EE 232E Project 3
Reinforcement Learning and Inverse Reinforcement Learning

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

May 21, 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)

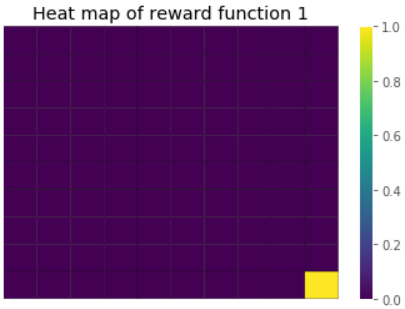

Figure 1: Heat map of reward function 1

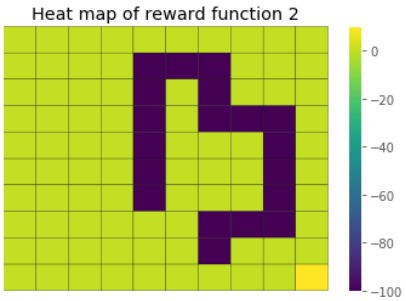

Figure 2: Heat map of reward function 2

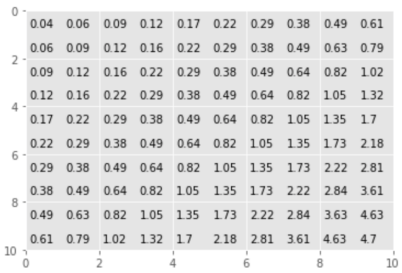

Figure 3: The optimal state value with reward function 1

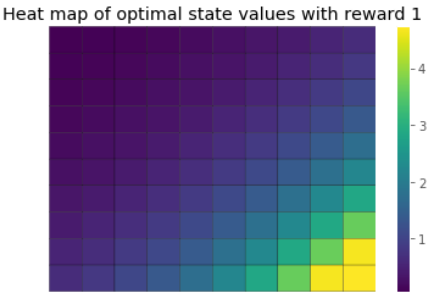

Figure 4: Heat map of optimal state values with reward function 1

- 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.

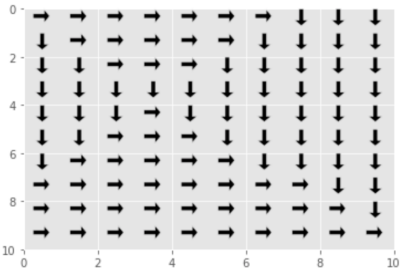

Figure 5: Optimal actions with reward function 1

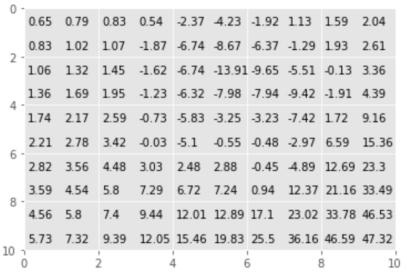

Figure 6: The optimal state value with reward function 2

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)

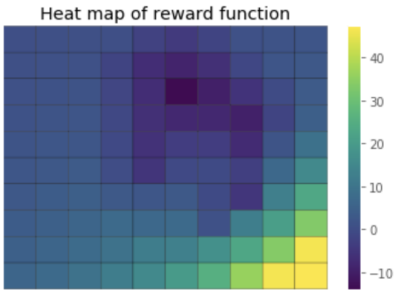

Figure 7: Heat map of optimal state values with reward function 2

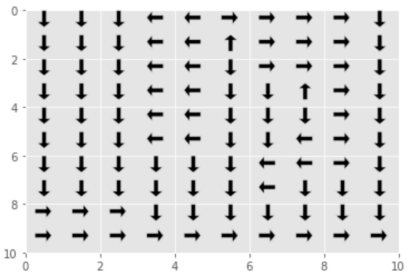

Figure 8: Optimal actions with reward function 2

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

Question 10: Express c, x, D in terms of R, P_a, P_{a_1} , t_i, u, λ and R_{max}

To recast the equation into block matrices, let $T = [t_1, t_2, ..., t_{|s|}]^T$, $U = [u_1, u_2, ..., u_{|s|}]^T$, $R = [R(s_1), ..., R(s_{|s|})]^T$.

Then the cost function can be expressed as $[0, I, \lambda * I, 0] * [\mathrm{R,T,U}]$

$$\max_x \begin{bmatrix} 0 & 1^T & -\lambda 1^T \end{bmatrix} \begin{bmatrix} R \\ T \\ U \end{bmatrix}$$

$$st: \begin{bmatrix} -(P_{a_1}-P_a)(I-\gamma P_{a_1})^{-1} & I & 0 \\ -(P_{a_1}-P_a)(I-\gamma P_{a_1})^{-1} & 0 & 0 \\ -I & 0 & -I \\ I & 0 & -I \\ I & 0 & 0 \\ -I & 0 & 0 \end{bmatrix} \begin{bmatrix} R \\ T \\ U \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ R_{max} \\ R_{max} \end{bmatrix}$$

Thus,

$$D = \begin{bmatrix} -(P_{a_1}-P_a)(I-\gamma P_{a_1})^{-1} & I & 0 \\ -(P_{a_1}-P_a)(I-\gamma P_{a_1})^{-1} & 0 & 0 \\ -I & 0 & -I \\ I & 0 & -I \\ I & 0 & 0 \\ -I & 0 & 0 \end{bmatrix}$$

$$x = \begin{bmatrix} R \\ T \\ U \end{bmatrix}$$

$$c = \begin{bmatrix} 0 \\ 1 \\ \lambda 1 \end{bmatrix}$$

2.2 Performance measure

Question 11: (30 points) Sweep λ from 0 to 5 to get 500 evenly spaced values for λ. For each value of λ compute $O_A(s)$ by following the process described above. For this problem, use the optimal policy of the agent found in question 5 to fill in the $O_E(s)$ values. Then use equation 3 to compute the accuracy of the IRL algorithm for this value of λ. You need to repeat the above process for all 500 values of λ to get 500 data points. Plot λ (x-axis) against Accuracy (y-axis). In this question, you should have 1 plot.

The plot of λ against Accuracy is shown in Fig. 9.

Question 12: (5 points) Use the plot in question 11 to compute the value of λ for which accuracy is maximum. For future reference we will denote this value as λ_{max}. Please report λ_{max}

From the plot, we can see the max accuracy is 0.75, and there are several λ which can reach the maximum accuracy. One of them is 0.

Question 13: for λ_{max}, generate heat maps of the ground truth reward and the extracted reward. Please note that the ground truth reward is the Reward function 1 and the extracted

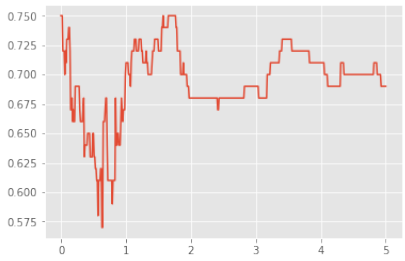

Figure 9: The plot of Accuracy with λ

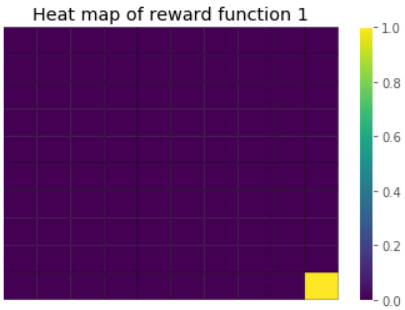

Figure 10: Heat map of ground truth reward

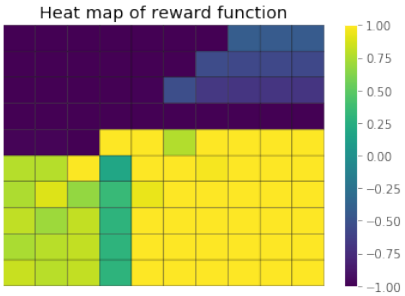

Figure 11: Heat map of extracted reward

reward is computed by solving the linear program given by equation 2 with the λ parameter set to λ_{max}. In this question, you should have 2 plots.

The heat map of ground truth reward and extracted reward are shown in Fig. 10 and Fig. 11, respectively.

Question 14: Use the extracted reward function computed in question 13, to compute the optimal values of the states in the 2-D grid. For computing the optimal values you need to use the optimal state-value function that you wrote in question 2. For visualization purpose, generate a heat map of the optimal state values across the 2-D grid (similar to the figure generated in question 3). In this question, you should have 1 plot.

The heat map of optimal values of the states computed from extracted reward function is shown in Fig. 12.

Question 15: (10 points) Compare the heat maps of Question 3 and Question 14 and provide a brief explanation on their similarities and differences.

In comparison to heat maps of optimal values from result of the RL algorithms, the corresponding heat maps of the result of IRL algorithm has the same tendency of value

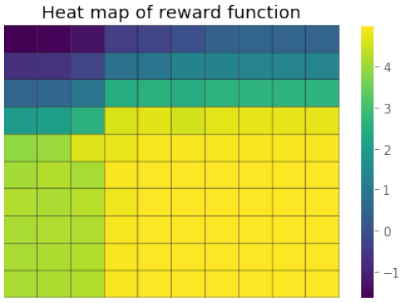

Figure 12: The heat map of optimal values

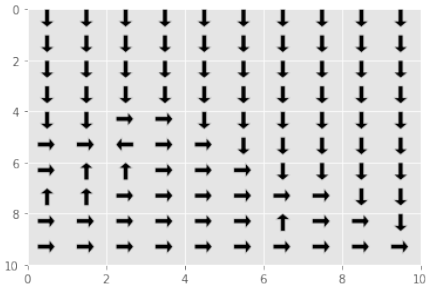

Figure 13: The arrows plot of optimal policy

distribution. We can see the top left corner's optimal values is the lowest and the bottom right corner has the highest value in both map, which is consistent with the value distribution of reward function. However, due to the difference of reward function they derived from, the result of RL algorithm has a much smoother transition from the top left corner to the bottom right corner and also is diagonal symmetric.

Question 16: Use the extracted reward function found in question 13 to compute the optimal policy of the agent. For computing the optimal policy of the agent you need to use the function that you wrote in question 5. 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 actions should be displayed using arrows. In this question, you should have 1 plot.

The arrows plot of optimal policy computed from extracted reward function is shown in Fig. 13.

Question 17: (10 points) Compare the figures of Question 5 and Question 16 and provide a brief explanation on their similarities and differences.

In conclusion, both figures have shown that the overall optimal policy is to make action starting from the top left corner eventually stop at the bottom right corner and most states on the right have optimal policy of going down and for states on the bottom they mostly aim to go right. On the other side, we can also that there are a few states of result of IRL algorithm has the opposite direction, which obviously is a mistake.

Question 18: (30 points) Sweep λ from 0 to 5 to get 500 evenly spaced values for λ. For each value of λ compute $O_A(s)$ by following the process described above. For this problem, use the optimal policy of the agent found in question 9 to fill in the $O_E(s)$ values. Then use equation 3 to compute the accuracy of the IRL algorithm for this value of λ. You need

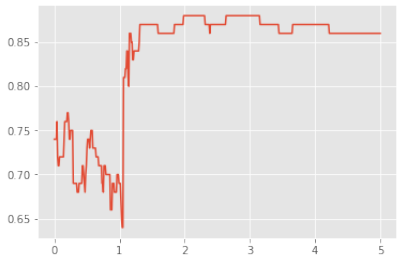

Figure 14: The plot of Accuracy with λ

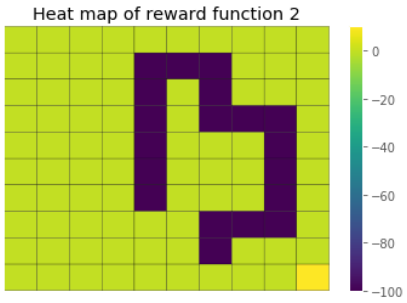

Figure 15: Heat map of ground truth reward

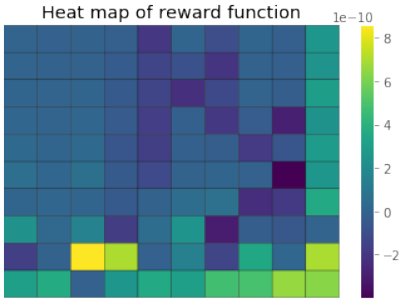

Figure 16: Heat map of extracted reward

to repeat the above process for all 500 values of λ to get 500 data points. Plot λ (x-axis) against Accuracy (y-axis). In this question, you should have 1 plot.

The plot of λ against Accuracy is shown in Fig. 14.

Question 19: (5 points) Use the plot in question 18 to compute the value of λ for which accuracy is maximum. For future reference we will denote this value as λ_{max}. Please report λ_maxmax

From the plot, we can see the max accuracy is 0.88, and there are several λ which can reach the maximum accuracy. One of them is 1.98397.

Question 20: for λ_{max}, generate heat maps of the ground truth reward and the extracted reward. Please note that the ground truth reward is the Reward function 2 and the extracted reward is computed by solving the linear program given by equation 2 with the λ parameter set to λ_{max}. In this question, you should have 2 plots.

The heat map of ground truth reward and extracted reward are shown in Fig. 15 and Fig. 16, respectively.

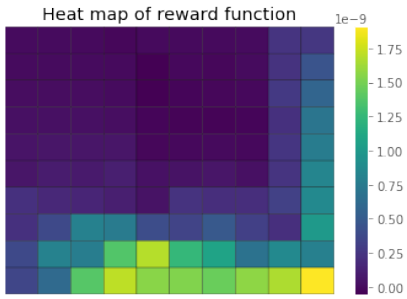

Figure 17: The heat map of optimal values

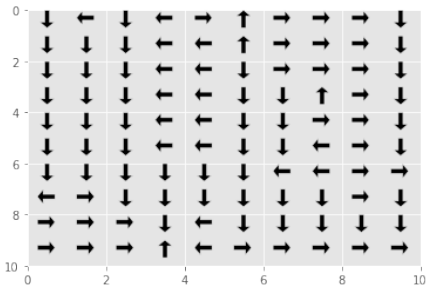

Figure 18: The arrows plot of optimal policy

Question 21: Use the extracted reward function computed in question 20, to compute the optimal values of the states in the 2-D grid. For computing the optimal values you need to use the optimal state-value function that you wrote in question 2. For visualization purpose, generate a heat map of the optimal state values across the 2-D grid (similar to the figure generated in question 7). In this question, you should have 1 plot.

The heat map of optimal values of the states computed from extracted reward function is shown in Fig. 17.

Question 22: (10 points) Compare the heat maps of Question 7 and Question 21 and provide a brief explanation on their similarities and differences.

Similarly with the previous result, the overall value distribution on the both map is pretty close. What we can observe from both optimal values and reward function is that the lowest optimal value exist on the states which have the lowest reward function and the same situation for highest optimal value. Unfortunately, in the result of IRL algorithm, the bottom left corner's value and the top right corner's value are a little bit higher than the top left corner's value, which should be the same.

Question 23: Use the extracted reward function found in question 20 to compute the optimal policy of the agent. For computing the optimal policy of the agent you need to use the function that you wrote in question 9. 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 actions should be displayed using arrows. In this question, you should have 1 plot.

The arrows plot of optimal policy computed from extracted reward function is shown in Fig. 18.

Question 24: (10 points) Compare the figures of Question 9 and Question 23 and provide

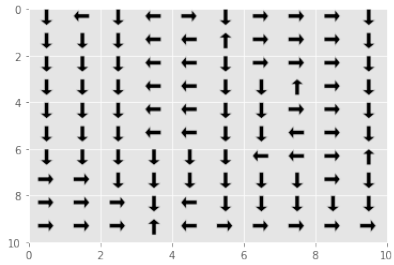

Figure 19: The plot of new optimal policy

a brief explanation on their similarities and differences.

Surprisingly, the similarity of those two figures is higher than the similarity of previous ones, even though the reward function is much more complex. For most states, they have the exactly the same optimal policy. Only some states on the edge of the map of on the cross of some arrow flow have some weird optimal policy, which will be discussed in the next question in detail.

Question 25: (50 points) From the figure in question 23, you should observe that the optimal policy of the agent has two major discrepancies. Please identify and provide the causes for these two discrepancies. One of the discrepancy can be fixed easily by a slight modification to the value iteration algorithm. Perform this modification and rerun the modified value iteration algorithm to compute the optimal policy of the agent. Also, recompute the maximum accuracy after this modification. Is there a change in maximum accuracy? The second discrepancy is harder to fix and is a limitation of the simple IRL algorithm. If you can provide a solution to the second discrepancy then we will give you a bonus of 50 points.

Discrepancy1: Observing the differences between results we got using two methods, we found that IRL sometimes generates the optimal policy that may lead the agent get out of the map before it reaches the destination (9,9) state.

Solution: So, in order to avoid this discrepancy, we add a boundary judgement condition function called judge boundary to original algorithm. Then we re-run the previous process, the new optimal policy plot is shown in Fig. 19. We can clearly see that those mistake arrows have gone. There are no optimal policy that could let the agent get out of the map except the end. And we also recompute the accuracy and we found that it doesn't change.

Discrepancy2: We also found that the optimal action we got for neighbor states may directly opposite to each other. In other words, agent may keep walking through only two states several times over and over again. The reason why this happens might be the algorithm can only get the local optimal solution.

Solution: We may need to add more constrains in the LP problem to restrict the condition to prevent sucking in the local optimal.