Markov Decision Problem and Reinforcement Learning Exercises

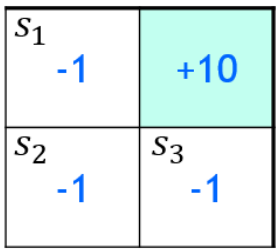
Prof. Jim Rehg
CS 3600 Introduction to Artificial Intelligence
School of Interactive Computing
Georgia Institute of Technology

February 26, 2018

These practice exercises are for your benefit in preparing for the exams. They will not be collected or graded. Solutions will be provided. Note that not all questions are representative of questions you will find on the exam, but the material covered by these questions will also be covered by the exams.

Question 1. **Markov Decision Problem**

Consider the simple grid-world MDP shown below. The agent starts at random in one of three states S_1 to S_3 and must reach the shaded terminal square and collect the reward of +10. All nonterminal states have a reward of -1. Assume no discounting ($\gamma = 1$). The available actions are *Left*, *Right*, *Up*, and *Down*. Use the standard probabilistic transition model for actions in which the probability of moving in the desired direction is 0.8 and there is 0.1 chance to move in each of the orthogonal directions (i.e. there is no probability of moving backwards).



(a) (6 pts) Assuming that the utilities are initialized to zero, perform *two rounds of value iteration*, showing the updated utilities after each round. In order to speeed convergence, use the most recent utility values that are available (i.e. use utilities computed earlier in the same iteration whenever they are available.)

(b) (3 pts) Use *policy iteration* to solve for the optimal value function in one step, by identifying the optimal policy and then using policy evaluation. *Hint:* Solve the linear system of equations given by the Bellman equations. Verify that these are indeed the optimal utilities by showing that they cause *value iteration* to terminate.

(c) (2 pts) Compute the table $Q^*(s, a)$ for the optimal policy π^* from part (b). The table is indexed by (s, a) pairs, where $Q^*(s, a)$ is the expected discounted reward obtained by starting in state s, collecting $R(s)$, taking action a, and subsequently following π^*.

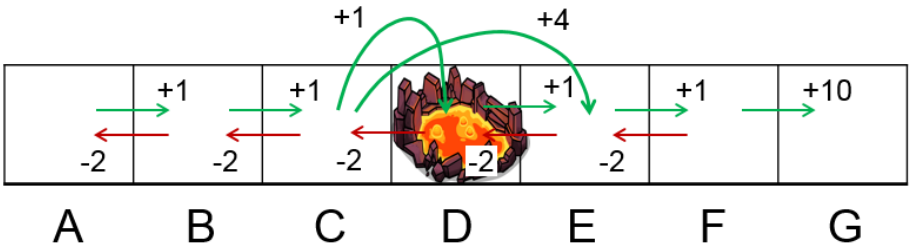
(d) (1 pts) You have obtained the following episodes from running your agent through the grid world, and would like to use Q-Learning.

Episode 1	Episode 2	Episode 3
$S_1, D, S_2, -1$	$S_2, R, S_3, -1$	$S_3, L, S_2, -1$
$S_2, R, S_3, -1$	$S_3, L, S_2, -1$	$S_2, U, S_1, -1$
$S_3, U, T, +10$	$S_2, R, S_3, -1$	$S_1, R, T, +10$
	$S_3, U, T, +10$	

Will Q-Learning be able to learn $Q^*(s, a)$ from part (c) based on these episodes? Why or why not?

Question 2. American Ninja MDP

You need to solve the following MDP to prepare for your upcoming appearance on the hit TV show *American Ninja Warrior*. Your objective is to start at state A, race down the track, jump the lava pit in state D, and reach the finish line (terminal state G). Actions are *Right*, *Left*, and *Jump*, but *Jump* can only be used in state C (and *Right* cannot be used there). Rewards for each state transition are shown in the figure. The discount is $\gamma = 1$.



All actions are *deterministic* except for *Jump*, which succeeds half the time, landing in state E, and fails half the time, landing in D. In summary, the action model is:

Right: Deterministically move to the right.
Left: Deterministically move to the left.
Jump: Stochastically jump to the right. This action is available for square C only.
$T(C, \textit{Jump}, E) = 0.5$ (jump succeeds)
$T(C, \textit{Jump}, D) = 0.5$ (jump fails)

(a) (2 pts) For the policy π of always moving forward (i.e., using actions *Right* or *Jump*), compute $V^{\pi}(C)$.

(b) (3 pts) Perform two iterations of value iteration and fill in the table below. All values are initialized to zero.

$V^2(B)$	
$Q^2(B, \textit{Right})$	
$Q^2(B, \textit{Left})$	

(c) (3 pts) You decide to use Q-Learning to obtain the optimal policy. After some number of iterations of Q-Learning, the Q table has the values given in the row "Initial" below. Apply the Q-Learning update rule to update the Q-values according to the four transitions in the episode given below. Q-values that are unaffected by a particular transition can be left blank. Use a learning rate α of 0.5. *Be sure to use the initial Q-values provided in the top row.*

Episode Data for Q-Learning

s	a	r	s	a	r	s	a	r	s	a	r	s
C	*Jump*	$+4$	E	*Right*	$+1$	F	*Left*	-2	E	*Right*	$+1$	F

	$Q(\text{C}, \textit{Left})$	$Q(\text{C}, \textit{Jump})$	$Q(\text{E}, \textit{Left})$	$Q(\text{E}, \textit{Right})$	$Q(\text{F}, \textit{Left})$	$Q(\text{F}, \textit{Right})$
Initial	-1	1	0	2	0	-2
Transition 1						
Transition 2						
Transition 3						
Transition 4						