EE 232E Project 3
Reinforcement Learning and Inverse Reinforcement Learning

Hengjie Yang, Sheng Chang, Wandi Cui, and Tianyi Liu

May 19, 2018

1 Reinforcement Learning (RL)

In this project, we implement the RL algorithm to explore its performance.

Question 1: (10 points) For visualization purpose, generate heat maps of Reward function 1 and Reward function 2. For the heat maps, make sure you display the coloring scale. You will have 2 plots for this question

The heat map of reward function 1 and 2 are plotted in Fig. 1 and 2, respectively.

1.1 Optimal policy learning using RL algorithms

Question 2: (40 points) Create the environment of the agent using the information provided in section 2. To be specific, create the MDP by setting up the state-space, action set, transition probabilities, discount factor, and reward function. For creating the environment, use the following set of parameters:

- Number of states = 100 (state space is a 10 by 10 square grid as displayed in figure 1)

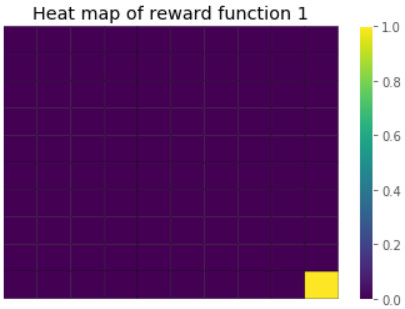

Figure 1: Heat map of reward function 1

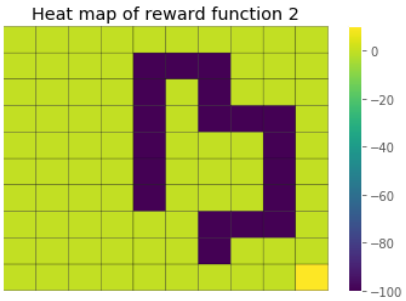

Figure 2: Heat map of reward function 2

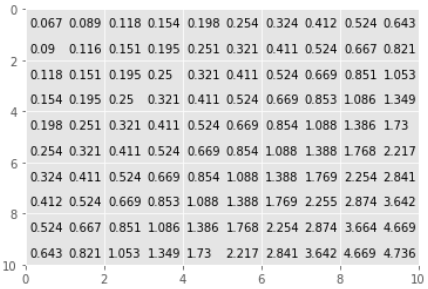

Figure 3: The optimal state value with reward function 1

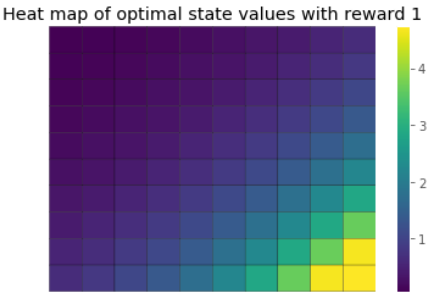

Figure 4: Heat map of optimal state values with reward function 1

- Number of actions = 4 (set of possible actions is displayed in figure 2)
- $w = 0.1$
- Discount factor $\gamma = 0.8$
- Reward function 1

After you have created the environment, then write an optimal state-value function that takes as input the environment of the agent and outputs the optimal value of each state in the grid. For the optimal state-value function, you have to implement the Initialization (lines 2-4) and Estimation (lines 5-13) steps of the Value Iteration algorithm. For the estimation step, use $\epsilon = 0.01$. For visualization purpose, you should generate figure similar to that of figure 1 but with the number of state replaced by the optimal value of that state. In this question, you should have 1 plot.

After implementing the RL algorithm, we obtain the grid of optimal state values in Fig. 3.

Question 3: (5 points) Generate a heat map of the optimal state values across the 2-D grid. For generating the heat map, you can use the same function provided in the hint earlier (see the hint after question 1).

After obtaining the optimal state value array with reward function 1, the corresponding heat map of it is shown in Fig. 4.

Question 4: (15 points) Explain the distribution of the optimal state values across the 2-D grid. (Hint: Use the figure generated in question 3 to explain)

Since we observe from reward function 1 that the reward function is symmetric with respect to the diagonal. So according to the RL algorithm, it follows that the optimal state value should also be symmetric with respect to the diagonal. Also, the value increases as they become close to the $(9, 9)$ state.

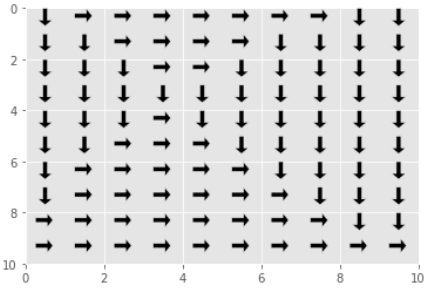

Figure 5: Optimal actions with reward function 1

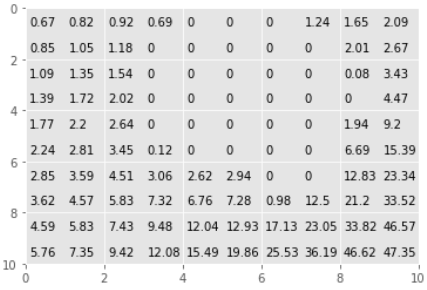

Figure 6: The optimal state value with reward function 2

Question 5: (30 points) Implement the computation step of the value iteration algorithm (lines 14-17) to compute the optimal policy of the agent navigating the 2-D state-space. For visualization purpose, you should generate a figure similar to that of figure 1 but with the number of state replaced by the optimal action at that state. The optimal actions should be displayed using arrows. Does the optimal policy of the agent match your intuition? Please provide a brief explanation. Is it possible for the agent to compute the optimal action to take at each state by observing the optimal values of it's neighboring states? In this question, you should have 1 plot.

After implementing the computation step, the optimal actions at each state are shown in Fig. 5. The optimal policy of the agent matches the intuition since the action is always towards the highest score. Yes, the optimal action is always towards the neighbor with the highest state value. Therefore, we can determine the optimal action by simply observing the values in the neighboring states.

Question 6: (10 points) Modify the environment of the agent by replacing Reward function 1 with Reward function 2. Use the optimal state-value function implemented in question 2 to compute the optimal value of each state in the grid. For visualization purpose, you should generate a figure similar to that of figure 1 but with the number of state replaced by the optimal value of that state. In this question, you should have 1 plot.

After replacing with reward function 2, the optimal value at each state is shown in Fig. 6.

Question 7: (10 points) Generate a heat map of the optimal state values (found in question 6) across the 2-D grid. For generating the heat map, you can use the same function provided in the hint earlier.

The heat map of the optimal state values is shown in Fig. 7.

Question 8: (20 points) Explain the distribution of the optimal state values across the 2-D grid. (Hint: Use the figure generated in question 7 to explain)

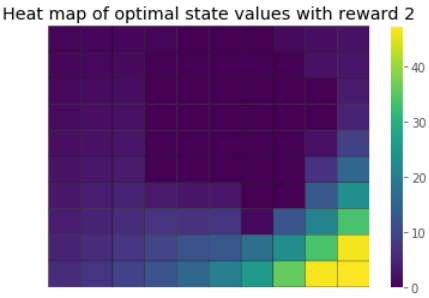

Figure 7: Heat map of optimal state values with reward function 2

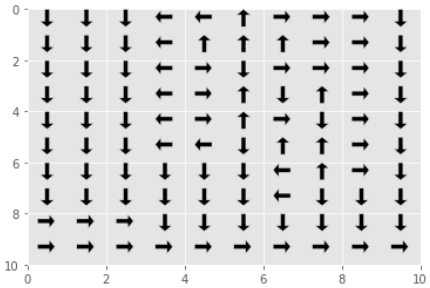

Figure 8: Optimal actions with reward function 2

The distribution of the optimal state values is that, in the area with the reward score of -100, the optimal state values are zero. In the area with the reward score of 0, the optimal state values are positive, and are increasing as they become close to the $(9, 9)$ state which has the highest reward score of 10. The $(9, 9)$ state has the highest state value as it has the highest reward score.

Question 9: (20 points) Implement the computation step of the value iteration algorithm (lines 14-17) to compute the optimal policy of the agent navigating the 2-D state-space. For visualization purpose, you should generate a figure similar to that of figure 1 but with the number of state replaced by the optimal action at that state. The optimal actions should be displayed using arrows. Does the optimal policy of the agent match your intuition? Please provide a brief explanation. In this question, you should have 1 plot.

After implementing the computation step, the optimal actions with reward function 2 are shown in Fig. 8. The optimal action policy matches the intuition as the actions are still towards the neighboring state with the highest state value.

2 Inverse Reinforcement Learning (IRL)

2.1 IRL algorithm

2.2 Performance measure