Reinforcement Learning
ECE 750 T40, Fall 2024, Assignment 1
Due: **Friday October 11**, 11:59pm

Collaboration/Groups: You may do your work individually or in pairs. To submit as a pair, you can choose your group partner in crowdmark and put your names together. You can also collaborate with other classmates on the right general approaches to use for questions, but your submitted worked must be only from members of your group. Note that looking up references online is allowed, but be sure to list the ones you use. Using a generative AI model like ChatGPT doesn't have sources, or validation, so that is not a valid reference. You also should not try to use such tools to generate your text or answers, first because you won't know if it's correct or not without checking other sources anyways, and second, because it isn't allowed.

Submission: Submit to the UWaterloo Crowdmark site using the link you received via email. Follow the instructions in Crowdmark for each question.

For each question, you should submit your answer as a separate image or pdf document on Crowdmark. To make mathematical notation it can be typed up and submitted as a pdf (for the math look at using LaTeX or MathJax languages (try Overleaf (https://www.overleaf.com/edu/uwaterloo) which is the best LaTeX editing tool, and UWaterloo has a site license). Most word-processors (MS Word, Google Docs) take input in one of these formats, including the piazza discussion group. You can also write out the solution by hand on paper or a tablet and submit images of those for each answer, *but be as neat as possible* in this case!

Some Useful LaTeX Examples (not all needed for this assignment):

- $\mathbb{E}[y|x,\theta]$: `\mathbb{E}[x|x,\theta]`
- $\sum_{x\in X}$: `\sum_{x\in X}`
- $P(X=x)$: `P(X=x)`
- $\mathbf{X}\sim\mathcal{N}(\mu,\sigma)$: `\mathbf{X}\sim \mathcal{N}(\mu, \sigma)`
- $x\leftarrow\mathbf{X}$: `X\sim \mathcal{N}(\mu, \sigma)`
- $\frac{\pi}{42}$: `\frac{\pi}{42}`
- $\acute{\alpha}\beta\delta\Delta$: `\Acute{\alpha}\beta\delta\Delta`
- $\bar{X}\tilde{\gamma}\hat{\Gamma}\omega^\beta$: `\bar{X} \tilde{\gamma} \hat{\Gamma}\omega^\beta`
- $O(\log n)\ldots\Omega(n^2) - \arg\max_a \nabla f(x,a)$
- $e=\sum_{n=0}^\infty \frac{1}{n!}\approx 2.718$: `e = \sum_{n=0}^\infty \frac{1}{n!} \approx 2.718`

MDPs

- Q3: **Policy Evaluation:** Figure 1 (left) (from section 3.14 of the Sutton textbook) shows a rectangular grid world representation of a simple finite MDP. The cells of the grid correspond to the states of the environment. We can give each state a name starting from s_0 in the top left corner, increasing to the right and wrapping at each row so that the last row goes from s_{20} to s_{24}. At each cell, four actions are possible: north, south, east, and west. Choosing an action which deterministically cause the agent to move one cell in the respective direction on the grid. Actions that would take the agent off the grid leave its location unchanged, but also result in a reward of -1. Other actions result in a reward of 0, except those that move the agent out of the special states A and B. From state A, all four actions yield a reward of +10 and take the agent to A'. From state B, all actions yield a reward of +5 and take the agent to B'.

  Given a discount factor $\gamma = 0.9$ and an initial policy with equal probabilities, the policy evaluation procedure resulted in the value function shown in Figure 1 (right).

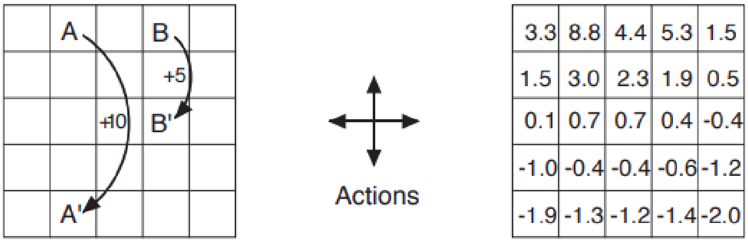
  First, you need to specify the general form of the Bellman equation and then expand it out in symbolic form for the given state id (eg. s_2, $p(s10|s5, S), r(s_5, E), V(s_0, E), \pi(N|s_2)$. etc.). Of course, having an equiprobably policy for all actions regardless of state isn't a very realistic policy. A general policy could have a different distribution for each (state,action) pair. So, next you will calculate the updated value for two particular states if the policy changed as specified in each sub-question. Note, you can confirm your calculation approach, if you use the original policy with a 25% chance of going in any direction you should arrive at the same value that is already in the table (up to two decimal places or so). (**Hint:** Try to understand what happens when the agent takes an action that brings it to hit one of the walls.)

  (a) **(3 points)** Write down the general Bellman equation for $v_\pi(s)$ for this domain and then expand it out explicitly for states s_5 and s_{12} using the state ids so that there are no summation ($\sum$) terms in the equation.

  (b) **(5 points)** Show the full calculation for state s_5 using the updated policy using the provided value function. In the updated policy $\pi(s_5, [N, S, E, W]) = [0.2, 0.2, 0.5, 0.1]$.

  (c) **(5 points)** Show the full calculation for state s_{12} using the updated policy using the provided value function. In the updated policy $\pi(s_5, [N, S, E, W]) = [0.6, 0.02, 0.08, 0.3]$.

  (d) **(3 points)** Are the policies in parts (b) and (c) better or worse than the original equiprobable policy?



3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

Figure 1: Gridworld example: exceptional reward dynamics (left) and state-value function for the equiprobable random policy (right)

1 Part 2

See Assignment 2 gitlab page for programming part of the assignment.