COMPSCI 687 - Fall 2022 Midterm

Nirupan Ananthamurugan

TOTAL POINTS

103 / 110

QUESTION 1

1 Question 1 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 2

2 Question 2 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 3

3 Question 3 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 4

4 Question 4 **5 / 5**

✓ **- 0 pts** *Correct*

QUESTION 5

5 Question 5 **10 / 10**

✓ **- 0 pts** *Correct*

QUESTION 6

6 Question 6 **10 / 10**

✓ **- 0 pts** *Correct*

QUESTION 7

7 Question 7 **7 / 12**

✓ **- 5 pts** *Arguing that the distance between points _can_ decrease does not mean that it _must_ decrease for the inequality to hold. If lambda were, e.g., equal to 1, even if the distance could decrease, the inequality would still hold if kept the same, so this part of the argument is missing.*

QUESTION 8

8 Question 8 **16 / 16**

✓ **- 0 pts** *Correct*

QUESTION 9

9 Question 9 **8 / 10**

- 2 *Point adjustment*

The agent would start, on average, in all states, since the algorithm is MC with exploring starts.

QUESTION 10

Question 10 22 pts

10.1 Q10.a **8 / 8**

✓ **- 0 pts** *Correct*

10.2 Q10b **12 / 12**

✓ **- 0 pts** *Correct*

10.3 Q10c **2 / 2**

✓ **- 0 pts** *Correct*

QUESTION 11

11 Extra 1 **3 / 3**

✓ **- 0 pts** *Correct*

QUESTION 12

12 Extra 2 **7 / 7**

✓ **- 0 pts** *Correct*

116/140

Name: NIRVPAN ANANTHAMURUGAN Student ID: 33591075

CMPSCI 687 Midterm - Fall 2022

(1 additional sheet)

Instructions: This midterm is **closed** notes. Do not use any notes or electronic devices. This exam will start at 7 pm and you have until 9 pm to complete it.

1. (5 points) Explain, in English, how the Policy Iteration algorithm works and what is its objective.

The policy iteration algorithm works in 2 steps:

1. Policy Evaluation: the policy is evaluated i.e. we compute $v^{\pi}(s)$ for all $s \in S$
2. Policy Improvement: we improve the policy by choosing the action that maximizes $v^{\pi}(s)$ computed above.

It's objective is to find the optimal policy (when $\pi_{i+1} = \pi_i$).

2. (5 points) Name the three classes of algorithms, studied in class, that may be used to perform Policy Evaluation.

1. Dynamic Programming
2. Monte Carlo Methods
3. Temporal Difference Method

3. (5 points) Write the Bellman Optimality Equation for the state-value function, v.

~~$v^{\pi^*}(s) = \sum \pi^*(s,a) \sum p(s,a,s') \left(R(s,a) + \gamma v^{\pi^*}(s)\right)$~~

$$v^*(s) = \max_a \sum_{s'} p(s,a,s') \left(R(s,a) + \gamma v^*(s')\right)$$

4. (5 points) Describe (using math) how to compute π^* assuming that you *do* know q^* and v^*, but that you *do not* know p and R.

Since π^* is the optimal policy, we expect the policy to choose the action at every state that maximizes the q value (i.e. action that has max. q-value).

$$\pi^*(s) = \arg\max_a q^*(s,a)$$

5. (10 points) Consider an MDP with a single nonterminal state, s. Assume that $\gamma = 1$. Suppose you observe one episode that lasts 10 steps, where the agent repeatedly transitions from s back to s until the end of the episode. The reward received by the agent at each step is +1, thus producing a return of 10. What are the First-Visit and Every-Visit Monte Carlo estimators of $v(s)$?

First Visit: $G_1 = 1 + \gamma 1 + \dots + \gamma^{\wedge} 1 = 10$

$G_1 = 10 \Rightarrow \boxed{v(s) = 10}$

Every Visit: $G = (G_1 + G_2 + G_3 + \dots + G_{10})/10$

$= (10 + 9 + 8 + \dots + 1)/10$

$= 55/10$ $\Rightarrow \boxed{v(s) = 5.5}$

6. (10 points) Write an expression for the probability that the action at time $t = 14$ is a_{14} given that the state at time $t = 13$ is s_{13} and the action at time $t = 12$ is a_{12}. Your solution <u>needs</u> to be derived from "first principles": you should repeatedly apply definitions and properties of probability distributions such as the ones discussed in the first homework, as well as the Markov Property (when appropriate), and then replace the relevant quantities with their corresponding definitions in RL (e.g., you can substitute $\Pr(A_0 = a | S_0 = s)$ with $\pi(s, a)$). Show all steps of your derivation. When writing your final answer, reorganize your terms and summations in "temporal" order, like in the first homework.

$$\Pr(A_{14} = a_{14} \mid S_{13} = s_{13}, A_{12} = a_{12})$$

$$= \Pr(A_{14} = a_{14} \mid S_{13} = s_{13}) \quad \text{(by Markov Property)}$$

$$= \sum_{a_{13}} \Pr(A_{13} = a_{13} \mid S_{13} = s_{13}) \Pr(A_{14} = a_{14} \mid S_{13} = s_{13}, A_{13} = a_{13})$$

$$= \sum_{a_{13}} \pi(s_{13}, a_{13}) \sum_{s_{14}} \Pr(S_{14} = s_{14} \mid S_{13} = s_{13}, A_{13} = a_{13}) \Pr(A_{14} = a_{14} \mid S_{14} = s_{14}) \quad \text{(again by Markov Property)}$$

$$= \boxed{\sum_{a_{13}} \pi(s_{13}, a_{13}) \sum_{s_{14}} p(s_{13}, a_{13}, s_{14}) \pi(s_{14}, a_{14})}$$

7. (12 points) Let f be an operator that is a contraction mapping. That is, there exists some $\lambda \in [0, 1)$ such that $d(f(x), f(y)) \leq \lambda d(x, y)$, for any $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, and where d is a distance function. In class, we formally explained why a contraction mapping has a unique fixed point. Prove that this is indeed the case; that is, that a contraction mapping cannot have two fixed points.

To prove this, let us assume that there are 2 fixed points, let's call them x^* and y^*. (Note: since the points are different, $d(x^*, y^*) \neq 0$)

If we apply the contraction mapping f on these 2 points and we get $f(x^*)$ and $f(y^*)$

then, $d(f(x^*), f(y^*)) \leq \lambda\, d(x^*, y^*)$

which means the points can further be converged (brought closer).

Hence, we can prove by contradiction that a contraction mapping CANNOT have two fixed points.

8. (16 points) Prove that multiplying all rewards (of a finite MDP with bounded rewards and $\gamma < 1$) by a constant positive scalar does not change which policies are optimal. **Hint:** *(i)* start from the definition of the expected return of a policy and consider what would happen if we were to multiply all rewards by a positive scalar; *(ii)* recall that optimal policies are policies that achieve the highest possible return; *(iii)* combine these findings/observations and construct a formal argument for why transforming the reward function in this way would not change which policies are optimal.

$\mathbb{E}[G_t^\pi] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t\right]$

Assume π is the optimal policy.

If we multiply each R_t with a scalar, let's say α.

$\Rightarrow J(\pi) = \gamma^0 R_0 + \gamma^1 R_1 + \gamma^2 R_2 + \dots$

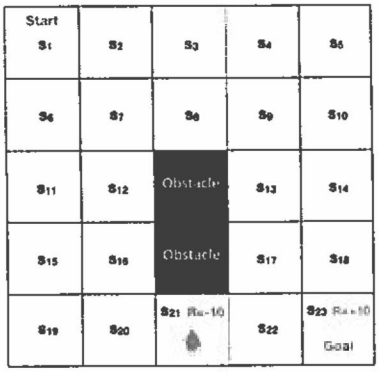
$\Rightarrow J(\pi^*) = \alpha\gamma^0 R_0 + \alpha\gamma^1 R_1 + \alpha\gamma^2 R_2 + \dots$

$= \alpha(\gamma^0 R_0 + \gamma^1 R_1 + \gamma^2 R_2 + \dots)$

i.e. for the same policy we got the new expected return to be α times the old expected return.

X Check additional sheets X

9. (10 points) Recall the 687-Gridworld domain described in class:



Actions:
attempt_up
attempt_down
attempt_left
attempt_right

When the agent attempts to move in a direction:
The agent succeeds, $p = 0.8$
The agent veers 90° right, $p = 0.05$
The agent veers 90° left, $p = 0.05$
The agent stays in place, $p = 0.1$

If the agent would ever hit a wall, it stays in its current position.

All unspecified rewards are zero. All specified rewards are for entering the state

Let π be a policy that always executes the action Attempt_Left. We wish to compute v^π using the *Monte Carlo with Exploring Starts* algorithm. What could go wrong if you were to use this algorithm to estimate the value function of this policy in the 687-Gridworld domain? What is one of the main assumptions made by Monte Carlo algorithms (so that they can be applied to a given problem) that is not satisfied in this case?

Let us assume $(S_{18}, \text{Attempt\_Left})$ is one of the many exploring starts (s,a). And let us say that when we run this algorithm the agent transitions into S_{23} (5% chance). Then we would add the return 10 to our list to compute averages.

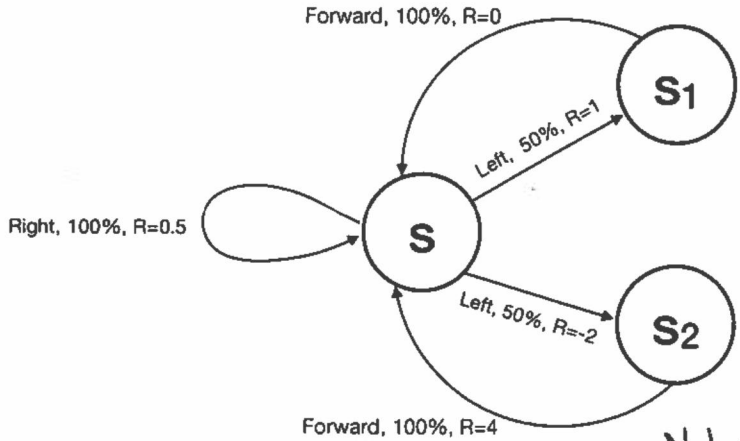
i.e. we get positive rewards in our estimate / evaluation.

However, since the start state is only S_1, if we always executed AttemptLeft, then we would never leave the first column

X Continued on additional sheets X

10. (22 points) Consider the MDP shown below. The only decision to be made is when the agent is in state s, when it can choose to execute either action Left or action Right. Edges between any two states, A and B, are annotated with a possible action, a_i, the probability $p(A, a_i, B)$, and the reward ~~$R(A, a_i)$~~, respectively. There are exactly two deterministic policies: π_{left} (which always executes action Left in state s) and π_{right} (which always executes action Right in state s). Assume that $\gamma = 0.5$.

$R(A, a_i, B)$



Note: Substitute all formulae $R(s,a)$ with $R(s,a,s')$

(Question 10a. 8 points) Assume we wish to compute $v^{\pi_{\text{left}}}$ using the Dynamic Programming Policy Evaluation algorithm. Assume that at the i-th iteration of this algorithm, the value function estimate, v_i, is $v_i(s) = 2$, $v_i(s_1) = 1$, and $v_i(s_2) = 4$. Simulate the execution of one iteration of the Policy Evaluation algorithm (showing your work) to obtain a new estimate of the value function, v_{i+1}. Based on this result, argue/explain whether the algorithm has converged to the true value function, $v^{\pi_{\text{left}}}$.

$$V_{i+1}(s) = \sum_a \pi(s,a) \sum_{s'} p(s,a,s') \left(R(s,a) + \gamma V_i(s')\right)$$

For s,

$$V_{i+1}(s) = 1\left(0.5\left(1 + 0.5 * 1\right) + 0.5\left(-2 + 0.5 * 4\right)\right)$$

$$= 0.5\,(1.5) + 0.5\,(0)$$

$$\Rightarrow \boxed{V_{i+1}(s) = 0.75}$$

For s_1,

$$V_{i+1}(s_1) = 1\left(1\left(0 + 0.5 * 2\right)\right)$$

$$\Rightarrow \boxed{V_{i+1}(s_1) = 1}$$

For s_2,

$$V_{i+1}(s_2) = 1\left(1\left(4 + 0.5 * 2\right)\right)$$

$$\Rightarrow \boxed{V_{i+1}(s_2) = 5}$$

Since $V_{i+1}(s) \neq V_i(s)$ for all $s \in S$, the algorithm has NOT YET converged to the true $V^{\pi_{\text{left}}}$.

(Question 10b. 12 points) Write the Bellman Equations for $v^{\pi_{\text{left}}}$ and $v^{\pi_{\text{right}}}$, for all states of the MDP, using the transition probabilities and rewards specified in the figure above. Then, use these equations to numerically compute $v^{\pi_{\text{left}}}(s)$ (the value of state s under policy π_{left}) and $v^{\pi_{\text{right}}}(s)$ (the value of s under policy π_{right}). Show your work.

$\pi_{\text{left}} = \pi_L$ & $\pi_{\text{right}} = \pi_R$

Please

X Check Additional Sheets for Answer ! X

$V^{\pi_{\text{left}}}(s) = \frac{2}{3}$	$V^{\pi_{\text{left}}}(s_1) = \frac{1}{3}$	$V^{\pi_{\text{left}}}(s_2) = \frac{13}{3}$
$V^{\pi_{\text{right}}}(s) = 1$	$V^{\pi_{\text{right}}}(s_1) = 0.5$	$V^{\pi_{\text{right}}}(s_2) = 4.5$

(Question 10c. 2 points) Assume that the agent is always initialized in state s. Based on the values computed above, argue/explain what is the optimal policy for this MDP: π_{left} or π_{right}?

Since $V^{\pi_{\text{right}}}(s) \geq V^{\pi_{\text{left}}}(s) \quad \forall\, s \in \{s, s_1, s_2\}$

$\pi_{\text{right}} \geq \pi_{\text{left}}$ [Here it is strictly $>$] (not $\geq$)

Here, π_{right} is the optimal policy for this MDP.

Extra Credit Questions (10 points)

1. (3 points) Assume an MDP with 1 state (s) and 3 actions (a_1, a_2, a_3). Assume that the current policy is $\pi(s)=a_2$. After evaluating π, we find that $q^\pi(s,a_1)=10$, $q^\pi(s,a_2)=13$, and $q^\pi(s,a_3)=9$. Show how to perform one step of Policy Improvement and present the new, updated policy, π'. What can be said about π'?

Policy Improvement $\Rightarrow$ $\pi' = \arg\max_a q^\pi(s,a)$

$\pi'(s) = a_2$ $\quad (\because q^\pi(s,a_2) \geq q^\pi(s,a)\ \forall a \in \{a_1, a_2, a_3\}$

Since $\pi'(s) = \pi(s) \Rightarrow \pi'$ is the optimal policy for this MDP.

2. (7 points) The First-Visit Monte Carlo algorithm uses an estimator, $\hat{v}(s)$, of the value of some state s, constructed by taking the average of all corresponding returns G_i^s, where G_i^s is the return observed from the first occurrence of state s in the i-th trajectory. In class we showed that First-Visit Monte Carlo is unbiased. Suppose we wish to construct a variant of this algorithm (which we will call Prime-Visits Monte Carlo) that only takes into account the returns G_i^s for which i is a prime number. In particular, assume you are given a function IsPrime(x) = 1 iff x is a prime number, and 0 otherwise. The Prime-Visits Monte Carlo estimator of the value of some state s is defined as $\hat{v}_{PV}(s) = \frac{1}{\sum_{i=1}^{n} \text{IsPrime(i)}} \sum_{i=1}^{n} (\text{IsPrime(i)}\ G_i^s)$. Is Prime-Visits Monte Carlo unbiased? If so, prove that this is the case. If not, formally argue/explain why.

Yes, Prime Visits Monte Carlo is <u>unbiased</u>.

Since $\hat{V}(s) = \frac{1}{n} \sum_{i=1}^{n} G_i^s$ is unbiased.

It follows that if we modify n i.e. reduce n or increase n, it still remains unbiased.

Essentially $\hat{V}_{PV}(s) = \frac{1}{\sum_{i=1}^{n} \text{IsPrime(i)}} \sum_{i=1}^{n} (\text{IsPrime(i)}\ G_i^s)$

does just that. The number of samples being collected is now lesser since we only take the prime trajectories into consideration.

(We are still using the return computed from the first visit of each of the prime trajectories.)

FRONT

NIRUPAN ANANTHAMURUGAN | 33591075

Q. 10b

Assume $\pi_{left} = \pi_L$ & $\pi_{right} = \pi_R$

$$V^{\pi_L}(s) = \sum_a \pi_L(s,a) \sum_{s'} p(s,a,s') \left(R(s,a) + \gamma V^{\pi_L}(s')\right)$$

$$= 1\left(0.5\left(1 + 0.5\, V^{\pi_L}(s_1)\right) + 0.5\left(-2 + 0.5\, V^{\pi_L}(s_2)\right)\right)$$

$$= 0.5 + 0.25 V^{\pi_L}(s_1) - 1 + 0.25 V^{\pi_L}(s_2)$$

$$\Rightarrow \boxed{V^{\pi_L}(s) = 0.25 V^{\pi_L}(s_1) + 0.25 V^{\pi_L}(s_2) - 0.5} \rightarrow (1)$$

$$V^{\pi_L}(s_1) = 1\left(1\left(0 + 0.5 V^{\pi_L}(s)\right)\right)$$

$$\Rightarrow \boxed{V^{\pi_L}(s_1) = 0.5 V^{\pi_L}(s)} \rightarrow (2)$$

$$V^{\pi_L}(s_2) = 1\left(1\left(4 + 0.5 V^{\pi_L}(s)\right)\right)$$

$$\Rightarrow \boxed{V^{\pi_L}(s_2) = 4 + 0.5 V^{\pi_L}(s)} \rightarrow (3)$$

Substituting (2) & (3) in (1)...

$$V^{\pi_L}(s) = 0.25\left(0.5 V^{\pi_L}(s)\right) + 0.25\left(4 + 0.5 V^{\pi_L}(s)\right) - 0.5$$

$$V^{\pi_L}(s) = 0.5\left(0.5 V^{\pi_L}(s)\right) + 1 - 0.5$$

$$V^{\pi_L}(s) = 0.25 V^{\pi_L}(s) + 0.5$$

$$V^{\pi_L}(s) = \frac{0.5}{0.75} = \frac{2}{3}$$

Substituting in (2) & (3), we get:

$$\boxed{V^{\pi_L}(s) = \frac{2}{3}} \qquad \boxed{V^{\pi_L}(s_1) = \frac{1}{3}} \qquad \boxed{V^{\pi_L}(s_2) = \frac{13}{3}}$$

NIRUPAN ANANTHAMURUGAN | 33591075

$$V^{\pi_R}(s) = \sum_a \pi_R(s,a) \sum_{s'} p(s,a,s') \left(R(s,a) + \gamma V^{\pi_R}(s')\right)$$

$$= 1\left(1\left(0.5 + 0.5\, V^{\pi_R}(s)\right)\right)$$

$$V^{\pi_R}(s) = 0.5\, V^{\pi_R}(s) + 0.5$$

$$\Rightarrow \boxed{V^{\pi_R}(s) = 1} \rightarrow (1)$$

$$V^{\pi_R}(s_1) = 1\left(1\left(0 + 0.5\, V^{\pi_R}(s)\right)\right)$$

$$\Rightarrow \boxed{V^{\pi_R}(s_1) = 0.5\, V^{\pi_R}(s)} \rightarrow (2)$$

$$V^{\pi_R}(s_2) = 1\left(1\left(4 + 0.5\, V^{\pi_R}(s)\right)\right)$$

$$\Rightarrow \boxed{V^{\pi_R}(s_2) = 4 + 0.5\, V^{\pi_R}(s)} \rightarrow (3)$$

Substituting (1) in (2) & (3), we get:

$$\boxed{V^{\pi_R}(s) = 1} \qquad \boxed{V^{\pi_R}(s_1) = 0.5} \qquad \boxed{V^{\pi_R}(s_2) = 4.5}$$

Q. 9 (continued) The assumption for MC method is that the MDP must be finite. Gridworld MDP is not finite MDP.

Q. 8 (continued) Since $J(\pi)$ was the max. return possible in the original MDP, $\alpha J(\pi)$ would be the max. return possible in the new MDP where all Rewards are multiplied with α.

Hence π would still be the optimal policy.

(Note: π_R could have also been the optimal π to start with)