2. Players and adversaries

I. Modelling Winning Probabilities in Stochastic Games

Giacomo Bergami

Newcaslte University

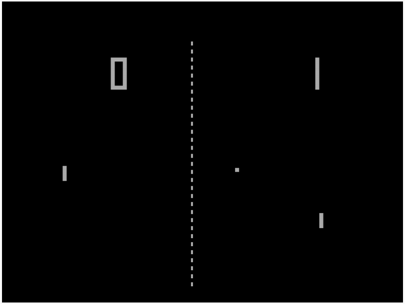
Objectives

- Understanding how probabilities can be used to model different game properties:
  1. Player's skills, **PvP**: Simplified Tennis (PONG)
  2. Player's chances at advancing the game: Snakes and Ladders.
- Using stochastic models for establishing the interconnection between probabilities and time:
  1. Which is the probability of ending/winning the game in n turns?
  2. Which is the average number of turns required for ending the game?
- Understanding how changes in the model and/or its probability distribution alters the previous questions, and their effect on gameplay.

Chances vs. Skills

- Chance allows a weaker player to beat a stronger one
- The outcome of the game should be mainly influenced by the skills
- Still, the skills of a player are hard to assess correctly.
- As a first approximation, we can model the player's skills as the chances of winning:
  - In a simplistic game, a player could only perform two actions: one, advancing the game towards the next turn, and the other making the player fail thus letting the opponent win.

Simplified Tennis (PONG) (1/3)

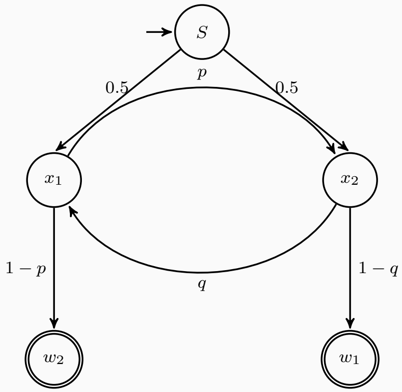


- One single graph represents one single (tennis) play (i.e., *round*).
- One of the two players, the *server*, starts the game (2 possible initial states). This player is randomly picked (S).
- We have 2 states identifying one of the two players actively playing (x_1 and x_2), and 2 states (w_1 and w_2) declaring the winning for the given tennis game (e.g., *round*).
- The transition probabilities (p, q) represent the player's expertise in performing the correct action

Simplified Tennis (PONG) (2/3)

- The game is *memoryless*: the ability of responding to a service is completely unrelated to the previous move.
- We can represent such a game via a probabilistic process, namely a Discrete Time Markov Chain (DTMC), represented as a transition matrix T.
- Such matrix can be graphically represented as a weighted directed graph.

Simplified Tennis (PONG) (3/3)



$$T = \begin{array}{cc} & \begin{array}{ccccc} S & x_1 & x_2 & w_1 & w_2 \end{array} \\ \begin{array}{c} S \\ x_1 \\ x_2 \\ w_1 \\ w_2 \end{array} & \begin{bmatrix} 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & p & 0 & 1-p \\ 0 & q & 0 & 1-q & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

- **State Space**: Determining the server (S), Player1's turn (x_1), Player2's turn (x_2), Player 1 wins (w_1), Player 2 wins (w_2).
- **Initial State**: S.
- **Actions & Transitions**:
  1. $P1$ serves, $S \xrightarrow{0.5} x_1$
  2. $P2$ serves, $S \xrightarrow{0.5} x_2$
  3. $P1$ responds, $x_1 \xrightarrow{p} x_2$
  4. $P2$ responds, $x_2 \xrightarrow{q} x_1$
  5. $P1$ does not catch, $x_1 \xrightarrow{1-p} w_2$
  6. $P2$ does not catch, $x_2 \xrightarrow{1-q} w_1$
- **Goal Test**: Either w_1 or w_2 are reached.

Average Hitting Time (1/2)

The *average hitting time* determines the mean number of turns x_i required to reach one of the goals for the first time from state i. We can prove that this reduces to solve the following *linear (equation) system* for all the states i:

$$\begin{cases} x_i = 0 & i \text{ goal} \\ -x_i + \sum_{j \neq i} T_{ij} x_j = -1 & \text{otherwise} \end{cases}$$

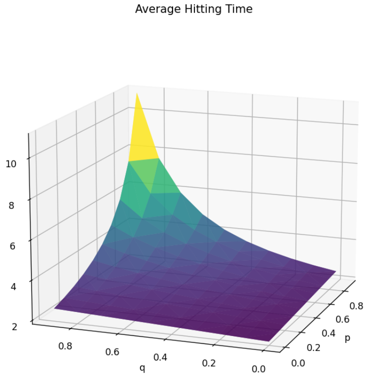
This reduces to solve a linear system $A\vec{x} = \vec{b}$, where:

$$\vec{b}_i = \begin{cases} 0 & i \text{ goal} \\ -1 & \text{otherwise} \end{cases} \qquad A_{ij} = \begin{cases} 1 & i = j, i \text{ goal} \\ 0 & i \neq j, i \text{ goal} \\ -1 & i \neq j, i \text{ not goal} \\ T_{ij} & \text{otherwise} \end{cases}$$

Average Hitting Time (2/2)

By solving the system, we obtain the following *Average Hitting Time*:

$$x_s = \frac{4 + p + q - 2pq}{2 - 2pq}$$



We now exploit *average hitting time* for determining how log one single round of the PONG game will take dependently from the success probabilities p and q for both players.

- If only one player is strong, it will take as little as ~ 2 turns after tossing the coin.
- The more experienced the two players are, the longer it will take!

Probability of winning in n turns (1/3)

In the previous slide we discussed how long should it take to end the game when p and q vary:

- This gives no information of the chances of winning of one player in a given amount of turns.
- To simplify the analysis, we assume that Player1 is an expert, $p = 0.8$, while we test varying opponent (Player2) skills.
- Furthermore, we want to analyse the probability of Player1 and Player2 of winning after n turns in the game:
- This can be assessed by assessing the Reachability Probability at step n.

Probability of winning in n turns (2/3)

Initial Probability Distribution μ:

- Given that in our game we have only one initial state, S, we will have an **initial probability distribution** μ as an empty vector except for $\mu_S = 1$.

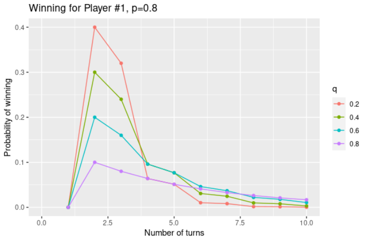
Accepting States/Goal Tests:

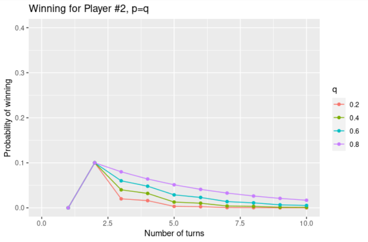
- Given that the two states declaring the winner, w_1 and w_2, have no outgoing edges and given that all the intermediate states can reach such a state, we want to assess the probability to reach such states.

Reachability Probability at step n:

- μT returns a vector $\mu^{(1)}$, where $\mu_i^{(1)}$ indicates the probability of reaching a state i in one step.
- The probability of Player1 winning in 1 turn is $\mu_{w_1}^{(1)} = 0$.
- The probability of reaching any game configuration in two steps is $\mu^{(2)} = \mu^{(1)}T = \mu TT = \mu T^2$:
- so, the probability of Player1 winning in n turns is given by $[\mu T^n]_{w_1}$

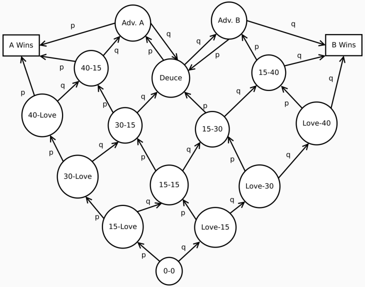
Probability of winning in n turns (3/3)





- We need to first toss the coin before assessing a winning probability.
- The two players have higher chances of winning at the first stages of the game.
- When both opponents have the same skills ($q = 0.8$), they both have the same distribution of chances of winning.
- The lower the skills for Player2, the higher are the changes of winning for Player1.

More complex games: Semi-Realistic Tennis



- Each PONG process could be one single nested process within the diagram on the left.
- Please observe: from theory, nested process are equivalent to un-nested ones, so:
  1. Replace each circle state c from the left with a whole PONG game instance, where
  2. the ingoing edges to w_1 (and w_2) state have now the same target ot $c \xrightarrow{p} c'$ (and $x \xrightarrow{q} c''$), and
  3. all the ingoing edges to a circle are redirected to each node S.
  4. Remove the unconnected w_1 and w_2.
- **Exercise**: compare the results of PONG with the ones with Semi-Realistic Tennis.