Reinforcement learning and Inverse Reinforcement learning

William Nafack UID : 405725778 Lea Alcantara UID : 005872120
Gaurav Singh UID : 305353434

May 22, 2022

1 Question 1

Figure 1 shows the two heatmaps of the reward functions.
Interpretation: Darker the box, the lower is the value of reward. In both the reward functions bottom right corner has the maximum reward.

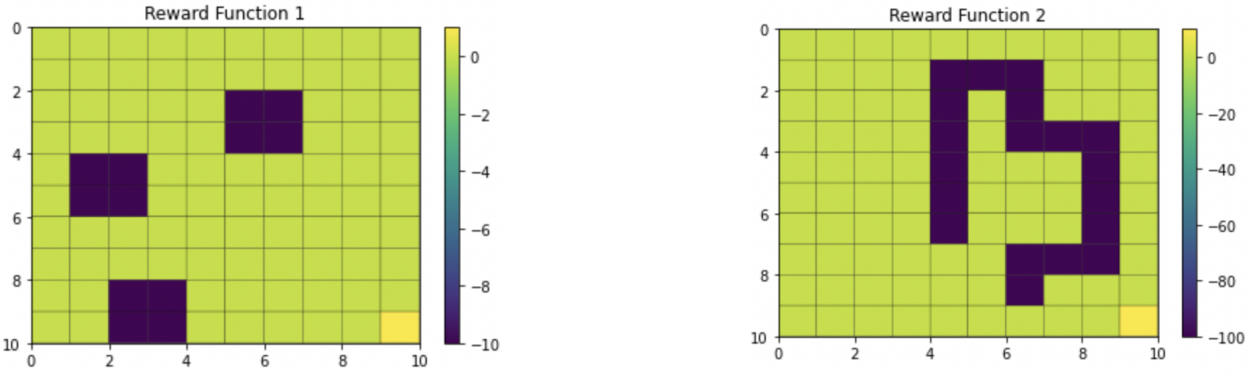


Figure 1: Heatmaps of Reward function 1 and 2

2 Question 2

Figure 2 shows a visualization of the optimal state values given w is 0.01 , γ is 0.8 and ϵ is 0.01 with reward function 1.

Figure 2

It takes $N = \mathbf{22}$ **steps** until convergence of the estimation process of the value iteration process. Figure below shows snapshots of the optimal state values across 5 different steps from 1 to N. These steps are **1, 6, 11, 16 and 21**.

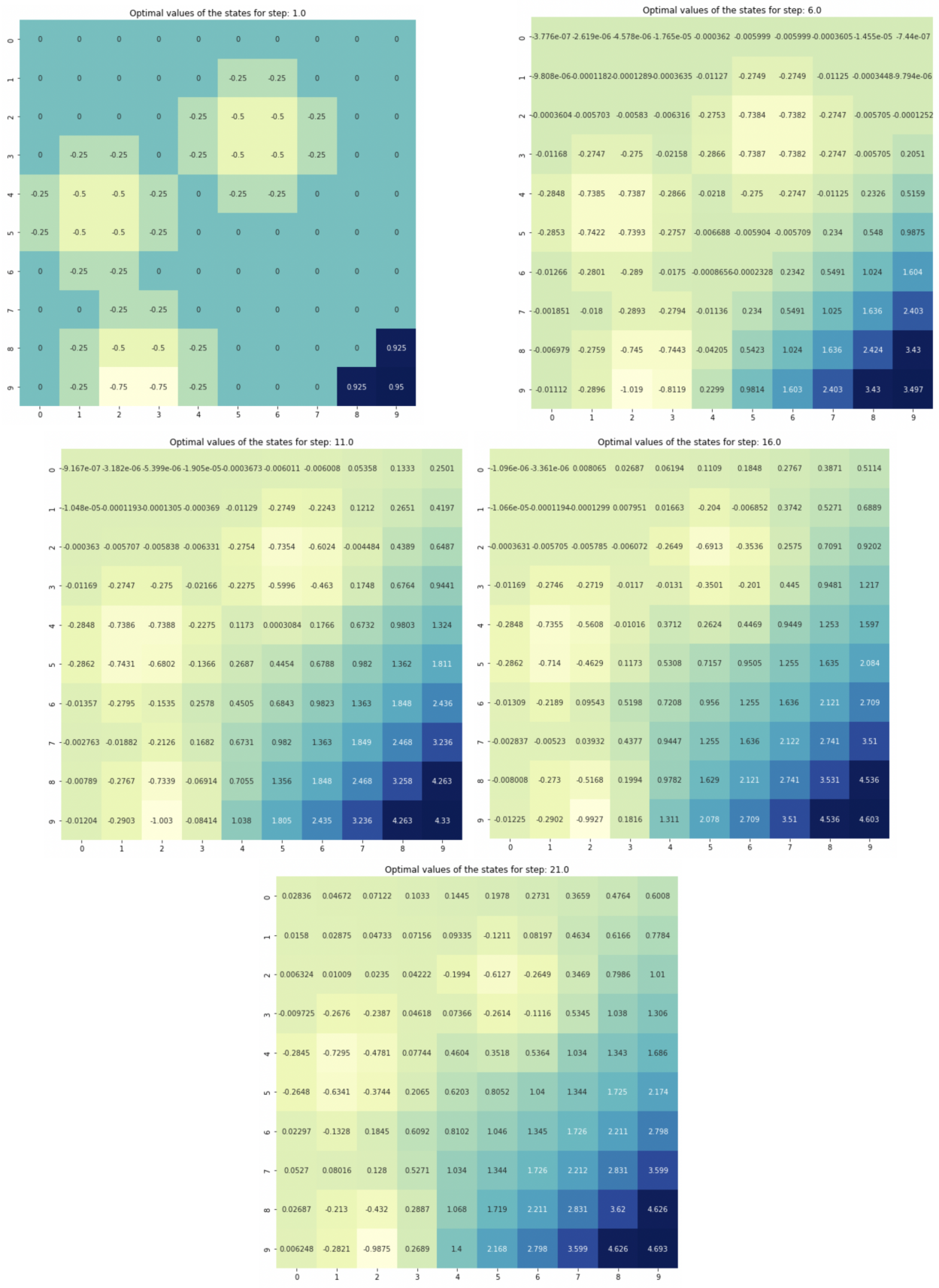


Figure 3: Snapshots of optimal state values across 5 steps.

We notice that the state values at state 99 and its neighbours has the highest optimal values through out the different steps as shown in figure 3. This is evident as from the reward function 1 in figure 1, state 99 received the highest reward. As you move away from state 99, the optimal values decreases and the states that had negative reward and its neighbours shows optimal values which are comparatively smaller to that of state 99 and its neighbours. This demonstrate that during learning, if the agent moves towards low reward location , the optimal values of the neighbours decreases and if the agent moves towards a high reward location, the optimal values will gradually increase.

Another point to notice is that at at the first step the state space is nearly sparse with only zero values in most states but as the algorithm continues through the next steps we progressively end up with a non sparse state space.

3 Question 3

Figure 4 shows the heatmap of the optimal state values with reward function 1 calculated above.

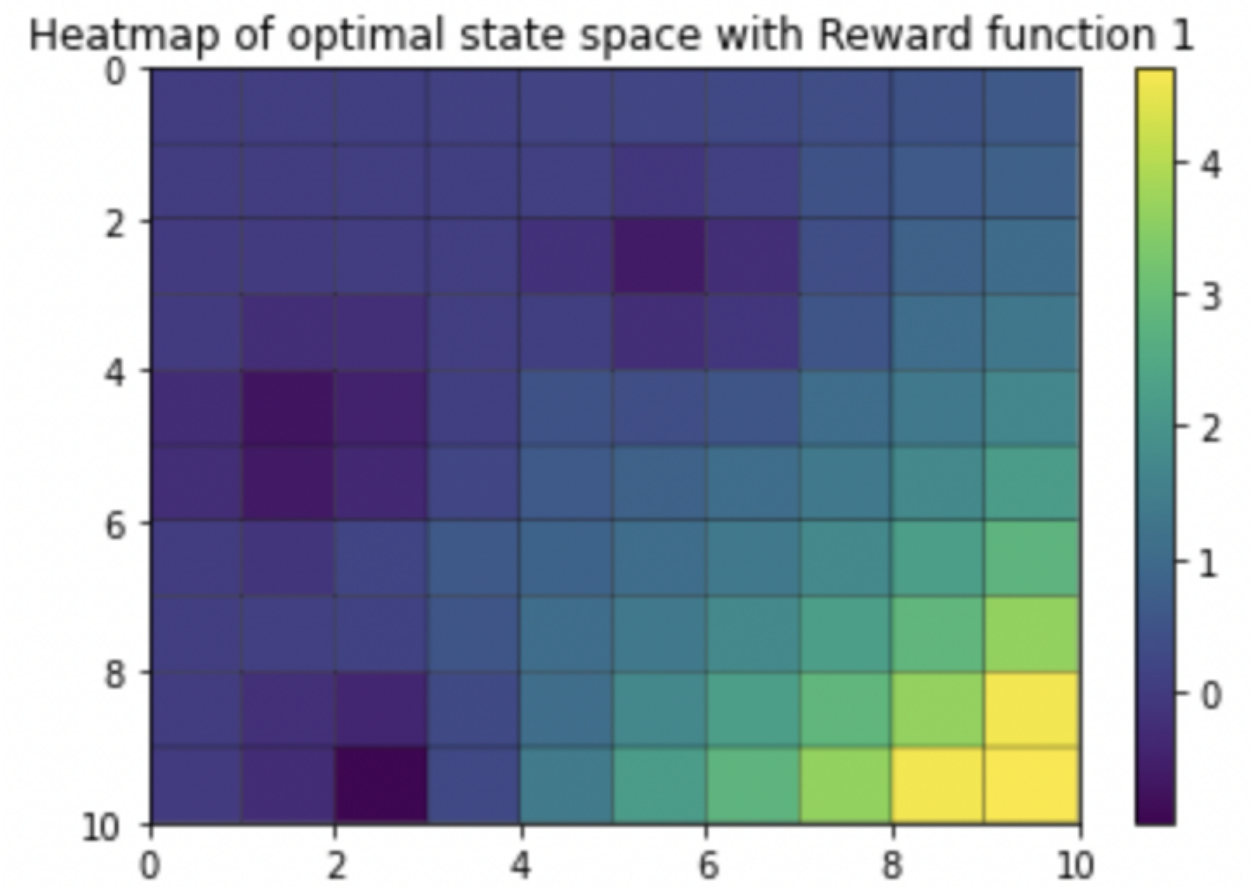


Figure 4: Heatmap of Optimal State Values with Reward Function 1

4 Question 4

From 4 we can make the following analysis. The darker regions on the heatmap shows the states with low optimal values and the brighter regions corresponds to the states with high optimal values. Furthermore, As you move away from the the state with high optimal values, the state values decreases progressively hence resulting in the bright colours at the bottom right shading continuously as we move left and up. This gradual

decay of the optimal values as we move from the state with the highest rewards can be due to the discount factor which discount future rewards.

In addition, the pattern that can be seen in the heatmap of the reward function 1 is also seen in the heatmap of the optimal values. For example we can identify the state with the highest optimal value at the bottom right in figure 4 is the same state with the highest reward on reward function 1 in figure 1.Also the neighbouring states are also brighter and fades away gradually as we move to low reward values and low optimal state values in both figures. This gives the intuition that we can extract the reward function by observing how the agent behaves and can be done through the process of Inverse Reinforcement Learning.

5 Question 5

Figure 5 shows the optimal action of the agent on the state space with reward function 1.

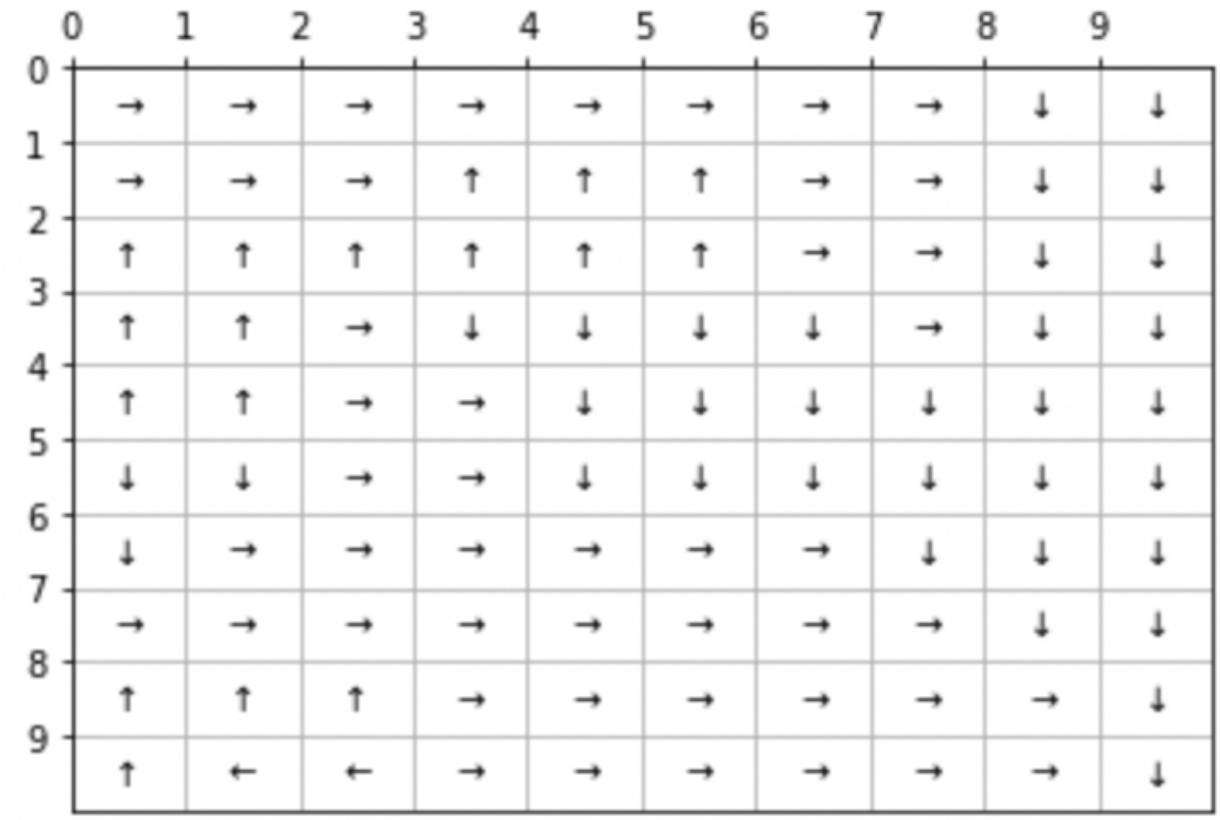


Figure 5: Optimal action taken by agent on the state space With Reward Function 1

We know that the goal of the agent is to maximize the the rewards it can get in the state space which is finite. Hence, given the reward function 1 we know that to be able to maximize this reward, the agent should move to state 99 with the highest reward as the other states have zero or negative rewards. From figure 5 , this is demonstrated as every path taken from one state will lead to the state 99 with the highest reward.

Also, It can be noticed that the agent moves away from the locations of negative rewards as it tries to find a path to the highest rewards. Another interesting point is that the agent always looks at the neighbours optimal values to decide the optimal action.

When we compare figure 2 with figure 5 we see that the arrows always point towards the state among the neighbours with the highest optimal values. By doing this iteratively, the agent can then build up to create

an optimal path / policy to maximize the reward. This is also supported mathematically as the optimal policy from the value iteration algorithm as shown below also relies on the optimal values of the neighbouring states.

$$\pi^*(S) = \underset{a \in A}{\text{argmax}} P^a_{s',s}[R^a_{s',s} + \gamma V^*(S')]$$

6 Question 6

Figure 6 shows a visualisation of the optimal state values given w is 0.01 , γ is 0.8 and ϵ is 0.01 with **Reward Function 2**

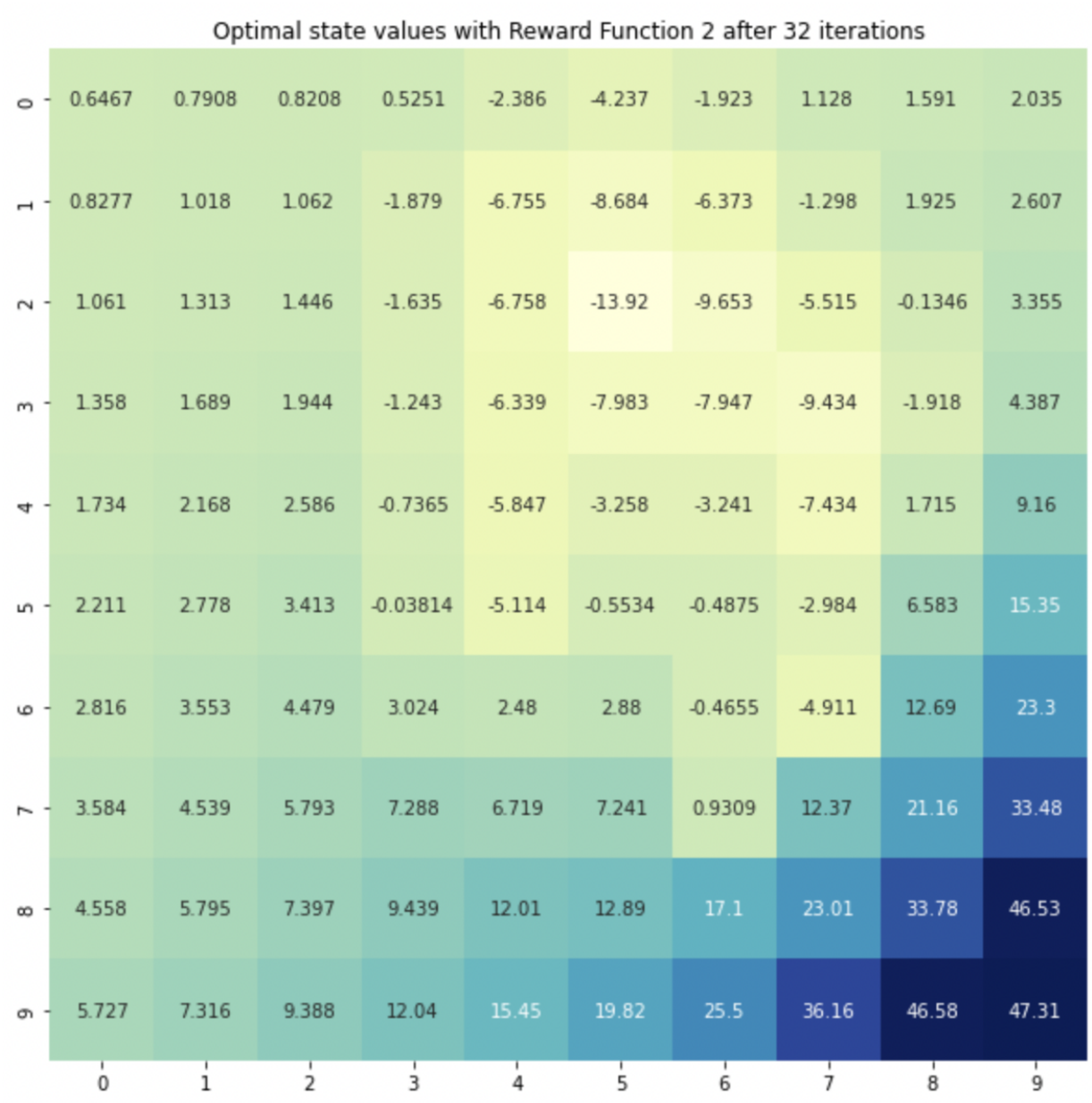


Figure 6: Optimal State Values With Reward Function 2

7 Question 7

Figure 7 shows the heatmap of the optimal state values with reward function 2 calculated above.

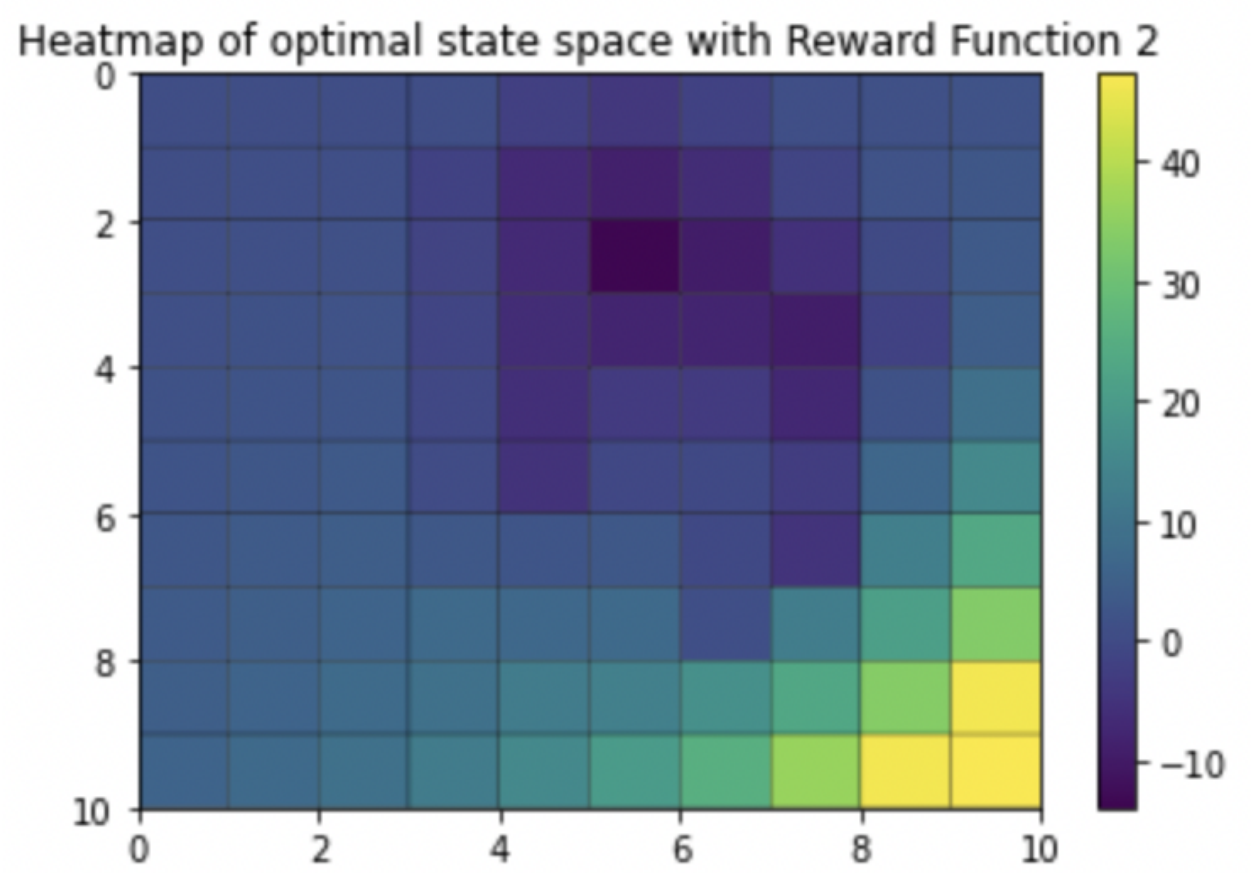


Figure 7: Heatmap of the Optimal State Values With Reward Function 2

From 4 we can make the following analysis similar to Question 4. The darker regions on the heatmap shows the states with low optimal values and the brighter regions corresponds to the states with high optimal values. As you move away from the the state with high optimal values, the state values decreases progressively hence resulting in the bright colours at the bottom right shading continuously as we move left and up. This gradual decay of the optimal values as we move from the state with the highest rewards can be due to the discount factor which discount future rewards.

Also, another pattern identified is the states with negative rewards forms a snake-like structure in the heatmap of reward function 2 in figure 1 and can also be visible in the heapmap of the optimal state values in figure 7. They are both in darker regions indicating low reward scores and low optimal values respectively. This gives another intuition that we can extract the reward function by observing how the agent behaves using the process of Inverse Reinforcement Learning.

8 Question 8

Figure 8 below shows the optimal action of the agent on the state space with reward function 2.

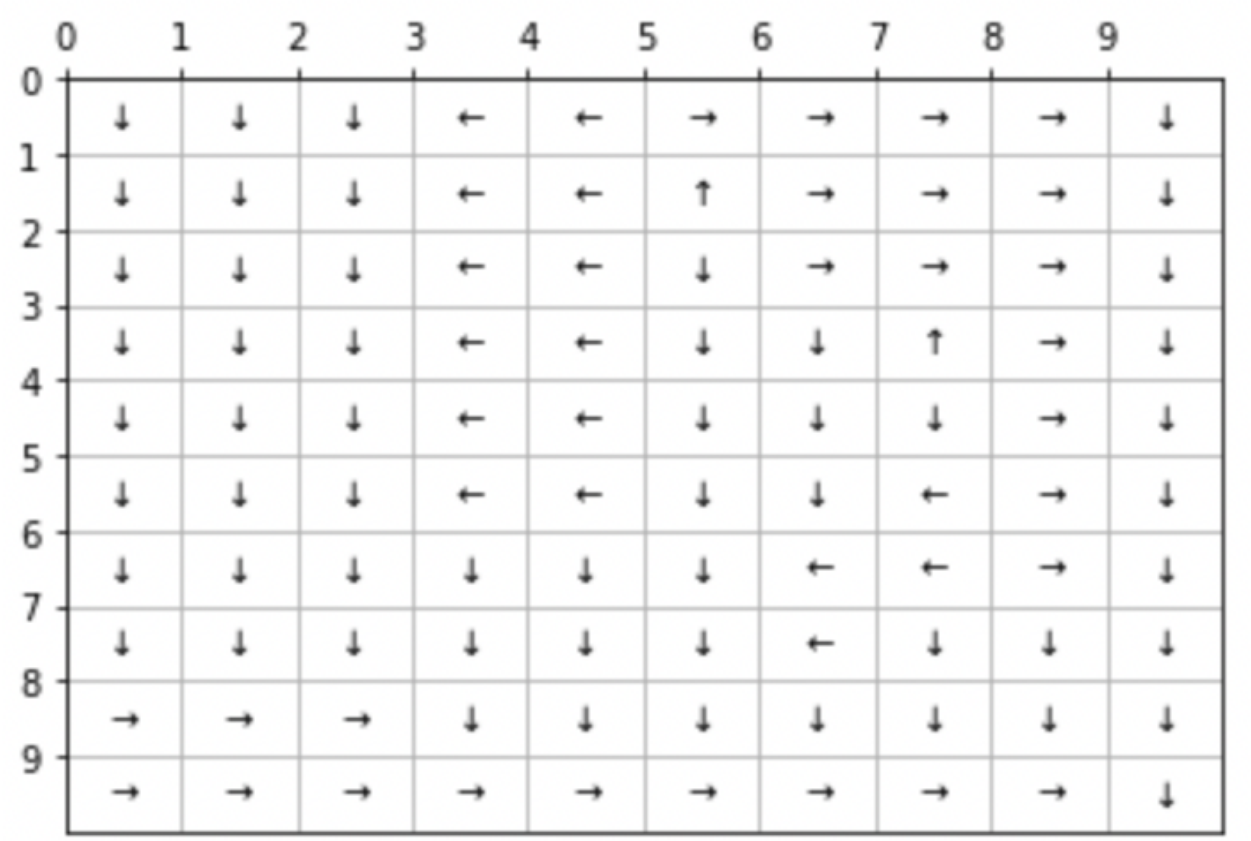


Figure 8: Optimal action taken by agent on the state space With Reward Function 2

From figure 8 we can make the following analysis similar to Question 5.We know that the goal of the agent is to maximize the the rewards it can get in the state space which is finite. Hence, given the reward function 1 we know that to be able to maximize this reward, the agent should move to state 99 with the highest reward as the other states have zero or negative rewards. From figure 5 , this is demonstrated as every path taken from one state will lead to the state 99 with the highest reward.

Also similarly to Question 5, when we compare figure 6 with figure 8 we see that the arrows always point towards the state among the neighbours with the highest optimal values. By doing this iteratively, the agent can then build up to create an optimal path / policy to maximize the reward. This is also supported mathematically as the optimal policy from the value iteration algorithm as shown below also relies on the optimal values of the neighbouring states.

$$\pi^*(S) = \underset{a \in A}{\operatorname{argmax}} P^a_{s',s}[R^a_{s',s} + \gamma V^*(S')]$$

In addition, we notice that the agent tries to move away from the states with negative rewards encapsulated in the snake-like structure. This gives us additional information that the agent does not necessarily have to move in the direction that has the shortest path to the reward but rather in the direction that maximizes the reward regardless of the path as per the optimal policy computation.

9 Question 9

Figure 9 and 10 below shows the state space with optimal values , heatmap of the optimal values and optimal action taken given given w is 0.6 , γ is 0.8 and ϵ is 0.01 with both **reward function 1** and **reward function 2**.

9.1 Reward Function 1

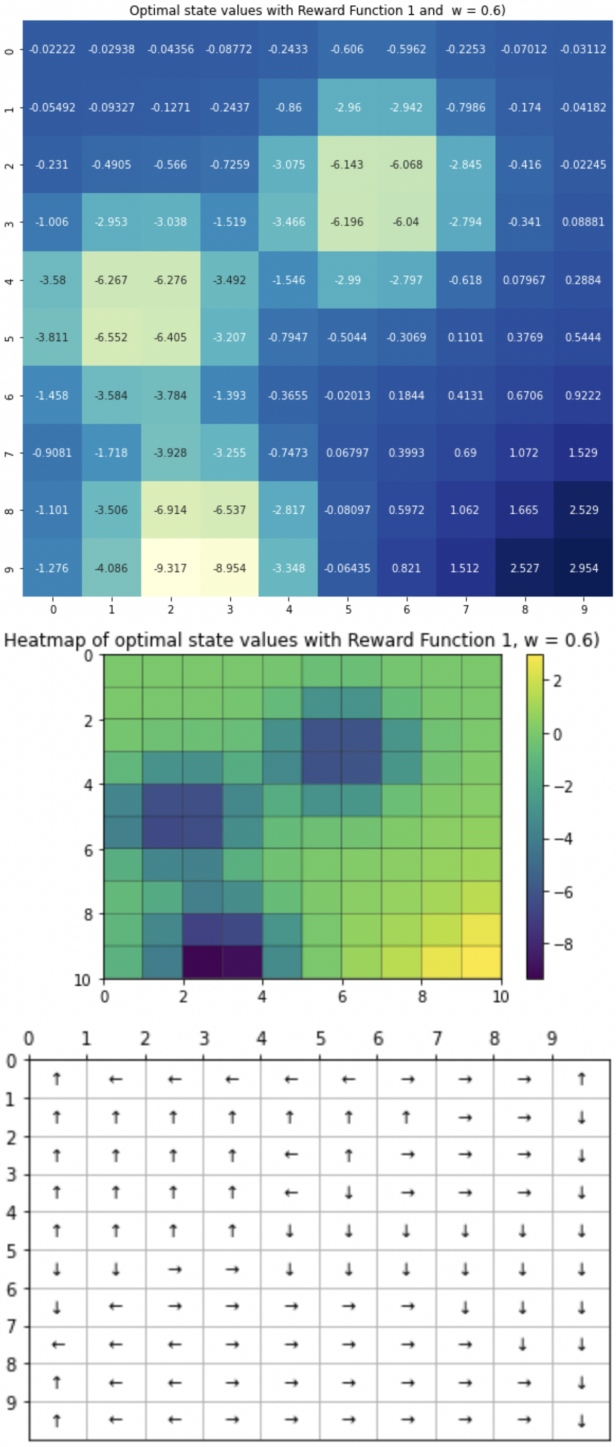


Figure 9: Plots of optimal values, Heatmap and optimal action with Reward Function 1

9.2 Reward Function 2

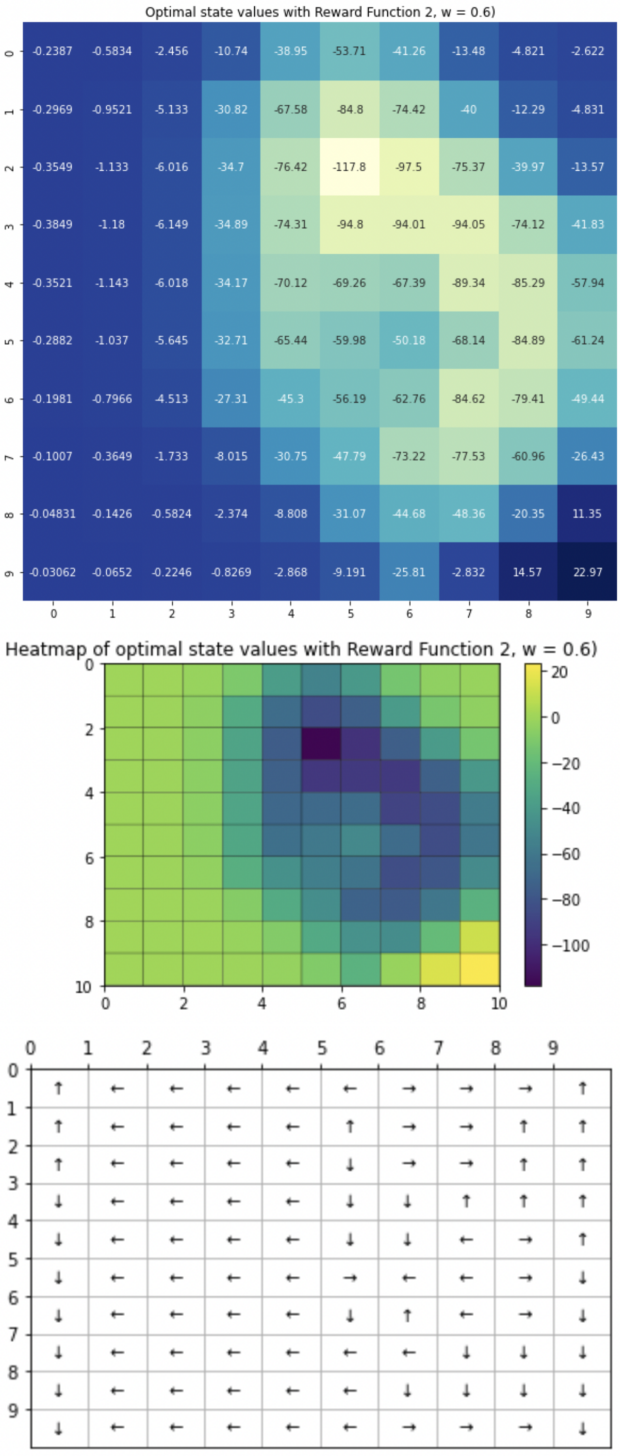


Figure 10: Plots of optimal values, Heatmap and optimal action with Reward Function 2

We can make the following observations from 9 and 10 when it comes to using w as 0.06. First, the optimal value with both reward functions at state 99 has decreased. The values at the states that had negative

reward has further decreased especially with reward function 2 where it is amplified making them negative values. As compared to the previous learning , we can see here that there is a high influence from the negative rewards affecting the agent in reaching the state with the highest reward.

From the optimal action plots , we see that the agent can move out of the state space. This may be as a result of the influence of the negative rewards. There are also only a few path that can lead the agent to state 99 with the highest reward. An explanation to this is that the states neighbouring state 99 do not offer any reward and hence are more influence by the states with negative rewards. There also appear to be instances in the optimal action plots where the agent is in a deadlock and simply move along the same limited path infinitely.

These results helps to understand the trade off of using different w values. Higher w allows you to explore more sub-optimal location that could lead to a global optima in the long-term. While this can be good in certain cases where more exploration of the state space is needed , it can be harsh on the learning process of the agent if it over weighs the use of the neighbouring optimal values to make the decision of which path to take.

In our use case here , a value of w as **0.1** is more appropriate as it allows the agent to learn an optimal path to the state with highest reward and making it less likely to be stuck in a sub-optimal location.

10 Question 10

In this project to extracting the reward the function we are using the LP formulation which is given as:

$$\begin{aligned} \underset{\mathbf{R},t_i,u_i}{\text{maximize}} \quad & \textstyle\sum_{i=1}^{|\mathcal{S}|}(t_i - \lambda u_i) \\ \text{subject to} \quad & [(\mathbf{P}_{a_1}(i) - \mathbf{P}_a(i))(\mathbf{I} - \gamma \mathbf{P}_{a_1})^{-1}\mathbf{R}] \geq t_i, \;\; \forall a \in \mathcal{A} \setminus a_1, \forall i \\ & (\mathbf{P}_{a_1} - \mathbf{P}_a)(\mathbf{I} - \gamma \mathbf{P}_{a_1})^{-1}\mathbf{R} \succeq 0, \;\; \forall a \in \mathcal{A} \setminus a_1 \\ & -\mathbf{u} \preceq \mathbf{R} \preceq \mathbf{u} \\ & |\mathbf{R}_i| \leq R_{max}, \;\; i = 1, 2, \cdots, |\mathcal{S}| \end{aligned}$$

P_a is the transition matrix corresponding to action a. λ is adjustable penalty coefficient. R is the reward vector whose size is same as number of states $|S|$. t_i and u_i are augmented variables needed for LP formulation. R_{max} is the maximum value of the ground truth reward to put a constraint on the limits of the reward. We need to extract R under given constraints. The above LP formulation can be simplified to below formulation by the change of variables.

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

C, x, D and b are given as:

$$x = \begin{bmatrix} t \\ u \\ r \end{bmatrix} \tag{1}$$

$$c = \begin{bmatrix} \mathbf{1}_{|S|\times 1} \\ -\lambda \cdot \mathbf{1}_{|S|\times 1} \\ \mathbf{0}_{|S|\times 1} \end{bmatrix} \tag{2}$$

$$b = \begin{bmatrix} \mathbf{0}_{|S|\times 1} \\ \mathbf{0}_{|S|\times 1} \\ \mathbf{0}_{|S|\times 1} \\ \mathbf{0}_{|S|\times 1} \\ \mathbf{0}_{|S|\times 1} \\ \mathbf{0}_{|S|\times 1} \\ \mathbf{0}_{|S|\times 1} \\ \mathbf{0}_{|S|\times 1} \\ R_{max} \cdot \mathbf{1}_{|S|\times 1} \\ R_{max} \cdot \mathbf{1}_{|S|\times 1} \end{bmatrix} \tag{3}$$

$$D = \begin{bmatrix} \mathbf{I}_{|S|\times|S|} & \mathbf{0}_{|S|\times|S|} & -\mathbf{P}_{a_1,a_2} \\ \mathbf{I}_{|S|\times|S|} & \mathbf{0}_{|S|\times|S|} & -\mathbf{P}_{a_1,a_3} \\ \mathbf{I}_{|S|\times|S|} & \mathbf{0}_{|S|\times|S|} & -\mathbf{P}_{a_1,a_4} \\ \mathbf{0}_{|S|\times|S|} & \mathbf{0}_{|S|\times|S|} & -\mathbf{P}_{a_1,a_2} \\ \mathbf{0}_{|S|\times|S|} & \mathbf{0}_{|S|\times|S|} & -\mathbf{P}_{a_1,a_3} \\ \mathbf{0}_{|S|\times|S|} & \mathbf{0}_{|S|\times|S|} & -\mathbf{P}_{a_1,a_4} \\ \mathbf{0}_{|S|\times|S|} & -\mathbf{I}_{|S|\times|S|} & -\mathbf{I}_{|S|\times|S|} \\ \mathbf{0}_{|S|\times|S|} & -\mathbf{I}_{|S|\times|S|} & \mathbf{I}_{|S|\times|S|} \\ \mathbf{0}_{|S|\times|S|} & \mathbf{0}_{|S|\times|S|} & \mathbf{I}_{|S|\times|S|} \\ \mathbf{0}_{|S|\times|S|} & \mathbf{0}_{|S|\times|S|} & -\mathbf{I}_{|S|\times|S|} \end{bmatrix} \tag{4}$$

$$P_{a_i,a_j} = \begin{bmatrix} (P_{a_i}(1) - P_{a_j}(1)) \cdot (I - \gamma P_{a_j})^{-1} \\ (P_{a_i}(2) - P_{a_j}(2)) \cdot (I - \gamma P_{a_j})^{-1} \\ . \\ . \\ . \\ (P_{a_i}(|S|) - P_{a_j}|S|) \cdot (I - \gamma P_{a_j})^{-1} \end{bmatrix} \tag{5}$$

11 Question 11

In this question we analysed the effect of λ on the accuracy of the IRL algorithm . λ is swept from (0 - 5) in steps of 0.01 to get 500 evenly spaced points. Accuracy is measuring how close the generated policy of extracted reward function O_a is to the expert policy O_e.

From below graph we can see that accuracy increases as λ increases for small values of λ and then it starts to drop. λ has a regularizing effect (Lasso regularization)which tends to find simple reward vectors. Large λ values tend to underfit the task.

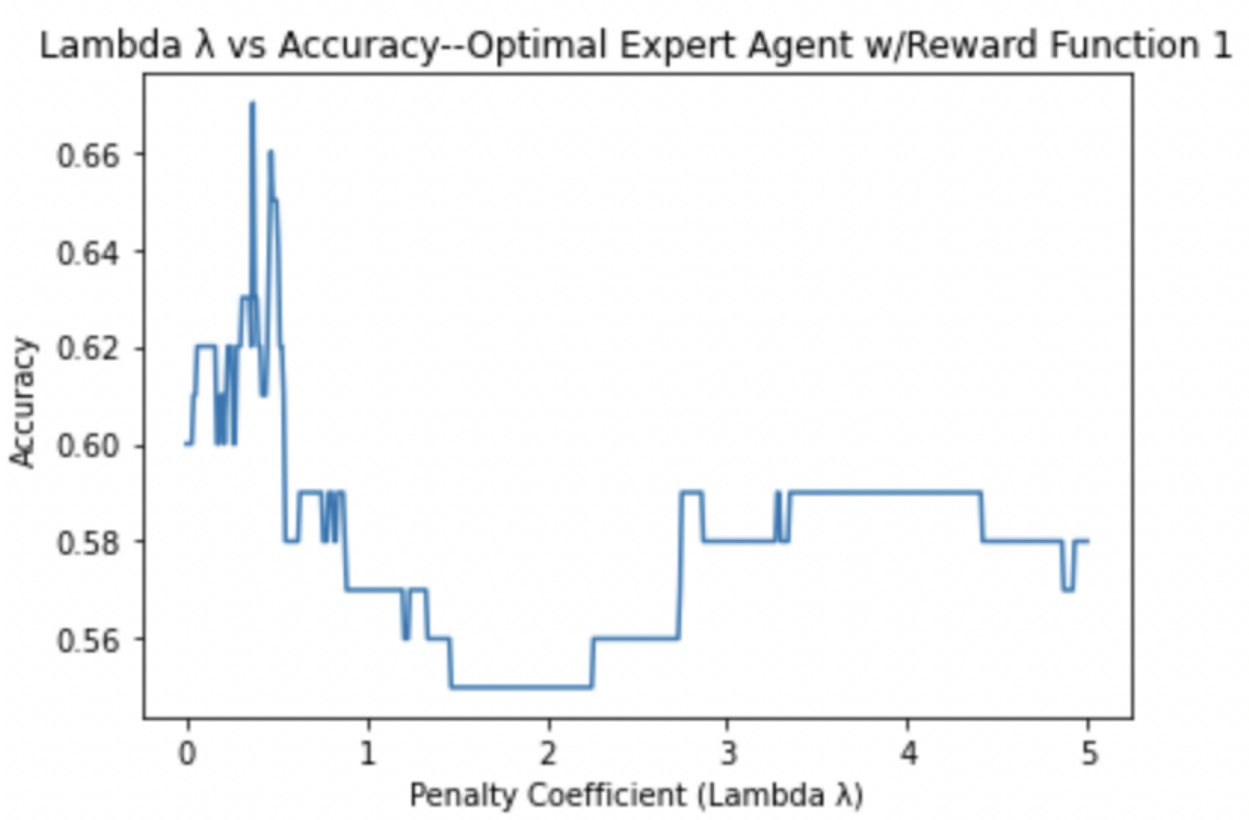


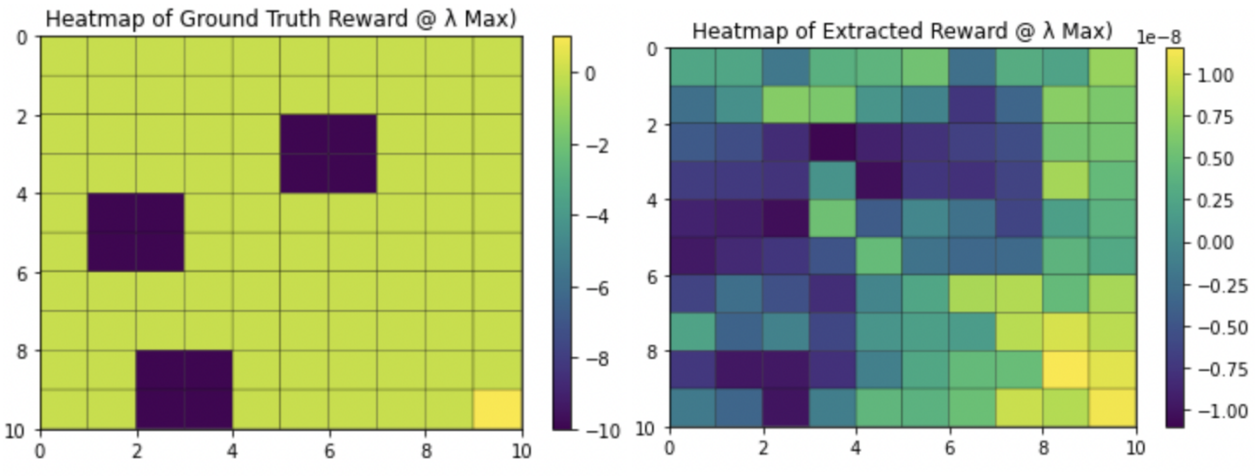
Figure 11: Plotting the penalty coefficient lambda versus Accuracy.

12 Question 12

Maximum accuracy obtained among different λ for reward function 1 is : $\lambda^{(1)}_{\mathbf{max}} = 0.37$ is $\mathbf{68}\%$.

13 Question 13

The heatmaps of the ground truth reward function 1 and extracted reward function are shown below. We can see that the key states are identified correctly. The states with high reward values (bottom right) and the blocks of lowest rewards are extracted correctly. The extracted reward function seems to have smooth transition from these key states to adjacent states as we can see similar patterns around the key states of the reward function.



(a) Ground Truth Reward Function

(b) Extracted Reward

Figure 12: Heat map functions

14 Question 14

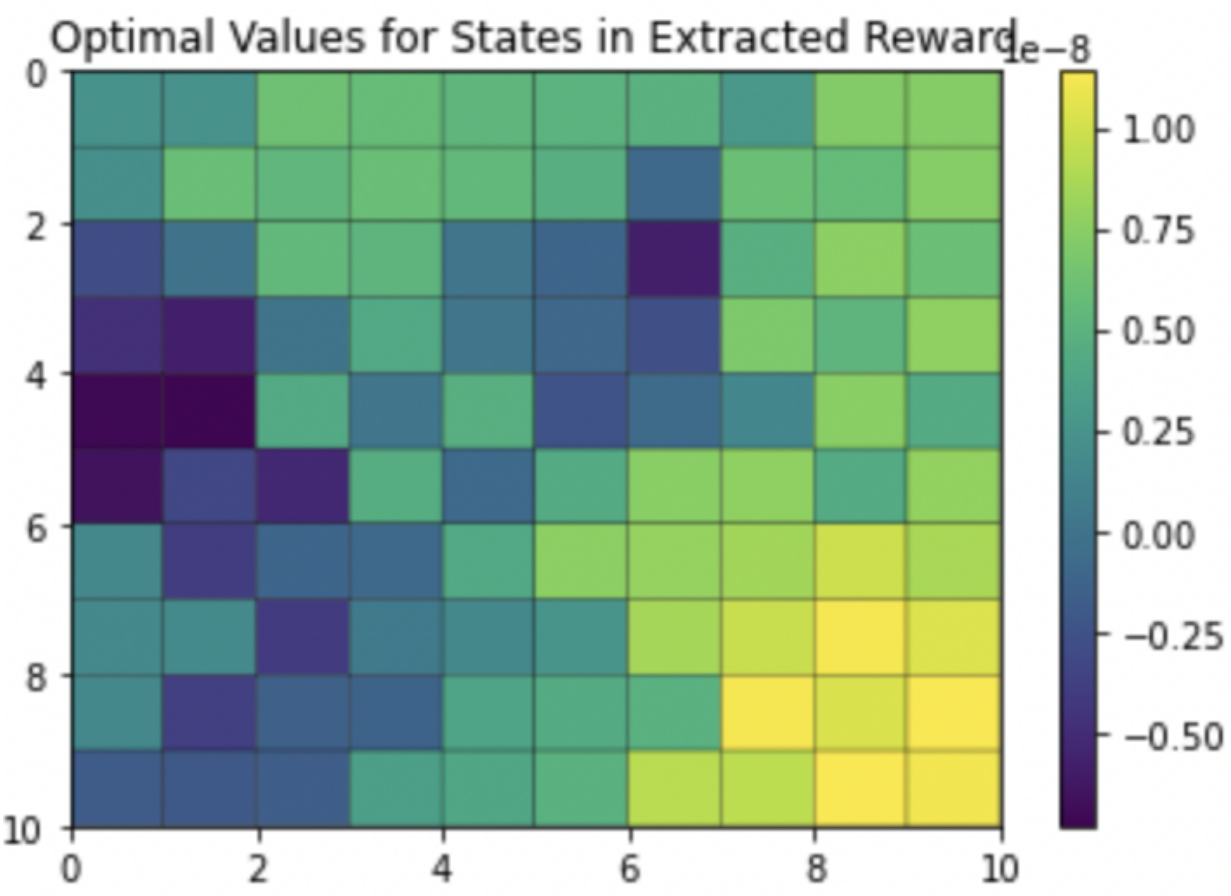


Figure 13: Heat Map of the Optimal Values of States for the Extracted Reward Function 1

15 Question 15

Comparison of the heatmaps obtained from Question 3 and Question 14.

Similarities:
1). The highest state values are found to be around state 99. This is expected because the state 99 has the highest reward. As we move away from state 99, we can see a gradual decrease in the state values in both the plots. The patterns are quite similar for both the plots. We can distinctively find 3 major regions of low state values from both the plots which correspond to low reward regions.

2). The gradual transition from high to low reward regions can be seen in the state values. Moving away from the high rewarding states show a decrease in the state values and vice versa in both the plots. The reason for this transitions is aligned with the fact that Bellman equation incorporated with discount factor tends to discount the future rewards.

Differences:
1). The scale of the optimal state values for both the plots is quite different. For expert reward, the optimal state values are of the order of 10 but for the extracted reward function its very low and of the order of 10^-9.

2). The regions in the optimal state values from question 3 are much more distinct and well defined as compared to that of question 14. This is because the values original reward function had well defined distinct reward values with 0, -10 and 1 but in the extracted reward function the values quite homogeneous such that it captures the transitions of the agent as it explores the environment and thus we don't see the clear well defined distinct regions in state values from extracted reward function.

16 Question 16

Optimal policy from the extracted reward function 1 with $\lambda^{(1)}_{max}$.

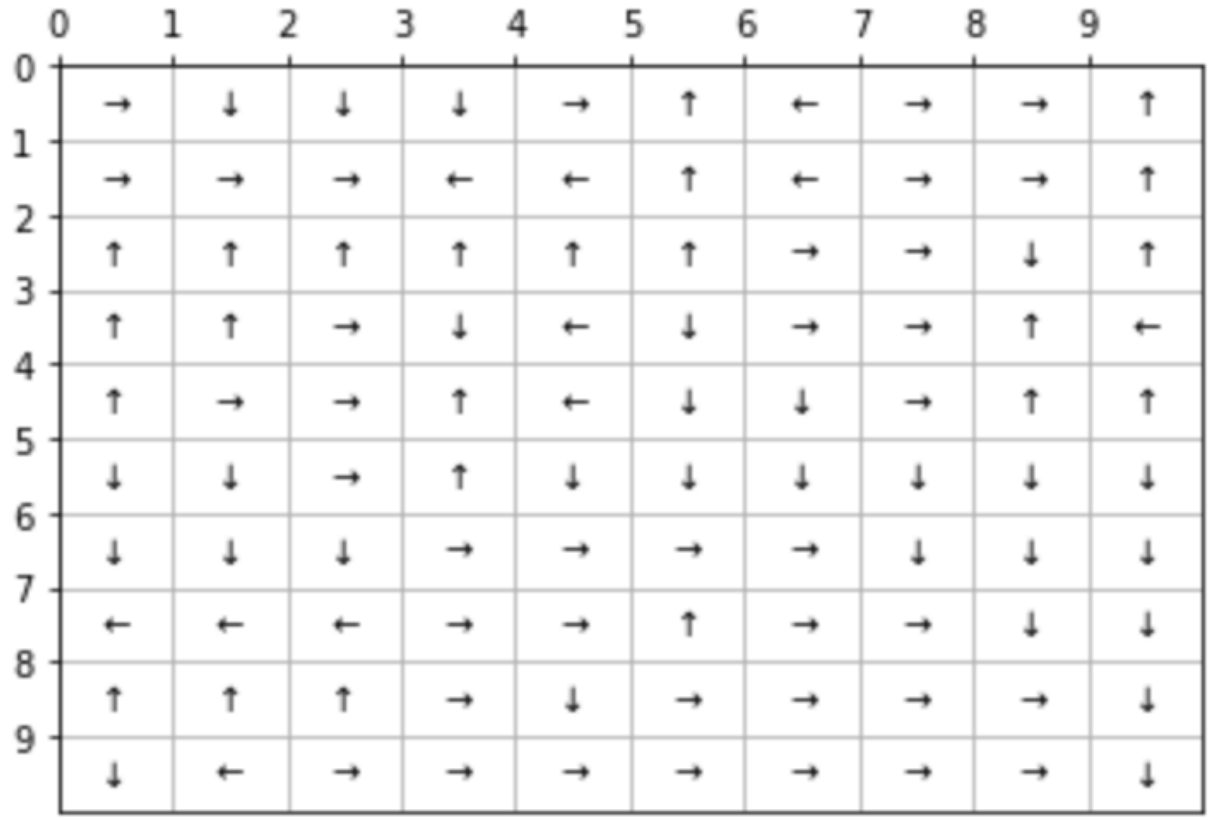


Figure 14: Actions taken by agent for extracted reward function 1

17 Question 17

Comparison between optimal policy obtained from Question 5 and Question 16.

Similarities:
1). In both the optimal policies obtained we see that the agent tries to take majority of the actions from right and down action space. This is because the maximal reward value is found in the bottom right position.

Differences:
1). The optimal policy from extracted reward function seems to throw agent off the grid for some states. Example: For state 7, is making the agent go left. This is not the case for the policy obtained from original reward function. This is because for extracted reward function the agent is taking suboptimal actions to maximise the future reward as seen from the extracted reward function and thus can throw the agent off the grid during these suboptimal actions.

2). The agent is guaranteed to reach the state with maximum reward following the policy from question 5 but this is not the case with policy from Question 16.

3). The actions in Question 16 seem to be more haphazard as compared to those from Question 5.

4). There are some suboptimal states which can be seen from Question 16 policy which will lead the agent to be stuck in a local loop. Example: State 21 and State 31.

18 Question 18

The plot of accuracy vs lambda (evenly spaced 500 points between [0, 5]) is shown below, for reward function 2.

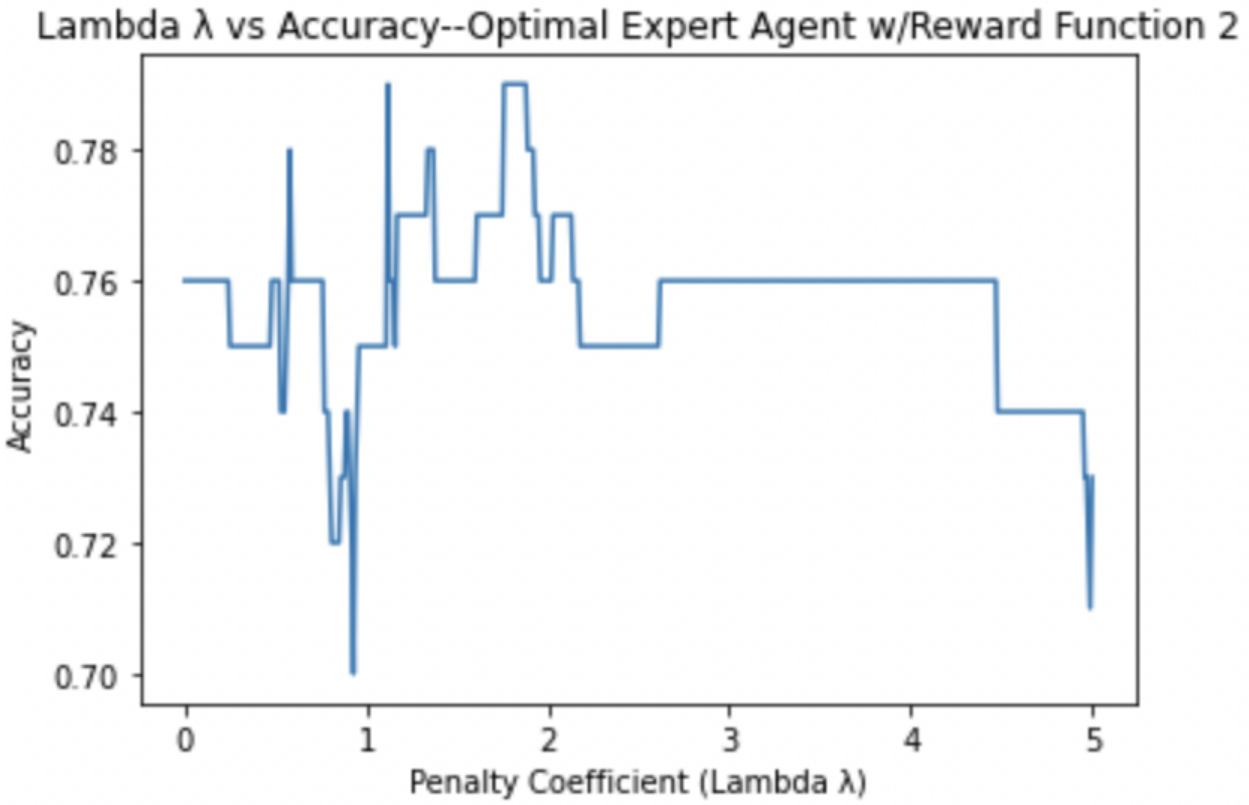


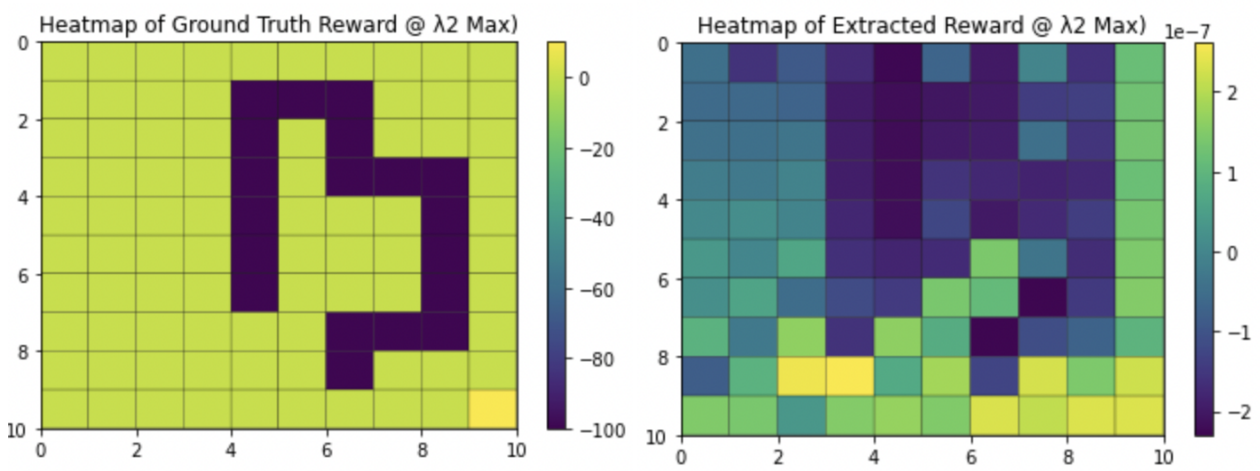
Figure 15: Plotting the penalty coefficient lambda versus Accuracy for Reward Function 2.

19 Question 19

Maximum accuracy obtained among different λ for reward function 2 is : $\lambda_{\mathbf{max}}^{(\mathbf{2})} = 1.12$ is $\mathbf{79}\%$.

20 Question 20

The heatmaps of the ground truth reward function 2 and extracted reward function are shown below. We can see that the key states are identified correctly. The states with high reward values (bottom right) and the blocks of lowest rewards are extracted correctly. The extracted reward function seems to have smooth transition from these key states to adjacent states as we can see similar patterns around the key states of the reward function.



(a) Ground Truth Reward Function

(b) Extracted Reward

Figure 16: Heat maps for Reward Function 2 with $\lambda_{max}^{(2)}$

21 Question 21

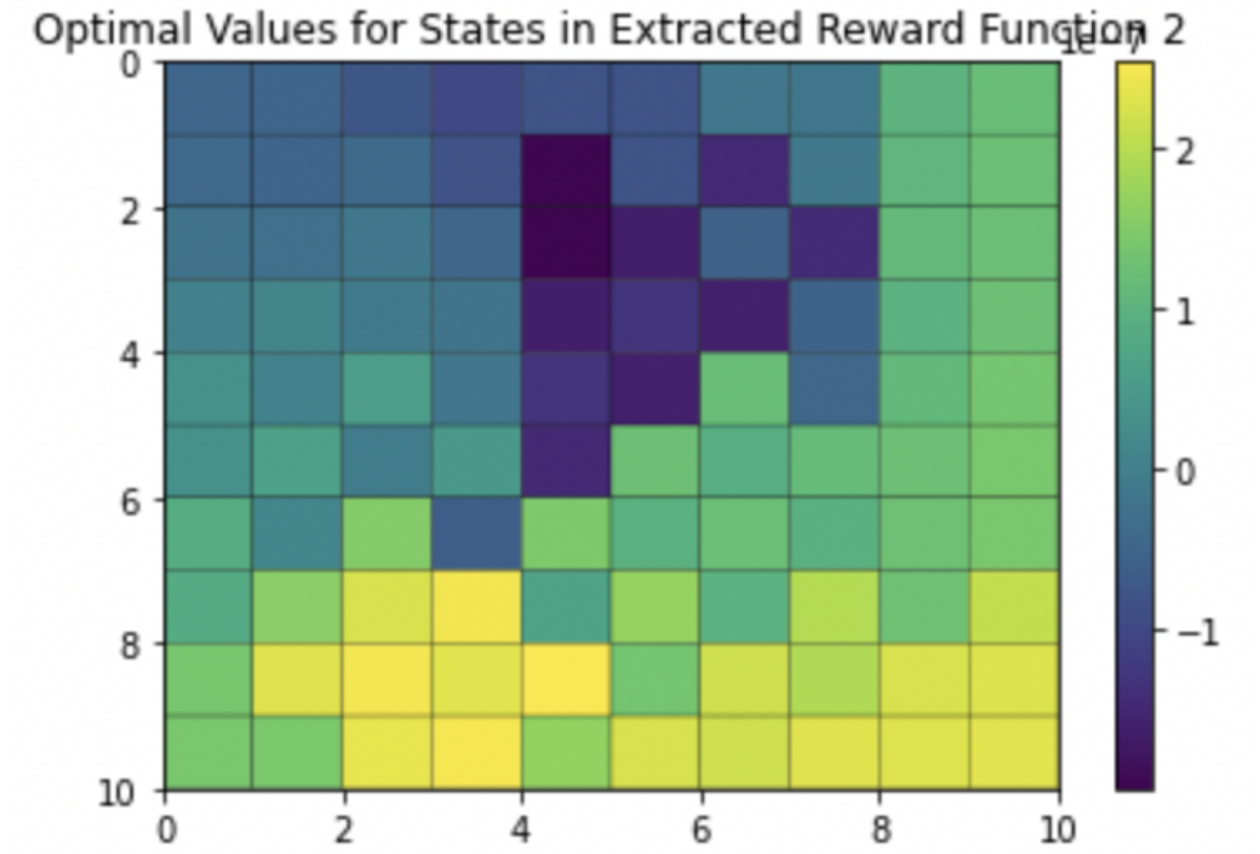


Figure 17: Heat Map of the Optimal Values of States for the Extracted Reward Function 2

22 Question 22

Comparison between the heatmaps from Question 7 and Question 21.
Similarities:

1). The state 99, is the state of highest reward and it can be seen from the optimal state values from both of the extracted reward function and expert reward function.

2). The region of negative rewards i.e states with low optimal values are quite similar in both the plots.

3). The gradual transition from high to low reward regions can be seen in the state values. Moving away from the high rewarding states show a decrease in the state values and vice versa in both the plots. The reason for this transitions is aligned with the fact that Bellman equation incorporated with discount factor tends to discount the future rewards.

Differences:

1). The scaling factor for the two heatmaps is very different. It's very low by the order of 10^8 as compared to the heatmap from Question 7. This is because optimal state values are calculated from the extracted reward function and not the original reward function.

2). The heatmap from Question 21 seems to have two distinct regions(bottom right and bottom left) of high state values as compared to heatmap from question 7.

3). The agent is more likely to move off grid seeing the heatmap from question 21 as compared to heatmap from question 7 as the rewards are spread and distributed over the board and not localized as compared to the baseline reward and state values, which might cause the agent to move off grid.

23 Question 23

Optimal policy from the extracted reward function 2 with $\lambda^{(2)}_{max}$.

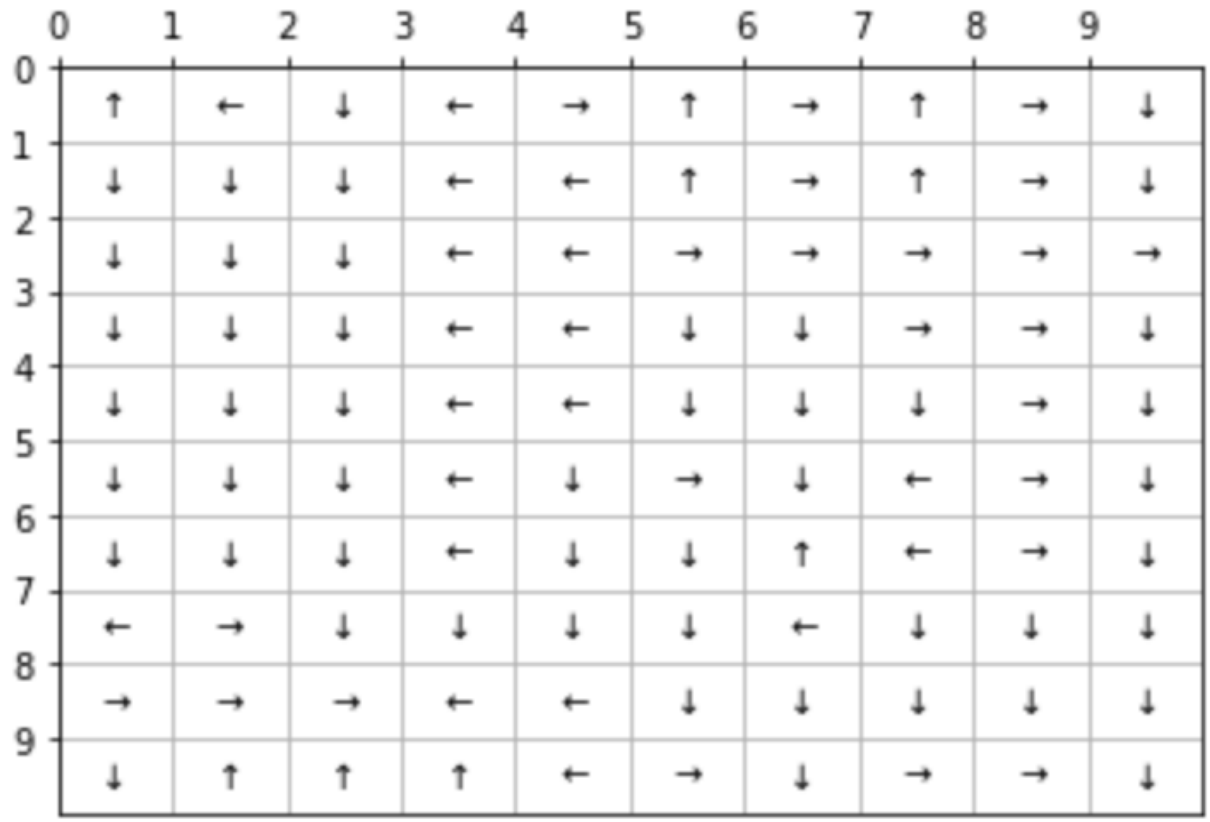


Figure 18: Optimal Policy of the Agent for extracted reward function 2

24 Question 24

Comparing the optimal policies from Question 9 and Question 23.
Similarities:

1). In both the action maps we see that the agent tries to move from the region of low reward or low state values towards region of high rewards.

2). We can see all four action types being used by the agent to reach the optimal reward regions in both of the plots.

3). In both of the plots there are a few actions in neighbouring states which tend to diverge the agent, which tells us about the incompatible states.

Differences:

1). The optimal policy from extracted reward function seems to throw agent off the grid for some states. Example: For state 7, is making the agent go left. This is not the case for the policy obtained from original reward function. This is because for extracted reward function the agent is taking suboptimal actions to maximise the future reward as seen from the extracted reward function and thus can throw the agent off the grid during these suboptimal actions.

2). The agent is guaranteed to reach the state with maximum reward following the policy from ques-

tion 9 but this is not the case with policy from Question 23.

3). The optimal policy actions in Question 23 seem to be more haphazard as compared to those from Question 9.

4). There are some suboptimal states which can be seen from Question 23 policy which will lead the agent to be stuck in a local loop. Example: State 28 and State 38 which is not there in optimal policy from Question 9.

25 Question 25

The discrepancies found from the above results are as follows:

1). The agent might fall into a loop between two suboptimal states and thus can never reach the maximum reward states. Example: State 28 and state 38 have converging actions (arrows) for Optimal policy from extracted reward function 2.

2). There are boundary actions in which can throw the agent off the grid. Example: Action at state 0 and state 10 per optimal policy found from extracted reward function 2 can throw agent off the grid.

Discrepancy 2 is handled by setting the state values for the edge/ boundary states to be $-\infty$. This would result in choosing the actions which will move the agent to in-grid states and thus ensure each action will lead the agent to the maximal reward without making it to move off grid.

The discrepancy 1 requires more algorithmic changes but one quick change that can be done is to decrease the ϵ (error tolerance) before algorithm stops. ϵ is changed from 0.01 to $\epsilon = 10^1 1$. This will make Value iteration to run for more iterations before convergence criterion is met and it will give the agent more time to get to better policy.

The results after adding the above mentioned fixes led to below results for IRL for the two reward functions. We can see there is a significant improvement in performance in terms of accuracy for the extracted policies.

For reward function 1:
The accuracy changed from 67% ($\lambda^{(1)}_{max} = 0.37$) to 75% ($\lambda^{(1)}_{max} = 1.05$)

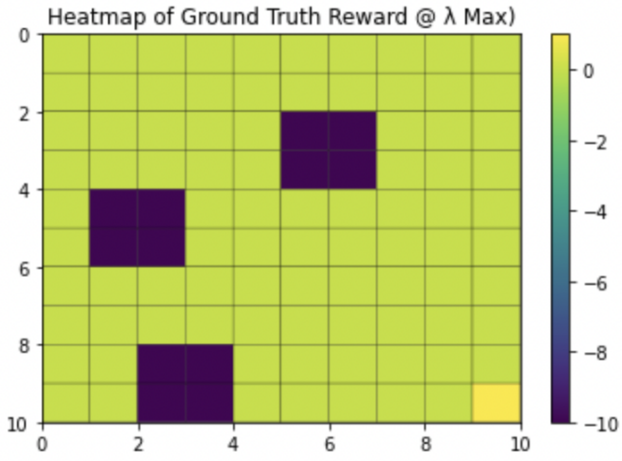
For reward function 2:
The accuracy changed from 76% ($\lambda^{(1)}_{max} = 1.12$) to 87% ($\lambda^{(2)}_{max} = 3.0$)

The comparison plots for Accuracy vs λ for both the extracted reward functions is shown below along with optimal state values and policies. The extracted reward functions are shown below. On comparison with the original reward functions and with the heatmaps of the reward functions without mentioned fixes we see they are more closer to the original functions. Also the dark regions(low rewards) and bright regions(high rewards) are captured more precisely and distinctively. We can see these extracted reward values tend to be more discrete now just like original reward functions as compared to those extracted from without the fixes which were distributed in the neighbouring states.

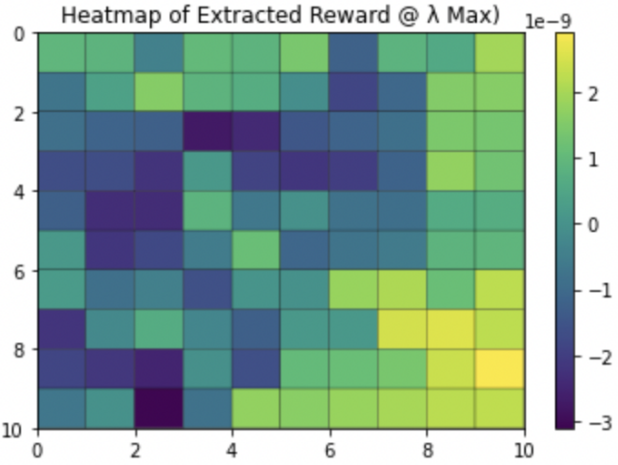
It is also seen that off-grid boundary actions in the optimal policy are reduced now i.e now actions are such that high rewards are achievable by agent without moving off the grid.

The local loops(deadlocks) are somewhat reduced but they are still there and require more algorithmic changes to mitigate.

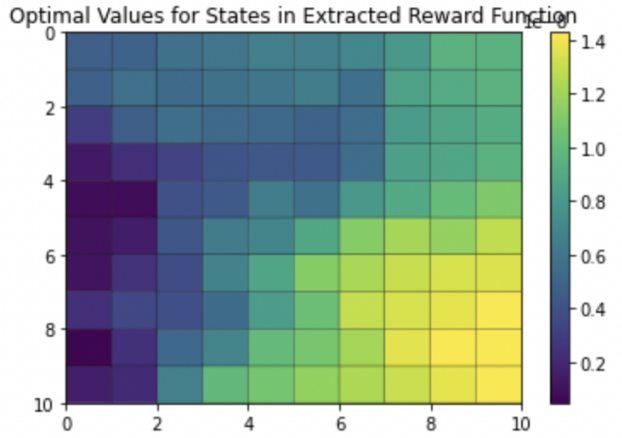
25.1 Effects of modifications for Reward Function 1



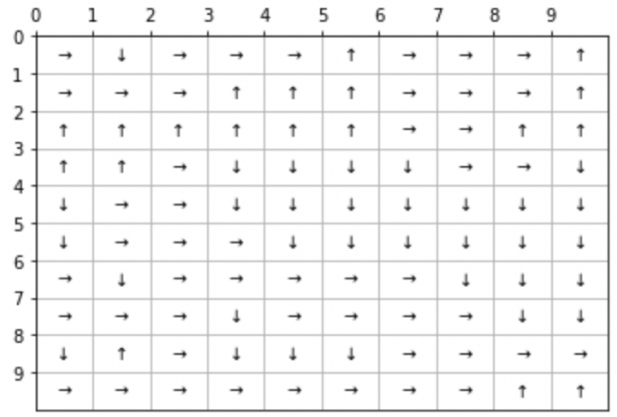
(a) Ground Truth Reward Function 1



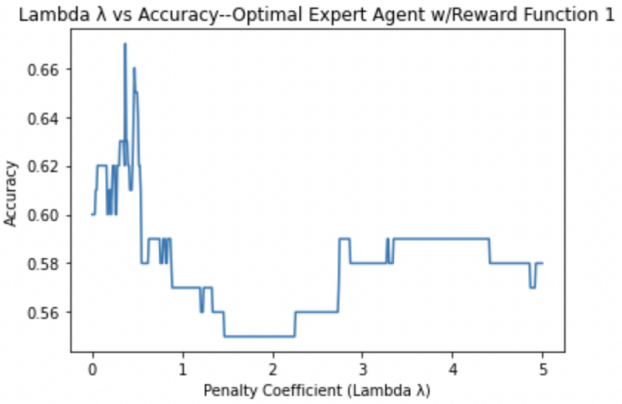
(b) Extracted Reward



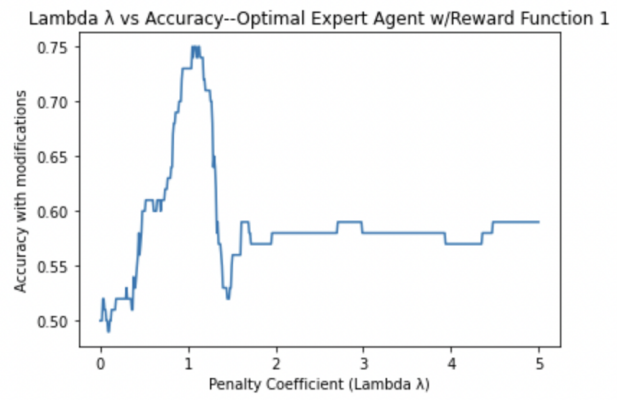
(c) Ground Truth Reward Function



(d) Optimal policy for extracted Reward function 1 after fixes

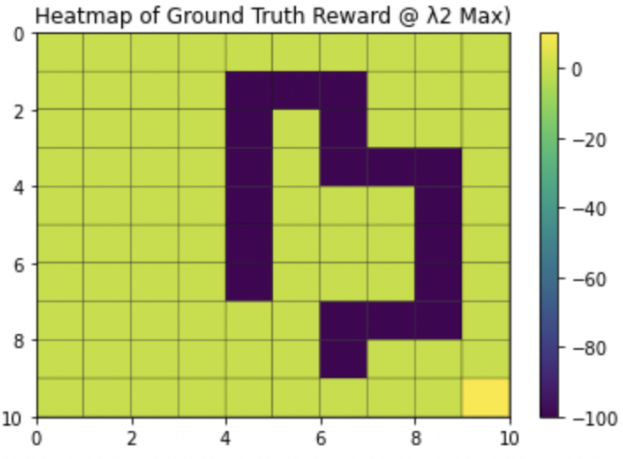


(e) Accuracy vs Lambda (Without improvements)

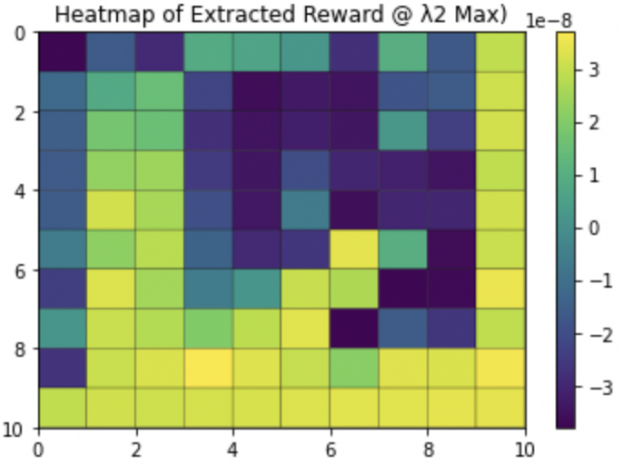


(f) Accuracy vs Lambda (With improvements)

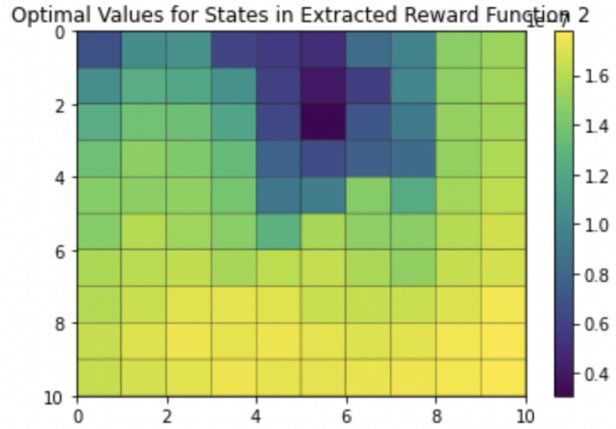
25.2 Effects of modifications for Reward Function 2



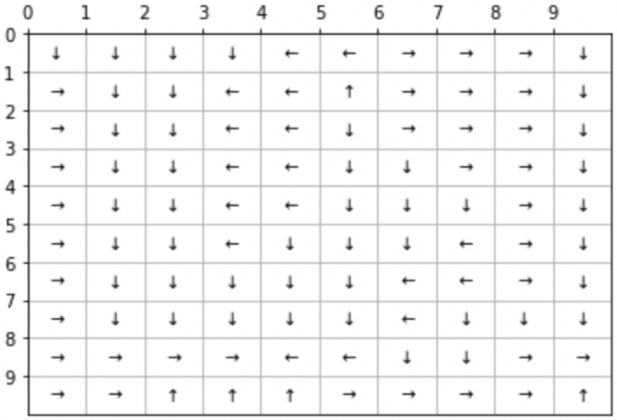
(a) Ground Truth Reward Function 1



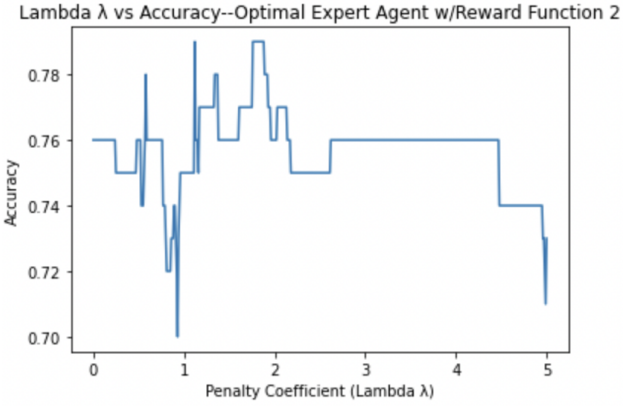
(b) Extracted Reward



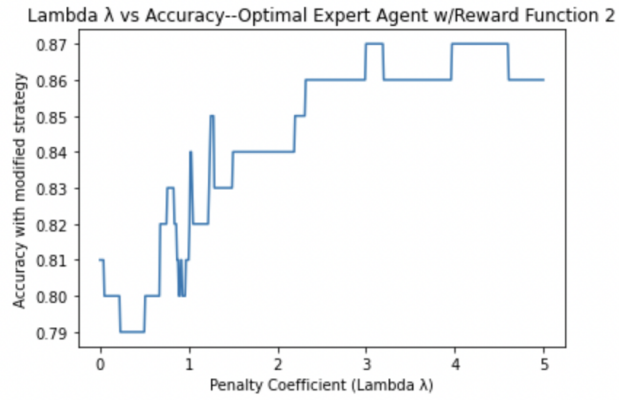
(c) Ground Truth Reward Function



(d) Optimal policy for extracted Reward function 2 after fixes



(e) Accuracy vs Lambda (Without improvements)



(f) Accuracy vs Lambda (With improvements)