INDENG 290: Problem Set 2

Due: October 14, 2022 at 3 pm PST

Homework Collaboration Policy.
Please feel free to discuss the homework problems in groups if you prefer, however, you must write/code you solutions independently.

All code must be written in Python and submitted on Gradescope along with the theoretical part of the assignment.

1. **Problem 1 (20 pts).** Consider the Snakes and Ladders game (single player version) that we covered in class. Compute the probability distribution function of finishing the game in X dice rolls as a result of simulation over 5000 runs. What is the expected number of dice rolls that is needed to finish the game?

2. **Problem 2 (10 pts).** Consider two Markov Decision Processes, M_1 and M_2, with corresponding reward functions R_1 and R_2. Suppose M_1 and M_2 are identical except that the rewards for R_2 are shifted by a constant from the rewards for R_1, i.e., for all states s, $R_2(s) = R_1(s) + c$, where c does not depend upon s. Prove that the optimal policy must be the same for both Markov Decision Processes.

3. **Problem 3 (40 pts).** Consider an array of $n+1$ lilypads on a pond, numbered 0 to n. A frog sits on a lilypad other than the lilypad numbered 0 or n. When on lilypads i $(1 \leq i \leq n-1)$, the frog can croak one of two sounds A or B. If it croaks A when on lilypad i $(1 \leq i \leq n-1)$, it is thrown to lilypad $i-1$ with probability $\frac{i}{n}$ and is thrown to lilypad $i+1$ with probability $\frac{n-i}{n}$. If it croaks B when on lilypad i $(1 \leq i \leq n-1)$, it is thrown to one of the lilypads $0, \ldots, i-1, i+1, \ldots n$ with uniform probability $\frac{1}{n}$. A snake, located on lilypad 0, will eat the frog if the frog lands on lilypad 0. The frog can escape the pond (and hence, escape the snake!) if it lands on lilypad n. What should the frog croak when on each of the lilypads $1, 2, \ldots, n-1$ in order to maximize the probability of escaping the pond (i.e., reaching lilypad n before reaching

lilypad 0)? Although there are more than one ways of solving this problem, we'd like to solve it by modeling it as an MDP and identify its Optimal Action Value function Q^*.

(a) Express with clear mathematical notation the state space, action space, transition probabilities and rewards of an MDP so that the above frog-escape problem can be solved by arriving at the Optimal Action Value Function Q^* of this MDP (20 pts).

(b) Write working Python code that models this MDP and solves the for the Optimal Action Value Function Q^* of this MDP. For $n = 3$, $n = 10$ and $n = 25$, plot a graph of $Q^*(s, a')$ as a function of the states of this MDP for each action a'. For every training episode, fix a state value function and use it to simulate total cumulative rewards of the frog-escape MDP with 10 random choices of the starting lilypad i. Then for every training episode, plot the average of total cumulative rewards over the 10 simulation episodes (20 pts).

4. **Problem 4 (30 pts).** Download depth 5 data for your favorite stock from LOBSTER. We are interested in formulating an MDP with an ultimate goal to be able to find an optimal execution strategy of a large client buy order of size X using a model-free RL method (but we will not code an RL method for it in this homework assignment).

   Assume that we will be operating in market replay regime – the transition probabilities of an MDP are implied by the historical data that you downloaded – with an assumption that there is no market impact and that our execution agent will be taking a decision every 10 seconds. Express with clear mathematical notation the proposed state space, action space and rewards of an MDP that needs to be formulated to find an optimal execution strategy of a large client buy order of size X. For the purpose of this exercise, please make sure that the state space consists of at least 10 state variables! Papers that we considered in class (listed on the syllabus and in the lecture notes) can help you with these formulations.

   Now write Python function *define_states* to implement your proposed market state variables as a function of time. Plot every state variable as a function of time. Also, please plot the histogram distribution of each of the observed market state variables. Which of the state variables do you expect to be 'useful' for learning an optimal execution agent? Why, or why not?