RL assignment3 report

520030910342 Jiyu Liu

1. Policy Gradient

$$
\begin{aligned}
&\mathbb{E}[\frac{\partial \log \pi_\theta(a|s)}{\partial \theta}] \\
&\triangleq \sum_{s\in\mathcal{S}} \rho^{\pi_\theta}(s) \sum_{a\in\mathcal{A}} \pi_\theta(a|s) \frac{\partial \log \pi_\theta(a|s)}{\partial \theta} f(s) \\
&= \sum_{s\in\mathcal{S}} \rho^{\pi_\theta}(s) f(s) \sum_{a\in\mathcal{A}} \pi_\theta(a|s) \frac{\partial \log \pi_\theta(a|s)}{\partial \theta} \\
&= \sum_{s\in\mathcal{S}} \rho^{\pi_\theta}(s) f(s) \sum_{a\in\mathcal{A}} \frac{\partial \pi_\theta(a|s)}{\partial \theta} \\
&= \sum_{s\in\mathcal{S}} \rho^{\pi_\theta}(s) f(s) \frac{\partial \sum_{a\in\mathcal{A}} \pi_\theta(a|s)}{\partial \theta}
\end{aligned}
$$

(Number of possible actions are obviously finite)

$$
\begin{aligned}
&= \sum_{s\in\mathcal{S}} \rho^{\pi_\theta}(s) f(s) \frac{\partial 1}{\partial \theta} \\
&= 0
\end{aligned}
$$

2. Implementation of the Dyna-Q and Dyna-Q+ algorithms

(a) According to my observation, within the range of (0, 10, 50), the larger the number of planning steps(n_planning) are, the better the performance is. The reason of this phenomenon is that the model with larger n_planning can simulate more possible future states and actions from certain visited state and action, thus has more subtle and accurate adjustment to the Q-table, allowing the agent to take better decisions in next steps. The

(b) In the basic maze, Dyna-Q has a better performance than Dyna-Q+, the possible reason of which is that the basic maze is very naive and does not need much exploration, so the exploration-encouraging algorithm Dyna-Q+ cannot perform as well as simple Dyna-Q.

In the blocking maze, with the value of kappa equal to **0.001**, I find that Dyna-Q+ outperforms Dyna-Q when time is less than 17500 but the opposite is true when time is larger than 17500. The reason might be that when time is relatively low Dyna-Q+ can find the optimal path faster than Dyna-Q, but when time is large Dyna-Q+ may do some unnecessary exploration, which results in inferior cumulative reward. However, I also notice that with the value of kappa equal to **0.0001**, the cumulative reward of Dyna-Q+ is always greater than Dyna-Q, indicating that a moderate additional weight of unvisited states can lead to better performance of Dyna-Q+.

In the shortcut maze, with the value of kappa equal to **0.001**, once time is larger than 3000, Dyna-Q+ outperforms Dyna-Q in cumulative reward. When time is less than 3000, there is only one entrance to the terminal state, so the agent doesn't need to do a lot of exploration. However, after 3000 time steps, a shorter path is opened up and the agent needs more exploration to discover the shorter path. This is where the adventage of Dyna-Q+ comes into play.

3. Plot

The output images can be seen in the folder './plot', which has three corresponding subfolders named 'basic', 'blocking' and 'shortcut'.

If you have any question about my report, code or images, please contact me at any moment.