Stanford CME 241 (Winter 2021) - Assignment 4

Assignments:

1. **Manual Value Iteration:** Consider a simple MDP with $\mathcal{S} = \{s_1, s_2, s_3\}, \mathcal{T} = \{s_3\}, \mathcal{A} = \{a_1, a_2\}$. The State Transition Probability function

$$\mathcal{P} : \mathcal{N} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$$

is defined as:

$$\mathcal{P}(s_1, a_1, s_1) = 0.2, \mathcal{P}(s_1, a_1, s_2) = 0.6, \mathcal{P}(s_1, a_1, s_3) = 0.2$$
$$\mathcal{P}(s_1, a_2, s_1) = 0.1, \mathcal{P}(s_1, a_2, s_2) = 0.2, \mathcal{P}(s_1, a_2, s_3) = 0.7$$
$$\mathcal{P}(s_2, a_1, s_1) = 0.3, \mathcal{P}(s_2, a_1, s_2) = 0.3, \mathcal{P}(s_2, a_1, s_3) = 0.4$$
$$\mathcal{P}(s_2, a_2, s_1) = 0.5, \mathcal{P}(s_2, a_2, s_2) = 0.3, \mathcal{P}(s_2, a_2, s_3) = 0.2$$

The Reward Function

$$\mathcal{R} : \mathcal{N} \times \mathcal{A} \to \mathbb{R}$$

is defined as:

$$\mathcal{R}(s_1, a_1) = 8.0, \mathcal{R}(s_1, a_2) = 10.0$$
$$\mathcal{R}(s_2, a_1) = 1.0, \mathcal{R}(s_2, a_2) = -1.0$$

Assume discount factor $\gamma = 1$.

Your task is to determine an Optimal Deterministic Policy *by manually working out* (not with code) simply the first two iterations of Value Iteration algorithm.

- Initialize the Value Function for each state to be it's max (over actions) reward, i.e., we initialize the Value Function to be $v_0(s_1) = 10.0, v_0(s_2) = 1.0, v_0(s_3) = 0.0$. Then manually calculate $q_k(\cdot, \cdot)$ and $v_k(\cdot)$ from $v_{k-1}(\cdot)$ using the Value Iteration update, and then calculate the greedy policy $\pi_k(\cdot)$ from $q_k(\cdot, \cdot)$ for $k = 1$ and $k = 2$ (hence, 2 iterations).
- Now argue that $\pi_k(\cdot)$ for $k > 2$ will be the same as $\pi_2(\cdot)$. Hint: You can make the argument by examining the structure of how you get $q_k(\cdot, \cdot)$ from $v_{k-1}(\cdot)$. With this argument, there is no need to go beyond the two iterations you performed above, and so you can establish $\pi_2(\cdot)$ as an Optimal Deterministic Policy for this MDP.

2. Let's revisit the frog-croaking Markov Decision Process (Question 3 on Assignment 3). Compute the Optimal Value Function and Optimal Policy using Policy Iteration algorithm as well as Value Iteration algorithm (use the functions policy_iteration and value_iteration from file rl/dynamic_programming.py). Analyze the computational efficiency of Policy Iteration versus Value Iteration in terms of speed of convergence to the Optimal Value Function. How do Policy Iteration and Value Iteration compare against the brute-force method of evaluating the MDP for all 2^n deterministic policies (as was suggested in Assignment 3)? Plot some graphs of convergence speed for different values of n (the number of lilypads).