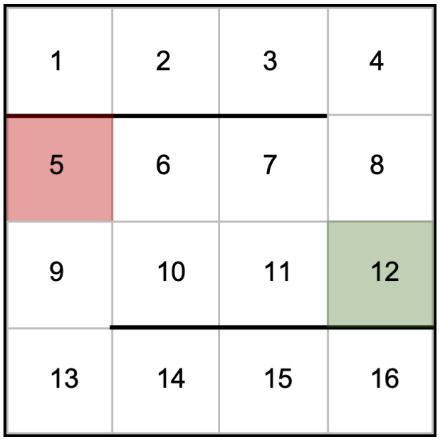
CP8319/CPS824: Reinforcement Learning (Winter 2021)
Assignment 1
Due: Monday February 18, 2021 (Midnight 11:59PM or 23:59)
Instructor: Nariman Farsad

Please Read Carefully:

- Submit your assignments on D2L before the deadline. The submission will be closed after the deadline.
- Show your work and write your solutions legibly. You can use applications such as Word or Latex to type up your solutions for the written portion of the assignment or use applications such as CamScanner to create a PDF file from your handwritten solutions. For coding problems you need to submit your python source files or Jupyter notebook, and have the main plots and answers to questions in the written portion.
- Do the assignment on your your own, i.e., do not copy. If two assignments are found to be copies of each other, they BOTH receive 0.
- These questions require thought, but do not require long answers. Please be as concise as possible.
- If you have a question about this homework, we encourage you to post your question on our D2L discussion board.
- Students in CP8319 are required to answer all questions in the assignment. CPS824 students are not required to answer the questions marked for CP8319 students. These questions will not be graded for CPS824 students and they can just attempt them for practice.
- For instructions on setting up the python environment for the assignment read the "README.md" file.

1. (35 points) **MDP Grid World**

Consider the following grid environment. Starting from any unshaded square, you can move up, down, left, or right. Actions are deterministic and always succeed (e.g. going left from state 16 goes to state 15) unless they will cause the agent to run into a wall. The thicker edges indicate walls, and attempting to move in the direction of a wall results in staying in the same square (e.g. going in any direction other than left from state 16 stays in 16). Taking any action from the green target square (no. 12) earns a reward of r_g (so $r(12, a) = r_g \ \forall a$) and ends the episode . Taking any action from the red square of death (no. 5) earns a reward of r_r (so $r(5, a) = r_r \ \forall a$) and ends the episode. Otherwise, from every other square, taking any action is associated with a reward $r_s \in \{-1, 0, +1\}$ (even if the action results in the agent staying in the same square). Assume the discount factor $\gamma = 1$, $r_g = +5$, and $r_r = -5$ unless otherwise specified.



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

(a) (5pts) Define the value of r_s that would cause the optimal policy to return the shortest path to the green target square (no. 12). Using this r_s, find the optimal value for each square.

(b) (5pts) Lets refer to the value function derived in (a) as $V_{old}^{\pi_g}$ and the policy as π_g. Suppose we are now in a new gridworld where all the rewards (r_s, r_g, and r_r) have +2 added to them. Consider still following π_g of the original gridworld, what will the new values $V_{new}^{\pi_g}$ be in this second gridworld?

(c) (10pts) **(CP8319 ONLY QUESTION)** Consider a general MDP with rewards, and transitions. Consider a discount factor of γ. For this case assume that the horizon is infinite (so there is no termination). A policy π in this MDP induces a value function V^{π} (lets refer to this as V_{old}^{π}). Now suppose we have a new MDP where the only difference is that all rewards have a constant c added to them. Can you come up with an expression for the new value function V_{new}^{π} induced by π in this second MDP in terms of V_{old}^{π}, c, and γ?

(d) (5pts) Lets go back to our gridworld from (a) with the default values for r_g, r_r, γ and with the value you specified for r_s. Suppose we now derived a second gridworld by adding a constant c to all rewards (r_s, r_g, and r_r) such that $r_s = +2$. How does the optimal policy change (Just give a one or two sentence description)? What do the values of the unshaded squares become?

(e) (5pts) Now take the second gridworld from part (d) and change γ such that $0 < \gamma < 1$. Can the optimal policy change and does it depend on your choice of gamma? (A brief description is sufficient, no formal proof or mathematical analysis required).

(f) (5pts) Lets go back to our gridworld from (a) with the default values for r_g, r_r, γ and with the value you specified for r_s. In this gridworld, our optimal policy from any unshaded square never terminates in the red square. Now suppose r_s can take on any real, non-infinite value and is not restricted to $\{+1, 0, -1\}$ anymore. Give a value of r_s such that there are unshaded squares starting from which following the optimal policy results in termination in the red square.

2. (25 points) **Value Iteration and Policy Iteration for MDP**

In this question, you will implement value iteration and policy iteration for the Frozen Lake environment from OpenAI Gym (click on OpenAIGym for details). We have provided custom versions of this environment in the starter code.

(a) **(coding)** (10 pts) Read through `vi_and_pi.py` and implement the functions: `policy_evaluation`, `policy_improvement` and `policy_iteration`. The stopping tolerance (defined as $\max_s |V_{old}(s) - V_{new}(s)|$) is tol $= 10^{-3}$. Use $\gamma = 0.9$. Return the optimal value function and the optimal policy.

(b) **(coding)** (10 pts) Implement `value_iteration` in `vi_and_pi.py`. The stopping tolerance is tol $= 10^{-3}$. Use $\gamma = 0.9$. Return the optimal value function and the optimal policy.

(c) **(written)** (5 pts) Run both methods on the Deterministic-4x4-FrozenLake-v0 and Stochastic-4x4-FrozenLake-v0 environments. In the second environment, the dynamics of the world are stochastic. How does stochasticity affect the number of iterations required, and the resulting policy?