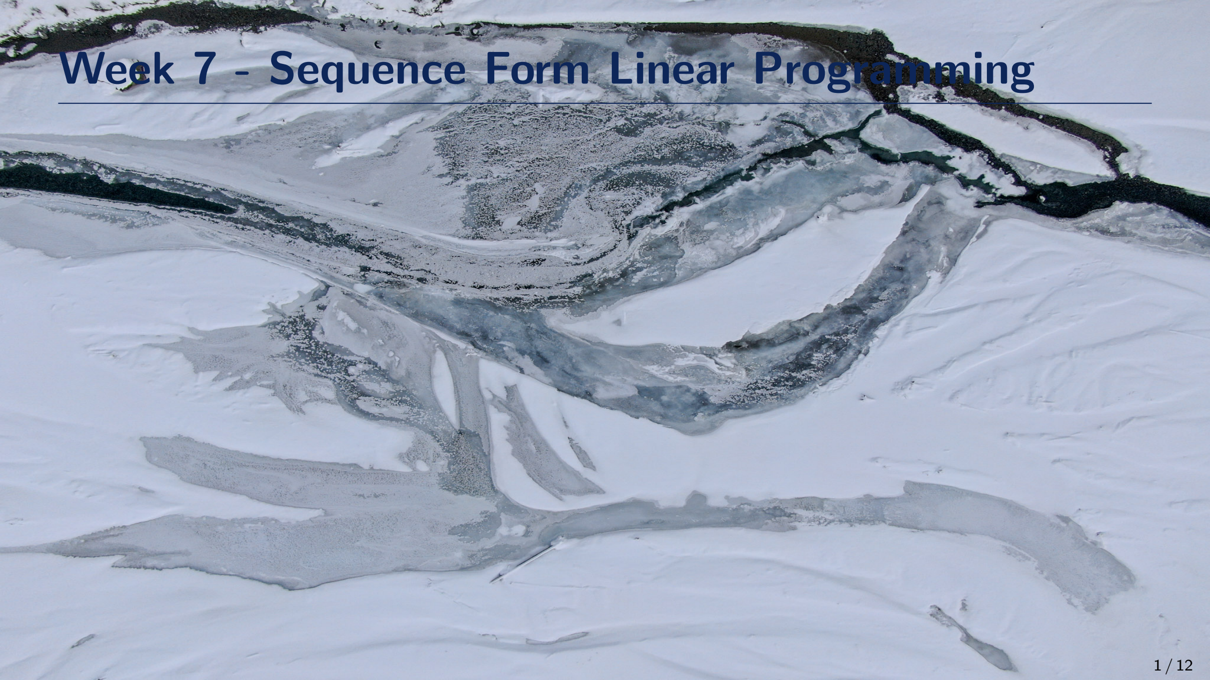
Week 7 - Sequence Form Linear Programming



Sequence Form

- The conversion to normal-form games allows us to solve extensive-form games using the techniques we already know (e.g., linear programming for two-player zero-sum case)
- Unfortunately, as we have seen, the constructed normal-form game can be exponentially large
- Can we somehow fix the fact that the constructed game can be exponential?
- The idea is to represent all paths - **sequences** for the players

Definition: Sequence

A sequence of moves of player i is the sequence of its actions on the path from the root (history $\varnothing$) to the node/history h, and is denoted $s_i(h)$.

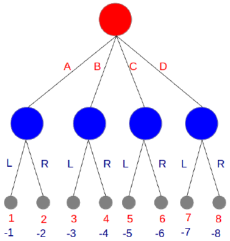
Sequence Form

- $s_1(A, K, check, bet, call) = (check, call)$
- $s_2(A, K, check, bet, call) = (bet)$
- Let S_i be the set of all sequence for a player
- Clearly, the size of S_i is linear in the size of the game tree
- Using the sequences, we can now conveniently formalize what **perfect recall** is

Definition: Perfect Recall

The game satisfies perfect recall if and only if for all players, $s_i(h_1) = s_i(h_2)$ for any two histories $h_1, h_2 \in \int_i$.

Sequence Form

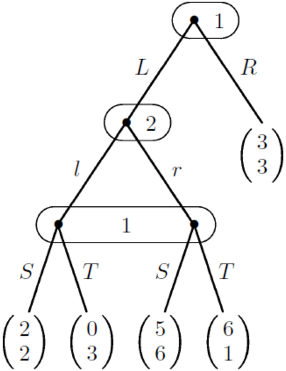


	A	B	C	D
(A-L, B-L, C-L, D-L)	1;-1	3;-3	5;-5	7;-7
(A-L, B-L, C-L, D-R)	1;-1	3;-3	5;-5	8;-8
(A-L, B-L, C-R, D-L)	1;-1	3;-3	6;-6	7;-7
(A-L, B-L, C-R, D-R)	1;-1	3;-3	6;-6	8;-8
(A-L, B-R, C-L, D-L)	1;-1	4;-4	5;-5	7;-7
(A-L, B-R, C-L, D-R)	1;-1	4;-4	5;-5	8;-8
(A-L, B-R, C-R, D-L)	1;-1	4;-4	6;-6	7;-7
(A-L, B-R, C-R, D-R)	1;-1	4;-4	6;-6	8;-8
(A-R, B-L, C-L, D-L)	2;-2	3;-3	5;-5	7;-7
(A-R, B-L, C-L, D-R)	2;-2	3;-3	5;-5	8;-8
(A-R, B-L, C-R, D-L)	2;-2	3;-3	6;-6	7;-7
(A-R, B-L, C-R, D-R)	2;-2	3;-3	6;-6	8;-8
(A-R, B-R, C-L, D-L)	2;-2	4;-4	5;-5	7;-7
(A-R, B-R, C-L, D-R)	2;-2	4;-4	5;-5	8;-8
(A-R, B-R, C-R, D-L)	2;-2	4;-4	6;-6	7;-7
(A-R, B-R, C-R, D-R)	2;-2	4;-4	6;-6	8;-8

	0	A	B	C	D
0	0;0	0;0	0;0	0;0	0;0
L_a	0;0	1;-1	0;0	0;0	0;0
R_a	0;0	2;-2	0;0	0;0	0;0
L_b	0;0	0;0	3;-3	0;0	0;0
R_b	0;0	0;0	4;-4	0;0	0;0
L_c	0;0	0;0	0;0	5;-5	0;0
R_c	0;0	0;0	0;0	6;-6	0;0
L_d	0;0	0;0	0;0	0;0	7;-7
R_d	0;0	0;0	0;0	0;0	8;-8

Figure: Normal form vs. sequence form

Extensive Form Games To Normal Form Games



$$A = \begin{array}{c} \begin{array}{cc} l & r \end{array} \\ \left[\begin{array}{cc} 2 & 5 \\ 0 & 6 \\ 3 & 3 \\ 3 & 3 \end{array}\right] \end{array} \begin{array}{l} \\ \langle L,S\rangle \\ \langle L,T\rangle \\ \langle R,S\rangle \\ \langle R,T\rangle \end{array} \qquad B = \begin{array}{c} \begin{array}{cc} l & r \end{array} \\ \left[\begin{array}{cc} 2 & 6 \\ 3 & 1 \\ 3 & 3 \\ 3 & 3 \end{array}\right] \end{array} \begin{array}{l} \\ \langle L,S\rangle \\ \langle L,T\rangle \\ \langle R,S\rangle \\ \langle R,T\rangle \end{array}$$

$$A = \begin{array}{c} \begin{array}{ccc} \emptyset & l & r \end{array} \\ \left[\begin{array}{ccc} & & \\ & & \\ 3 & & \\ & 2 & 5 \\ & 0 & 6 \end{array}\right] \end{array} \begin{array}{l} \\ \emptyset \\ L \\ R \\ LS \\ LT \end{array} \qquad B = \begin{array}{c} \begin{array}{ccc} \emptyset & l & r \end{array} \\ \left[\begin{array}{ccc} & & \\ & & \\ 3 & & \\ & 2 & 6 \\ & 3 & 1 \end{array}\right] \end{array} \begin{array}{l} \\ \emptyset \\ L \\ R \\ LS \\ LT \end{array}$$

Figure: Normal form, sequence form, the payoff matrices

Sequence Form

- We can see that the sequence form is a valid, lossless representation
- Can we just use it directly to create another linear program to solve the game?
- But we can't choose the rows/columns of the payoff matrix A/B arbitrarily as in normal-form game!
- Denote the probability the players assigns to his sequence $s_i(h)$ as $x(s_i(h))$.
- See that x now might not be a probability distribution to form a correct strategy! (find an example)
- Let's add some constraints for the sequence's probabilities, so that these probabilities form a valid strategy

$$\sum_{a \in A(h)} x(s_i(h), a) = x(s_i(h))$$

$$x(\varnothing) = 1$$

Sequence Form

$$\sum_{a \in A(h)} x(s_i(h), a) = x(s_i(h))$$

$$x(\varnothing) = 1$$

- Note that these restrictions are linear
- To compute the expected payoff given the sequence strategies, we just need to go through all the terminal nodes, and compute the reach probability of that node

$$\sum_{t \in Z} \sigma_c(t)\sigma_1(t)\sigma_2(t)u_i(t)$$

- Now we can formalize the the utility matrix A (for player 1)

$$A_{x,y} = \sum_{t \in Z \,|\, s_1(t)=x, s_2(t)=y} \sigma_c(t)\sigma_1(t)\sigma_2(t)u_1(t)$$

Sequence Form

- We can write $\sum_{a \in A(h)} x(s_i(h), a) = x(s_i(h)),\ x(\varnothing) = 1$ as $Ex = e$
- Similarly, we can use $Fy = f$ for the second player

$$\begin{bmatrix} 1 & & & & \\ -1 & 1 & 1 & & \\ & -1 & & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Figure: $Ex = e$, $Fy = f$

- To compute a best response (given a fixed strategy y)

$$\begin{aligned} \max_x \quad & x^T A y \\ \text{subject to} \quad & Ex = e,\ x \geq 0 \end{aligned}$$

- Now consider the dual problem

$$\begin{aligned} \min_x \quad & e^T u \\ \text{subject to} \quad & E^T u \geq Ay \end{aligned}$$

Sequence Form

- The dual problem

$$\begin{array}{ll} \min\limits_{x} & e^T u \\ \text{subject to} & E^T u \geq Ay \end{array}$$

- This dual finds the best response (for Player 1).
- In the case of a zero-sum game, Player 2 wants to minimize this value.
- The strategy of Player 2 is denoted as y

Sequence Form

$$\begin{aligned} \min_{u} \quad & e^T u \\ \text{subject to} \quad & E^T u \geq Ay \end{aligned}$$

- We need to make sure that y forms a valid strategy: $Fy = f$
- Now y won't be fixed, but Player 2 chooses the strategy it wants to play

Final Linear Program

$$\begin{aligned} \min_{u,y} \quad & e^T u \\ \text{subject to} \quad & E^T u \geq Ay,\ Fy = f, y \geq 0 \end{aligned}$$

Theorem - Sequence Form LP

The solution to the "Final Linear Program" corresponds to a Nash equilibrium (sequence form) for two-player zero-sum games

Corolllary

There's a polynomial algorithm to compute a Nash equilibrium for two-player zero-sum extensive-form games

Week 7 Homework

This week, your task is to implement:

- a function that converts extensive-form games to normal-form games
- a function that finds Nash equilibrium of a given extensive-form game using Sequence Form LP