Tutorial 5

22 March 2023 14:00

Game Theory

→ lower bounds on efficiency of randomized algorithms

→ Example: AND-OR Tree evaluation

$$
\begin{array}{c|ccc}
 & \multicolumn{3}{c}{\text{Bob's}} \\
 & R & P & S \\
\hline
R & 0 & -1 & 1 \\
\text{Alice's } P & 1 & 0 & -1 \\
\to S & \boxed{-1} & 1 & 0
\end{array}
$$

→ Alice's goal is to maximize the outcome

→ Bob's goal is to minimize the 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 $\max_i \min_j \{M_{ij}\} = O_A$

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

There are games for which $O_A = O_B$

$$
\begin{array}{cccc}
 & & & \downarrow \\
 & 0 & -1 & \boxed{-2} \\
 & 1 & 0 & \boxed{-1} \\
\to & \boxed{2} & \boxed{1} & \boxed{0}
\end{array}
$$

MIXED STRATEGIES

Alice's strategy = probability distribution over rows p.

Bob's strategy = probabili ——— " — columns q.

} column vectors

$$p^T M q = \sum_{ij} p_i q_j M_{ij}$$ = <u>expected</u> value of game M with strategies p and q.

For fixed Alice's strategy p Alice is guaranteed to achieve an expectation $\min_q p^T M q$

for Bob 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 — " — $\min_q \max_p p^T M q = O_B$

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

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

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

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

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

---

deterministic algorithms
randomized algorithm "choose from"

	A_1	A_2	A_2	......	A_m
I_1	$C(I_1, A_1)$				
I_2					
$\vdots$					

	A_1	A_2	A_2	......	A_m "randomized algorithm "choose from"
I_1	$C(I_1, A_1)$				
I_2					
⋮					
I_n					

Choice of inputs with probability p and probability of choosing a deterministic algorithm q.

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

and randomized algorithm characterized by q.

$$= p^T M q$$

vN's thm:

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

Loomis' thm:

$$\cancel{\max_{p}} \min_{A_i \in \mathcal{A}} E\left(C(I_p, A_i)\right) = \cancel{\min_{q}} \max_{i \in I} E\left(C[I_i, A_q]\right)$$

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

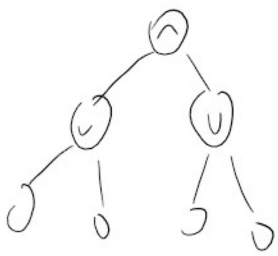
for each input distribution p find the best deterministic algorithm

lower bound

for each randomized algorithm A_q find the worst input

interested in this

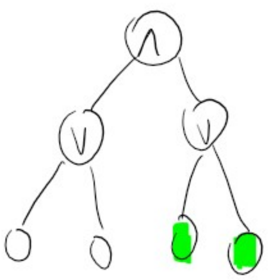
Tree evaluation



Example of input distribution

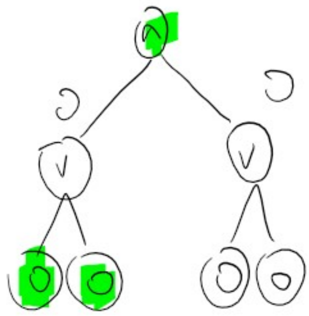
all 0's w.p. $1/2$

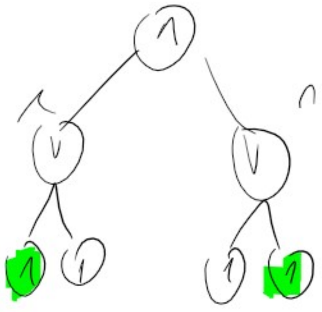
all 1's w.p. $1/2$



4 choices of 1st leaf to visit

4 choices of 1st leaf to visit
2 choices of 2nd leaf to visit = 8 traversal paths = 8 deterministic algs to choose from





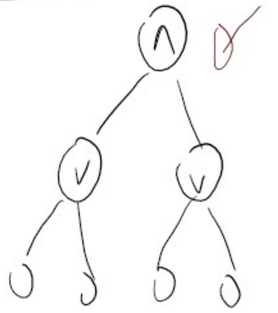
$\Downarrow$

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

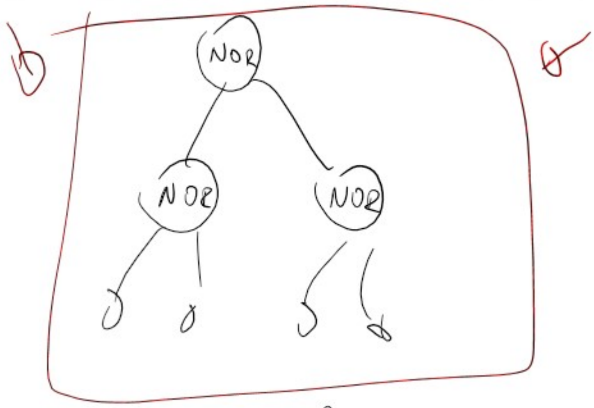
$\Downarrow$ Loomis thm

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

$$2^k \longrightarrow 3^k$$



$\approx$



00	1
01	0
10	0
11	0

Pr(leaf = 1) = p

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

$(1-p)^2 = p$

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

$\Rightarrow$  evaluates to 1 w.p. $(1-p)^2 = p$

$p \cdot 1 + (1-p) \cdot 2 \rightarrow 1{,}61$

(left subtree ~~has~~ value 1)

$(1{,}61)^{2k} = (1{,}61^2)^k = (2.59)^k < 3^k$

$<$

2^k