CS 188 Spring 2022 | Introduction to Artificial Intelligence | Written HW 11

Due: Friday 04/29/2022 at 10:59pm (submit via Gradescope).

Policy: Can be solved in groups (acknowledge collaborators) but must be written up individually

Submission: It is recommended that your submission be a PDF that matches this template. You may also fill out this template digitally (e.g. using a tablet). **However, if you do not use this template, you will still need to write down the below four fields on the first page of your submission.**

First name	
Last name	
SID	
Collaborators	

For staff use only:

Q1. MDPs and RL	/26
Total	/26

Q1. [26 pts] MDPs and RL

The agent is in a 2×4 gridworld as shown in the figure. We start from square 1 and finish in square 8. When square 8 is reached, we receive a reward of $+10$ at the game end. For anything else, we receive a constant reward of -1 (you can think of this as a time penalty).

1	2	3	4
5	6	7	8

The actions in this MDP include: up, down, left and right. The agent cannot take actions that take them off the board. In the table below, we provide initial non-zero estimates of Q values (Q values for invalid actions are left as blanks):

Table 1

	action=up	action=down	action=left	action=right
state=1		Q(1, down)=4		Q(1, right)=3
state=2		Q(2, down)=6	Q(2, left)=4	Q(2, right)=5
state=3		Q(3, down)=8	Q(3, left)=5	Q(3, right)=7
state=4		Q(4, down)=9	Q(4, left)=6	
state=5	Q(5, up)=5			Q(5, right)=6
state=6	Q(6, up)=4		Q(6, left)=5	Q(6, right)=7
state=7	Q(7, up)=6		Q(7, left)=6	Q(7, right)=8

(a) Your friend Adam guesses that the actions in this MDP are fully deterministic (e.g. taking down from 2 will land you in 6 with probability 1 and everywhere else with probability 0). Since we have full knowledge of T and R, we can thus use the Bellman equation to improve (i.e., further update) the initial Q estimates.

Adam tells you to use the following update rule for Q values, where he assumes that your policy is greedy and thus does $\max_a Q(s, a)$. The update rule he prescribes is as follows:

$$Q_{k+1}(s,a) = \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma \max_{a'} Q_k(s',a')]$$

(i) [1 pt] Perform one update of $Q(3, \text{left})$ using the equation above, where $\gamma = 0.8$. You may break ties in any way.

(ii) [1 pt] Perform one update of $Q(3, \text{down})$ using the equation above, where $\gamma = 0.8$.

(iii) [3 pts] For the Q update rule prescribed above, how is it different from the Q learning update that we saw in lecture, which is $Q_{k+1}(s, a) = (1 - \alpha)Q_k(s, a) + \alpha * \text{sample}$?

(b) After observing the agent for a while, Adam realized that his assumption of T being deterministic is wrong in one specific way: **when the agent tries to legally move down, it occasionally ends up moving left instead** (except from grid 1 where moving left results in out-of-bound). All other movements are still deterministic.

Suppose we have run the Q updates outlined in the equation above until convergence, to get $Q^*_{wrong}(s, a)$ under the original assumption of the wrong (deterministic) T. Suppose $Q^*_{correct}(s, a)$ denotes the Q values under the new correct T. Note that you don't explicitly know the exact probabilities associated with this new T, but you know that it qualitatively differs in the way described above. As prompted below, list the set of (s, a) pairs where $Q^*_{wrong}(s, a)$ is either an over-estimate or under-estimate of $Q^*_{correct}(s, a)$.

(i) [3 pts] List of (s, a) where $Q^*_{wrong}(s, a)$ is an over-estimate. Explain why.

(ii) [3 pts] List of (s, a) where $Q^*_{wrong}(s, a)$ is an under-estimate (and why):

(c) [2 pts] Suppose that we have a mysterious oracle that can give us either all the correct Q-values $Q(s,a)$ or all the correct state values $V(s)$. Which one do you prefer to be given if you want to use it to find the optimal policy, and why?

(d) [2 pts] Suppose that you perform actions in this grid and observe the following episode: 3, right, 4, down, 8 (terminal).

With learning rate $\alpha = 0.2$, discount $\gamma = 0.8$, perform an update of $Q(3, right)$ and $Q(4, down)$. Note that here, we update Q values based on the sampled actions as in TD learning, rather than the greedy actions.

(e) [2 pts] One way to encourage an agent to perform more exploration in the world is known as the "ϵ-greedy" algorithm. For any given policy $\pi(s)$, this algorithm says to take the original action $a = \pi(s)$ with probability $(1-\epsilon)$, and to take a random action (drawn from a uniform distribution over all legal actions) with probability ϵ. If ϵ can be tuned, would you assign it to be a high or low value at the beginning of training? What about at the end of the training? Please answer both questions and justify your choices.

(f) Instead of using the "ϵ-greedy" algorithm, we will now do some interesting exploration with softmax. We first introduce a new type of policy: A stochastic policy $\pi(a|s)$ represents the probability of action a being prescribed, conditioned on the current state. In other words, the policy is a now a distribution over possible actions, rather than a function that outputs a deterministic action.

Let's define a new policy as follows:

$$\pi(a|s) = \frac{e^{Q(s,a)}}{\sum_{a'} e^{Q(s,a')}}$$

(i) [2 pts] Suppose we are at square 3 in the grid and we want to use the originally provided Q values from the table. What is the probability that this policy will tell us to go right? What is the probability that this policy will tell us to go left? Note that the sum over actions prescribed above refers to a sum over legal actions. You may leave your answer in terms of e.

(ii) [2 pts] How is this exploration strategy qualitatively different from "ϵ-greedy"?

(g) Your friend Cody argues that we could still explicitly calculate Q updates (like Adam's approach in part (a)) even if we don't know the true underlying transition function $T(s, a, s')$, because he believes that our T can be roughly approximated from samples.

(i) [3 pts] Suppose you collect 1,000 transitions from $s = 3, a = Down$, in the form of (s_{start}, a, s_{end}). Describe how you can use these samples to compute $T_{approx}(s = 3, a = Down, s')$, which is an approximation of the true underlying (unknown) $T(s, a, s')$.

(s =3, a = Down, s'= 6)	(s = 3, a= Down, s'=7)
99	901

(ii) [2 pts] Now perform one step of q-value iteration based on your transition model computed above.