ECE 232E

Spring 2019

# Project 3 Reinforcement Learning

Kehkashan Sadiq Fazal
(305220750)

Joanna Itzel Navarro
(303861189)

Raja Mathanky Sankaranarayanan
(705227740)

Zhijie Yao
(404843943)

June 1, 2019

## 2 Reinforcement learning (RL)

**QUESTION 1: 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 two heat maps obtained are shown below for the two Reward Functions:

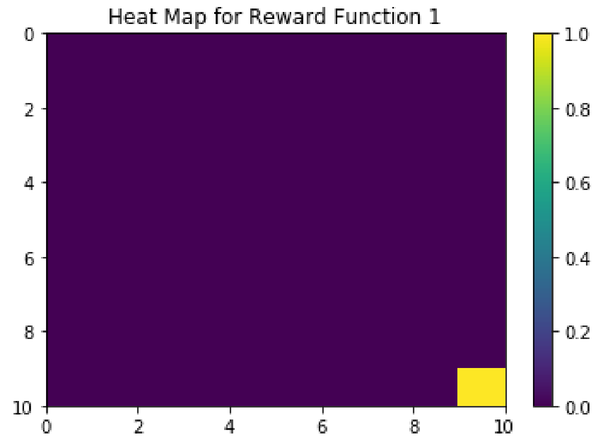


Figure 1: Heat Map for Reward Function 1

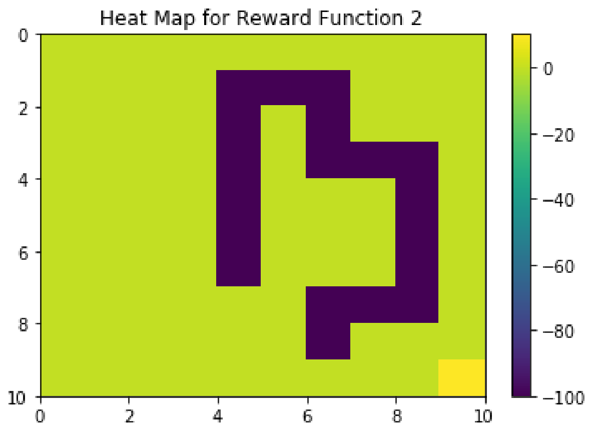


Figure 2: Heat Map for Reward Function 2

## 3 Optimal policy learning using RL algorithms

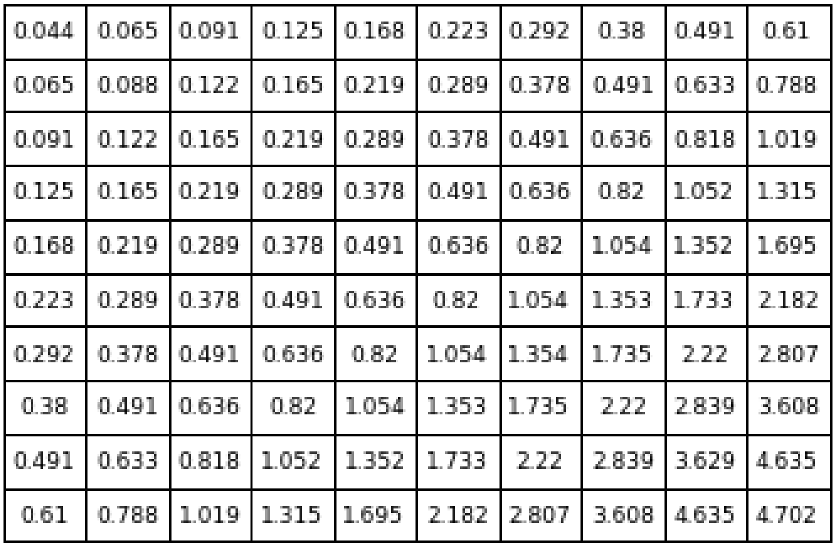
**QUESTION 2: Create the environment of the agent using the information provided in section 2. To be specific, create the MDP by setting up the state-space, action set, transition probabilities, discount factor, and reward function. For creating the environment, use the following set of parameters:**

- **Number of states = 100 (state space is a 10 by 10 square grid as displayed in figure 1)**
- **Number of actions = 4 (set of possible actions is displayed in figure 2)**
- **w = 0.1**
- **Discount factor = 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 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.**

With the given environment, we tried to realize the initialization and estimation part for the value iteration procedure. We then created a table to demonstrate the optimal value of each state. The table is shown below:



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.044 | 0.065 | 0.091 | 0.125 | 0.168 | 0.223 | 0.292 | 0.38 | 0.491 | 0.61 |
| 0.065 | 0.088 | 0.122 | 0.165 | 0.219 | 0.289 | 0.378 | 0.491 | 0.633 | 0.788 |
| 0.091 | 0.122 | 0.165 | 0.219 | 0.289 | 0.378 | 0.491 | 0.636 | 0.818 | 1.019 |
| 0.125 | 0.165 | 0.219 | 0.289 | 0.378 | 0.491 | 0.636 | 0.82 | 1.052 | 1.315 |
| 0.168 | 0.219 | 0.289 | 0.378 | 0.491 | 0.636 | 0.82 | 1.054 | 1.352 | 1.695 |
| 0.223 | 0.289 | 0.378 | 0.491 | 0.636 | 0.82 | 1.054 | 1.353 | 1.733 | 2.182 |
| 0.292 | 0.378 | 0.491 | 0.636 | 0.82 | 1.054 | 1.354 | 1.735 | 2.22 | 2.807 |
| 0.38 | 0.491 | 0.636 | 0.82 | 1.054 | 1.353 | 1.735 | 2.22 | 2.839 | 3.608 |
| 0.491 | 0.633 | 0.818 | 1.052 | 1.352 | 1.733 | 2.22 | 2.839 | 3.629 | 4.635 |
| 0.61 | 0.788 | 1.019 | 1.315 | 1.695 | 2.182 | 2.807 | 3.608 | 4.635 | 4.702 |

Figure 3: Table representing the optimal value of each state

**QUESTION 3: 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).**

The generated heat map is as shown below for the optimal state values:

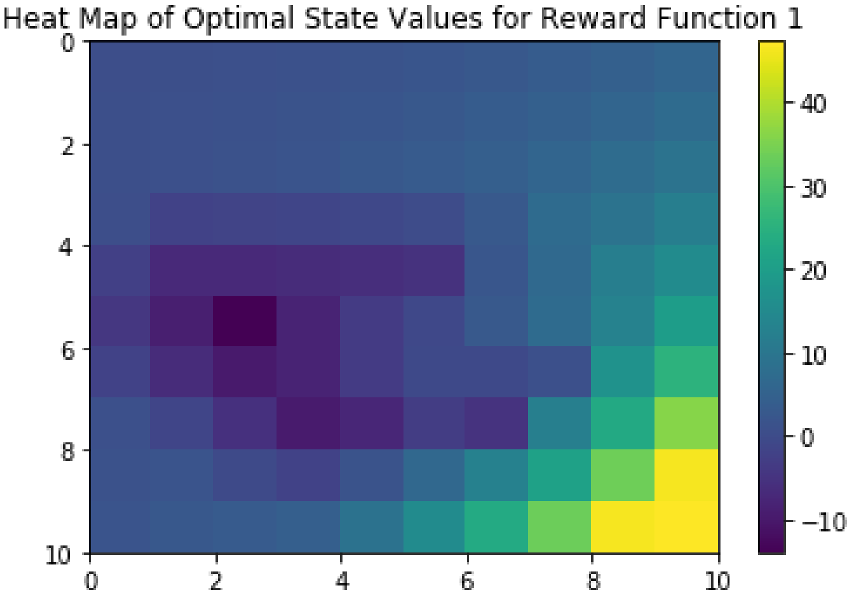


Figure 4: Heat Map for Optimal State of values across 2D grid

**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)**

To explain the distribution of optimal state values, we first need to understand the correlation. We see that the higher optimal state value a state has, the higher is the reward that it can obtain. This occurrence of optimal state values representation in can be seen in Figure 4, which we will use as a reference point for our discussion.

The initial reward function only has value on the last state (number of state = 99). According to the update function for values in the estimation step, we can find that each state value is influenced by its 4 neighbors (and itself). Therefore, the values are impacted accordingly, again as can be seen from Figure 4.

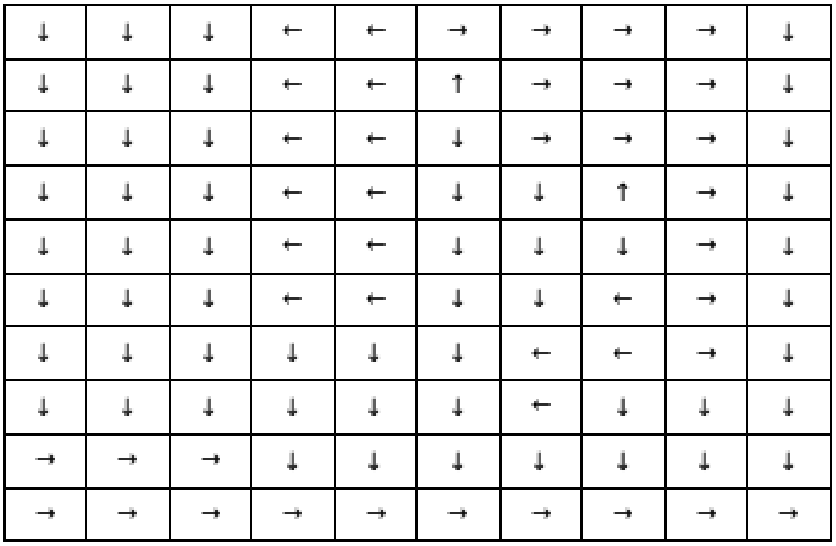
On the other hand, due to the discount factor = 0.8 in the same function, there exists a decay from one state to its neighbor states. Therefore, the final optimal state has comparably low value for states on the top-left part of the 2D grid.

The distribution in the 2-D map indicates that the further away the state is from the reward, the smaller the value will be. This is consistent with the idea of a value representing the expected cumulative discounted reward.

**QUESTION 5 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.**

The Optimal Action representationis as shown below:



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ↓ | ↓ | ↓ | ← | ← | → | → | → | → | ↓ |
| ↓ | ↓ | ↓ | ← | ← | ↑ | → | → | → | ↓ |
| ↓ | ↓ | ↓ | ← | ← | ↓ | → | → | → | ↓ |
| ↓ | ↓ | ↓ | ← | ← | ↓ | ↓ | ↑ | → | ↓ |
| ↓ | ↓ | ↓ | ← | ← | ↓ | ↓ | ↓ | → | ↓ |
| ↓ | ↓ | ↓ | ← | ← | ↓ | ↓ | ← | → | ↓ |
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ← | ← | → | ↓ |
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ← | ↓ | ↓ | ↓ |
| → | → | → | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| → | → | → | → | → | → | → | → | → | → |

Figure 5: Optimal action for each state

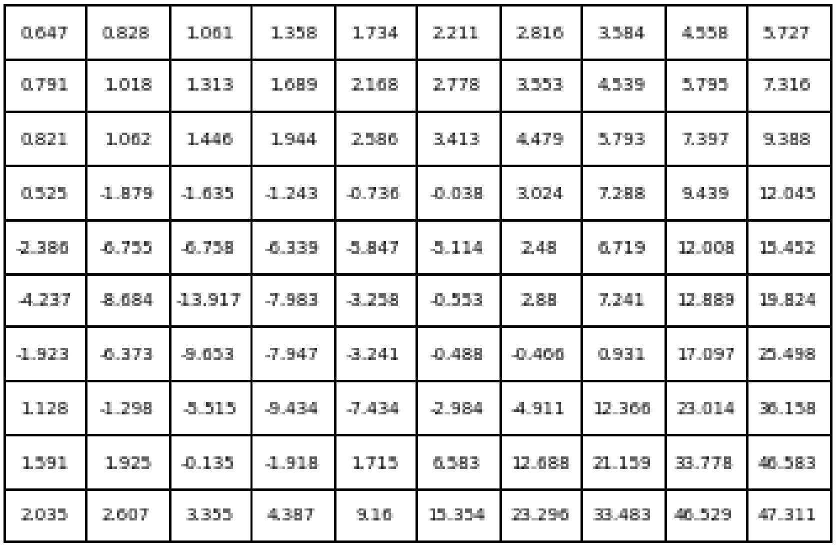
This optimal policy of the agent matches our intuition. The upper states from the table tend to take optimal actions to go down, and the left states from the table tend to go right. We can see from Figure 5 that every state follows this form. This can also be seen from the heatmap generated in Figure 4, the arrows correspond to being directed to the hottest areas which is essentially the most optimal route.

The reason for this form can be described as: since the last state has the highest optimal value, all the other states tend to take actions to come nearer to the last state in order to get a higher reward.

Therefore, it is possible for the agent to compute the optimal action to take by observing the optimal values of its neighboring states.

**QUESTION 6 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.**

The optimal value representation for Reward Function 2 is as shown below:

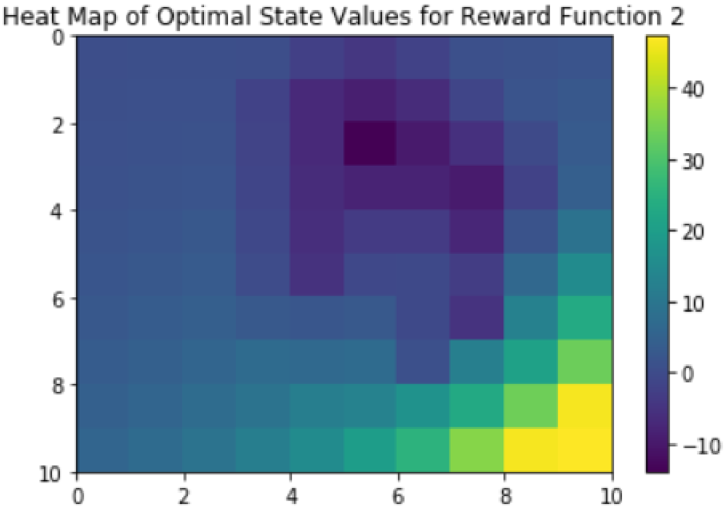


| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.647 | 0.828 | 1.061 | 1.358 | 1.734 | 2.211 | 2.816 | 3.584 | 4.558 | 5.727 |
| 0.791 | 1.018 | 1.313 | 1.689 | 2.168 | 2.778 | 3.553 | 4.539 | 5.795 | 7.316 |
| 0.821 | 1.062 | 1.446 | 1.944 | 2.586 | 3.413 | 4.479 | 5.793 | 7.397 | 9.388 |
| 0.525 | -1.879 | -1.635 | -1.243 | -0.736 | -0.038 | 3.024 | 7.288 | 9.439 | 12.045 |
| -2.386 | -6.755 | -6.758 | -6.339 | -5.847 | -5.114 | 2.48 | 6.719 | 12.008 | 15.452 |
| -4.237 | -8.684 | -13.917 | -7.983 | -3.258 | -0.553 | 2.88 | 7.241 | 12.889 | 19.824 |
| -1.923 | -6.373 | -9.653 | -7.947 | -3.241 | -0.488 | -0.466 | 0.931 | 17.097 | 25.498 |
| 1.128 | -1.298 | -5.515 | -9.434 | -7.434 | -2.984 | -4.911 | 12.366 | 23.014 | 36.158 |
| 1.591 | 1.925 | -0.135 | -1.918 | 1.715 | 6.583 | 12.688 | 21.159 | 33.778 | 46.583 |
| 2.035 | 2.607 | 3.355 | 4.387 | 9.16 | 15.354 | 23.296 | 33.483 | 46.529 | 47.311 |

Figure 6: Optimal value of each state

**QUESTION 7 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.**

In order to show Figure 6 in a more visualized way, we also generate its heat map across the 2D grid. The heat map is shown below:

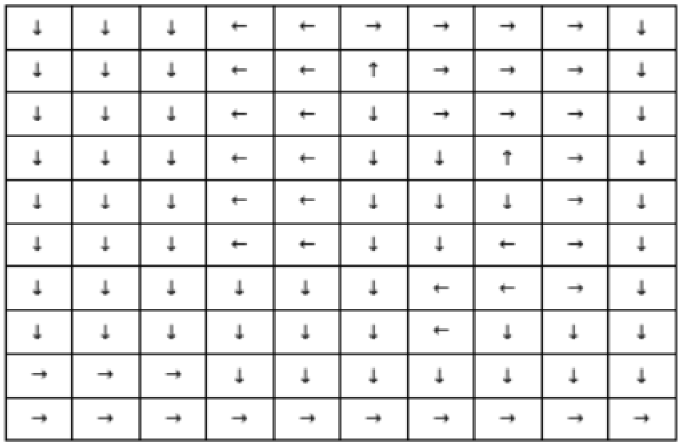


**QUESTION 8 Explain the distribution of the optimal state values across the 2-D grid. (Hint: Use the figure generated in question 7 to explain)**

In this case, since the reward function has some rewards to states in addition to the last state i.e. state 99, it is obvious that the optimal state values obtained in this case will not be same as when using reward function 1. In reward function 2, we observe that some of the states have a negative reward i.e. -100. This means that if the agent moves to that state, it will be getting a negative reward, which is obviously not desirable! Hence, the agent should try to avoid these states. Essentially this means that such states should have lesser optimal state values that the states to which the agent actually wants to go. These states mostly beget a negative optimum state value. This is what is depicted in the above heat map of optimal state values. Here too, the reward for reaching the 99th state is the highest. Thus, the agent will always try to get to this state by avoiding the states as much as possible which give negative reward. Such states with negative optimal values are depicted in darker shades in the heat map. This is very prominently observed in the 52nd state of the grid which begets the darkest colour. The darkest colour signifies that it has the least optimal value. This can be justified because this state is surrounded by states on three sides (up: 51st state, left: 42nd state, right: 62nd state) which have -100 reward. That is why the optimal value of this state is very low, because it is highly possible that if the agent comes to this state then it will most likely land up in one of the neighbouring states which have -100 reward. Hence, the agent should avoid this state. That is why it is justified for this state to have a very low optimal state value. Likewise, for all other states which have dark colour in the above heat map, their optimal state values are also low as compared to the states where the agent will actually benefit. Other states which have also been shown in the shades of purple colour (but a lighter shade) signify that the optimal values is close to 0 but positive. This is starkly observed in the upper left part of the grid. If the agent lands up in some state in the bottom right part of the grid, then it will be easier for him to go to the 99th state which has the highest reward. Therefore, these states in the bottom right part of the grid are signified by a high optimal value and a light shade which essentially means that it is beneficial for the agent to visit these states. Such is the distribution of the optimal state values across the 2-D grid.

**QUESTION 8 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.**

In this part, we again implement the computation step for the value iteration algorithm and get the optimal actions, just as what we did in question 5. Then, we visualize the actions using the arrows in the state table. The figure is shown below:



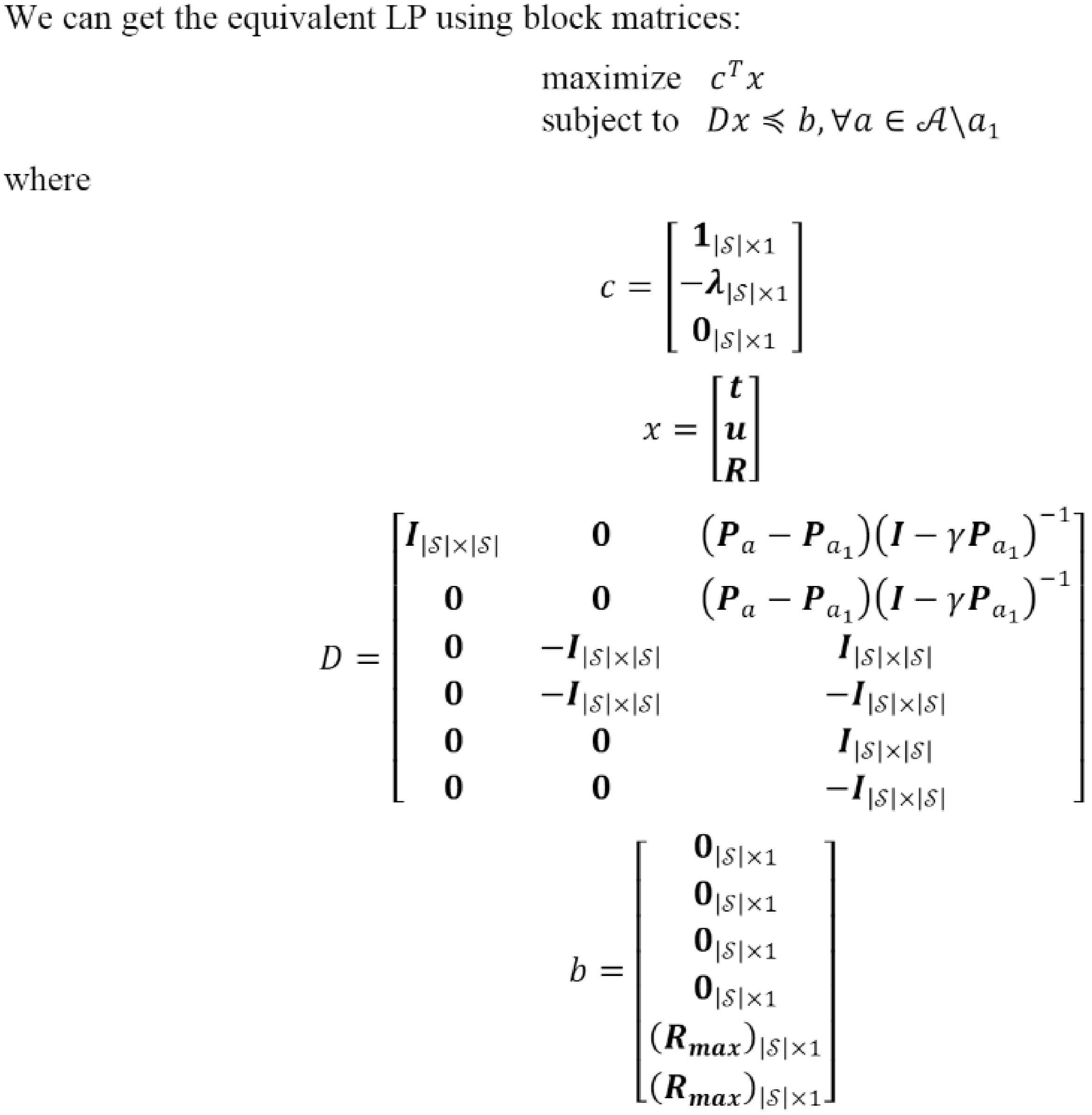
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ↓ | ↓ | ↓ | ← | ← | → | → | → | → | ↓ |
| ↓ | ↓ | ↓ | ← | ← | ↑ | → | → | → | ↓ |
| ↓ | ↓ | ↓ | ← | ← | ↓ | → | → | → | ↓ |
| ↓ | ↓ | ↓ | ← | ← | ↓ | ↓ | ↑ | → | ↓ |
| ↓ | ↓ | ↓ | ← | ← | ↓ | ↓ | ↓ | → | ↓ |
| ↓ | ↓ | ↓ | ← | ← | ↓ | ↓ | ← | → | ↓ |
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ← | ← | → | ↓ |
| ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ← | ↓ | ↓ | ↓ |
| → | → | → | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| → | → | → | → | → | → | → | → | → | → |

Figure 7: Optimal action for each state

Since the final state has the largest optimal value, the overall trend is that the upper state tends to go down and the left-part state tends to go right. However, since there exist negative optimal values in the right-half plane, the trend for optimal actions is interrupted by these points as well. Within the neighborhood of these low values, the neighbors tend to flow to states with higher values, but not these low values. So, the overall mode is interrupted for these points. Therefore, this optimal policy still follows our intuition in this question.

# 4 Inverse Reinforcement learning(IRL)

**Question 10 Express c, x,D in terms of** $R, P_a, P_{a1}, t_i, u, \lambda and R_{max}$



We can get the equivalent LP using block matrices:

$$\begin{aligned} &\text{maximize} && c^T x \\ &\text{subject to} && Dx \preccurlyeq b, \forall a \in \mathcal{A} \backslash a_1 \end{aligned}$$

where

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

$$x = \begin{bmatrix} \boldsymbol{t} \\ \boldsymbol{u} \\ \boldsymbol{R} \end{bmatrix}$$

$$D = \begin{bmatrix} \boldsymbol{I}_{|\mathcal{S}|\times|\mathcal{S}|} & \mathbf{0} & \left(\boldsymbol{P}_a - \boldsymbol{P}_{a_1}\right)\left(\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1}\right)^{-1} \\ \mathbf{0} & \mathbf{0} & \left(\boldsymbol{P}_a - \boldsymbol{P}_{a_1}\right)\left(\boldsymbol{I} - \gamma \boldsymbol{P}_{a_1}\right)^{-1} \\ \mathbf{0} & -\boldsymbol{I}_{|\mathcal{S}|\times|\mathcal{S}|} & \boldsymbol{I}_{|\mathcal{S}|\times|\mathcal{S}|} \\ \mathbf{0} & -\boldsymbol{I}_{|\mathcal{S}|\times|\mathcal{S}|} & -\boldsymbol{I}_{|\mathcal{S}|\times|\mathcal{S}|} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{|\mathcal{S}|\times|\mathcal{S}|} \\ \mathbf{0} & \mathbf{0} & -\boldsymbol{I}_{|\mathcal{S}|\times|\mathcal{S}|} \end{bmatrix}$$

$$b = \begin{bmatrix} \mathbf{0}_{|\mathcal{S}|\times 1} \\ \mathbf{0}_{|\mathcal{S}|\times 1} \\ \mathbf{0}_{|\mathcal{S}|\times 1} \\ \mathbf{0}_{|\mathcal{S}|\times 1} \\ (\boldsymbol{R_{max}})_{|\mathcal{S}|\times 1} \\ (\boldsymbol{R_{max}})_{|\mathcal{S}|\times 1} \end{bmatrix}$$

**Question 11** **Sweep $\lambda$ from 0 to 5 to get 500 evenly spaced values for $\lambda$. For each value of $\lambda$ compute OA(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 OE(s) values. Then use equation 3 to compute the accuracy of the IRL algorithm for this value of $\lambda$. You need to repeat the above process for all 500 values of $\lambda$ to get 500 data points. Plot $\lambda$ (x-axis) against Accuracy (y-axis). In this question, you should have 1 plot.**

In this question, we use the optimal policy of the expert in question 5 to fill $O_E(S)$

values. Then we sweep $\lambda$ from 0 to 5 with 500 evenly distributed points. For each $\lambda$ , we use the optimal policy of the agents (with extracted reward function 1) to fill $O_A(S)$ values. Next, we calculate the Accuracy with each value of $\lambda$ and plot $\lambda$ against Accuracy as below:

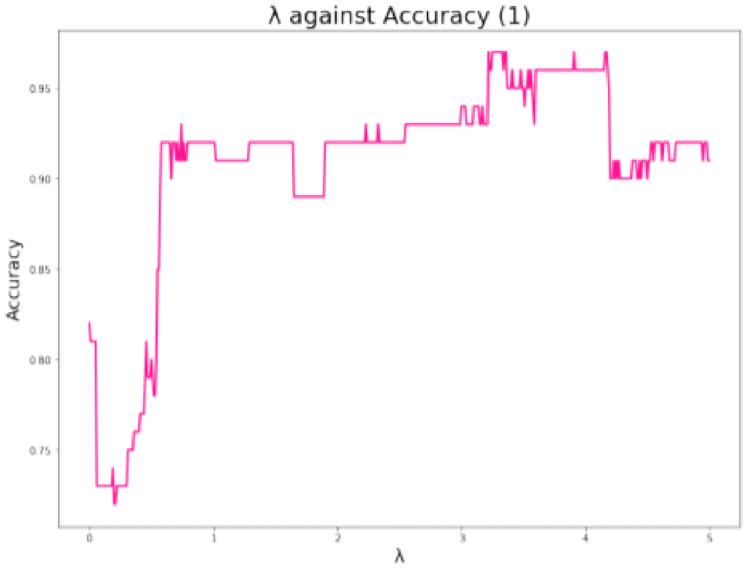


Figure 8: $\lambda against Accuracy with Extracted Reward Functions$

From the figure above, we find that the curve raised up to about 0.97 first and then decreased. Besides, there is a sharp decrease at $\lambda = 3.22$, which indicates that the maximum effect of adjustable penalty coefficient $\lambda$ happens around this area.

**Question 12 Use the plot in question 11 to compute the value of $\lambda$ for which accuracy is maximum. For future reference we will denote this value as $\lambda_{max}^{(1)}$. Please report $\lambda_{max}^{(1)}$**

From the plot in the last question, we can get the value of $\lambda$ with maximum accuracy:

$$\lambda_{max} = 3.22$$

**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 parameter set to $\lambda_{max}^{(1)}$. In this question, you should have 2 plots.**

The heat maps of the ground truth reward and extracted reward are show below:

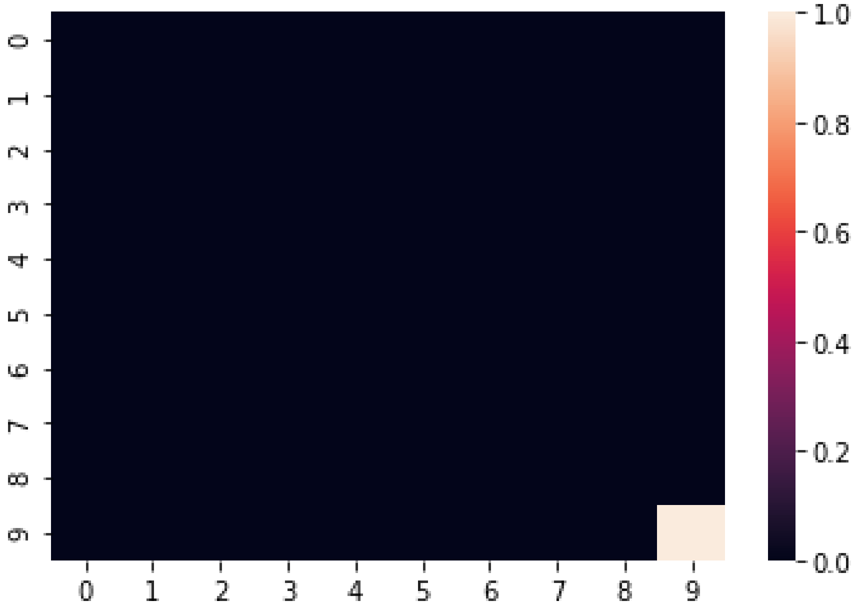


Figure 9: Heat map of ground truth reward

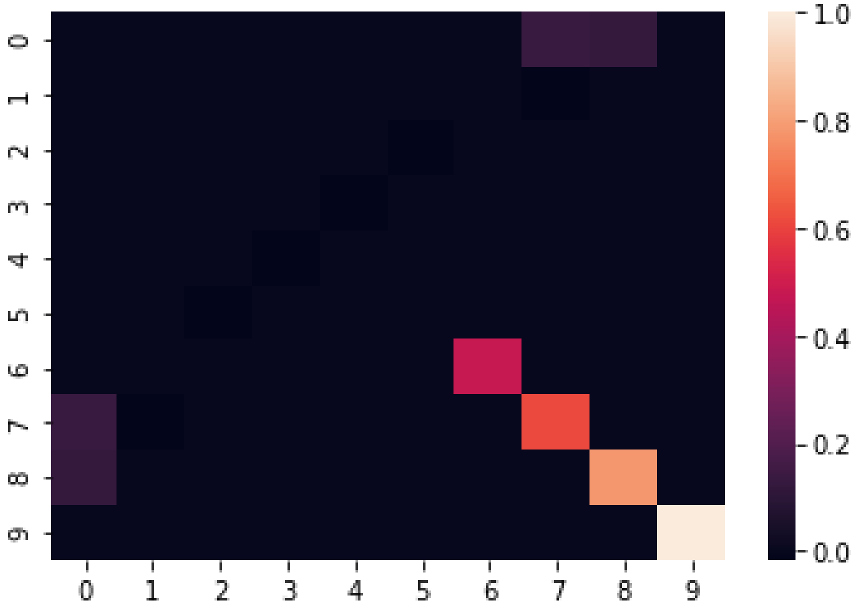


Figure 10: Heat map of extracted reward

**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 the optimal state values across the 2-D grid is shows below:

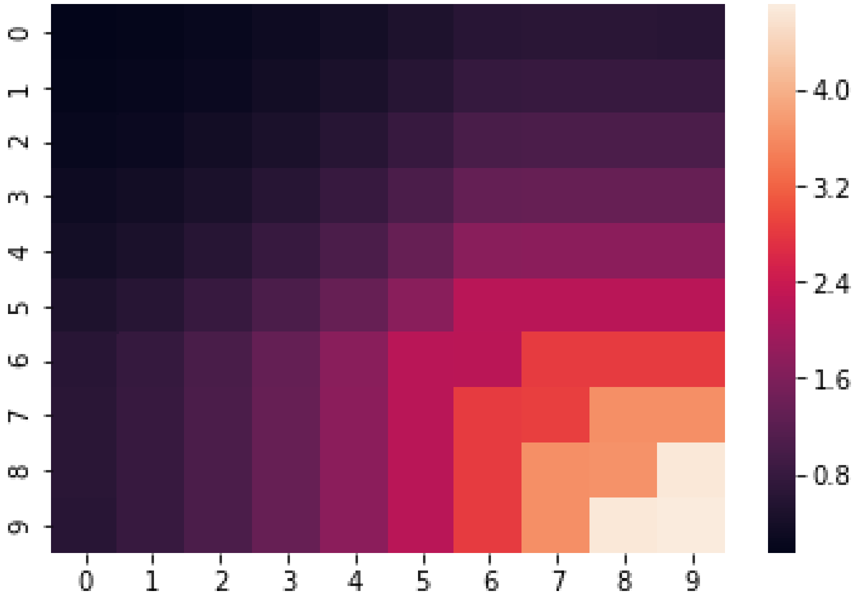


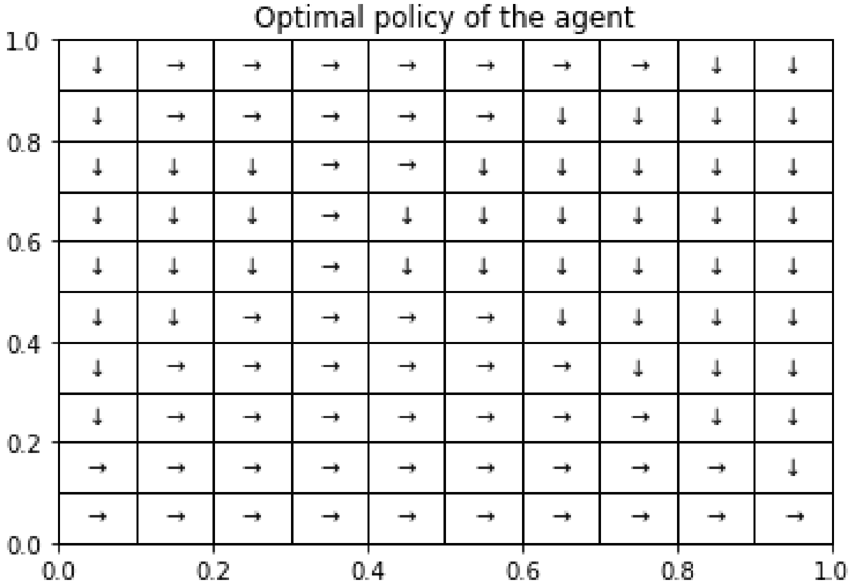
Figure 11: Heat map of optimal state values with extracted reward function 1

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

For the heat maps of question 3 and question 14, we can find that both maps increases from top left to the bottom right corner. The optimal state value of the extracted reward function and ground truth function are the same. The difference is that the heat map of question 3 has only positive value, but there are some negative value in the question 14. Because the L1 regularization, the heat map of extracted reward function is smoother than the ground truth function.

**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 optimal policy of the agent use the extracted reward function is shows below:



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ↓ | → | → | → | → | → | → | → | ↓ | ↓ |
| ↓ | → | → | → | → | → | ↓ | ↓ | ↓ | ↓ |
| ↓ | ↓ | ↓ | → | → | ↓ | ↓ | ↓ | ↓ | ↓ |
| ↓ | ↓ | ↓ | → | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| ↓ | ↓ | ↓ | → | ↓ | ↓ | ↓ | ↓ | ↓ | ↓ |
| ↓ | ↓ | → | → | → | → | ↓ | ↓ | ↓ | ↓ |
| ↓ | → | → | → | → | → | → | ↓ | ↓ | ↓ |
| ↓ | → | → | → | → | → | → | → | ↓ | ↓ |
| → | → | → | → | → | → | → | → | → | ↓ |
| → | → | → | → | → | → | → | → | → | → |

Figure 12: Optimal policy of the agent

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

By compare the figures of question 5 and question 16, we can find that the trend of the policy is almost the same that from top left to bottom right. But there are some different policy may caused by fluctuation in the extracted reward function.

**Question 18 Sweep $\lambda$ from 0 to 5 to get 500 evenly spaced values for $\lambda$. For each value of $\lambda$ 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 $\lambda$. You need to repeat the above process for all 500 values of $\lambda$ to get 500 data points. Plot $\lambda(x - axis)$ against Accuracy $(y - axis)$. In this question, you should have 1 plot.**

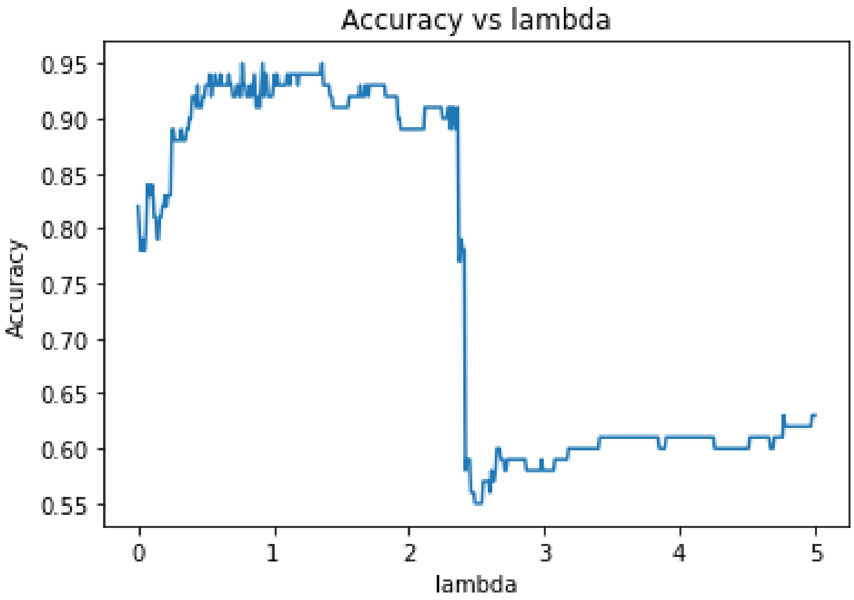
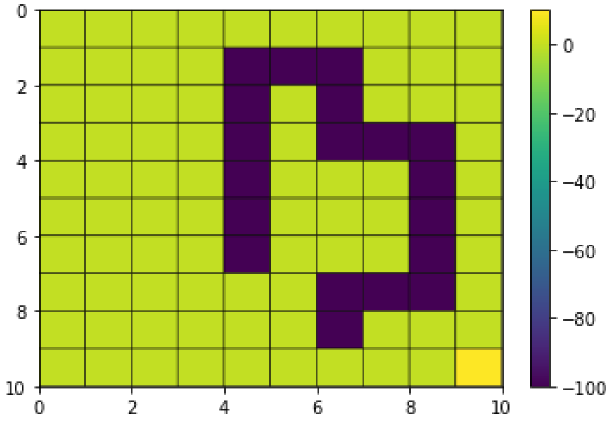


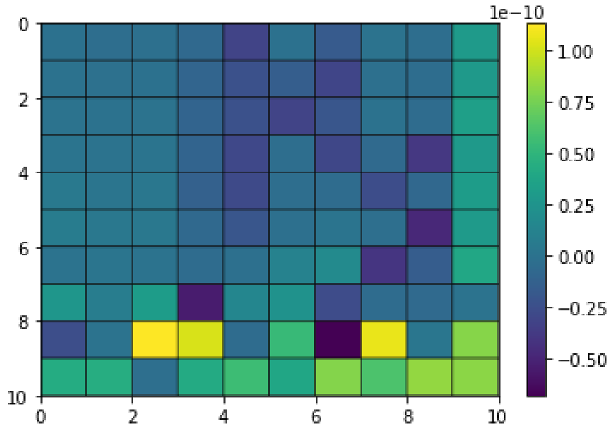
Figure 13: Lambda against Accuracy

**Question 19 Use the plot in question 18 to compute the value of $\lambda$ for which accuracy is maximum. For future reference we will denote this value as $\lambda_{max}^{(2)}$. Please report $\lambda_{max}^{(2)}$**

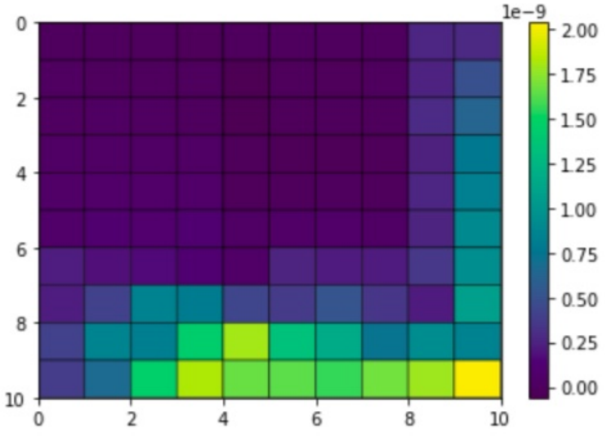
The plot above suggests that there more than one $\lambda$ values for which accuracy is maximum, but $\lambda_{max}^{(2)} \approx 0.95$.

**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 $\lambda$ parameter set to $\lambda_{max}^{(2)}$. In this question, you should have 2 plots.**





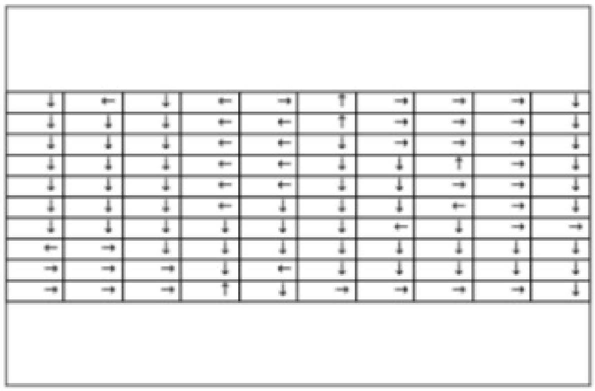
**Question 21 Use the extracted reward function computed in question 20, to compute the optimal values of the states in the 2-D grid. For computing the optimal values you need to use the optimal state-value function that you wrote in question 2. For visualization purpose, generate a heat map of the optimal 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.**

The heat map in Question 7 and Question 21 both have their higher values in the bottom right corner. One of the major differences in the range of values. Question 7's heat map has a maximum value of approximately 45, and Question 21's heat map has a maximum value of approximately 2. The spread of values also differs for both. Question 7's has a concentration of low values in the top middle of the graph, and Question 21's has a relatively even spread of low values (aside from the bottom and far right row).

**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.**



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

Both optimal policies walk towards the positive state and away from the negative state. Two major differences are the number of targets and the convergence place for both question (and this is caused by the reward function).

**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 first major discrepancy for the optimal policy is the coverage center for the extracted rewards. This results in the nearby states to be affected and respond and take action. To address this, the e-greedy algorithm can be used, or the discount value can be reduced.

The second major discrepancy is that there is action taking place for states on edges out of the border. This is caused by small reward values in proximity to the states. To address this, the optimal policy function's border boundaries can be modified. After making this change, the accuracy did change.