CS 221, Fall 2020
Quiz 4 Solutions

1. **[0 points]** You are allowed to consult online lecture notes, personal notes, and use a basic calculator during this quiz. You may not use the internet to access any websites other than Gradescope or the course website. You may not communicate with others to give or receive aid during or after the quiz. **I agree to abide by these rules and the Stanford Honor Code.**

   (a) Agree

   (b) Disagree

   Answer: (a)

2. **[2 points]** Which of the following are components of the definition of a Markov decision process (MDP)?

   (a) A transition distribution

   (b) A policy

   (c) An agent

   (d) A discount factor γ

   (e) An exploration/exploitation trade-off factor ϵ

   Answer: (a), (d) See slide 10: https://stanford-cs221.github.io/autumn2020/modules/module.html#include=mdps/modeling.js&mode=print1pp

3. **[2 points]** If you're given an oracle that can compute the optimal value, $V_{\text{opt}}(s)$, of any state s in any MDP in constant time, then you can compute the min cost path for __Blank 1__ by invoking the oracle __Blank 2__.

   Blank 1 options:

   (a) no graph search problems (because graph search is a much harder problem than computing the optimal value of a state an MDP)

   (b) only graph search problems with only negative edge costs

   (c) only graph search problems with only positive edge costs

   (d) any graph search problem

   Answer: (d) The oracle is powerful enough to solve any graph search problem, see slide 12: https://stanford-cs221.github.io/autumn2020/modules/module.html#include=mdps/modeling.js&mode=print1pp

   Blank 2 options:

   (a) once

   (b) $O(n)$ times (where n is the length of the min cost path)

   (c) $O(nb)$ times (where n is the length of the min cost path and b is the maximum number of actions available in any state along the min cost path)

   (d) $O(n^2b)$ times (where n is the length of the min cost path and b is the maximum number of actions available in any state along the min cost path)

   (e) $O(n^2b^2)$ times (where n is the length of the min cost path and b is the maximum number of actions available in any state along the min cost path)

   (f) none of the above

   Answer: (c) The oracle must be invoked once for every action from every state along the min cost path.

4. **[2 points]** Select each choice that, by itself, guarantees the convergence of value iteration (without additional assumptions).

   (a) The MDP has an even number of end states.

   (b) The discount factor is less than one.

   (c) The MDP graph is tree structured.

   (d) The MDP graph contains at least one loop.

   (e) The end state can only be reached from a single state action pair.

   (f) There is a non-zero probability of transitioning from any state action pair to the end state.

   Answer: (b), (c), (f) See slide 14: https://stanford-cs221.github.io/autumn2020/modules/module.html#include=mdps/value-iteration.js&mode=print1pp

5. **[2 points]** When running an epsilon-greedy algorithm it is common to

   (a) **Decrease** epsilon to **decrease** exploitation as time passes.

   (b) **Decrease** epsilon to **increase** exploitation as time passes.

   (c) **Increase** epsilon to **decrease** exploitation as time passes.

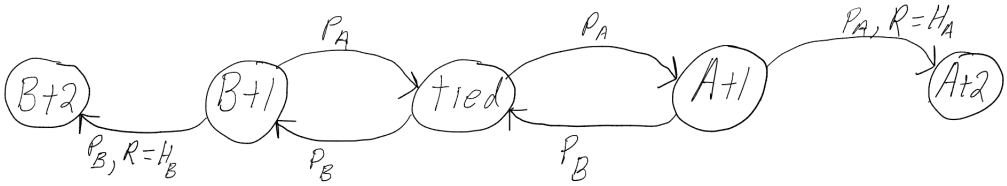
   (d) **Increase** epsilon to **increase** exploitation as time passes.

   (e) Cycle epsilon to encourage waves of exploitation and exploration.

   Answer: (b) See slide 10: https://stanford-cs221.github.io/autumn2020/modules/module.html#include=mdps/epsilon-greedy.js&mode=print1pp

6. **[3 points]** Two friends of yours are playing ping pong, friend A and friend B. Friend A wins every point with probability P_a and friend B wins every point with probability $P_b = 1 - P_a$. You promised to buy a bottle of coke for friend A if they win and you promised to buy friend B a cheeseburger if they win. Right now your two friends are tied with a score of 10-10. Ping pong is played to 11 points, but a player must win by two points. In expectation, how much money will you have to spend to buy your winning friend a prize given that $P_a = .7$, a bottle of coke costs \$1.50, and a cheeseburger costs \$10? (Assume that your friends won't stop playing until one wins. Hint 1: you may want to draw an MDP with a single action available in each state and compute the value of a particular state. Hint 2: be sure to enter a positive number.)

   Answer:



$$V_{B+1} = P_b H_B + P_a V_{\text{tied}}, \quad V_{\text{tied}} = P_a V_{A+1} + P_b V_{B+1}, \quad V_{A+1} = P_a H_A + P_b V_{\text{tied}}$$
$$V_{\text{tied}} = P_a^2 H_A + P_b^2 H_B + 2 P_a P_b V_{\text{tied}}$$
$$V_{\text{tied}} = \frac{P_a^2 H_A + P_b^2 H_B}{1 - 2 P_a P_b}$$
$$V_{\text{tied}} \approx 2.819$$

7. **[4 points total]** You love burritos from your local joint, but are particular about the ingredients. Your ideal burrito contains cold black beans (that aren't steaming), hot carnitas (that are steaming), and fresh salsa. Burritos are ordered at a counter. The server first asks if you'd like black or pinto beans. In the past you've tried ordering (1) black beans or (2) whichever type of bean is colder. Then the server asks if you'd like carnitas or chicken. You've tried ordering (1) carnitas or (2) whichever is hotter. Finally the server asks for your salsa preference. Note that only after you've ordered can you see the beans or meat scooped onto your burrito to check which variety you're getting and whether they're hot (based on whether they're steaming).

   To systematically improve your ordering strategy (or policy π), you decide to model the burrito ordering process as an MDP. States represent the composition of your burrito as it is made. This MDP has discount factor 1 and a non-zero reward function only at end states, which correspond to a complete burrito. You write down Q-value estimates when using your standard ordering strategy, some of which are shown in Table 1.

state (s)	action (a)	$\hat{Q}_\pi(s,a)$
start	order black beans	10
start	order colder beans	8
cold black beans	order carnitas	14
cold black beans	order hotter meat	12
hot black beans	order carnitas	7
hot black beans	order hotter meat	8
cold pinto beans	order carnitas	10
cold pinto beans	order hotter meat	5

Table 1: Your initial Q-value estimates when following your standard policy π. Note that you model the temperature of your food as a binary variable, either hot and steaming or cold and not steaming.

Then you collect data using your standard ordering strategy. On your **first data collection trip** you order black beans, are given hot black beans, then order whichever meat is hotter, are given hot carnitas, finally order salsa roja, and are served a burrito whose quality is worth a reward of 12.

(a) **[2 points]** You decide to use Q-learning to estimate Q-values for the optimal policy. You use $\hat{Q}_\pi(s,a)$ (given in Table 1) as your initial estimate of the optimal Q-values, $\hat{Q}_{\text{opt}}(s,a)$. Using the step size $\eta = .5$, what will your updated estimate of $Q_{\text{opt}}(\text{start}, \text{order black beans})$ be after only processing data from your first data collection trip?

Answer: $\hat{Q}_{\text{opt}}(\text{start}, \text{order black beans}) \leftarrow .5 * 10 + .5 * 8 = 9$

See slide 4: https://stanford-cs221.github.io/autumn2020/modules/module.html#include=mdps/q-learning.js&mode=print1pp

(b) **[2 points]** To double check your initial Q-value estimates, you decide to use model free Monte Carlo to compute Q-value estimates from scratch using only newly collected data (and ignoring the values from Table 1). You make a **second data collection trip** where you order black beans, are given cold black beans, then order carnitas, are given hot carnitas, finally order salsa verde, and are served a burrito whose quality is worth a reward of 15. What would your model free Monte Carlo estimate of $Q_\pi(\text{start}, \text{order black beans})$ be, using data from **both** the first and second data collection trips?

Answer: $\hat{Q}_\pi(\text{start}, \text{order black beans}) \leftarrow .5 * 12 + .5 * 15 = 13.5$

See slides 4 and 8: https://stanford-cs221.github.io/autumn2020/modules/module.html#include=mdps/model-free-monte-carlo.js&mode=print1pp