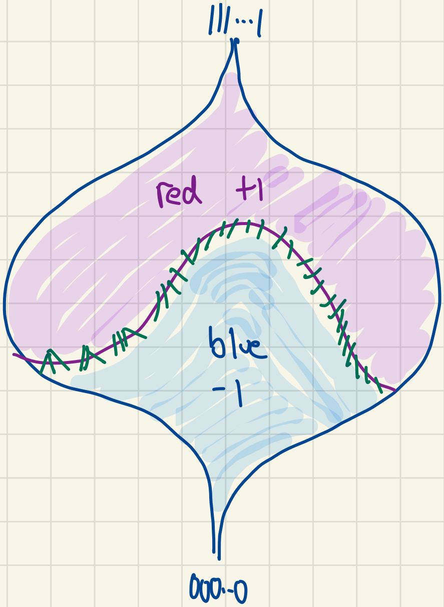
Case 2: otherwise $\Pr[f(x) = 1] \in [\frac{1}{4}, \frac{3}{4}]$

Let's first look at monotone fctns in a different way:

excuse for a detour

Monotone Functions on Boolean Cube:

A "graph" view



monotone $\Rightarrow$ no blue above any red

$x \le y$ if $\forall i \; x_i \le y_i$

f monotone if

$\forall x \le y, \; f(x) \le f(y)$

Influence of f:

$$\mathrm{Inf}_i(f) = \frac{\#\text{ red-blue edges in } i^{th} \text{ dir}}{2^{n-1}}$$

$$= \Pr_x\left[f(x) \ne f(x^{\oplus i})\right]$$

x with i^{th} bit flipped

$$\mathrm{Inf}(f) = \frac{\#\text{ red-blue edges}}{2^n}$$

$$= \sum_{i=1}^{n} \mathrm{Inf}_i(f)$$

Thm 1 f monotone $\Rightarrow$ $\inf_i(f) = \hat{f}(\{i\})$

Thm 2 majority fctn $f(x) \equiv \text{sign}\left(\sum_{i=1}^{n} x_i\right)$ (odd n)

maximizes influence among monotone fctns

Pfs on h.w.

Plan:

note: $\inf_i(f) = \hat{f}(\{i\})$ (Thm 1)

$$= 2 \cdot \Pr[f(x) = \underbrace{\chi_{\{i\}}(x)}_{x_i}] - 1$$

early Fourier lecture: agreement vs. Fourier coeffs

so showing $\text{Inf}_i(f) \geq \Omega\left(\frac{1}{n}\right)$

is equivalent to showing

weak learner

$$\Pr[f(x) = x_i] \geq \frac{1}{2} + \frac{\inf_i(f)}{2} \gtrsim \frac{1}{2} + \Omega\left(\frac{1}{n}\right)$$

such an i would give us our theorem!

To show that such an i exists, will use a cool tool:

Canonical Path Argument

Plan (1) define canonical path for every red-blue pair of nodes

(such a path must cross at least one red-blue edge)

(2) Show upper bound on # of c.p.'s passing through any edge

(in particular, any red-blue edge)

(3) Conclude lower bound on # of red-blue edges.

Part I: define canonical path for every red-blue pair of nodes

def $\forall (x,y)$ s.t. x red & y blue

"canonical path from x to y" is:

scan bits left to right

flipping where needed

each flip $\rightsquigarrow$ step in path

example:

	dimension	1	2	3	4
x	=	-1	$+1$	$+1$	$+1$
w	=	$+1$	$+1$	$+1$	$+1$
z	=	$+1$	-1	$+1$	$+1$
y	=	$+1$	-1	$+1$	-1

$x \to w \to z \to y$

each step has Hamming distance 1

note: c.p.'s can go up & down

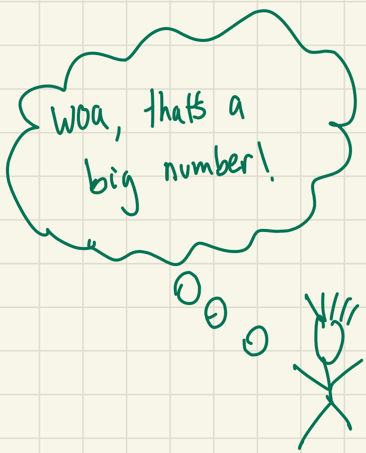
e.g. $x \to w$ is up step $w \to z$ is downstep

Big question:

How many red-blue x,y pairs have canonical paths?

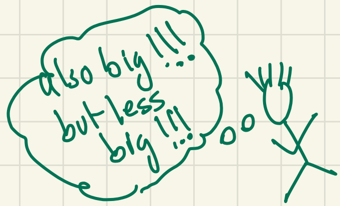
recall, $\Pr[f(x)=1] \in [\frac{1}{4}, \frac{3}{4}]$

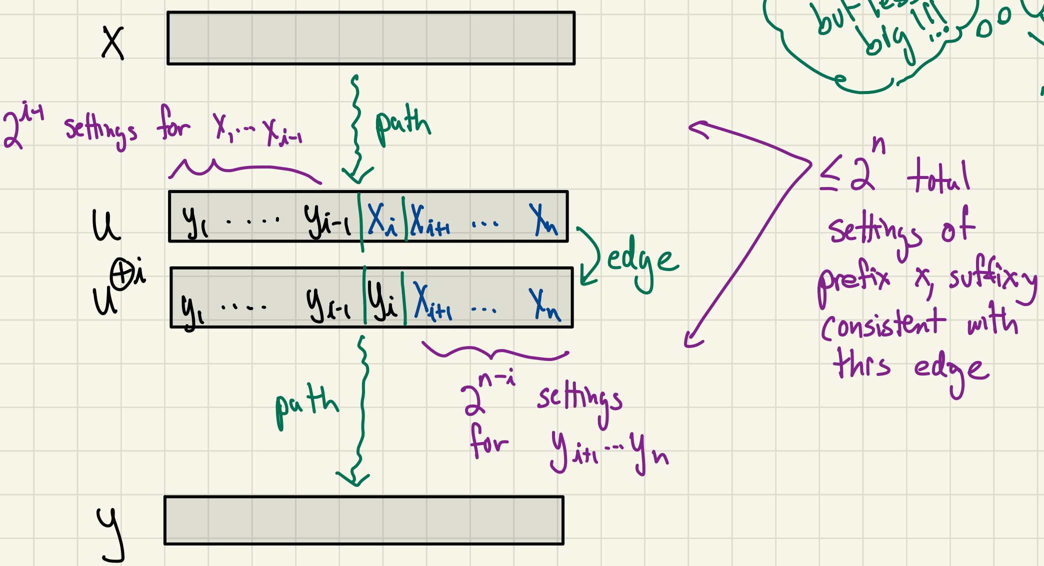
$$\#\text{paths} \geq \underbrace{\frac{1}{4}\cdot 2^n}_{\text{l.b. on \#red}} \cdot \underbrace{\frac{1}{4}\cdot 2^n}_{\text{l.b. on \#blue}} = \frac{1}{16}\cdot 2^{2n}$$



Part II: Show upper bound on # of c.p.'s passing through any edge

for any red-blue edge e, how many x-y pairs can cross it with canonical x-y path?





Main point: all canonical paths crossing $u, u^{\oplus i}$ agree on $y_1 \cdots y_{i-1}$ & $x_{i+1} \cdots x_n$

$\Rightarrow \leq 2^n$ possible paths for each $x_1 \cdots x_i$, $y_i \cdots y_n$

i.e. started at ++++

example:

$(-+++)$ or $(++++)$ must come from node with this suffix

two options for x

$$\begin{matrix} + & + & + & + \\ + & - & + & + \end{matrix} \qquad e = (++++,\ +-++)$$

must go to node with this prefix

difference in $i=2$

$(+---)$ or $(+--+)$ or $(+-+-)$ or $(+-++)$

4 options for y

Part III: Conclude lower bound on # of red-blue edges.

(# red-blue edges) × (max # canonical paths that use each edge)

$\geq$ # red-blue canonical paths

since each crosses ≥ 1 red-blue edge

$$\Rightarrow \# \text{ red blue edges} \geq \frac{\overbrace{\frac{1}{16}\cdot 2^{2n}}^{\text{l.b. on \# r-b pairs}}}{\underbrace{2^n}_{\text{u.b. on \# canonical paths crossing any edge}}} = \frac{1}{16}\cdot 2^n$$

$\Rightarrow \exists\, i$ s.t. $\geq \frac{2^n}{16}\cdot\frac{1}{n}$ red-blue edges in direction i

$$\Rightarrow \exists i \text{ s.t. } \mathrm{Inf}_i(f) = \hat{f}(\{i\}) = 2 \cdot \Pr[f(x) = x_i] - 1$$

$$\geq \frac{\frac{2^n}{16n}}{2^{n-1}} = \frac{1}{8n}$$

total # edges in dir i

$$\Rightarrow \exists i \text{ s.t. } \Pr[f(x) = x_i] \geq \frac{1}{2} + \frac{1}{16n}$$

✓

$\blacksquare$

Other uses of canonical path arguments:

- routing
- expansion/conductance of hypercube/other Markov chains

What good is weak learning?

unclear

here can only weakly learn on uniform distribution

ability to weakly learn on all distributions

$\Longrightarrow$ ability to strongly learn [Schapire]

"boosting"

Weak vs. Strong Learning

<u>Def.</u> Algorithm $\mathcal{A}$ "weakly PAC learns" concept class $\mathcal{C}$ if $\exists \gamma > 0$

s.t. $\forall c \in \mathcal{C}$ + $\forall$ dists $\mathcal{D}$

$\forall \delta > 0$ ← ($\delta = \frac{1}{4}$ or $\frac{1}{n^2}$ doesn't affect)

with prob $\geq 1-\delta$

given examples of c

$\mathcal{A}$ outputs h s.t. $\Pr_{\mathcal{D}}[h(x) = c(x)] \geq \underbrace{\frac{1}{2}}_{\text{not good compared to } 1-\varepsilon \text{ or } 99\%} + \frac{\gamma}{2}$

↑ advantage over guessing

It was first conjectured that weak learning is easier than strong (i.e. $\exists$ fctns that can weakly learn but not strongly learn)

Surprise!!

Can "boost" a weak learner

Thm if $\mathcal{C}$ can be weakly learned on

any dist $\mathcal{D}$ then $\mathcal{C}$ can be

$\underbrace{\text{(strongly)}}_{\text{i.e. } \forall \varepsilon}$ learned

dependence on γ ?

δ ?

ε ?

Will prove for case of $\mathcal{D}_0 = \mathcal{U}$

Applications:

1) "theoretical"

- uniform distribution algorithms for
  poly term DNF
  weight- w poly threshold fctns
  (Boosting + KM)

low degree alg doesn't work well

- Ave case vs. worst case complexity

2) practical: "Boosting"

Freund- Schapire

Good & Bad Ideas

1) simulate weak learner several times on same distribution & take

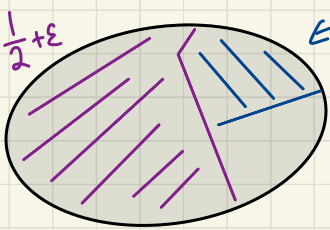
majority answer

<u>or</u>

best answer

- gives better confidence
- but doesn't reduce error – what if always get same answer?

2) filter out examples on which current hypothesis does well & run weak learner on part where you do badly



← $\frac{1}{2}+\varepsilon$ of non-purple

<u>Problem</u>: given <u>new</u> example, how do you know which section it is in?

3) Keep some samples on which you are ok in your filtering.
Always use majority vote on previous hypotheses to predict value of new samples.

history: Schapire, Freund-Schapire, Impagliazzo-Servedio-Klivans

Filtering Procedures:

- decide which samples to keep vs. throw out
- samples on which you guess

  correctly: needed for checking future hypotheses

  incorrectly: needed for improvement