CMPSCI 687 Homework 1

Due September 20, 2018, 11pm Eastern Time

Instructions: This homework assignment consists of a written portion and a programming portion. Collaboration is not allowed on any part of this assignment. Submissions must be typed (hand written and scanned submissions will not be accepted). You must use LaTeX. The assignment should be submitted on Moodle as a .zip (.gz, .tar.gz, etc.) file containing your answers in a .pdf file and a folder with your source code. Include with your source code instructions for how to run your code. You may not use any reinforcement learning or machine learning specific libraries in your code (you may use libraries like C++ Eigen and numpy though). If you are unsure whether you can use a library, ask on Piazza. If you submit by September 25, you will not lose any credit. The automated system will not accept assignments after 11:55pm on September 25.

Part One: Written (65 Points Total)

1. (Your grade will be a zero on this assignment if this question is not answered correctly) Read the class syllabus carefully, including the academic honesty policy. To affirm that you have read the syllabus, type your name as the answer to this problem.
   Ans: Bochen Xu

2. (15 Points) Given an MDP $M = (\mathcal{S}, \mathcal{A}, P, d_R, d_0, \gamma)$ and a fixed policy, π, the probability that the action at time $t = 0$ is $a \in \mathcal{A}$ is:

$$\Pr(A_0 = a) = \sum_{s \in \mathcal{S}} d_0(s)\pi(s, a). \tag{1}$$

   Write similar expressions (using only $\mathcal{S}, \mathcal{A}, P, R, d_0, \gamma$, and π) for the following:

   - The probability that the state at time $t = 3$ is either $s \in \mathcal{S}$ or $s' \in \mathcal{S}$.
     Ans: $Pr(S_3 = s || S_3 = s')$
     $= \sum_{s_0, s_1, s_2 \in \mathcal{S}} d_0(s_0)\pi(s_0, a_0)P(s_0, a_0, s_1)\pi(s_1, a_1)P(s_1, a_1, s_2)\pi(s_2, a_2)(P(s_2, a_2, s) + P(s_2, a_2, s'))$
   - The probability that the action at time $t = 16$ is $a' \in \mathcal{A}$ given that the action at time $t = 15$ is $a \in \mathcal{A}$ and the state at time $t = 14$ is $s \in \mathcal{S}$.
     Ans: $Pr(A_{16} = a' | A_{15} = a, S_{14} = s)$
     $= \pi(s, a_{14})P(s, a_{14}, s_{15})\pi(s_{15}, a)P(s_{15}, a, s_{16})\pi(s_16, a')$
   - The expected reward at time $t = 6$ given that the action at time $t = 3$ is $a \in \mathcal{A}$, and the state at time $t = 5$ is $s \in \mathcal{S}$.
   - The probability that the initial state was $s \in \mathcal{S}$ given that the state at time $t = 1$ is $s' \in \mathcal{S}$.

- The probability that the action at time $t = 5$ is $a \in \mathcal{A}$ given that the initial state is $s \in \mathcal{S}$, the state at time $t = 5$ is $s' \in \mathcal{S}$, and the action at time $t = 6$ is $a' \in \mathcal{A}$.
  Ans: $Pr(A_5 = a | S_0 = s, S_5 = s', A_6 = a') = \pi(s', a) \sum_{S_6 \in \mathcal{S}} P(s', a, s_6)\pi(s_6, a')$

3. (3 Points) In 687-Gridworld, if we changed how rewards are generated so that hitting a wall (i.e., when the agent would enter an obstacle state, and is placed back where it started) results in a reward of -1, then what is $\mathbf{E}[R_t | S_t = 17, A_t = \text{AL}, S_{t+1} = 17]$?
   Ans: $-8/9$

4. (2 Points) How many stochastic policies are there for an MDP with $|\mathcal{S}| < \infty$ and $|\mathcal{A}| < \infty$? (You may write your answer in terms of $|\mathcal{S}|$ and $|\mathcal{A}|$).
   Ans: ∞

5. (5 Points) Create an MDP (which may not have finite state or action sets) that does *not* have an optimal policy. The rewards for your MDP must be bounded.
   Ans: $\mathcal{S} = \{S_0, S_1\}$
   $\mathcal{A} = \{a_i, i \in \mathcal{N}_{>0}\}$
   $P(S_1, a, S_0) = 0, P(S_1, a, S_1) = 1$ for all $a \in \mathcal{A}$
   $P(S_0, a_i, S_0) = 1 - P(S_0, a_i, S_1), P(S_0, a_i, S_1) = \sum_{m=1}^{i} \prod_{n=1}^{m} (0.1)^n$
   $d_R(S_0, a, S_1) = 1$ for all $a \in \mathcal{A}$, 0 elsewhere
   $d_0(S_0) = 1, d_0(S_1) = 0, \gamma = 1$ for any policy, we can always replace an action a_i in it with a_{i+1}, increase the chance to switch from S_0 to S_1, hence the expected reward, so there is no optimal policy.

6. (3 Points) Read about the Pendulum domain, described in Section 5.1 of this paper (Reinforcement Learning in Continuous Time and Space by Kenji Doya). Consider a variant where the initial state has the pendulum hanging down with zero angular velocity always (a deterministic initial state where the pendulum is hanging straight down with no velocity) and a variant where the initial angle is chosen uniformly randomly in $[-\pi, \pi]$ and the initial velocity is zero. Which variant do you expect an agent to require more episodes to solve? Why?
   Ans: The one with randomly selected initial state needs more episode, because it keeps changing.

7. (1 Point) How many episodes do you expect an agent should need in order to find near-optimal policies for the gridworld and pendulum domains?
   Ans : 200

8. (5 Points) Select a problem that we have not talked about in class, where the agent does not fully observe the state. Describe how this problem can be formulated as an MDP by specifying $(\mathcal{S}, \mathcal{A}, P, [d_r \text{ or } R], d_0, \gamma)$ (your specifications of these terms may use English rather than math, but be

precise).

9. (5 Points) Create an MDP for which there exist at least two optimal policies that have different variance of their returns. Describe the two optimal policies and derive the expected value and variance of their returns.
   Ans: Let there be two states, S_0 the initial state and S_1 the terminal state. There are two actions a, a', both takes you from S_0 to S_1. a gives you 1 or -1 reward with 50% chance each, while a' gives you 10 or -10 with 50% chance each. $\gamma = 1$.
   The two optimal policies is just take the two actions, both with expected reward 0. The policy of taking a has variance 1, while the policy of taking a' has variance 100.

10. (2 Points) Create an MDP that always terminates, but which has no terminal states.
    Ans: Let us have state S_0, S_1, S_∞ and action a, a takes you from S_0 to S_1 or from S_1 to S_0 with probability 0.1, and takes you to S_∞ when you are at S_0 ar S_1 with probability 0.9. The initial state is S_0, reward is always 0, and $\gamma = 1$. There is no terminal state, but the only policy is keep taking action a, and the probability of reaching S_∞ is $0.9 + 0.09 + 0.009 + ... = 1$.

11. (2 Points) The sequence of states that results from running a fixed policy is a *Markov chain*. A state in a *Markov chain* has *period* k if every return to the state must occur in multiples of k time steps. More formally,

    $$k = \gcd\{t > 0 : \Pr(S_t = s | S_0 = s) > 0\}.$$

    Create an MDP and a policy that result in a state having a period of 3.
    Ans: $\mathcal{S} = S_0, S_1, S_2, S_3, S_\infty$ There are two actions: a takes you from S_0 to S_1, S_1 to S_2, S_2 to S_3, S_3 to S_1; a' always takes you to S_∞, reward is always 0, initial state is S_0, and $\gamma = 1$. A policy that always takes action a will make the state S_3 has period 3.

12. (5 Points) A Markov chain is *irreducible* if it is possible to get to any state from any state. An MDP is *irreducible* if the Markov chain associated with every deterministic policy is irredicible. A Markov chain is *aperiodic* if the period of every state is $k = 1$. The state of a Markov chain is *positive recurrent* if the expected time until the state recurs is finite. A Markov chain is *positive recurrent* if all states are positive recurrent. A Markov chain is *ergodic* if it is *aperiodic* and *positive recurrent*. An MDP is *ergodic* if the Markov chain associated with every deterministic policy is ergodic. Create an MDP that is ergodic, but *not* irreducible.

13. (3 Points) Create an MDP that is not ergodic.
    Ans: I need more thinking on this...

For the following questions, Let's think about a game of snake and ladder, where you have 100 blocks on the board, numbered from 1 to 100 (100 states and S_∞), and you move forward x steps when you roll x with a 6-side die(the only action is "roll"). if you are in state with a snake head, you fall to the snake tail in the next time step no matter what you do. Simillarly, if you are at the bottom of a ladder, you go to the top of the ladder in the next step no matter what you do. The game ends when you reach block 100 (S_100 is the terminal state).

14. (2 Points) Describe a real-world problem and how it can be reasonably modeled as an MDP where R_t is *not* a deterministic function of S_t, A_t, and S_{t+1}. Ans: say in this game, for each time step, I roll a 6-side die and give you that many chocolate cake. We can set R_t to be the number of cakes you get, which is completely random.

15. (2 Points) Describe a real-world problem and how it can be reasonably modeled as an MDP where R_t is a deterministic function of S_t.
Ans: say in this game, for each time step, I give you x chocolate cakes after you leave block x. We can set R_t to be the number of cakes you get, which completely depends on S_t.

16. (2 Points) Describe a real-world problem and how it can be reasonably modeled as an MDP where R_t is a deterministic function of S_{t+1}.
Ans: say in this game, for each time step, I give you x chocolate cakes after you enter block x. We can set R_t to be the number of cakes you get, which completely depends on S_{t+1}.

17. (2 Points) Describe a real-world problem and how it can be reasonably modeled as an MDP where the reward function, R, would be known.
Ans: say in this game, I give you 1 chocolate cake when you reach black 100, R is known to you.

18. (2 Points) Describe a real-world problem and how it can be reasonably modeled as an MDP where the reward function, R, would *not* be known.
Ans: say in this game, for each time step, I give you x chocolate cakes based on a lottery program output, but you do not know how the program works. We can set R_t to be the number of cakes you get, which is unknown to you.

19. (2 Points) Describe a real-world problem and how it can be reasonably modeled as an MDP where the transition function, P, would be known.

Ans: in this game, the rule is simple, you roll the die and move that many steps, unless you slide down the snake or climb up the ladder, so you know P.

20. (2 Points) Describe a real-world problem and how it can be reasonably modeled as an MDP where the transition function, P, would *not* be known.
Ans: I replace the die with a "magic die" program which you can input 1 to 6 before it simulates a die roll, but you find out that it is broken: it is not giving you the number you just input. Is there a bug? You are confused and now P is unknown to you.

Part Two: Programming (25 Points Total)

Implement the 687-Gridworld domain described in class and in the class notes. Have the agent select actions uniformly randomly.

- (5 Points) Have the agent uniformly randomly select actions. Run 10,000 episodes. Report the mean, standard deviation, maximum, and minimum of the observed discounted returns.
Ans:
mean:-0.633 std:2.397 max:4.305 min:-43.752

- (5 Points) Find an optimal policy (you may do this any way you choose, including by reasoning through the problem yourself). Report the optimal policy here. Comment on whether it is unique.
Ans:
1.AR 2.AR 3.AR 4.AR 5.AD
6.AR 7.AR 8.AR 9.AR 10.AD
11.AU 12.AU [ ] 13.AD 14.AD
15.AU 16.AU [ ] 17.AD 18.AD
19.AU 20.AL 21.AR 22.AR

- (10 Points) Run the optimal policy that you found in the previous question for 10,000 episodes. Report the mean, standard deviation, maximum, and minimum of the observed discounted returns.
Ans:
mean:3.924 std:0.682 max:4.783 min:1.216

- (5 Points) Using simulations, empirically estimate the probability that $S_{19} = 21$ (the state with water) given that $S_8 = 18$ (the state above the goal) when running the uniform random policy. Describe how you estimated this quantity (there is *not* a typo in this problem, nor an oversight).
Ans: place the robot at state 18 and see if it gets to state 21 after 11 time

steps. Do this for 100000 times and calculate the ratio to estimate the probability. It is about 0.063.