Tutorial 5

17 March 2022 12:05

GAME THEORY

→ lower bound on efficiency of randomized algorithms

→ Example: Tree evaluation

		Bob's R	P	S
	R	0	-1	1
Alice's	P	1	0	-1
→	S	$\boxed{-1}$	1	0

→ Alice is trying to maximize the outcome

→ Bob is trying to minimize the outcome

← Game evaluation matrix

Generally GEM $\{M_{ij}\}$ of real numbers.

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

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

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

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

There are games for which $O_A = O_B$

→	0	-1	$\boxed{-2}$
→	1	0	$\boxed{-1}$
→ →	2	1	$\boxed{0}$

MIXED STRATEGIES

Alice's strategy = probability distribution over rows p

Alice's strategy = probability distribution over columns

Column

Alice's strategy = probability distribution over rows p
Bob's strategy = probability distribution over columns q
} Column vectors

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

For fixed strategy p Alice is guaranteed to achieve an expectation $\min_q p^T M q$

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

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

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

Loomi's theorem

$$\forall M \quad \max_p \min_{e} p^T M e_{e} = \min_q \max_j e_j^T M q$$

$e_i = (0, \dots, 1, \dots 0)$ ← ith position

for fixed p: $\overbrace{p^T M}^{a^T} q \Rightarrow \min_q a^T \cdot q = a_1 q_1 + a_2 q_2 + a_i q_i + a_n q_n$

to minimize over q find smallest a_i and set $q_i = 1$ all others to 0.

Inputs ↓	A_1	A_2	A_3	– – – –	A_n
I_0	$C(I_0, A_1)$	$C(I_0, A_2)$			

↙ deterministic algorithms our randomized algorithm is choosing from gorithms

Our randomized algorithm is choosing from

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

Choice of inputs with probability p and probability of choosing deterministic algorithms q.

$$E\big(C(I_p, A_q)\big) = \text{expected running time for input distribution } p \text{ and}$$

and randomized algorithm characterized by q.

$$= p^T M q.$$

vN's thm:

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

Loomis' thm:

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

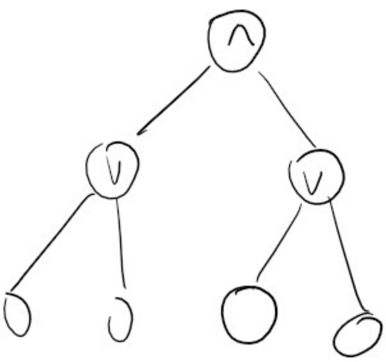
$$\boxed{\forall_{p,q}} \quad \min_{A_i \in \mathcal{A}} E(C(I_p, A_i)) \leq \max_{i \in I} E(C[I_i, A_q])$$

for chosen input distribution p find the best deterministic algorithm $\leq$ for all randomized algorithms A_q find the worst input

lower bound

typically interested in this

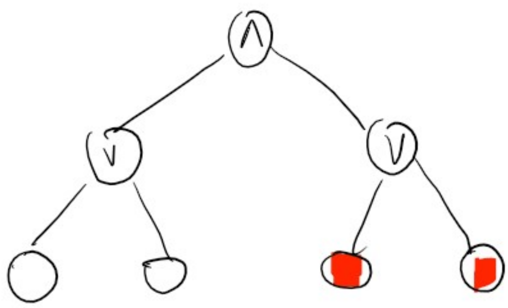
Tree evaluation



Example of input distribution

all 0's w.p. 1/2

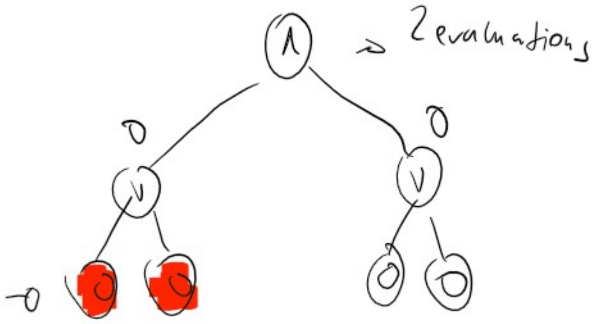
all 1's w.p. 1/2

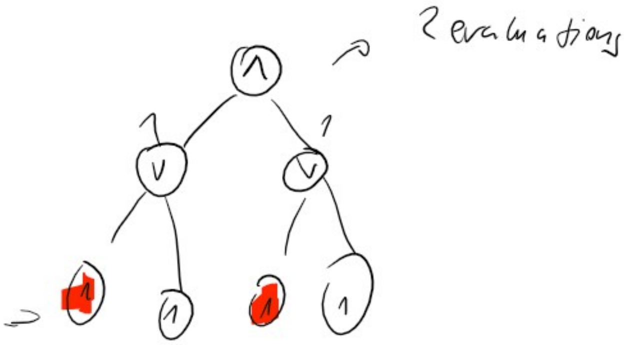


4 choices of firs leave

2 choices of first leaf in the other tree

= 8 traversal paths = 8 deterministic algorithms.





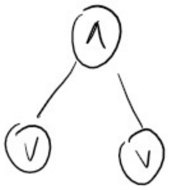
$\Downarrow$

the best deterministic algorithm for our chosen inputs takes 2^k evaluations

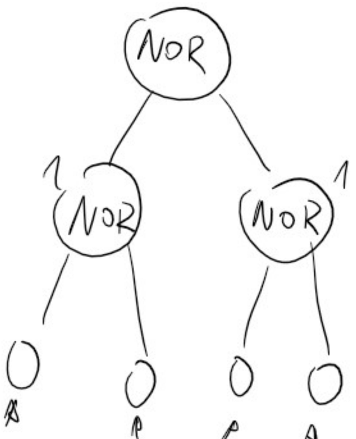
$\Downarrow$ Loomi's thm

Any randomized algorithm needs at least 2^k evaluations in the worst case.

$2^k < \longrightarrow \quad < 3^k$



$\approx$



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

$\shortparallel$

$(a \text{ nor } b) \text{ nor } (c \text{ nor } d)$

a b	a NOR b	
$\boxed{0\ 0}$	1	✓
0 1	0	
1 0	0	
1 1	0	

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

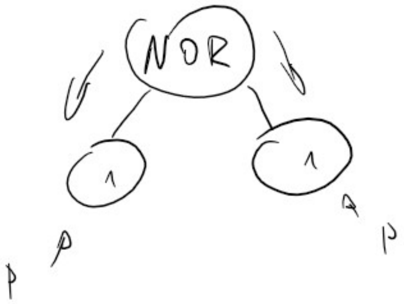
$Pr(\text{leaf} = 0) = 1 - p$

$$(1-p)^2 = p \quad \Rightarrow$$

evaluates to 1 with probability p

and we can use induction

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



$$p \cdot 1 + (1-p) \cdot 2 \overset{\text{plug in } p}{>} 1{,}61$$

left leaf has input value 1

left leaf has input value 0

0	0	→	2 step	$(1-p)^2 = p$
0	1	→	2 step	$p \cdot (1-p)$
1	0	→	1 steps	$p(1-p)$
1	1	→	1 steps	p^2

$$p\left[(1-p)+p\right]$$

$$\text{Expected} = 2p + 2p - 2p^2 + p - p^2 + p^2$$

$$= 5p - 3p^2 = p(5-3p)$$

$$p = \frac{3-\sqrt{5}}{2} \qquad = \frac{3-\sqrt{5}}{2}\left(5 - 3\left(\frac{3-\sqrt{5}}{2}\right)\right) > 1{,}61 \quad \checkmark$$

$$(1{,}61)^{2t} = \left[1.61^2\right]^{t} = (2.5921)^{t}$$