Reinforcement Learning: Midterm 2 Solution April 15, 2020

Name: _____

Start time: _____ Finish time: _____ Total minutes_____

Answer the questions in the answer spaces provided on the question sheets. If you run out of room for an answer, note in the answer space that it is continued on another page, and continue on a blank sheet.

No use of a computing device is allowed other than as a timer, clock, music player, simple (non-programmable) scientific calculator (for example, https://www.calculator.net/scientific-calculator.html), or for word processing (editing this PDF file or writing your answers in some word processor). This is a closed-book exam. You may have one 8.5x11 single-sided handwritten page of notes.

If you think something about a question is open to interpretation, write any assumptions you've made as part of answering the question.

Be concise in your answers; you need not try to fill in all or even most of the space provided for an answer. Show your work, though, since partial credit may be awarded.

You have four contiguous hours to complete this exam starting from when you look at any page other than the first. The four hours is much more than I think you need, but should allow for any needed time for dealing with computer issues in creating your final PDF.

Submit the completed exam to Gradescope no later than 1 PM, April 15, 2020. You may submit:

- An edited version of the PDF with answers added (PDF editor on iPad for example, with handwritten answers, or Preview on Mac with text annotation).
- A scanned paper version of the PDF (that you've handwritten answers on). Make sure the scanned version is legible before you submit it.
- A PDF output of some word processor. For example, you can write your answers in LaTeX, or Microsoft Word (with Equations), or Google Docs (with Auto-Latex addon). You need not repeat the question: just provide your answer.

Good luck!

5 points 1. Circle all of the following methods that use bootstrapping to estimate values:

A. Q-learning

B. Sarsa

C. Expected Sarsa

D. Tree Backup

E. Monte Carlo Tree Search

6 points 2. Circle all of the following statements that are true about Monte Carlo Tree Search (MCTS):

A. MCTS works only for episodic tasks

B. MCTS does planning in the background

C. MCTS does planning at decision time

D. MCTS can be used with a random rollout policy

E. MCTS can be used with a non-random rollout policy

F. The intent of MCTS is to select an action better than that of the underlying rollout policy

5 points 3. When using Temporal Difference learning, why is it better to learn action values (Q-values) rather than state values (V-values)?

Solution: Because it is easy to determine a policy from the Q-values. If you only have state values, that doesn't help determine the correct action to take from a given state unless you know the dynamics of the environment.

5 points 4. Briefly describe what *bootstrapping* is in the context of Reinforcement Learning.

Solution: Bootstrapping is the process of updating value estimates of states (or state/action pairs) on the basis of value estimates of other states (or state/action pairs).

24 points

5. You are given an environment with 1 state, x, and 2 actions, b and c. T is the terminal state. Your TD algorithm generates the following episode using the policy π when interacting with its environment:

Timestep	Reward	State	Action
0		x	b
1	16	x	c
2	12	x	b
3	16	T	

- The policy π is given by: $\pi(b|x) = 0.9, \pi(c|x) = 0.1$
- The current values of q are: $q(x, b) = 1$ and $q(x, c) = 2$.
- the discount factor, γ, is $\frac{1}{2}$.
- the step size, α, is 0.1

Show the values of $q(x, b)$ and $q(x, c)$ after their *first* update using 1-step Sarsa, 2-step Sarsa, 2-step Expected Sarsa, and 2-step Tree Backup. Note: you should update $q(x, b)$ and $q(x, c)$ only once per learning algorithm. **Show your work** and carry out your calculations to *two* decimal places.

Learning Algorithm	$q(x, b)$ after its first update	$q(x, c)$ after its first update
1-step Sarsa	**2.6**	**3.13**
2-step Sarsa	**3.13**	**3.8**
2-step Expected Sarsa	**3.13**	**3.8**
2-step Tree Backup	**2.61**	**3.73**

Solution:

0.1 1-step Sarsa

After the first timestep:

$$
\begin{aligned}
q(S_0, A_0) &= q(S_0, A_0) + \alpha[R_1 + \gamma q(S_1, A_1)) - q(S_0, A_0)] \\
q(x, b) &= q(x, b) + \alpha[R_1 + \gamma q(x, c) - q(x, b)] \\
&= 1 + .1[16 + +.5 \times 2 - 1] \\
&= 1 + .1[16 + 1 - 1] \\
&= 2.6
\end{aligned}
$$

After the second timestep,

$$\begin{aligned} q(S_1, A_1) &= q(S_1, A_1) + \alpha[R_2 + \gamma q(S_2, A_2)) - q(S_1, A_1)] \\ q(x, c) &= q(x, c) + \alpha[R_2 + \gamma q(x, b) - q(x, c)] \\ &= 2 + .1[12 + +.5 \times 2.6 - 2] \\ &= 2 + .1[12 + 1.3 - 2] \\ &= 2 + .1[11.3] \\ &= 3.13 \end{aligned}$$

0.2 2-step Sarsa

After the second timestep:

$$\begin{aligned} q(S_0, A_0) &= q(S_0, A_0) + \alpha[R_1 + \gamma R_2 + \gamma^2 q(S_2, A_2) - q(S_0, A_0)] \\ q(x, b) &= q(x, b) + \alpha[R_1 + \gamma R_2 + \gamma^2 q(x, b) - q(x, b)] \\ &= 1 + .1[16 + .5 \times 12 + .5^2 \times 1 - 1] \\ &= 1 + .1[16 + 6 + .25 - 1] \\ &= 1 + .1[21.25] \\ &= 3.13 \end{aligned}$$

After the third timestep,

$$\begin{aligned} q(S_1, A_1) &= q(S_1, A_1) + \alpha[R_2 + \gamma R_3 + \gamma^2 q(S_3, A_3)) - q(S_1, A_1)] \\ q(x, c) &= q(x, c) + \alpha[R_2 + \gamma R_3 + \gamma^2 q(T, \cdot) - q(x, c)] \\ &= 2 + .1[12 + .5 \times 16 + .5^2 \times 0 - 2] \\ &= 2 + .1[12 + 8 - 2] \\ &= 3.8 \end{aligned}$$

0.3 2-step Expected Sarsa

After the second timestep:

$$q(S_0, A_0) = q(S_0, A_0) + \alpha[R_1 + \gamma R_2 + \gamma^2(\sum_a \pi(a|S_2)q(S_2, a)) - q(S_0, A_0)]$$

$$\begin{aligned} q(x, b) &= q(x, b) + \alpha[R_1 + \gamma R_2 + \gamma^2(\sum_a \pi(a|x)q(x, a)) - q(x, b)] \\ &= 1 + .1[16 + .5 \times 12 + .5^2(.9 \times 1 + .1 \times 2) - 1] \\ &= 1 + .1[16 + 6 + .25(1.1) - 1] \\ &= 1 + .1[16 + 6 + .28 - 1] \\ &= 1 + .1[21.28] \\ &= 3.13 \end{aligned}$$

After the third timestep,

$$q(S_1, A_1) = q(S_1, A_1) + \alpha[R_2 + \gamma R_3 + \gamma^2(\sum_a \pi(a|S_3)q(S_3, a)) - q(S_1, A_1)]$$

$$\begin{aligned} q(x, c) &= q(x, c) + \alpha[R_2 + \gamma R_3 + \gamma^2(\sum_a \pi(a|T)q(T, a)) - q(x, c)] \\ &= 2 + .1[12 + .5 \times 16 + .5^2(\sum_a 0.5 \times 0) - 2] \\ &= 2 + .1[12 + 8 - 2] \\ &= 3.8 \end{aligned}$$

0.4 2-step Tree Backup

After the second timestep:

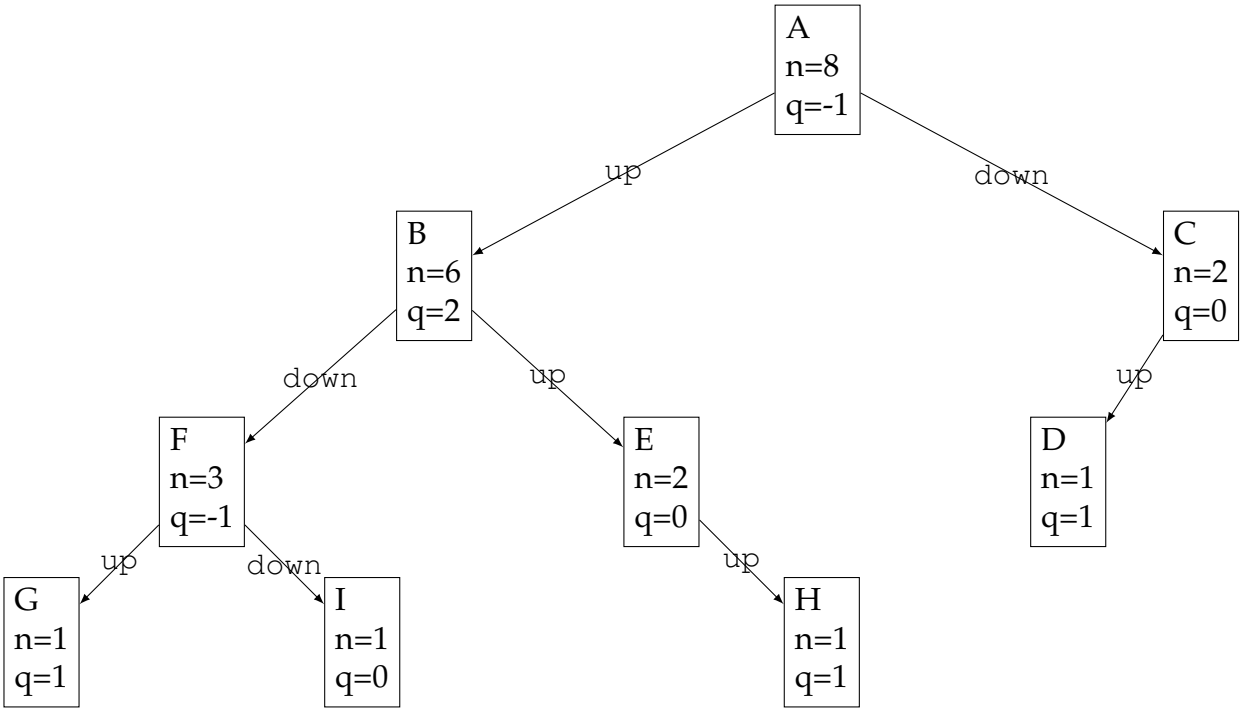
$$
\begin{aligned}
q(S_0, A_0) &= q(S_0, A_0) + \alpha[R_1 + \gamma \sum_{a \neq A_1} \pi(a|S_1) q(S_1, a) \\
&\quad + \gamma \pi(A_1|S_1)(R_2 + \gamma \sum_a \pi(a|S_2) q(S_2, a)) - q(S_0, A_0)] \\
q(x, b) &= q(x, b) + \alpha[R_1 + \gamma \sum_{a \neq c} \pi(a|x) q(x, a) \\
&\quad + \gamma \pi(c|x)(R_2 + \gamma \sum_a \pi(a|x) q(x, a)) - q(x, b)] \\
&= q(x, b) + \alpha[R_1 + \gamma \pi(b|x) q(x, b) \\
&\quad + \gamma \pi(c|x)(R_2 + \gamma(\pi(b|x) q(x, b) + \pi(c|x) q(x, c))) - q(x, b)] \\
&= 1 + .1[16 + .5 \times .9 \times 1 \\
&\quad + .5 \times .1(12 + .9 \times 1 + .1 \times 2) - 1] \\
&= 1 + .1[16 + .45 + .05(12 + .5(.9 + .2)) - 1] \\
&= 1 + .1[16 + .45 + .05(12.55) - 1] \\
&= 1 + .1[16 + .45 + .6275 - 1] \\
&= 1 + .1[16.08] \\
&= 2.61
\end{aligned}
$$

After the third timestep,

$$
\begin{aligned}
q(S_1, A_1) &= q(S_1, A_1) + \alpha[R_2 + \gamma \sum_{a \neq A_2} \pi(a|S_2) q(S_2, a) \\
&\quad + \gamma \pi(A_2|S_2)(R_3 + \sum_a \pi(a|S_3) q(S_3, a)) - q(S_1, A_1)] \\
q(x, c) &= q(x, c) + \alpha[R_2 + \gamma \sum_{a \neq b} \pi(a|x) q(x, a) \\
&\quad + \gamma \pi(c|x)(R_3 + \sum_a \pi(a|T) q(T, a)) - q(x, c)] \\
&= q(x, c) + \alpha[R_2 + \gamma \pi(c|x) q(x, c) \\
&\quad + \gamma \pi(c|x)(R_3 + 0) - q(x, b)] \\
&= 2 + .1[12 + .5 \times .1 \times 2 + .5 \times .9(16 + 0) - 2] \\
&= 2 + .1[12 + .1 + .45 * 16 - 2] \\
&= 2 + .1[12 + .1 + 7.2 - 2] \\
&= 2 + .1[17.3] \\
&= 3.73
\end{aligned}
$$

10 points

6. You are using Monte Carlo Tree Search to decide on the next action for a two-person competitive game with 2 actions at each state (`up` and `down`). It is player 1's turn to play in state A. The state of the tree so far is as follows (each node consists of state identifier, n value, and q value):



Remember that the formula for the UCT value for a node, v, is:

$$UCT(v) = \frac{q(v)}{n(v)} + c\sqrt{\frac{\ln n(v.parent)}{n(v)}}$$

Assume the constant c in the UCT formula is 0.5.

(a) What is the node that is next selected (show your work)?

Solution: We start at the root node, A. Since A is fully expanded, we use the UCT formula for its children, B, and C.

$$UCT(B) = \frac{2}{6} + 0.5\sqrt{\frac{\ln 8}{6}} = .63$$

$$UCT(C) = \frac{0}{2} + 0.5\sqrt{\frac{\ln 8}{2}} = .51$$

Since UCT(B) > UCT(A), we move to node B. Since B is fully expanded, we

use the UCT formula for its children, F and E.

$$UCT(F) = \frac{-1}{3} + 0.5\sqrt{\frac{\ln 6}{3}} = .05$$

$$UCT(E) = \frac{0}{2} + 0.5\sqrt{\frac{\ln 6}{2}} = .47$$

Since UCT(E) > UCT(F), we move to node E. Since E is not fully expanded, E is the next selected node.

(b) What are the details of the node that gets expanded from the selected node?

Solution: We'll add a new node with action `down` to E (call it J) with n=0 and q=0.

(c) Assuming that the simulation (rollout) from the expanded node gives a value of 1 (that is, player 1 wins), backup that value to all of the affected nodes.

Solution: Since A is a state reached from player 2 playing, B is a state reached from player 1 playing, then E is a state reached from player 2 playing, and J is a state reached from player 1 playing. Thus, we will increment q for states B and J, and decrement it for states A and E.

We'll increment n for states A, B, E, and J.

New vales: A: n=9, q=-1 - 1 = -2
B: n=7, q=2 + 1 = 3
E: n=3, q=0 - 1 = -1
J: n=1, q=0 + 1 = 1

(d) Assume that after this final rollout, we've run out of time to run the MCTS simulation and must now choose an action for player 1 from state A. What action will be chosen and why?

Solution: If we're using the highest n value, we'll choose action `up` since that leads to state B and n(B) > n(C).

If we're using the highest $\frac{q}{n}$ value, we'll also choose action `up` since that leads to state B and $\frac{q(B)}{n(B)} > \frac{q(C)}{n(C)}$.

5 points 7. What's the main difference between the Dyna-Q and Dyna-Q+ algorithms?

Solution: The Dyna-Q+ algorithm adds an exploration bonus to the reward based on how long it's been since the state/action pair has been selected in the actual environment.

4 points 8. Direct reinforcement learning updates a *policy* based on *interactions with the environment*. Planning (indirect reinforcement learning), however, updates a **policy** based on **a model**.

4 (bonus) 9. According to `https://ourworldindata.org/coronavirus`, the number of Covid-19 deaths in Italy approximately doubled from March 16, 2020 (370) to March 22, 2020 (795). **Using *the Rule of 72***, approximate the daily percentage increase.

Solution: There are six days from March 16, 2020 to March 22, 2020. 72% divided by 6 is 12%, so the daily percentage increase is approximately 12%.