ENPM808Z: Cognitive Robotics (Spring 2024)
Assignment 3: Theory
Due: Apr 22nd, 11:00AM *(NOT Extendable)*

March 31, 2024

Problem 1 — 8 points

True/False with justification for the tabular Q-learning algorithm.

(a) If the learning rate α in the update rule is decreased, then the Q-values will change less in an iteration (all else being equal). (2 points)

(b) If the ϵ parameter in the ϵ-greedy exploration strategy is increased, then the agent will explore less often than exploit the known policy. (2 points)

(c) We will get the optimal Q value using approximate Q-learning. (2 points)

Problem 2 — 18 points

A compulsive gambler begins with a capital of $10. At every time instance, the gambler can place a wager of $x where x is less than or equal to the gambler's current capital. That is, initially the gambler can wager either $1, $2, ..., $10.

The bet is simple. You toss a coin. If its heads, the gambler gets back $2x (that is, the original $x that was wagered plus an extra $x). If its tails, the gambler loses $x that the gambler wagered.

For example, suppose the gambler had $10 and bet $3. If its heads, the gambler will have $13. If its tails, the gambler will have $7.

Since the gambler is compulsive, the gambler will continue betting until the capital is at least $100. If the capital reaches $0, the gambler goes bust and cannot wager anymore. The coin may be a biased coin but we assume we know the odds of getting heads (say p) and tails (1-p) where $0 < p < 1$ is known.

Design an algorithm that can help the gambler decide how many dollars to bet at each time by formulating this as an MDP:

(a) What is the state representation? (3 points)

(b) What are the set of actions? (3 points)

(c) What is the reward function? (3 points)

(d) What is the transition function? (3 points)

(e) What discount factor (if any) you would use? (3 points)

(f) What algorithm can be used to solve this problem? (3 points)

Problem 3 — 24 points

For this problem assume that the discount factor $\gamma = 1$. The environment in which the agent moves can be seen in Figure 1. The agent starts from the start state S_1. Double squares denote exit states from which the only action the agent can take is exit. By taking the exit action, the agent collects the reward listed in the double box and then moves to a terminal state where no further rewards can be collected. In all other states (the single boxes), the agent can move to any neighboring state, obtaining a zero reward. For example, from state S_1 the agent can go right by taking action $\rightarrow$.

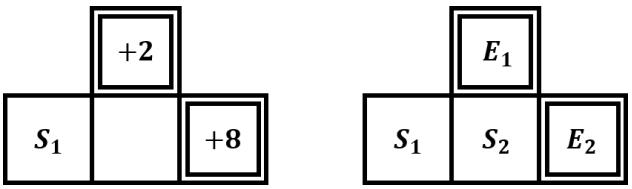

Figure 1: (Left) Start state and rewards for exit actions. (Right) State names.

(a) Given the trajectory in Table 1, please apply direct evaluation and report the V-values estimated for states S_1 and S_2? (8 points)

Now we would like to implement Q-learning to derive an optimal policy. When we run Q-learning, we initialize the Q-values to zero. Assume the sequence of transitions and associated rewards shown in table 1, where X denotes the terminal state.

Table 1: Transitions and associated rewards.

s	a	s'	r
S_1	$\rightarrow$	S_2	0
S_2	$\uparrow$	E_1	0
E_1	$exit$	X	+2
S_1	$\rightarrow$	S_2	0
S_2	$\rightarrow$	E_2	0
E_2	$exit$	X	+8

(b) Which of the following Q-values are non-zero after running Q-learning on the transition-reward pairs above, assuming that we go through the sequence above only one time (i.e. **every** state's Q-value is updated only once)? Select all that apply. (8 points)

A $Q(S_1, \rightarrow)$

B $Q(S_2, \uparrow)$

C $Q(S_2, \rightarrow)$

D $Q(E_1, exit)$

E $Q(E_2, exit)$

(C) Assume we use a learning rate α of 0.5. If we run Q-learning on the dataset above for an infinite number of iterations, then what are the following Q-values convergence to? If a Q-value does not converge, write $None$ for that value. (8 points)

A $Q(S_1, \rightarrow)$

B $Q(S_2, \uparrow)$

C $Q(S_2, \rightarrow)$

D $Q(S_2, \leftarrow)$