DDA 4230: Reinforcement learning **CUHKSZ**

Assignment 2 Reference Answer

TA: Bo Yue, Hengming Zhang

Due Date: Oct. 31th, 11:59 pm
Total points available: 100 points

Note: *Please note that external references are allowed only if you give an appropriate reference. There is no required format of reference. Please elaborate on your answers as well (do not just give a number, etc).*

Problem 1: Value and Policy Iteration Conceptual Questions [20 points]

1. When does policy iteration end? Immediately after policy iteration ends (without performing additional computation), do we have the values of the optimal policy? [10 points]
2. What changes if, during policy iteration, you only run one iteration of Bellman update in policy evaluation instead of running it until convergence? Do you still get an optimal policy? [10 points]

Problem 2: Value and Policy Iteration Computation [30 points]

Consider a simple MDP with 3 states s_1, s_2, s_3 and 2 actions a_1, a_2. The transition probabilities and expected rewards are shown in Figure 1 (e.g., The transition probability of $P(s_1, a_1, s_2) = 0.6$, and the reward of $R(s_1, a_1) = 8.0$). Assume discount factor $\gamma = 1$.

```
s1: {
  a1: ({s1: 0.2, s2: 0.6, s3: 0.2}, 8.0),
  a2: ({s1: 0.1, s2: 0.2, s3: 0.7}, 10.0)
},
s2: {
  a1: ({s1: 0.3, s2: 0.3, s3: 0.4}, 1.0),
  a2: ({s1: 0.5, s2: 0.3, s3: 0.2}, -1.0)
},
s3: {
  a1: ({s3: 1.0}, 0.0),
  a2: ({s3: 1.0}, 0.0)
}
```

Figure 1: A simple MDP example for Problem 2.

1. Initialize the value function for each state to be it's max (over actions) reward, i.e., we initialize the Value Function to be $V_0(s_1) = 10.0, V_0(s_2) = 1.0, V_0(s_3) = 0.0$. Then manually performs two iterations of value iterations. Show the final values for each state and the computation process. [15 points]
2. Let $\pi_k(s)$ denotes the extracted policy after k iterations of value iteration. Argue that $\pi_k(s) = \pi_2(s)$ for all states s. [15 points]

Problem 3: Q-Learning [50 points]

Now we consider the following gridworld as an MDP with Q-learning algorithm:

X	Y
Z	W

In state X, the available actions are right ($\rightarrow$) and down ($\downarrow$). In state Y, the options are left ($\leftarrow$) and down ($\downarrow$). For states Z and W, the only permissible action is to exit, which transitions the agent to the done state, concluding the rollout. It is also important to note that all actions in this MDP are deterministic and will always succeed.

Now, consider the following two episodes. We represent the reward values of each sample as r_1, r_2, r_3, r_4, r_5 (assuming $r_1, r_2, r_3, r_4, r_5 \geq 0$).

Episode 1:

s	a	s'	r
X	$\rightarrow$	Y	r_1
Y	$\downarrow$	W	r_3
W	$exit$	$done$	r_5

Episode 2:

s	a	s'	r
X	$\downarrow$	Z	r_2
Z	$exit$	$done$	r_4

1. Suppose that $\gamma = 0.5, r_1 = r_2 = r_3 = 0, r_4 = 1, r_5 = 10$. Calculate the following values after repeatedly processing the episodes one at a time using Q-Learning until convergence. Assume that all Q-values are initialized to 0 and that $\alpha = 1$.(Please write the calculation process.) [20 points]

- $Q(X, \rightarrow) =$
- $Q(X, \downarrow) =$

2. Now assume that $\gamma = 1$, $\alpha = 1$ and the agent starts from state X. You don't know the rewards, but you know that you want the optimal behavior to be ($X \rightarrow Y \rightarrow W \rightarrow done$). You have the following reward setups denoted by S_i at your disposal:

- $S_1 : (r_1 = 0, r_2 = 0, r_3 = 0, r_4 = 0, r_5 = 0)$
- $S_2 : (r_1 = 0, r_2 = 0, r_3 = 0, r_4 = 1, r_5 = 10)$
- $S_3 : (r_1 = 5, r_2 = 5, r_3 = 5, r_4 = 5, r_5 = 5)$
- $S_4 : (r_1 = 1, r_2 = 0, r_3 = 0, r_4 = 2, r_5 = 1)$

(i) Using the same episodes, which of the above reward setups **could possibly result in, but do not guarantee**, the optimal behavior after Q-learning converges? (more than one. Please explain the reasons for your answers.) [10 points]

(ii) Using the same episodes, which of the above reward setup(s) will guarantee the optimal behavior after Q-learning converges? (more than one. Please explain the reasons for your answers.) [10 points]

3. Now we add one more transition to the original table, where $r = \max\{r_1, r_2, r_3, r_4, r_5\}$. Assume that $\gamma = 1$ and $\alpha = 1$. The modified episodes and reward setups (S_i) are shown below: [10 points]

Episode 1:

s	a	s'	r
X	$\rightarrow$	Y	r_1
Y	$\downarrow$	W	r_3
W	$exit$	$done$	r_5

Episode 2:

s	a	s'	r
Y	$\leftarrow$	X	r
X	$\downarrow$	Z	r_2
Z	$exit$	$done$	r_4

- $S_1 : (r_1 = 0, r_2 = 0, r_3 = 0, r_4 = 0, r_5 = 0)$
- $S_2 : (r_1 = 0, r_2 = 0, r_3 = 0, r_4 = 1, r_5 = 10)$
- $S_3 : (r_1 = 5, r_2 = 5, r_3 = 5, r_4 = 5, r_5 = 5)$
- $S_4 : (r_1 = 1, r_2 = 0, r_3 = 0, r_4 = 2, r_5 = 1)$

(i) Are there any of the above reward setup(s) that will guarantee optimal behavior after Q-learning converges? Please provide a specific analysis.