ECE232 Project3
Reinforcement learning and Inverse Reinforcement learning

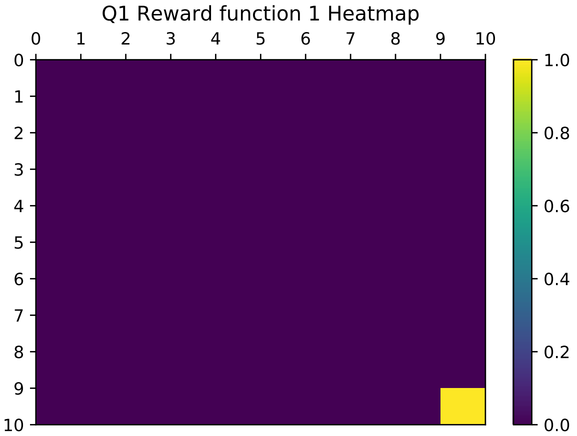
Juo-Wen Fu 805025223
Alex Hung 705035114
Te-Yuan Liu 805027255
Wesly Tzuo 704946416

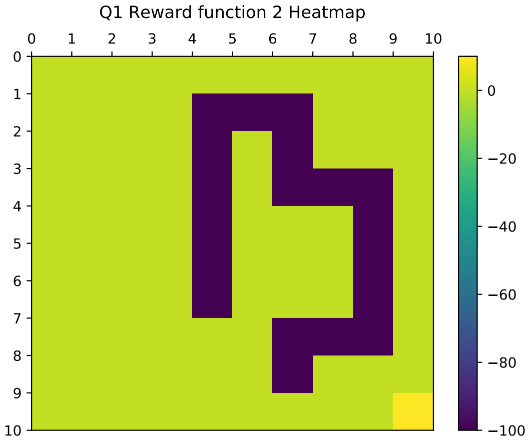
May 22, 2018

1 Introduction

Question 1: Generate heat maps of Reward function 1 and Reward function 2

In this question, we can use *pcolor* function to assign colors to different values automatically. In addition, we can use *colorbar* function so that we can observer the color and its corresponding value.

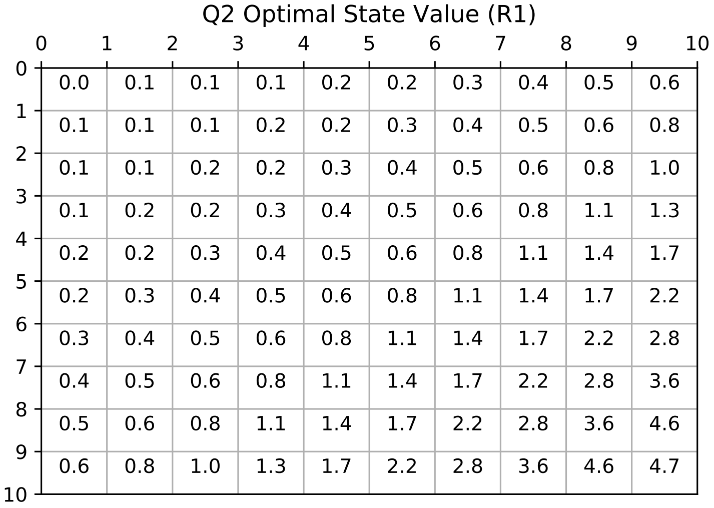




Question 2: create the MDP with following parameters:
Number of State = 100
Number of Action = 4
w = 0.1
Discount Factor = 0.8
Reward Function 1
After you have created the environment, generate a figure with the number of state replaced by the optimal value of that state using Value Iteration algorithm.

Here we implement the value iteration algorithm.
We can observe from the algorithm that the reward of a state will ”spread” to its neighbors based on the transition probabilities (multiply by a discount factor). Since Reward function 1 only has

one positive value on the bottom-right corner, we can imagine that the optimal state value will also have highest value on the bottom-right corner.
As you can see from the figure, the bottom-right corner has the highest value



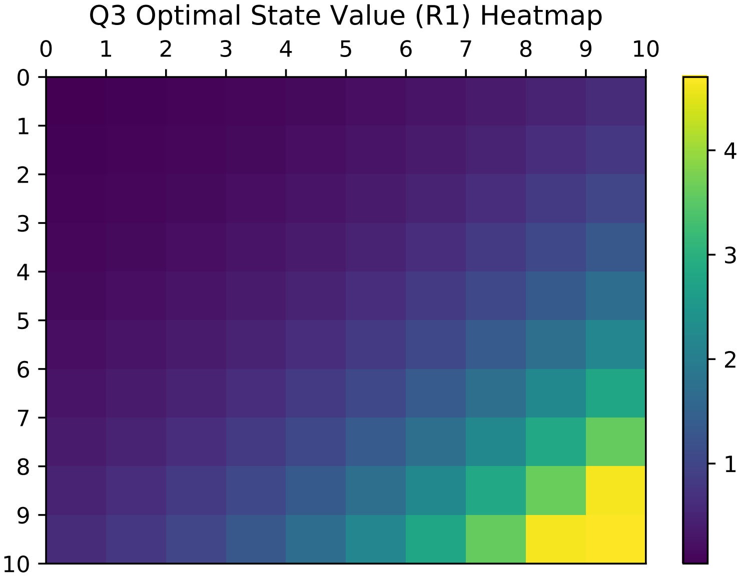
Q2 Optimal State Value (R1)

0
1
2
3
4
5
6
7
8
9
10

0.0	0.1	0.1	0.1	0.2	0.2	0.3	0.4	0.5	0.6
0.1	0.1	0.1	0.2	0.2	0.3	0.4	0.5	0.6	0.8
0.1	0.1	0.2	0.2	0.3	0.4	0.5	0.6	0.8	1.0
0.1	0.2	0.2	0.3	0.4	0.5	0.6	0.8	1.1	1.3
0.2	0.2	0.3	0.4	0.5	0.6	0.8	1.1	1.4	1.7
0.2	0.3	0.4	0.5	0.6	0.8	1.1	1.4	1.7	2.2
0.3	0.4	0.5	0.6	0.8	1.1	1.4	1.7	2.2	2.8
0.4	0.5	0.6	0.8	1.1	1.4	1.7	2.2	2.8	3.6
0.5	0.6	0.8	1.1	1.4	1.7	2.2	2.8	3.6	4.6
0.6	0.8	1.0	1.3	1.7	2.2	2.8	3.6	4.6	4.7

Question 3: Generate a heat map of the optimal state values across the 2-D grid.

Same as question 1, we can use *pcolor* function to assign colors to different values automatically and *colorbar* function to show the colors and its corresponding value.
Compare to the result we get in question 2, this figure allows us to observe the distribution of the optimal state values across the 2-D grid more easily.



Q3 Optimal State Value (R1) Heatmap

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

As we already mentioned in the question 2. We can observe from the value iteration algorithm that the reward of a state will ”spread” to its neighbors based on the transition probabilities (multiply by a discount factor). Since Reward function 1 only has one positive value on the bottom-right corner and the transition probabilities from neighbors are roughly equal, the optimal state value should also have highest value on the bottom-right corner and keeps decreasing as the distance increases.

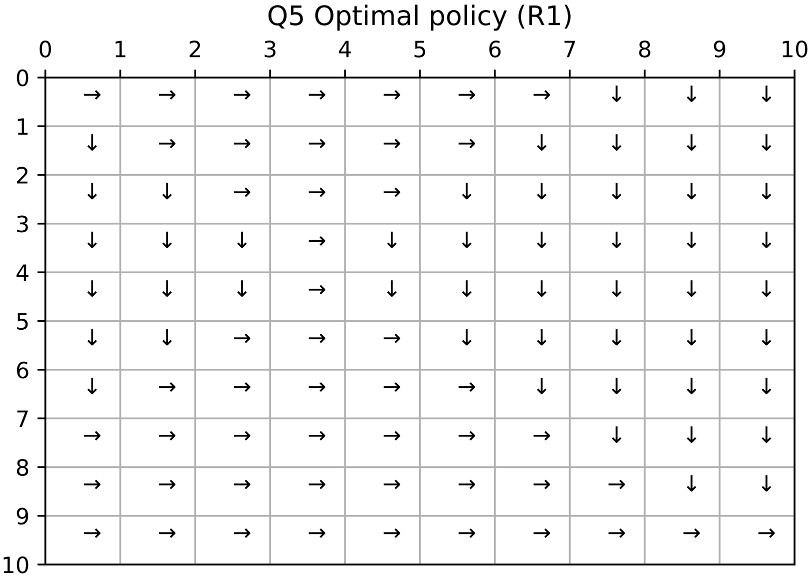
Question 5: Compute the optimal policy of the agent navigating the 2-D state-space. Generate a figure with the number of state replaced by the optimal action at that state. The optimal actions should be displayed using arrows. Does the optimal policy of the agent match your intuition? Please provide 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?

We can observe from computation step that the optimal policy of a state is based on the Reward and the discounted state value of its neighbor (assume the transition probabilities are equal). From this intuition, the optimal policy of any state should be either ”Right” or ”Down” (except the bottom-right corner) since there is no ”obstacle” on the 2-D grid.

According to these observations and the figure we get in question 3, we can conclude that:

1. The optimal policy of the agent match our intuition.

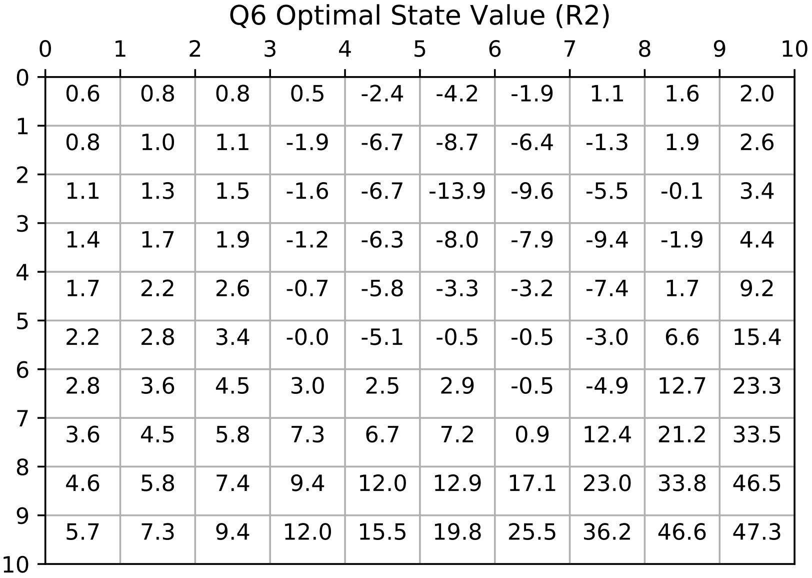
2. Yes. from the result we got in question 2 we can observe that the optimal action of a state fits the optimal values of it's neighboring states. That is, the optimal action is toward to the neighbor with the highest state value.



→	→	→	→	→	→	→	↓	↓	↓
↓	→	→	→	→	→	↓	↓	↓	↓
↓	↓	→	→	→	↓	↓	↓	↓	↓
↓	↓	↓	→	↓	↓	↓	↓	↓	↓
↓	↓	↓	→	↓	↓	↓	↓	↓	↓
↓	↓	→	→	→	↓	↓	↓	↓	↓
↓	→	→	→	→	→	↓	↓	↓	↓
→	→	→	→	→	→	→	↓	↓	↓
→	→	→	→	→	→	→	→	↓	↓
→	→	→	→	→	→	→	→	→	→

Question 6: Modify the environment of the agent by replacing Reward function 1 with Reward function 2. Generate a figure with the number of state replaced by the optimal value of that state using Value Iteration algorithm.

Here we use the same value iteration algorithm as in question 2. Except that we change the reward function 1 to reward function 2.



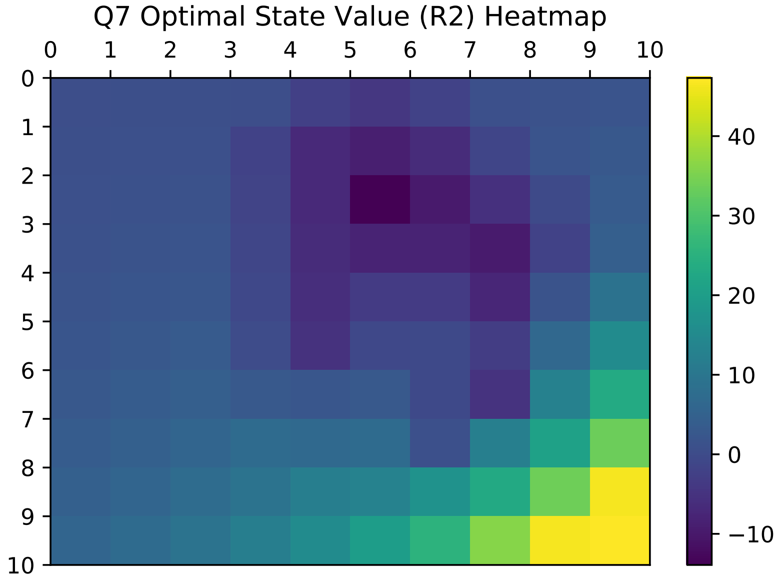
Q6 Optimal State Value (R2)

0
1
2
3
4
5
6
7
8
9
10

0.6	0.8	0.8	0.5	-2.4	-4.2	-1.9	1.1	1.6	2.0
0.8	1.0	1.1	-1.9	-6.7	-8.7	-6.4	-1.3	1.9	2.6
1.1	1.3	1.5	-1.6	-6.7	-13.9	-9.6	-5.5	-0.1	3.4
1.4	1.7	1.9	-1.2	-6.3	-8.0	-7.9	-9.4	-1.9	4.4
1.7	2.2	2.6	-0.7	-5.8	-3.3	-3.2	-7.4	1.7	9.2
2.2	2.8	3.4	-0.0	-5.1	-0.5	-0.5	-3.0	6.6	15.4
2.8	3.6	4.5	3.0	2.5	2.9	-0.5	-4.9	12.7	23.3
3.6	4.5	5.8	7.3	6.7	7.2	0.9	12.4	21.2	33.5
4.6	5.8	7.4	9.4	12.0	12.9	17.1	23.0	33.8	46.5
5.7	7.3	9.4	12.0	15.5	19.8	25.5	36.2	46.6	47.3

Question 7: Generate a heat map of the optimal state values (found in question 6) across the 2-D grid.

Same as question 1, we can use *pcolor* function to assign colors to different values automatically and *colorbar* function to show the colors and its corresponding value.
Compare to the result we get in question 2, this figure allows us to observe the distribution of the optimal state values across the 2-D grid more easily.



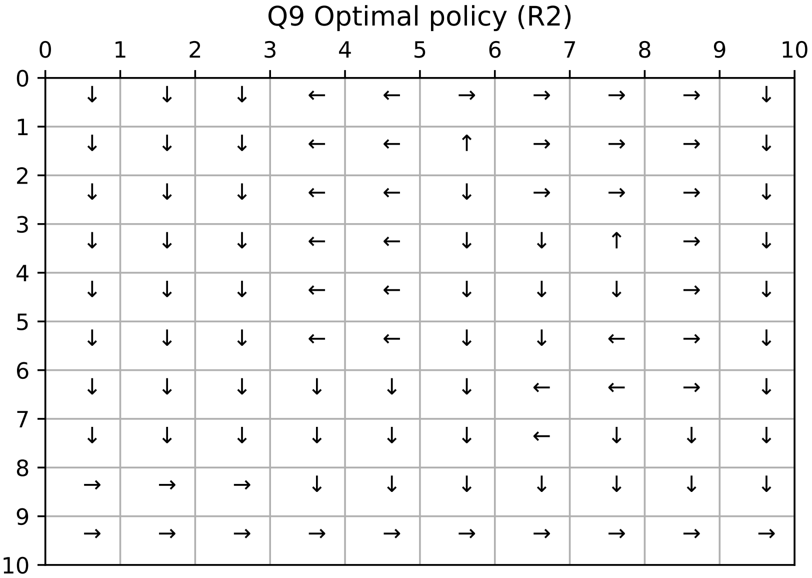
Question 8: Explain the distribution of the optimal state values across the 2-D grid. (Hint: Use the figure generated in question 7 to explain)

As we mention in question 4, We can observe from the value iteration algorithm that the reward of a state will ”spread” to its neighbors based on the transition probabilities (multiply by a discount factor). According to this observation, we can imagine that the state values around the *walls* (negative rewards) will be smaller. The remaining part will be similar to the state values with reward function 1 since we only have one positive value on the bottom-right corner.

Question 9: Compute the optimal policy of the agent navigating the 2-D state-space. Generate a figure 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.

Same as question 5, we can observe from computation step that the optimal policy of a state is based on the Reward and the discounted state value of its neighbor (assume the transition probabilities are equal).
From this intuition, the optimal policy should be:
1. For those states *within* the *walls* (negative rewards) area, the optimal policy should toward to the *gap* on the bottom.
2. For those states *on* the walls, the optimal policy should point away from the walls area.
3. For the remaining states, the optimal policy should be ”Right” or ”Down” as in question 5.

The figure shows that the optimal policy we get basically matches our intuition.



Question 10: Express $\mathbf{c}, \mathbf{x}, \mathbf{D}$ in terms of R, $\mathbf{P}_a$, $\mathbf{P}_{a1}$, t_i, u, and R_{max}

The LP formulation can be recast into

$$\begin{aligned} \underset{\mathbf{x}}{\text{maximize}} \quad & \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{D}\mathbf{x} \preceq 0, \forall a \in \mathcal{A} \setminus a_1 \end{aligned}$$

if we set

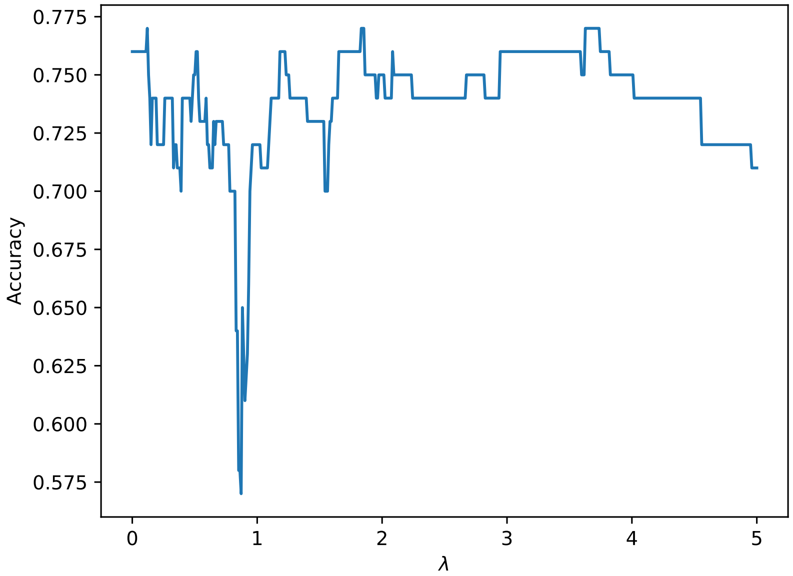
$$\mathbf{c} = \begin{bmatrix} \mathbf{0}_{|\mathcal{S}|\times 1} \\ \mathbf{I}_{|\mathcal{S}|\times 1} \\ -\lambda_{|\mathcal{S}|\times 1} \\ \mathbf{0}_{|\mathcal{S}|\times 1} \end{bmatrix}$$

$$\mathbf{x} = \begin{bmatrix} \mathbf{R} \\ \mathbf{T} \\ \mathbf{u} \\ \mathbf{R}_{max} \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} -(\mathbf{P}_{a1}(i) - \mathbf{P}_a(i))(\mathbf{I} - \gamma\mathbf{P}_{a1})^{-1} & \mathbf{I}_{|\mathcal{S}|} & \mathbf{0} & \mathbf{0} \\ -(\mathbf{P}_{a1} - \mathbf{P}_a)(\mathbf{I} - \gamma\mathbf{P}_{a1})^{-1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{I}_{|\mathcal{S}|} & \mathbf{0} & -\mathbf{I}_{|\mathcal{S}|} & \mathbf{0} \\ \mathbf{I}_{|\mathcal{S}|} & \mathbf{0} & -\mathbf{I}_{|\mathcal{S}|} & \mathbf{0} \\ -\mathbf{I}_{|\mathcal{S}|} & \mathbf{0} & \mathbf{0} & -\mathbf{I}_{|\mathcal{S}|} \\ \mathbf{I}_{|\mathcal{S}|} & \mathbf{0} & \mathbf{0} & -\mathbf{I}_{|\mathcal{S}|} \end{bmatrix}$$

where $\mathbf{T}$ is a vector of t_i and $\mathcal{R}_{max}$ is a vector of size $|\mathbf{S}| \times 1$ and elements all with values R_{max}

Question 11: 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.

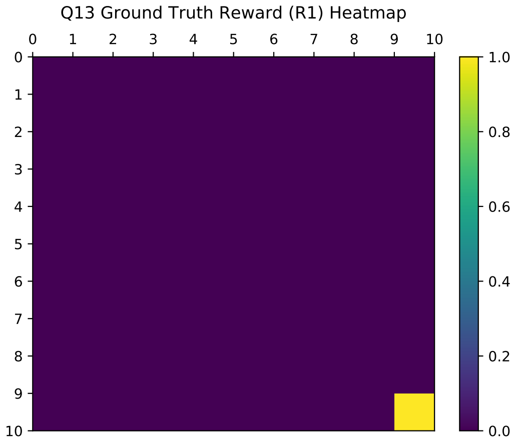


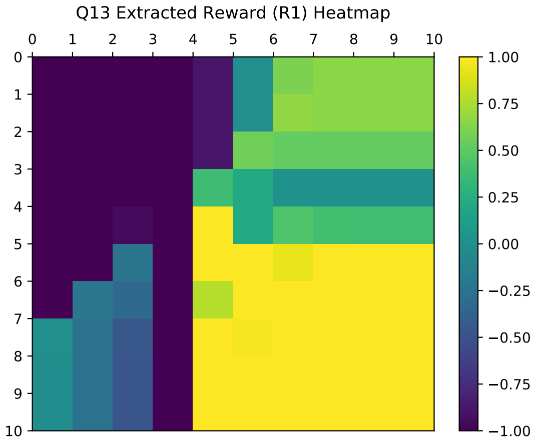
We tune λ between 0 to 5 and get 500 accuracies. As the graph shows, most of the accuracies fall between 0.700 and 0.775.

Question 12: 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 $\lambda_{max}^{(1)}$. Please report $\lambda_{max}^{(1)}$

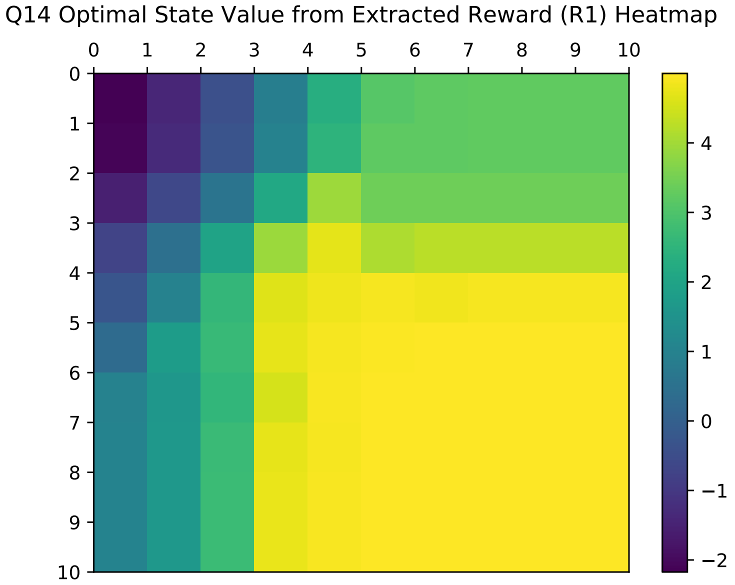
$\lambda_{max}^{(1)} = 0.120240480962$.

Question 13: For $\lambda_{max}^{(1)}$, 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 reward is computed by solving the linear program given by equation 2 with the λ parameter set to $\lambda_{max}^{(1)}$. In this question, you should have 2 plots.

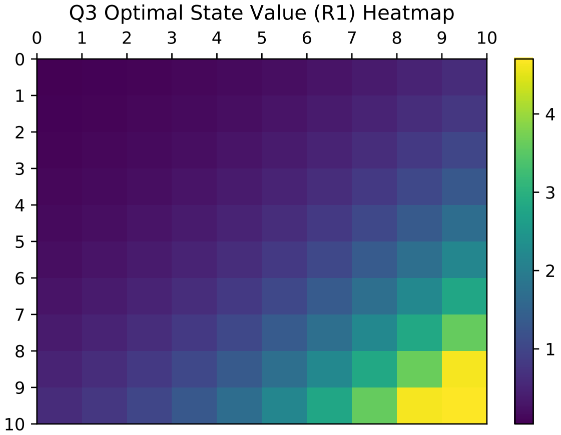


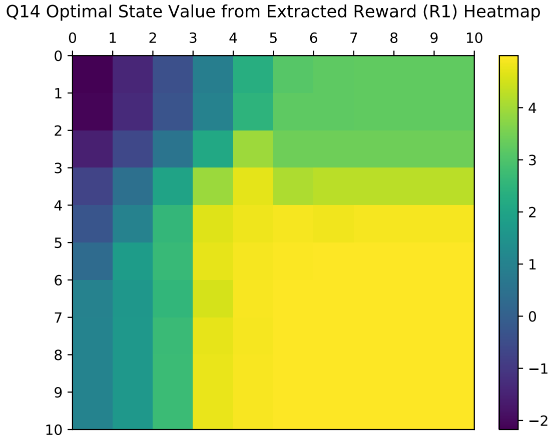


Question 14: Use the extracted reward function computed in question 13, to compute the optimal values of the states in the 2-D grid. For computing the optimal values you need to use the optimal state-value function that you wrote in question 2. For visualization purpose, generate a heat map of the optimal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



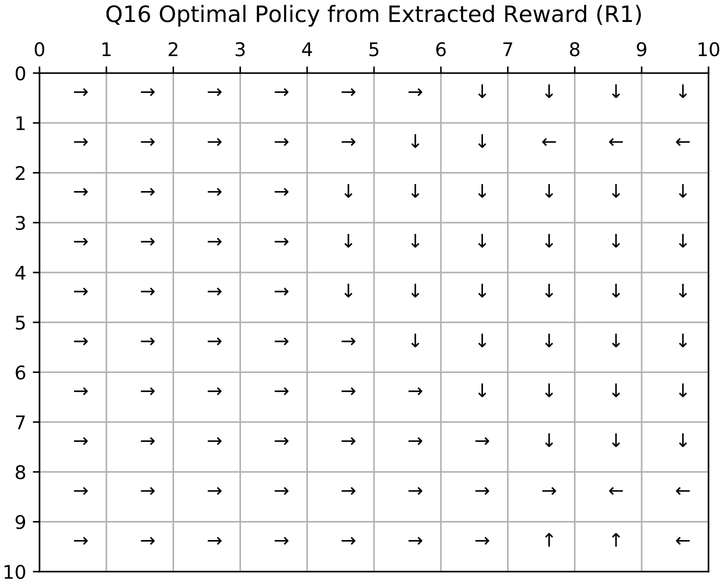
Question 15: Compare the heat maps of Question 3 and Question 14 and provide a brief explanation on their similarities and differences.


Q3 Optimal State Value (R1) Heatmap


Q14 Optimal State Value from Extracted Reward (R1) Heatmap

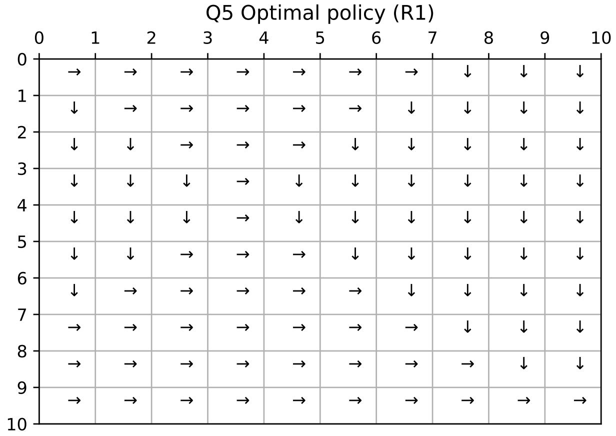
Both heat maps in question 3 and question 14 have higher values in square grids near the bottom-right corner and have lower values in squares near the upper-lefter corner. This is because the ground truth has reward value of 1 in (9,9) and the extracted reward have higher value near bottom-left corner. The differences is, question 14 has a larger region of higher optimal state values. This is also because the extracted reward has more square grids with value larger than 0.

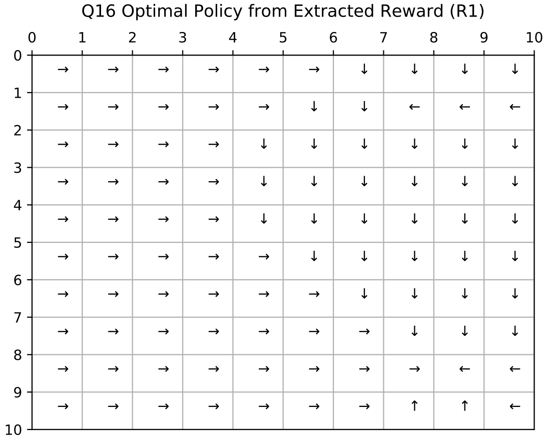
Question 16: Use the extracted reward function found in question 13 to compute the optimal policy of the agent. For computing the optimal policy of the agent you need to use the function that you wrote in question 5. For visualization purpose, you should generate a figure similar to that of figure 1 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.


Q16 Optimal Policy from Extracted Reward (R1)

→	→	→	→	→	→	↓	↓	↓	↓
→	→	→	→	→	↓	↓	←	←	←
→	→	→	→	↓	↓	↓	↓	↓	↓
→	→	→	→	↓	↓	↓	↓	↓	↓
→	→	→	→	↓	↓	↓	↓	↓	↓
→	→	→	→	→	↓	↓	↓	↓	↓
→	→	→	→	→	→	↓	↓	↓	↓
→	→	→	→	→	→	→	↓	↓	↓
→	→	→	→	→	→	→	→	←	←
→	→	→	→	→	→	→	↑	↑	←

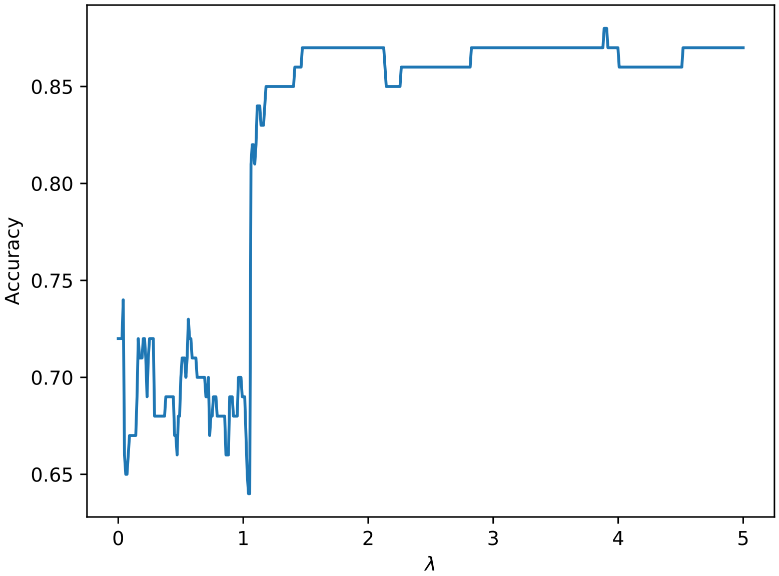
Question 17: Compare the figures of Question 5 and Question 16 and provide a brief explanation on their similarities and differences.





The similar part of question 5 and question 16 is that most of their policies are downwards and rights. This because the most of the larger extracted values are near the bottom-right corner. The difference between question 5 and question 16 is that in question 5, all of the policies are downs and rights while in question 16, there are 6 lefts and two ups. This is because not only does (9, 9) has a reward value of 1 but also other the square grids near the bottom-right have value larger than 0. If we look in detail, most of the different directions appear near the boundary of the non-zero value region. Thus, the agent might not only moves towards (9,9), but also moves toward other squares.

Question 18: 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 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.

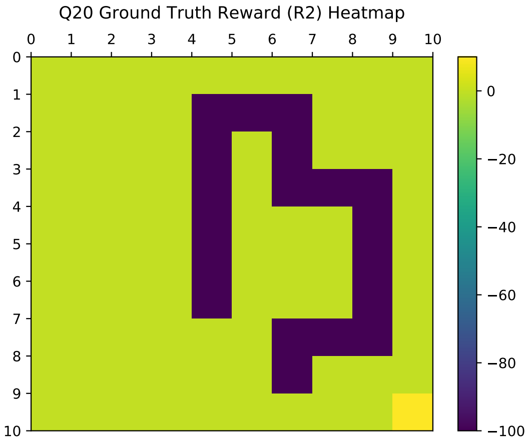


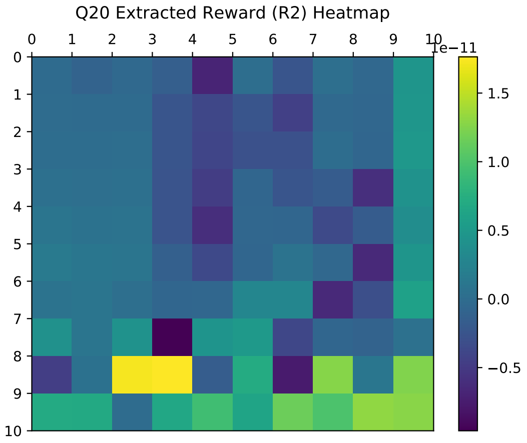
We tune λ between 0 to 5 and get 500 accuracies. As the graph shows, most of the accuracies are greater than 0.85 for $\lambda > 1.5$.

Question 19: 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 $\lambda_{max}^{(2)}$. Please report $\lambda_{max}^{(2)}$

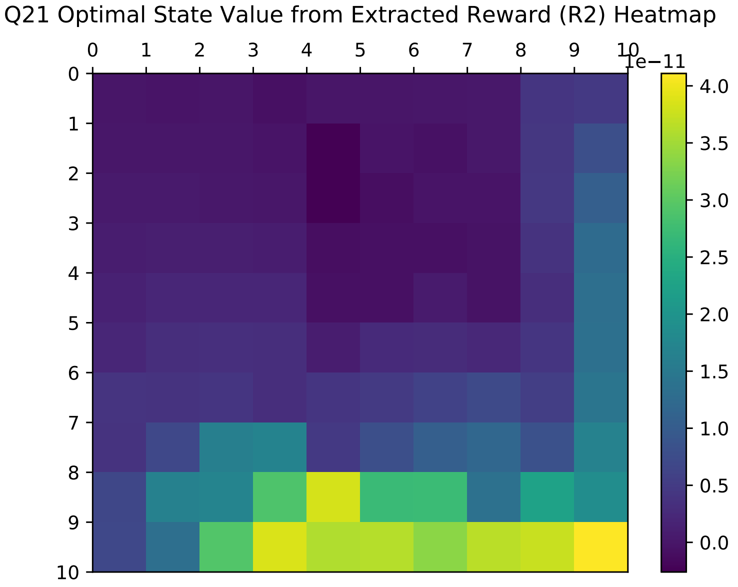
$\lambda_{max}^{(2)} = 3.8877755511$.

Question 20: For $\lambda_{max}^{(2)}$, 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 $\lambda_{max}^{(2)}$. In this question, you should have 2 plots.

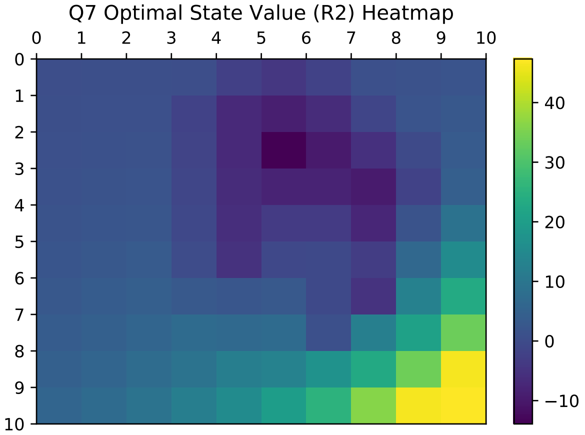


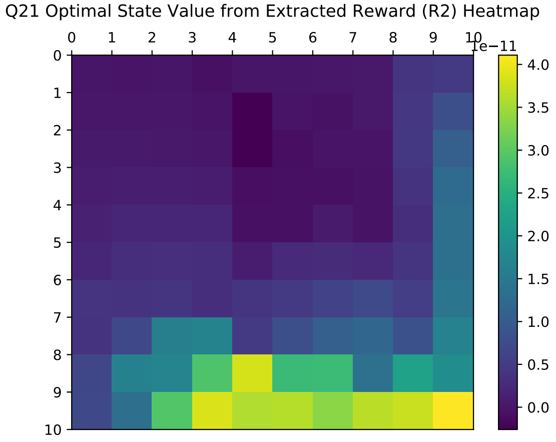


Question 21: Use the extracted reward function computed in question 20, to compute the optimal values of the states in the 2-D grid. For computing the optimal values you need to use the optimal state-value function that you wrote in question 2. For visualization purpose, generate a heat map of the optimal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



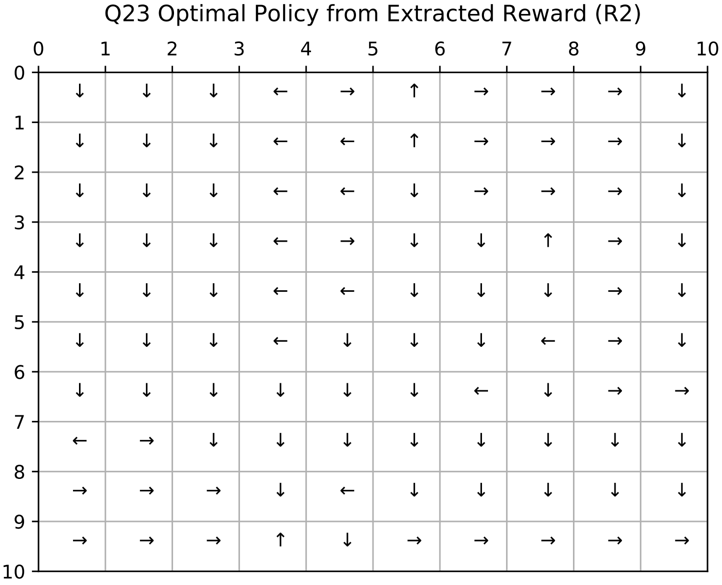
Question 22: Compare the heat maps of Question 7 and Question 21 and provide a brief explanation on their similarities and differences.


Q7 Optimal State Value (R2) Heatmap


Q21 Optimal State Value from Extracted Reward (R2) Heatmap

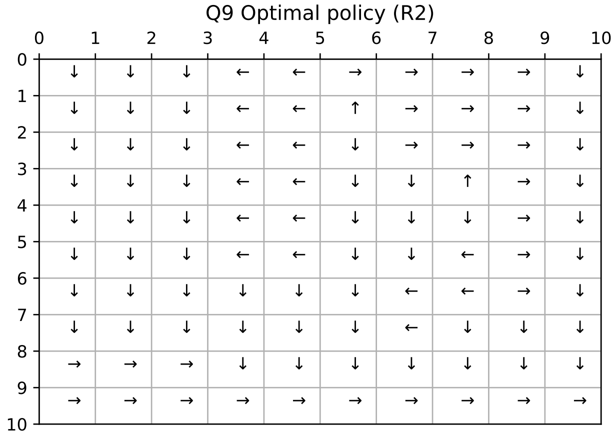
Both heat maps have higher optimal state value in the square grids near the lower right corner and lower optimal state values around grid (2,5). This is because the ground truth reward and the extracted reward both have high values of reward at the bottom right corner and low values of reward in upper middle right part. The major difference between the two heat maps is that the heat map from Q21 have some high values near the bottom middle left. This comes from the fact that the extracted reward shown in Q20 have the highest values at grid (8,2) and (8,3). Furthermore, the grid with lower state values are less pronounced in the heat map of Q21 than that of Q7. This can be seen from that the low values of the extracted reward are not significantly lower than the rest compare to the ground truth reward. Finally, the magnitude of the optimal state values from extracted reward are lower. This is because the average magnitude of the extracted reward is lower than the ground truth reward.

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.

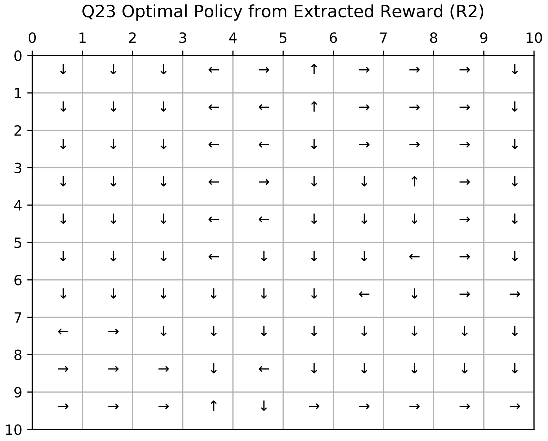

Q23 Optimal Policy from Extracted Reward (R2)

↓	↓	↓	←	→	↑	→	→	→	↓
↓	↓	↓	←	←	↑	→	→	→	↓
↓	↓	↓	←	←	↓	→	→	→	↓
↓	↓	↓	←	→	↓	↓	↑	→	↓
↓	↓	↓	←	←	↓	↓	↓	→	↓
↓	↓	↓	←	↓	↓	↓	←	→	↓
↓	↓	↓	↓	↓	↓	←	↓	→	→
←	→	↓	↓	↓	↓	↓	↓	↓	↓
→	→	→	↓	←	↓	↓	↓	↓	↓
→	→	→	↑	↓	→	→	→	→	→

Question 24: Compare the figures of Question 9 and Question 23 and provide a brief explanation on their similarities and differences.


Q9 Optimal policy (R2)

↓	↓	↓	←	←	→	→	→	→	↓
↓	↓	↓	←	←	↑	→	→	→	↓
↓	↓	↓	←	←	↓	→	→	→	↓
↓	↓	↓	←	←	↓	↓	↑	→	↓
↓	↓	↓	←	←	↓	↓	↓	→	↓
↓	↓	↓	←	←	↓	↓	←	→	↓
↓	↓	↓	↓	↓	↓	←	←	→	↓
↓	↓	↓	↓	↓	↓	←	↓	↓	↓
→	→	→	↓	↓	↓	↓	↓	↓	↓
→	→	→	→	→	→	→	→	→	→

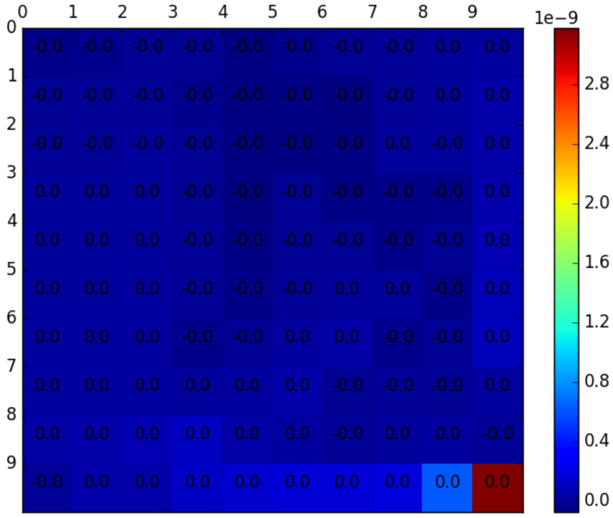

Q23 Optimal Policy from Extracted Reward (R2)

↓	↓	↓	←	→	↑	→	→	→	↓
↓	↓	↓	←	←	↑	→	→	→	↓
↓	↓	↓	←	←	↓	→	→	→	↓
↓	↓	↓	←	→	↓	↓	↑	→	↓
↓	↓	↓	←	←	↓	↓	↓	→	↓
↓	↓	↓	←	↓	↓	↓	←	→	↓
↓	↓	↓	↓	↓	↓	←	↓	→	→
←	→	↓	↓	↓	↓	↓	↓	↓	↓
→	→	→	↓	←	↓	↓	↓	↓	↓
→	→	→	↑	↓	→	→	→	→	→

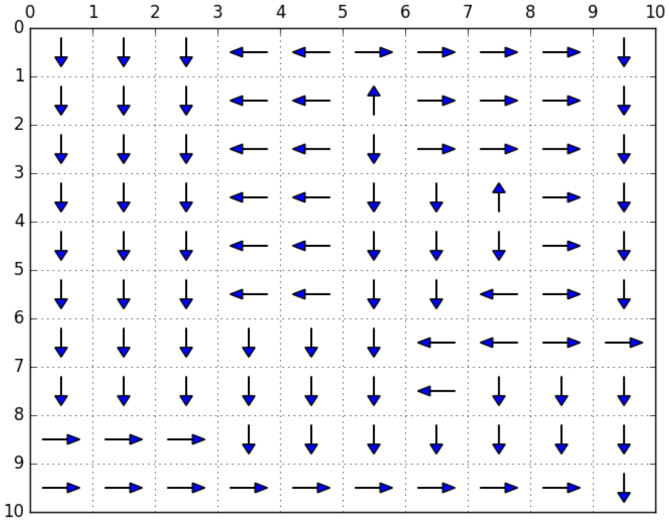
The similar part of question 9 and question 23 is that most of their policies are downwards and rights. This is because the highest optimal state value is at the lower right corner. One difference is that near the state with high extracted reward, some of the policy computed from extracted reward is reverse compare to that from the policy computed from the ground truth reward (e.g. grid (8,4) and (9,3)) because the grids with high extracted reward at grid (8,2) and (8,3) attracts the agent to move toward them. Finally, some of the policy for the boundary states from the extracted reward is telling the agent to stay on the same grid (i.e. move off the grid), which does not occur for the policy from the ground truth reward.

Question 25: 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 re-run 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.

The two optimal policy graphs have 13 differences in action, which happen at state 7, 10, 17, 39, 40, 45, 48, 49, 50, 67, 74, 76, 96. We can classify them into two discrepancy categories, one is moving towards the state with -100 reward, and the other one is not moving towards those states with negative reward. Actions in state 45, 67, 74, and 76 lead to states with negative reward while others do not. We think that the unexpected behavior of moving comes from the misleading extracted reward heat map shown before and the lack of updates in the value iteration algorithm due to the fixed threshold epsilon. The state 99 possesses the highest reward in the ground truth reward heat map of reward function 2, but it is obvious that the state 99 in the extracted reward heat map does not possess the highest reward, and we have come up an idea to fix this issue. We propose to modify the discount factor value from 0.8 to 0.2 in both inverse reinforcement learning procedure and the value iteration algorithm. The modification comes from the thought that extraction of reward function cares more about the immediate state reward so a lower discount factor value can enhance this goal. The maximum accuracy rises from 0.87 to 0.99 with lamda equals 2.054.
The updated extracted reward heat map is shown below.



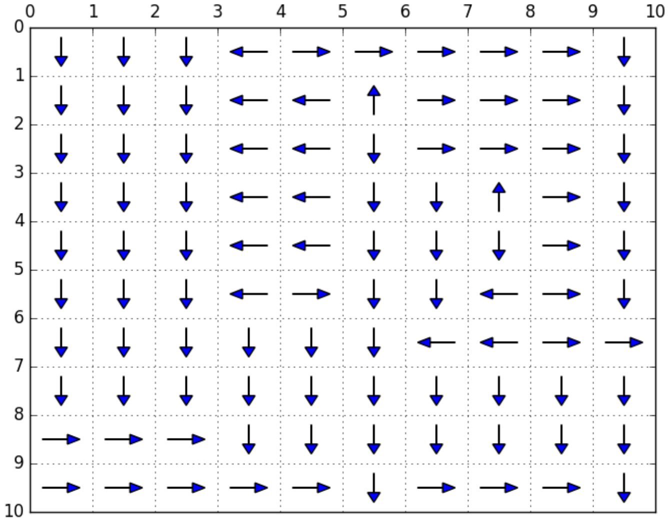
The optimal policy of the agent is shown below.



It is apparent that the updated extracted reward heat map is closer to the ground truth reward heat map and the accuracy is improved. Therefore, we have verified our thoughts.

The other issue is the lack of updates in the value iteration algorithm because epsilon is much larger than delta according to our observation and there is no update. To remedy this shortage, we modify epsilon to be 0.05 multiplying the maximum absolute value on the reward map. We fix the discount factor value to 0.5 and conduct experiment to verify our thoughts. The maximum accuracy rises from 0.91 to 0.95.

The updated optimal policy of the agent is shown below.



The updated version makes improvement. Therefore, we have verified our thoughts.