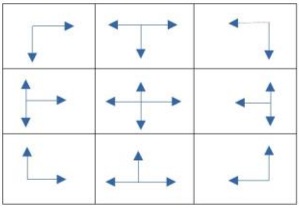
Tutorial 2 - Markov Decision Processes

Exercise 1

Consider a 3x3 grid world where an agent can move between cells. The agent's goal is to navigate this world while maximizing its rewards.



- The agent can take four possible actions: Up (↑), Down (↓), Left (←), or Right (→).
- The possible actions deterministically cause the agent to move one cell in the respective directions.
- If the agent decides to take a non-permitted action, it stays in the same state (cell), and gets a reward of -1.
- Otherwise, the agent transits to an adjacent cell with rewards given as follows.

s	(1,1)	(1,1)	(1,2)	(1,2)	(1,2)	(1,3)	(1,3)	(2,1)	(2,1)	(2,1)	(2,2)	(2,2)
s'	(1,2)	(2,1)	(1,1)	(1,3)	(2,2)	(1,2)	(2,3)	(1,1)	(2,2)	(3,1)	(1,2)	(2,1)
r	1	2/3	1/2	3/2	2	1/2	5/2	1/3	4/3	3/2	1/4	1/3

s	(2,2)	(2,2)	(2,3)	(2,3)	(2,3)	(3,1)	(3,1)	(3,2)	(3,2)	(3,2)	(3,3)	(3,3)
s'	(2,3)	(3,2)	(1,3)	(2,2)	(3,3)	(2,1)	(3,2)	(2,2)	(3,1)	(3,3)	(2,3)	(3,2)
r	3/2	3	1/4	1	7/2	1/2	3/2	4/5	1	3	1/2	4/5

- We assume a discount rate $\gamma = 0.7$
- We consider the following three policies:
  - π_1: If row ≠ 3: go down (↓), otherwise: go right (→).
  - π_2:
    - if row=2 and column=2: take the four possible actions with (1/4,1/4,1/4,1/4).
    - if row ≠ 2 and column ≠ 2: take the two actions with positive reward, with probabilities (1/2,1/2).
    - if (row=2 and column ≠ 2) or (row ≠ 2 and column=2): go to (2,2).
  - π_3: equidistributed directions with probability 1/4, for all the states.

Questions

1. Formalize the problem as a Markov decision process (MDP).
2. Determine the size of the state and action spaces.
3. At a given time step t, suppose the agent is in one of the following states and selects one of the following actions. What are the possible rewards and subsequent states for the three policies?

Current state	Selected action	π_1	π_2	π_3
(1,1)	up			
	down			
	right			
	left			
(2,2)	up			
	down			
	right			
	left			
(3,1)	up			
	down			
	right			
	left			

4. Is this problem a continuing or episodic task? Justify your answer.
5. How to transform this problem into an episodic task?
6. Initialize the Policies π_1, π_2, and π_3 with their respective action probabilities
7. Define the reward function that assigns a scalar value to each state-action pair
8. Define the state transition function that returns the next state based on the current state and the executed action.
9. Define select_action() function that selects an action based on the current state and a given policy
10. Define the get_trajectory() function that simulates a single episode by following the specified policy. This function returns a trajectory of sequential experiences, where each experience is represented as a tuple containing the (state, action, reward).
11. Define the get_returns() function that returns the discounted returns for each state in the given trajectory

12. Given that the Monte Carlo method computes the value function of each policy by averaging the discounted returns for each state generated from a set of simulated episodes. Compute the value function of each policy
    a. Initial value function for all states is 0.
    b. The starting state is (1,1)
    c. Episode length limit =20
    d. $V_\pi(s) = \frac{\sum_{i=1}^{N} G_i(s)}{N}$
        - N is the number of discounted returns for state
        - G(s) is the discounted return starting from state s
        - $V_\pi(s)$ is the value function (expected return) of state $\boldsymbol{s}$ under policy π
13. If discount rate γ=1 which of the following must be true?
    a. The agent only cares about the most immediate reward.
    b. The reward is not discounted.
    c. The agent cares more about future reward than present reward.

Exercise 2

We consider a system of two states s_1 and s_2. The system transits to any of the two states by taking actions in a set of two actions a_1 and a_2, with a discount factor γ = 0.8.
The transition probabilities and the resulted rewards are given as follows.

s	a	s'	$p(s'\|s,a)$	$r(s'\|s,a)$
s_1	a_1	s_1	0.7	−1
s_1	a_1	s_2	0.3	1
s_1	a_2	s_1	0.8	−1/2
s_1	a_2	s_2	0.2	3/2
s_2	a_1	s_1	0.9	−2/3
s_2	a_1	s_2	0.1	5/4
s_2	a_2	s_1	0.5	−1
s_2	a_2	s_2	0.5	1

1. Write the Bellman optimality equation (system of two equations and two variables).
2. Solve the system (Bellman optimality equation) using the fixed-point iteration method.
3. Derive the optimal policies.