IE-456/556 & EEE-448/548
Reinforcement Learning and Dynamic Programming
Homework Assignment 3 – Training RoboDry – Due July 22 23:59

RoboDry is an RL robot that needs to learn to take freshly painted cups from a painting machine, dry them, and place the dried cups on a shelf. We assign a $+10$ reward for storing dry cups on the shelf and a $+5$ reward for storing wet cups. The robot can attempt to dry any held cup, but there is a small chance (0.1) that the cup will break during the process, incurring a high cost (-20). If the robot decides to grab a new cup from the painting machine while already having a cup, it will drop and break the held cup (-20). The robot must now try to determine which policy to use for this task.

1. **Modeling**: Model a cup drying process as an MDP with the states of s_0: "Not Holding Cups", s_1: "Holding A Wet Cup", s_2: "Holding A Dry Cup", s_3: "Job Done", and the actions of a_0: "Grab", a_1: "Dry", and a_2: "Store". Make a transition graph for this problem, or write the reward and transition functions as a matrix. (*Hint*: We can create an episodic MDP in our RL modeling by designing our model to have at least one terminal state for all actions. In this example we can consider s_3 as the terminal state. The agent will go through this process over and over to learn the optimal pattern of performing the job successfully.)

2. **Learning**: Search for an optimal policy for this MDP using the following algorithms with the discount of $\gamma = 1$.

   (a) Policy Iteration (with the initial policy of a_0 for all states)

   (b) Value Iteration (with the initial value of zero for all states)

   (c) First-Visit Monte Carlo (with the initial value of zero for all state-action pairs)

   (d) SARSA (with the initial value of zero for all state-action pairs)

   (e) Q-Learning (with the initial value of zero for all state-action pairs)

   Note 1: You can perform one of the followings in the search for the optimal solution:

   (1) Find the optimal policy with each approach and compare their results and your observation of the convergence rate.

   (2) Perform three steps of each algorithm on paper by considering the following paths:

   - For Monte-Carlo:
     Step 1: $(s_2, a_1) \rightarrow (s_2, a_0) \rightarrow (s_1, a_0) \rightarrow (s_1, a_2) \rightarrow s_3$
     Step 2: $(s_1, a_0) \rightarrow (s_1, a_2) \rightarrow s_3$
     Step 3: $(s_0, a_0) \rightarrow (s_1, a_2) \rightarrow s_3$
   - For TD control:
     Step 1: $(s_2, a_2) \rightarrow s_3$
     Step 2: $(s_1, a_1) \rightarrow (s_2, a_2) \rightarrow s_3$
     Step 3: $(s_2, a_0) \rightarrow (s_1, a_2) \rightarrow s_3$

Note 2: While using ϵ-greedy exploration in an on-policy learning algorithm (especially for on-policy MC), you may need to define a limit to the number of actions you take before the episode terminates.

3. Which of the above methods should I use to find an optimal policy if I didn't know the probability of breaking cups during drying?

4. What is the effect of increasing ϵ while using an ϵ-greedy algorithm?