HW: MDP and Reinforcement Learning

1. **Optimal Meal Decisions with MDPs and Q-Learning**

   You are deciding between two meal options for lunch at your favorite cafeteria: Healthy Meal or Junk Food. You enjoy junk food (action), but you worry about the consequences on your health (state). This decision can be modeled as an MDP with two states: Healthy and Unhealthy, and two actions: Eat Healthy and Eat Junk Food.

   (1) States (S):
   - Healthy: You feel energetic and in good shape.
   - Unhealthy: You feel lethargic and unwell.

   (2) Actions (A):
   - Eat Healthy: You prioritize a balanced and nutritious meal.
   - Eat Junk Food: You indulge in a tasty but less nutritious meal.

   (3) Rewards ($R(s, a)$):
   - Eating healthy rewards you with long-term energy and satisfaction.
   - Eating junk food rewards you with immediate satisfaction but risks making you unhealthy.

   (4) Transition Probabilities ($P(s'|s, a)$):
   - Eating healthy increases the likelihood of staying healthy.
   - Eating junk food increases the likelihood of becoming unhealthy.

   (5) Discount Factor (γ):
   - $\gamma = 0.9$

   The rewards and transition probabilities for this MDP are given in the tables below:

   Rewards Table

State	Action	Reward
Healthy	Eat Healthy	8
Healthy	Eat Junk Food	10
Unhealthy	Eat Healthy	4
Unhealthy	Eat Junk Food	2

Transition Probabilities Table

From State	Action	Probability of Transitioning to State s'
		Healthy (s')
Healthy	Eat Healthy	0.9
Healthy	Eat Junk Food	0.6
Unhealthy	Eat Healthy	0.7
Unhealthy	Eat Junk Food	0.3

Questions:

(1) Value Iteration

- Perform value iteration for three iterations, starting with $V_0(s) = 0$ for all states.
- Fill in the following table with the results for each iteration:

Iteration	$V^*(Healthy)$	$V^*(Unhealthy)$
0	0	0
1		
2		
3		

(2) Policy Iteration

- Using the same MDP, perform policy iteration to determine the optimal policy.

  Assume the initial policy is as follows:
  - Healthy: Always eat junk food.
  - Unhealthy: Always eat healthy.
- Update the policy and value function for each iteration and indicate the final optimal policy.

Iteration	Policy($\pi(Healthy)$)	Policy($\pi(Unhealthy)$)
0		
1		
2		
Final		

(3) Q-Learning

Using the following sequence of actions and outcomes, update the Q-values step by step.

Assume:

- Learning rate $\alpha = 0.5$,

- Discount factor $\gamma = 0.9$,
- Initial $Q(s,a) = 0$ for all s, a.

Step	State	Action	Next State	Reward
1	Healthy	Eat Junk Food	Healthy	10
2	Healthy	Eat Healthy	Healthy	8
3	Healthy	Eat Junk Food	Unhealthy	10
4	Unhealthy	Eat Healthy	Healthy	4
5	Healthy	Eat Junk Food	Unhealthy	10

Fill in the Q-value table after each step using the Q-learning update rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma_{a'}^{\max} Q(s',a') - Q(s,a)] \cdot$$

Step	0	1	2	3	4	5
$Q(Healthy, Eat\ Healthy)$	0					
$Q(Healthy, Eat\ Junk\ Food)$	0					
$Q(Unhealthy, Eat\ Healthy)$	0					
$Q(Unhealthy, Eat\ Junk\ Food)$	0					

(4) Compare Policies

- Compare the optimal policies derived from value iteration, policy iteration, and Q-learning. Are they the same? Why or why not?

2. **Problem Setup:**

Let C be a set of cities joined by a set of roads $R(i,j)$, where $R(i,j)$ means you can take a road from city i to city j. Traversing the road from i to j takes time $t_{i,j}$, but traffic can cause additional delays.

Specifically:

- If you are in city i and try to go to city j, you succeed with probability $1 - \alpha_i$ and fail with probability α_i. The success or failure does not depend on previous trials.
- If you **fail** to travel to j, it takes $u_{i,j} > 0$ time, and you remain in city i.
- If you **succeed**, it takes $v_{i,j} > u_{i,j}$, time to reach city j.

Your goal is to minimize the expected commute time from a starting city $c_1 \in C$ to a destination city $c_2 \in C$.

Questions:

(1) Formulate this problem as an MDP.

- What are the states, actions, rewards, and transitions?

(2) Construct a deterministic state space model such that:

- The optimal policy in the MDP is the same as the minimum-cost path in this deterministic state space model.
- The expected utility (total commute time) of the MDP policy matches the utility of the deterministic state space model.

Ensure the MDP and the deterministic state space model share the same set of states and actions.

3. Use Agent Type: **Q-learning** at the RL Playground to observe and analyze how specific parameter combinations affect the agent's learning performance and behavior. Discuss insights gained from the results and explain the underlying reasons. (https://alazareva.github.io/rl_playground/)

Parameter set 1: $\alpha = 0.5$ $\epsilon = 0.05$ $\gamma = 0.0$
Parameter set 2: $\alpha = 0.5$ $\epsilon = 0.05$ $\gamma = 0.9$
Parameter set 3: $\alpha = 0.5$ $\epsilon = 0.05$ $\gamma = 1.0$
Parameter set 4: $\alpha = 0.5$ $\epsilon = 0$ $\gamma = 0.9$

Questions:

(1) Parameter Set 1: $\gamma = 0.0$

How does setting $\gamma = 0.0$ affect the agent's preference for short-term vs. long-term rewards? Does the agent still learn an effective policy?

(2) Parameter Set 2: $\gamma = 0.9$

How does a high discount factor encourage balancing immediate rewards with long-term gains? How does the agent's behavior differ compared to $\gamma = 0.0$?

(3) Parameter Set 3: $\gamma = 1.0$

When $\gamma = 1.0$, how does placing full emphasis on future rewards impact the stability of learning and the learned policy?

(4) Parameter Set 4: $\epsilon = 0.0$

What happens when there is no exploration ($\epsilon = 0$)? Does the agent discover the optimal policy? Why or why not?