CS221 Fall 2018 Homework [blackjack]

SUNet ID: [chiwang]
Name: [Chi Wang]

By turning in this assignment, I agree by the Stanford honor code and declare that all of this is my own work.

Problem 1

1.a:

Answer: $V_{opt}(s)$

state	iteration 0	iteration 1	iteration 2
-2	0	0	0
-1	0	15	14
0	0	-5	13.45
1	0	26.5	23
2	0	0	0

1.b:

Answer: π_{opt}

state	π_{opt}
-1	-1
0	1
1	1

Problem 2

2.b:

Answer: Since the given MDP is acyclic, we could use topology sort to organize MDP states into a direct graph. With this direct graph, in stead of using value iteration, we could use below formula to calculate V_{opt}:

$$V_{opt}(s) = max_{a \in Actions(s)} Q_{opt}(s,a) \quad or \quad 0 \tag{1}$$

$$Q_{opt}(s,a) = \sum_{s'} T(s,a,s')[Reward(s,a,s') + \gamma V_{opt}(s')] \tag{2}$$

We could either use top down or bottom up dynamic programming approach to compute V_{opt} for each state (node in direct graph) with only a single pass over all the (s, a, s') triples.

2.c:

Answer:
$T'(s,a,s') = \gamma T(s,a,s') \cup \{T(s,a,o) = (1-\gamma)\}$
$Reward'(s,a,s') = \frac{1}{\gamma} Reward(s,a,s') \cup \{Reward(s,a,o)\}$

First, we know below things for new state o:

$$V_{opt}(o) = 0 \quad (3)$$
$$Reward(s,a,o) = 0 \quad (4)$$

Then, let's compute the $V_{opt}(s)$ with above transitions and rewards for new MDP.

$$V_{opt}(0) = 0 \quad (5)$$
$$Q_{opt}(s) = \sum_{s'} T'(s,a,s')[Reward'(s,a,s') + V_{opt}(s')] \quad (6)$$
$$= \sum_{s'} \gamma T(s,a,s')[Reward'(s,a,s') + V_{opt}(s')] + T(s,a,o)[Reward(s,a,o) + V_{opt}(o)] \quad (7)$$
$$= \sum_{s'} \gamma T(s,a,s')[Reward'(s,a,s') + V_{opt}(s')] \quad (8)$$
$$= \sum_{s'} T(s,a,s')[\gamma \frac{1}{\gamma} Reward(s,a,s') + \gamma V_{opt}(s')] \quad (9)$$
$$= \sum_{s'} T(s,a,s')[Reward(s,a,s') + \gamma V_{opt}(s')] \quad (10)$$

Since the Q value for each state are same between new MDP and old MDP, they have the same optimal value.

Problem 4

4.b:

Answer: Here are some stats of comparing value iteration vs Q-learning:
For small MDP:
Total state count is 40, there are 5 π_{opt} in Q-learning result are different with value iteration result, match rate is 87.5%
For Large MDP:
Total state count is 32569, there are 8216 π_{opt} in Q-learning result are different with value iteration result, match rate is 74.8%

The problem here which makes Q-learning didn't perform well is the extractor function (identityFeatureExtractor) we employed provides no generalization, the feature it generates just show the existence of the (state, action) pair. We need to have a feature list with domain knowledge to make Q-learning result more close to the optimum solution.

4.d:

Answer: For black jack case, after changing to a smaller threshold from 21 to 17, the optimal value become smaller. If change to a bigger threshold, the optimal reward remain the same.

Here is some stats after doing experiment, we found that the optimal value for MDP is 7.33, the expected reward of simulation for newThreshold MDP is 7.2; expected reward of simulation for newThreshold MDP is 8.4.

Some interpretation:
The optimal value of simulation result by using FixedRLAlgorithm for newThresholdMDP is very close to optimal value of originalMDP, the reason is FixedRLAlgorithm choose actions by using same optimal policy of originalMDP, so even it has higher threshold, it will has same result as originalMDP.

We could see optimal value of Q-learning is bigger than FixedRLAlgorithm and optimal value of originalMDP, the reason is Q-learning is an off-policy learning algorithm, it suppose to find bigger optimal value when threshold is bigger.