CS221 Winter - 2019 Homework 4

Name: Dat Nguyen
Date: 2/27/2019

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

Problem 1: Value Iteration

(a) The value for Q at iteration 1 is

$$Q^{(1)}(0,-1) = 0.8(-5 + 1 \times 0) + 0.2(-5 + 1 \times 0) = -5$$
$$Q^{(1)}(0,1) = 0.7(-5 + 1 \times 0) + 0.3(-5 + 1 \times 0) = -5$$

$$Q^{(1)}(1,-1) = 0.8(-5 + 1 \times 0) + 0.2(100 + 1 \times 0) = 16$$
$$Q^{(1)}(1,1) = 0.7(-5 + 1 \times 0) + 0.3(100 + 1 \times 0) = 26.5$$

$$Q^{(1)}(-1,-1) = 0.8(20 + 1 \times 0) + 0.2(-5 + 1 \times 0) = 15$$
$$Q^{(1)}(-1,1) = 0.7(20 + 1 \times 0) + 0.3(-5 + 1 \times 0) = 12.5$$

The value for Q at iteration 2 is

$$Q^{(2)}(0,-1) = 0.8(-5 + 1 \times 15) + 0.2(-5 + 1 \times 26.5) = 12.3$$
$$Q^{(2)}(0,1) = 0.7(-5 + 1 \times 15) + 0.3(-5 + 1 \times 26.5) = 13.45$$

$$Q^{(2)}(1,-1) = 0.8(-5 + 1 \times -5) + 0.2(100 + 1 \times 0) = 12$$
$$Q^{(2)}(1,1) = 0.7(-5 + 1 \times -5) + 0.3(100 + 1 \times 0) = 23$$

$$Q^{(2)}(-1,-1) = 0.8(20 + 1 \times 0) + 0.2(-5 - 5) = 14$$
$$Q^{(2)}(-1,1) = 0.7(20 + 1 \times 0) + 0.3(-5 - 5) = 11$$

From that we have the value of V in each iteration

iter/state	-2	-1	0	1	2
0	0	0	0	0	0
1	0	15	-5	26.5	0
2	0	14	13.45	23	0

(b) The resulting optimal policy π_{opt} is

$$\pi_{\text{opt}}(-1) = -1$$
$$\pi_{\text{opt}}(0) = 1$$
$$\pi_{\text{opt}}(1) = 1$$

Problem 2: Transforming MDPs

(b) We will calculate V_{opt} of every state by starting from the end state which has $V_{\text{opt}} = 0$ and going backward the topological order. Since at a considered state, V_{opt} of every state going from that state has already been calculated, we can calculate V_{opt} for this state.

(c) We make a new transition from every state to the end state o with probability $1 - \gamma$ and we define $T'(s, a, s') = T(s, a, s') * \gamma$ and $\text{Reward}'(s, a, s') = \frac{\text{Reward}(s,a,s')}{\gamma}$. Then for each iteration of value iteration $V_{\text{opt}}(s) = V'_{\text{opt}}(s)$ because

$$V_{\text{opt}}^{(t)}(s) = \max_{a \in \text{Action}(s)} \sum_{s'} T(s, a, s')[\text{Reward}(s, a, s') + \gamma V_{\text{opt}}^{t-1}(s')]$$

end

$$\begin{aligned}
V_{\text{opt}}^{\prime(t)}(s) &= \max_{a \in \text{Action}'(s)} \Big(\sum_{s'} T'(s, a, s')[\text{Reward}'(s, a, s') + V_{\text{opt}}^{\prime(t-1)}(s')] + (1 - \gamma) \times 0 \Big) \\
&= \max_{a \in \text{Action}(s)} \sum_{s'} \gamma T(s, a, s')[\frac{\text{Reward}(s, a, s')}{\gamma} + V_{\text{opt}}^{\prime(t-1)}(s')] \\
&= \max_{a \in \text{Action}(s)} \sum_{s'} T(s, a, s')[\text{Reward}(s, a, s') + \gamma V_{\text{opt}}^{\prime(t-1)}(s')]
\end{aligned}$$

Since we can initialize $V^{(0)}(s)$ and $V'^{(0)}(s)$ to be the same and the update rules in each iteration are the same, the convergence value of $V_{\text{opt}}(s)$ and $V'_{\text{opt}}(s)$ are also the same.

Problem 4: Learning to Play Blackjack

(b) Using identityFeatureExtractor() on smallMDP with 30000 trials yields 88.89% of actions which match that of value iteration. On the other hand, using identityFeatureExtractor() on largeMDP with 30000 trials yields only 67.21% of actions which match that of value iteration. I think the reason for this low performance is that using identityFeatureExptractor() requires learning separated value for each state/action pair which is quite large (over 8000 pairs) in the cases of largeMDP. Therefore 30000 trials might not be enough to learn accurately the q value.

(d) The expected reward of value iteration trained on originalMDP performing on newThresholdMDP is 6.59.

The expected reward corresponding to Q-learning is 12, which is better because the Q-learning adapts its optimal Q-value during the simulation for the newThresholdMDP problem while value iteration just keeps its optimal actions on the original MDP problem.