Tutorial 5

01 April 2021 16:03

GAME THEORY

→ Lower bound on efficiency of randomized algorithms

→ Example: Tree evaluation

Bob

Alice	R	P	S
R	0	-1	1
P	1	0	-1
S	$\boxed{-1}$	1	0

→ Alice is trying to maximize outcome

→ Bob is trying to minimize outcome

Game evaluation matrix

Generally GEM $[M_{ij}]$ of real numbers

If Alice chooses strategy i, in the worst case she gets $\min_j \{M_{ij}\}$

If Bob chooses strategy j, in the worst case he gets $\max_i \{M_{ij}\}$

Alice's best strategy is $\max_i \min_j \{M_{ij}\} = O_A$

Bob's best strategy is $\min_j \max_i \{M_{ij}\} = O_B$

Bob's best strategy is $\min_j \max_i \{M_{ij}\} = O_B$

There are games for which $O_A = O_B$ (equilibrium)

$$\begin{array}{cc} & \\ \text{Alice} & \begin{matrix} 0 & -1 & \boxed{-2} \\ 1 & 0 & \boxed{-1} \\ \boxed{2} & \boxed{1} & \boxed{0} \end{matrix} \end{array}$$

MIXED STRATEGIES

Alice's strategy = probability distribution over rows p

Bob's Strategy = probability distribution over columns q

column vectors

$$\boxed{p^T M q} = \sum_{ij} p_i q_j M_{ij} = \text{Expected value of Game } M \text{ with strategies } p \text{ and } q.$$

For fixed strategy of Alice p she is guaranteed to achieve at least $\min_q p^T M q$.

fixed q guarantees Bob $\max_p p^T M q$

Alice's best strategy $\max_p \min_q p^T M q = O_A$

Bob's best strategy $\min_q \max_p p^T M q = O_B$

Von Neumann's theorem

$$\forall M \quad \max_p \min_q p^T M q = \min_q \max_p p^T M q$$

Loomis' theorem

$$\forall M \quad \max_p \min_k p^T M e_k = \min_q \max_j e_j^T M q$$

$e_i = (0, \ldots, 1, \ldots 0)$ ← i^{th} position

Proof sketch

$a^T = p^T M$

for fixed p : $\min_q \overbrace{p^T M}q = \min_q a^T \cdot q = a_1 q_1 + \cdots + a_n q_n$

find smallest a_i

and set $q_i = 1$ and others $q = 0$.

---

deterministic algorithms

✓ our randomized algorithm is choosing from

inputs ↓

	A_1	A_2	A_3	$\ldots$	A_n
I_0	$C(I_0, A_1)$	$C(I_0, A_2)$			
I_1	$C(I_1, A_1)$				
$\vdots$					
I_n					

In |

Choose probability of inputs p

and probability of choosing deterministic algorithms q

$E(C(I_p, A_q))$ = expected running time for input distribution p

and rand. algorithm characterized by q.

$$= p^T M q$$

vN's thm:

$$\max_p \min_q E(C(I_p, A_q)) = \min_q \max_p E(C(I_p, A_q))$$

Loomis' thm:

$$\cancel{\max_p} \min_{A_i \in A} E(C(I_p, A_i)) = \cancel{\min_q} \max_{i \in I} E(C[I_i, A_q])$$

$$\forall_{i,q} \quad \min_{A_i \in A} E(C(I_p, A_i)) \leq \max_{i \in I} E(C[I_i, A_q])$$

for chosen input distribution p find the best deterministic algorithm

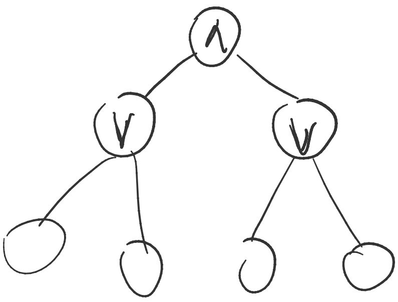
for given rand algorithm A_q find the worst input find the best deterministic algorithm

lower bound!

find the worst input

We are interested in this

Tree evaluation example

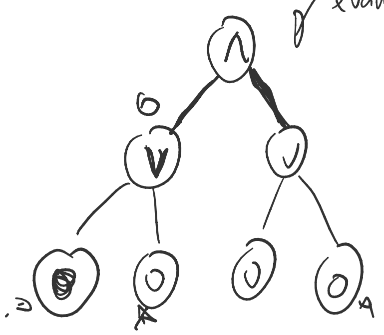


Example of input distribution

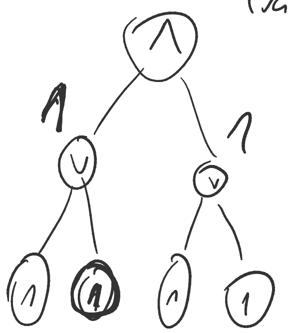
all 0's w.p. $1/2$ ⇝ (all nodes evaluate to 0)

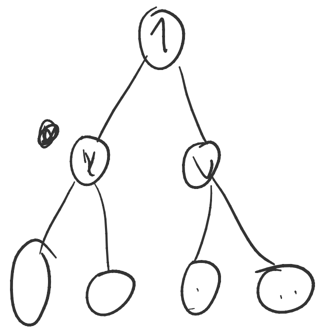
all 1's w.p. $1/2$ ⇝ (all nodes evaluate to 1)

2 leaf evaluations



2 leaf evaluations

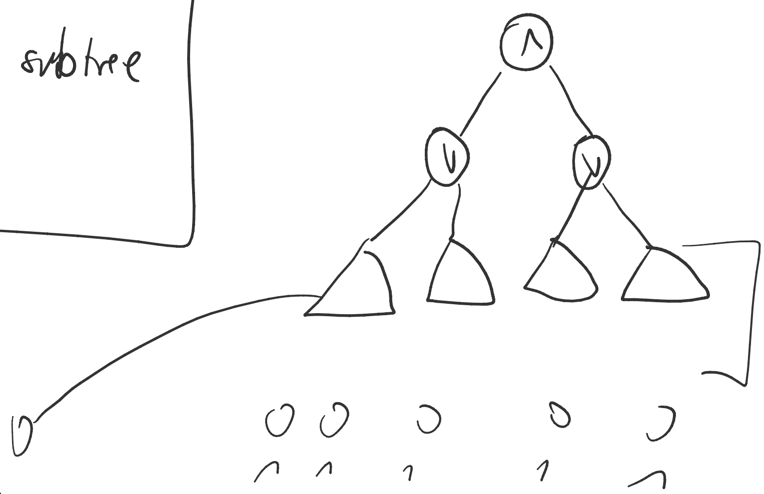




4 choices of first leaf

2 choices for leaf in the other subtree

= 8 algorithms



All subtrees evaluate to 0 w.p. $1/2$

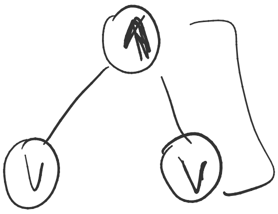
All subtrees evaluate to 0 w.p. 1/2

All subtrees evaluate to 1 w.p. 1/2

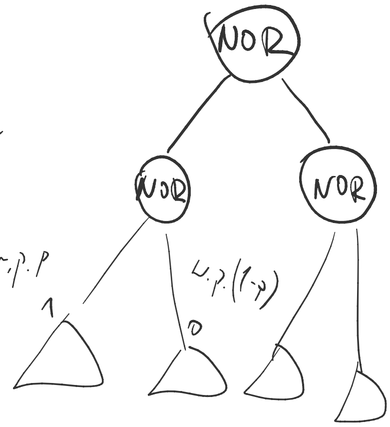
the best deterministic algorithm for any chosen input takes 2^k evaluations.

$\Downarrow$ min max theorem (Loomis's

Any randomized algorithm needs at least 2^k leaf evaluations.



$\approx$



$(a \vee b) \wedge (c \vee d)$

$\|$

$(a \text{ NOR } b) \text{ Nor } (c \text{ NOR } D)$

ab	a NOR b
~~00~~ 00	1
01	0
10	0
11	0

$P_r(\text{leaf} = 1) = p$

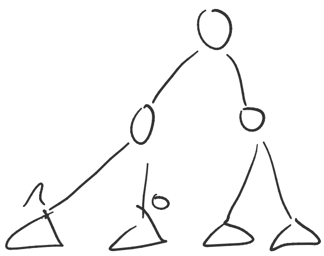
$P_r(\text{leaf} = 0) = 1 - p$

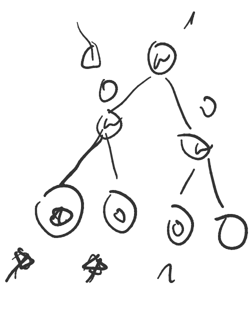
$(1-p)^2 = p \Rightarrow$ probability that

evaluates to 1 is equal to p

$\Downarrow$

$$p = \frac{3-\sqrt{5}}{2}$$

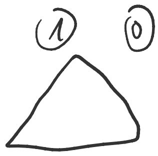


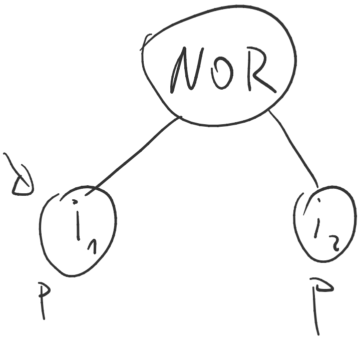


	$\Downarrow$		$\Downarrow$ (round down)
$(1-p)^4$ →	0000	4	0,14
$(1-p)^3 \cdot p$	0001	4	0,09
$(1-p)^3 \cdot p$	0010	3	⋮
	0011	3	⋮
	⋮		

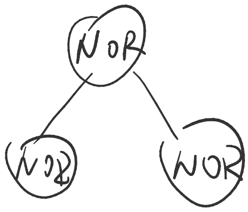
Expectation $2 < 2{,}52 < 3$

$< 2{,}61 < 3$





$p + (1-p) \cdot 2$ → each nor will take on average $p + (1-p) \cdot 2 = (2-p) < 1{,}61$



$p \cdot (2-p) + (1-p) \cdot 2 \cdot (2-p) = (2-p)^2$

$< 2{,}61$

=