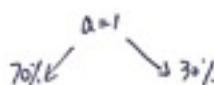
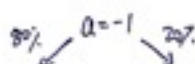


Song Han

1. Value iteration

states :



$V_{opt}^{(0)}$

0 0 0 0 0

$\pi_{opt}^{(0)}$

- - - - -

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

$$\begin{aligned} \text{①} \xrightarrow{a=-1} & 80\% (20+0) + 20\% (-5+0) = 15 \\ \text{①} \xrightarrow{a=1} & 70\% (20+0) + 30\% (-5+0) = 12.5 \end{aligned} \Rightarrow a = -1$$

$$\begin{aligned} \text{②} \xrightarrow{a=-1} & 80\% (-5+0) + 20\% (-5+0) = -5 \\ \text{②} \xrightarrow{a=1} & 70\% (-5+0) + 30\% (-5+0) = -5 \end{aligned} \Rightarrow a = -1 \text{ or } 1$$

$$\begin{aligned} \text{③} \xrightarrow{a=-1} & 80\% (-5+0) + 20\% (100+0) = 16 \\ \text{③} \xrightarrow{a=1} & 70\% (-5+0) + 30\% (100+0) = 26.5 \end{aligned} \Rightarrow a = 1$$

update

$V_{opt}^{(1)}$

0 15 -5 26.5 0

$\pi_{opt}^{(1)}$

- -1 -1 or +1 1 -

$$\begin{aligned} \text{①} \xrightarrow{a=-1} & 80\% (20+0) + 20\% (-5 + -5) = 14 \\ \text{①} \xrightarrow{a=1} & 70\% (20+0) + 30\% (-5 + 5) = 11 \end{aligned} \Rightarrow a = -1$$

$$\begin{aligned} \text{②} \xrightarrow{a=-1} & 80\% (-5+15) + 20\% (-5 + 26.5) = 12.3 \\ \text{②} \xrightarrow{a=1} & 70\% (-5+15) + 30\% (-5 + 26.5) = 13.45 \end{aligned} \Rightarrow a = 1$$

$$\begin{aligned} \text{③} \xrightarrow{a=-1} & 80\% (-5-5) + 20\% (0 + 100) = 12 \\ \text{③} \xrightarrow{a=1} & 70\% (-5-5) + 30\% (0 + 100) = 23 \end{aligned} \Rightarrow a = 1$$

update

$V_{opt}^{(2)}$

0 14 13.45 23 0

$\pi_{opt}^{(2)}$

- -1 1 1 -

4 (b)

For the small data set, only 1 out of the 27 states have different actions. The percentage of different items is 3.7%. The reinforcement learning performed pretty good.

For the large data set, 937 out of 2745 states have different actions. The percentage of different items is 34%. The reinforcement learning performed bad, because the large problem has far more states, and the identityFeatureExtractor is a bad choice, because the feature it generates is very sparse, and does not generalize. So the function approximation is bad

4 (d)

FixedRLAlgorithm with policy obtained from original MDP gets average reward of average of 6.84. However, when using Q-Learning, the average reward is 9.38. Q-learning is better result because it adapts to the new problem and adjusts the weights according to what it see, while FixedRLAlgorithm will stick to a fixed policy and can not adapt to the new problem.