ECE 232E Project 3
Reinforcement Learning and Inverse Reinforcement Learning
Spring 2019

Group Members:

Asavari Limaye	UID: 605224431	Pooja Janagal Nagaraja	UID: 405222664
Rupa Mahadevan	UID: 005225216	Vaishnavi Ravindran	UID: 805227216

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 first reward function and its heat map are reported below:

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

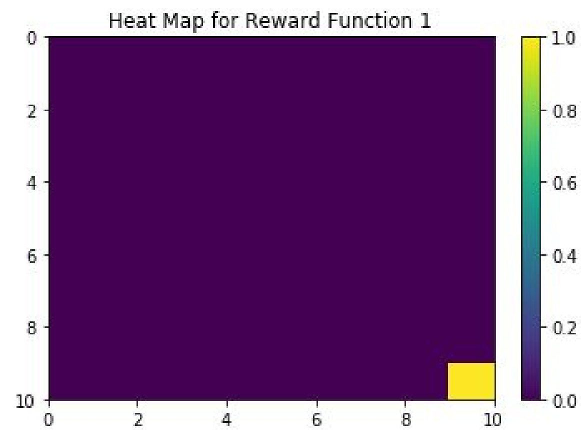


Figure 1.1: Heat Map for the first reward function

0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-100.0	-100.0	-100.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-100.0	0.0	-100.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	-100.0	0.0	-100.0	-100.0	-100.0	0.0
0.0	0.0	0.0	0.0	-100.0	0.0	0.0	0.0	-100.0	0.0
0.0	0.0	0.0	0.0	-100.0	0.0	0.0	0.0	-100.0	0.0
0.0	0.0	0.0	0.0	-100.0	0.0	0.0	0.0	-100.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	-100.0	-100.0	-100.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	-100.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.0

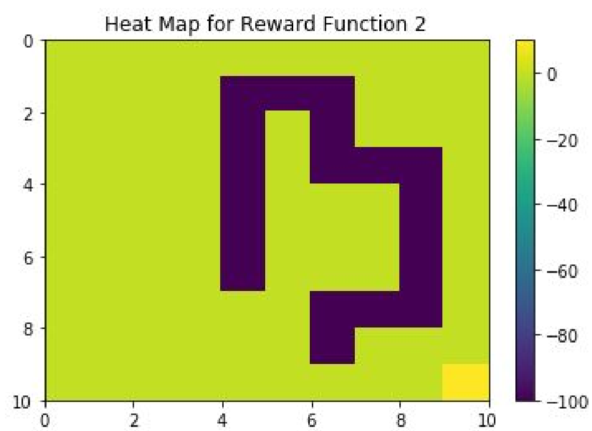


Figure 1.2: Heat Map for the second reward function

Question 2: 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.

An environment for the agent has been created in the following manner:

The state space has been set up as below:

0.0	10.0	20.0	30.0	40.0	50.0	60.0	70.0	80.0	90.0
1.0	11.0	21.0	31.0	41.0	51.0	61.0	71.0	81.0	91.0
2.0	12.0	22.0	32.0	42.0	52.0	62.0	72.0	82.0	92.0
3.0	13.0	23.0	33.0	43.0	53.0	63.0	73.0	83.0	93.0
4.0	14.0	24.0	34.0	44.0	54.0	64.0	74.0	84.0	94.0
5.0	15.0	25.0	35.0	45.0	55.0	65.0	75.0	85.0	95.0
6.0	16.0	26.0	36.0	46.0	56.0	66.0	76.0	86.0	96.0
7.0	17.0	27.0	37.0	47.0	57.0	67.0	77.0	87.0	97.0
8.0	18.0	28.0	38.0	48.0	58.0	68.0	78.0	88.0	98.0
9.0	19.0	29.0	39.0	49.0	59.0	69.0	79.0	89.0	99.0

The following 4 actions can be used by the agent: L = -10 R = 10 U = -1 D = 1

The discount factor, gamma, is set to 8
The weight, w, is set to 4

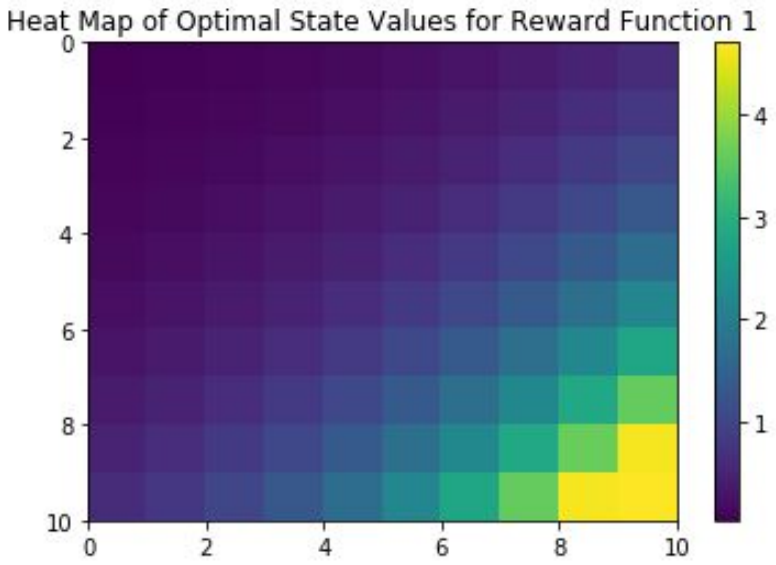
Using reward function 1, and the transition probabilities as defined in the specification, the optimal value for each state is determined using the Value iteration algorithm. The epsilon value or threshold used is 0.01. The optimal values obtained and the corresponding heat map have been reported below:

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 2.1: The optimal state values for reward function 1
obtained using value iteration algorithm

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

The heat map for the optimal state values are shown below



Question 4: Explain the distribution of the optimal state values across the 2-D grid.

We know from the reward function that the goal of the agent should be to move towards the bottom right. The reward function has no barriers or negative values. From Fig 2.1, it is seen that the optimal values keep increasing from left to right and top to bottom. The maximum optimal value is in the last cell at the bottom right. Hence, the optimal values at the bottom right are comparably larger than in the remaining grid. The optimal state values are indeed in accordance to what our intuition for the agent's moves should be like.

The values are also increasing along the diagonal of the grid which indicates that the optimal policy that the agent should use to get maximum reward is along the diagonal. It is also observed that the optimal values are symmetric across the grid since the agent has only one target and the top right and bottom left are at equal distances from the bottom right (target).

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 that the agent should follow has been reported in Figure 5.1.

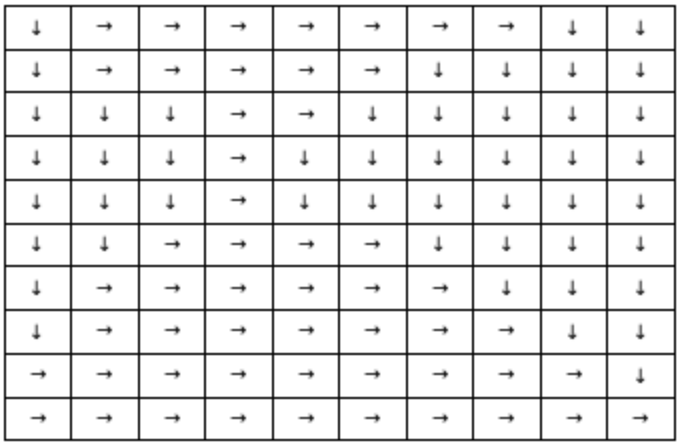


Figure 5.1: Optimal Policy for Reward Function 1

From Fig 5.1, it is seen that the optimal policy matches our intuition of how the agent should traverse through the grid. It is seen that the direction of the arrows along the diagonal ensures that the agent reaches the bottom right cell which is in accordance to the explanation of the distribution of the optimal values.

Yes, the agent can look at the neighbouring optimal values and decide to which cell it should move. The steps taken to reach that cell is given by the optimal policy. For example, in the optimal state values in Figure 2.1, when then agent begins in the state 0, the agent can choose either of its neighbours- right or down since they both have the same optimal policy. Once the agent makes this choice, the agent is pushed towards the diagonal since the maximum state value is for the state 11. From state 11, the agent can move either to the right or down. But once that choice has been made, the agent is pushed to the state 23. Thus, we can see that the neighbours can be used to determine where the agent should move. And by building on the previous steps, it is seen that the neighbours push the agent along the diagonal to reach the bottom right of the grid.

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 gure 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.

Fig 6.1 displays the optimal state values computed using the reward function 2 using the value iteration algorithm.

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 6.1: The optimal state values using reward function 2

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

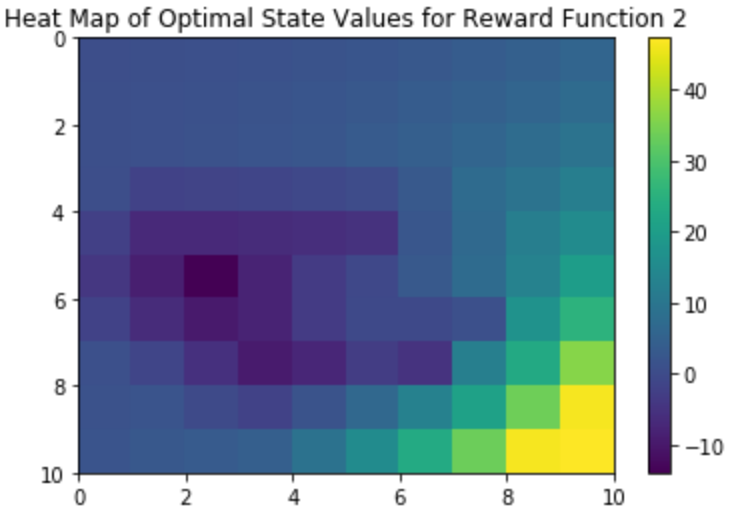


Figure 7.1: Heat map for optimal state values

Question 8: Explain the distribution of the optimal state values across the 2-D grid.

From Fig 6.1, it is seen that the optimal values include negative values as well since the reward function includes some negative values (or barriers). As in the previous case, the maximum reward is at the bottom right and hence the agent should learn to navigate to the bottom right of the grid. The optimal value is also the maximum for the bottom right.

There are several portions in the optimal values grid where there are negative values which implies that the agent will be prevented from going to those regions or penalized for it. It can

also be seen that the regions which have most negative values are not the cells with the negative reward itself but the neighbours. This prevents the agent from moving further and further along the wrong path. This is very intuitive.

It is also seen that the optimal values are not symmetric as in the previous case since there are negative rewards or obstacles along the path. The agent should learn to navigate by not nearing these regions. The obstacles themselves are not symmetric.

Question 9: 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.

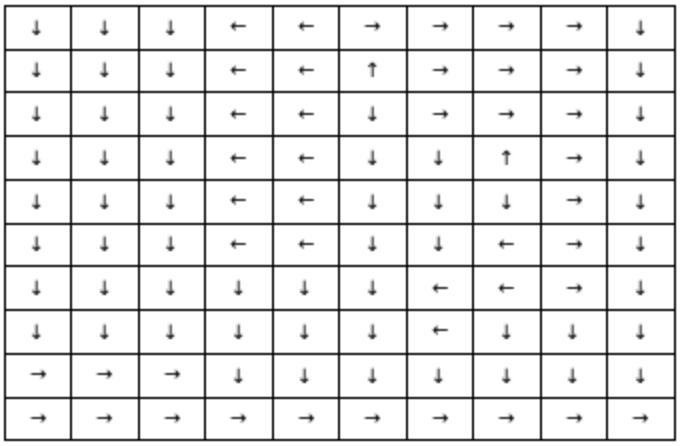


Figure 10.1: Optimal policy for agent with reward function 2

The optimal policy is in accordance with our intuition of the optimal state values and the reward function. It is seen that the agent is directed downwards and then to the right which is the best way to reach the cell with the highest reward. The agent cannot be directed to the right side of the grid initially since the negative rewards occupy the middle of the grid.

Another interesting observation that can be made is that the optimal policy diverges out of the region of negative rewards. In other words, the agent is pushed out of the region that contains the negative reward. This is also very intuitive since the agent's goal has to be to maximize the reward. This policy moves the agent out of the region where the reward reduces.

Hence the agent tries to move towards the bottom right while keeping away from regions of negative reward.

Question 10: Express c; x;D in terms of R, P_a, P_{a1}, t_i, u , and Rmax

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

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

$$D = \begin{bmatrix} \mathbf{I}_{|S|\times|S|} & \mathbf{0}_{|S|\times|S|} & -(\mathbf{P_{a1}} - \mathbf{P_a})(\mathbf{I} - \gamma\mathbf{P_{a1}})^{-1} \\ \mathbf{0}_{|S|\times|S|} & \mathbf{0}_{|S|\times|S|} & -(\mathbf{P_{a1}} - \mathbf{P_a})(\mathbf{I} - \gamma\mathbf{P_{a1}})^{-1} \\ \mathbf{0}_{|S|\times|S|} & -\mathbf{I}_{|S|\times|S|} & \mathbf{I}_{|S|\times|S|} \\ \mathbf{0}_{|S|\times|S|} & -\mathbf{I}_{|S|\times|S|} & -\mathbf{I}_{|S|\times|S|} \\ \mathbf{0}_{|S|\times|S|} & \mathbf{0}_{|S|\times|S|} & \mathbf{I}_{|S|\times|S|} \\ \mathbf{0}_{|S|\times|S|} & \mathbf{0}_{|S|\times|S|} & -\mathbf{I}_{|S|\times|S|} \end{bmatrix}$$

Question 11:

We get the following plot of lambda (ranging from 0 to 5 in steps of 500) vs derived policy accuracy after solving the equation above for the LP problem:

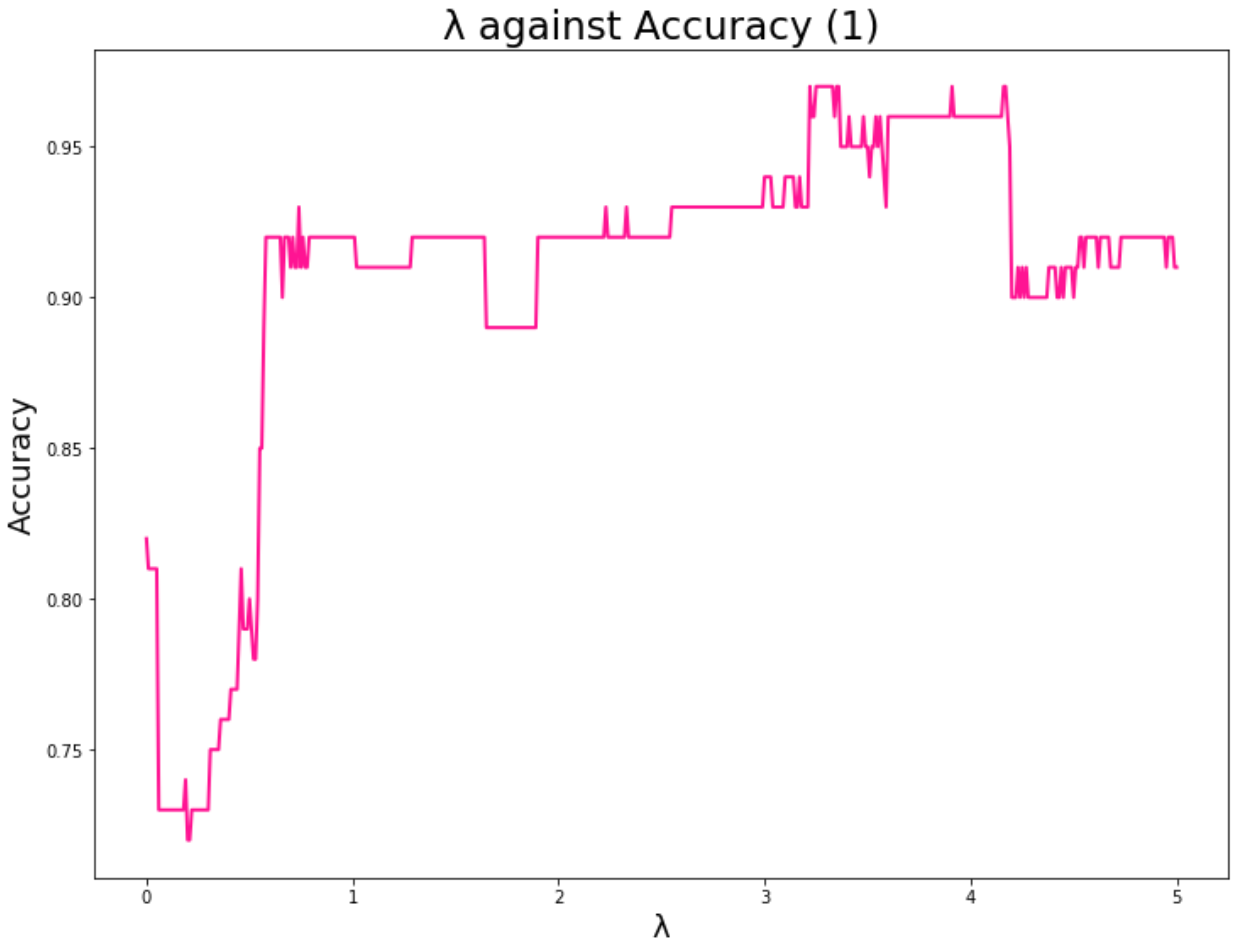


Figure. 8

Question 12:

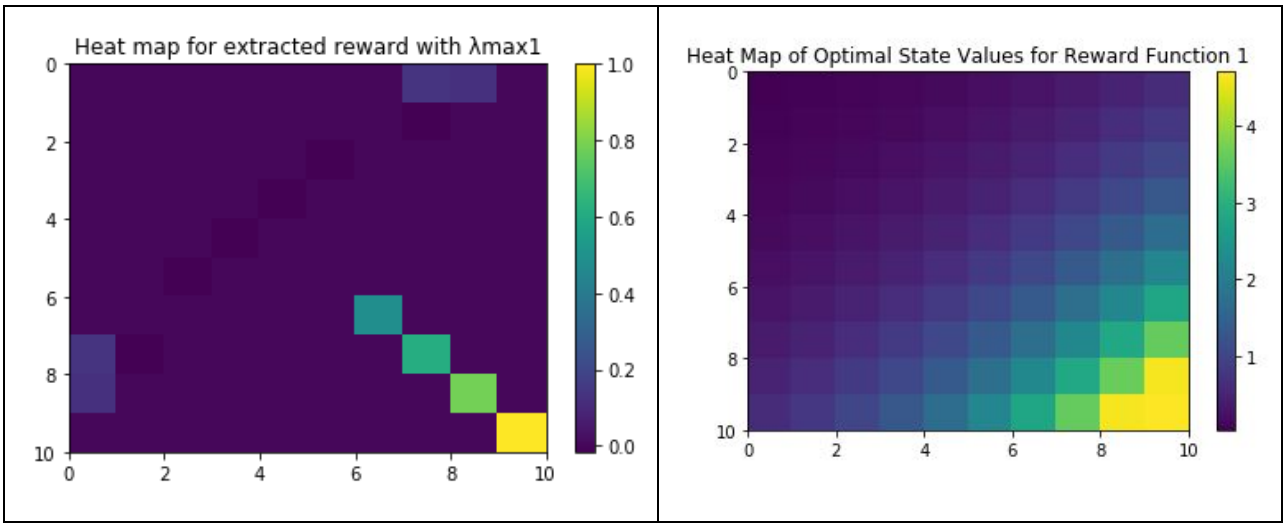
Since there are multiple values of lambda for which we get maximum accuracy we report the maximum of those values and cal this λ(1) max :

λ(1) max = 3.2199999999999753

Question 13:

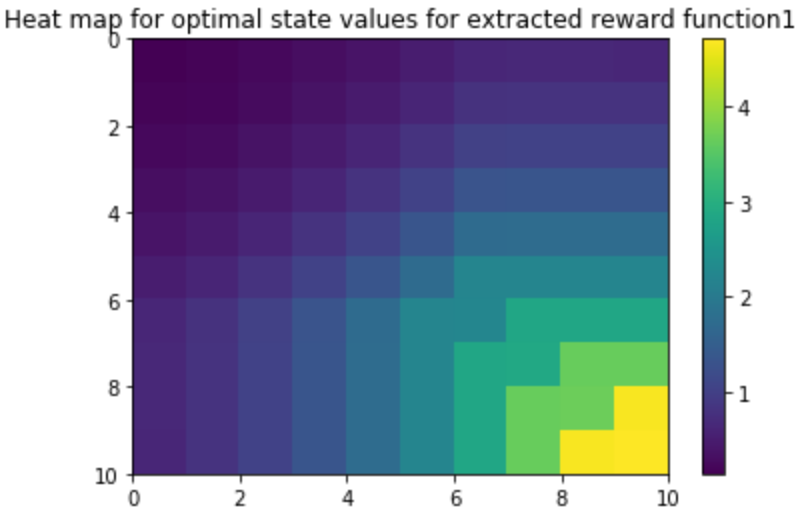
We know set λ = λ(1) max (3.2199) and then get the heat map for the extracted reward and compare it with the heat map for ground truth reward:

Derived Reward got from solving LP	Ground truth reward 1



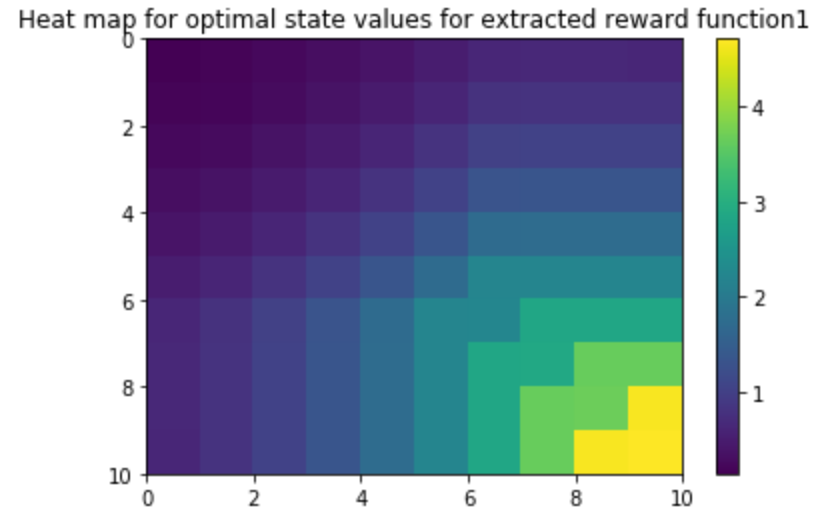
Question 14:

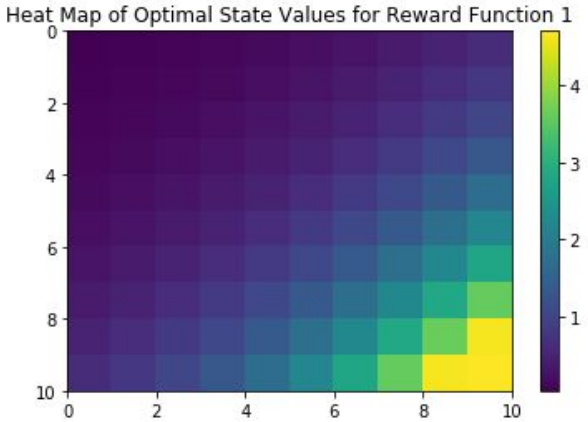
We report the heat map for the optimal state values got by running the value iteration algorithm



Question 15

We compare the heat maps of the optimal state values for the extracted reward function 1 and the ground truth reward function 1 below:





From these two heat maps, it is clear that both heat maps have a similar pattern or trend where we see it increasing from the top-left corner to the bottom right corner gradually. Since question 3 showed us that the heat maps for the extracted reward and ground truth reward, it makes sense that their optimal state values are also similar.

The primary difference between the two is that the ground truth reward has states with same value along the diagonal line from top left to bottom right but the extracted reward has similar state values in a diamond shape. The reason for this is because in the ground truth reward, there is exactly one state which has the highest reward and therefore this has an equal effect on other states.However, in extracted reward, there a four different states with gradually decreasing reward in the bottom right corner which have effects on the states around them resulting in a diamond shape.

Question 16:

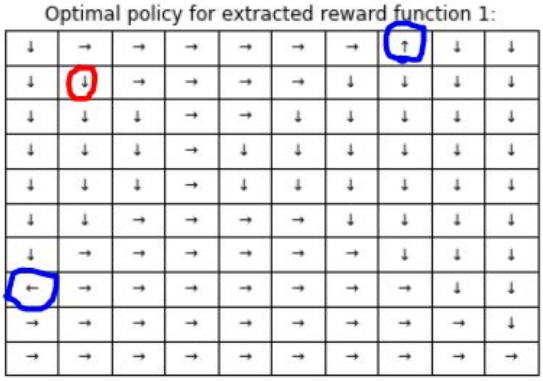
The optimal policy for the agent as got by the extracted reward function is :

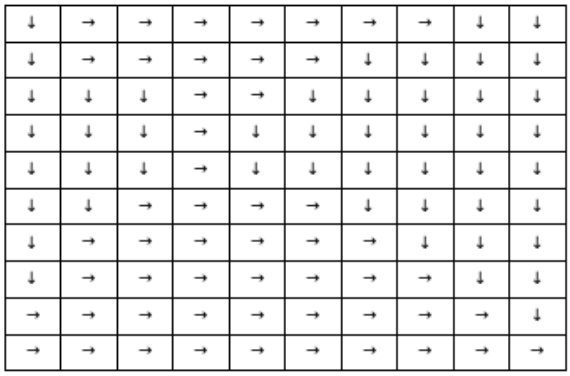
Optimal policy for extracted reward function 1:

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

Question 17:

Optimal policy of derived reward 1	*Optimal policy of ground truth reward 1*
Optimal policy for extracted reward function 1:	Optimal policy for ground truth reward function 1:





From the figures above we can see that the optimal policies for derived reward and ground truth reward are very similar except for differences shown with the circles. This makes sense since we compared the optimal state values in question 15 and saw that they were very similar indicating that the corresponding policies would be similar too.

There are few difference however as shown with circles. For cells (7,8) and (7,80) (in blue), if we compare the corresponding circles in these policy maps to the circles in the value function we see that they are caused due to differences in the extracted reward function. They are different since if we look at the optimal state values around them we notice that they are very small. As a

result, since these are edge states they stay in the current state to maintain the higher values because as they spread to their neighbours they decrease such that the values around them are lower than the states themselves. For the cell in red however, the difference is caused because for this state the values of two directions are the same , that is, this state has a value of 0.122 for both the right and bottom direction and hence it is not a true difference.

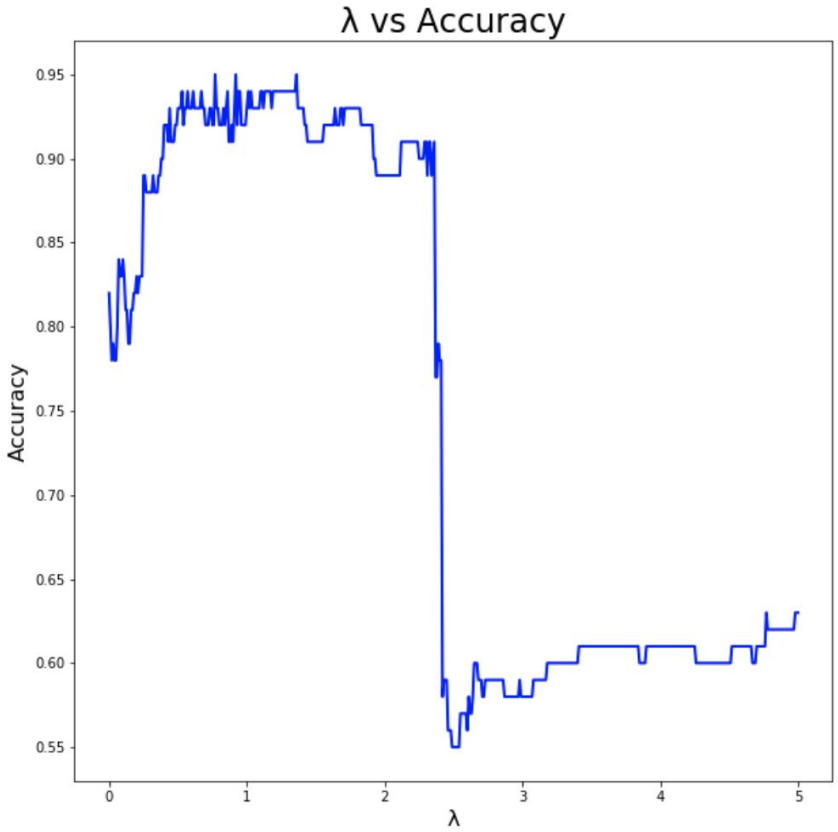
Question 18:

We have OA(s) depicting Optimal action of the agent at state s. OE(s) depicts Optimal action of the expert at state s. We get 500 evenly spaced values for λ between 0 and 5. For each value of λ, we calculate OA(s). We use the optimal policy of the agent from question 9 for OE(s) values. We use the following equation to compute accuracy:

m(s) = 1 if OA(s) = OE(s) else it is 0.

Accuracy = $(\sum_{s \in S}(m(s))/(|S|)$

We perform the above process for all 500 values of λ. We plot the λ values against the accuracies. The resulting plot:



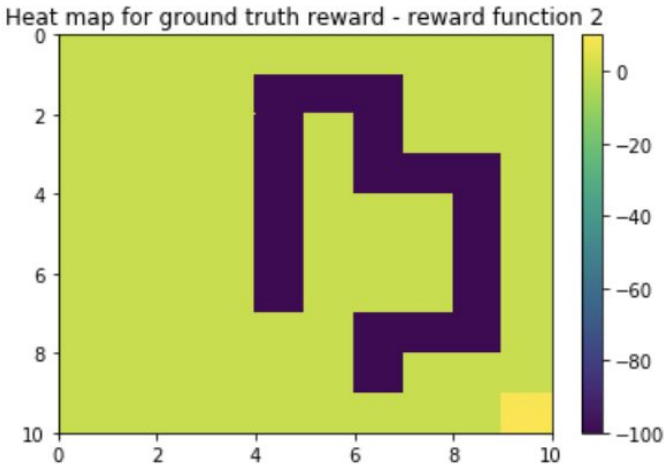
Question 19:

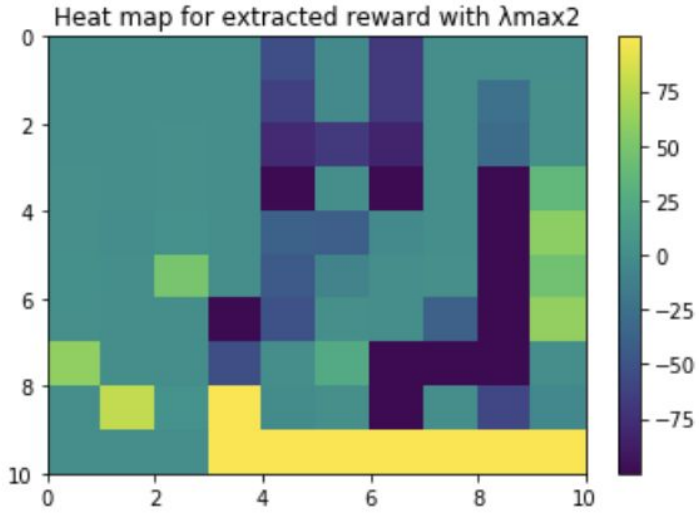
From the above plot, we can see that the maximum accuracy is 0.95 and the corresponding λ is 0.77. We will refer to this as $\lambda_{max}^{(2)}$.

$\lambda_{max}^{(2)} = 0.77$

Question 20:

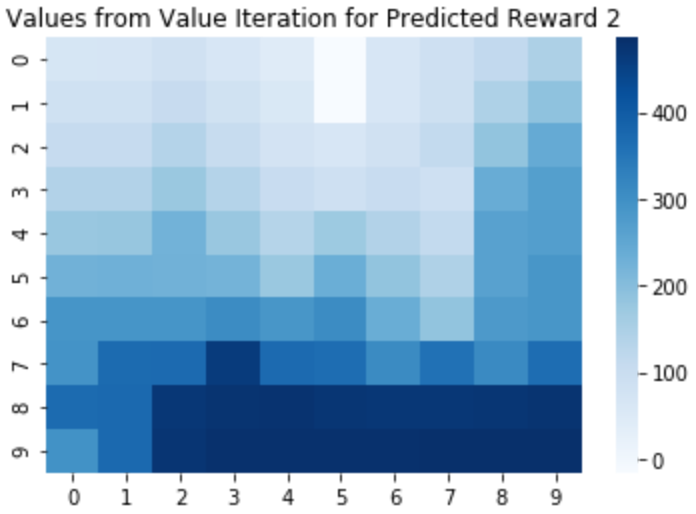
Here, we obtain two heatmaps. One for the ground truth reward which is the reward function 2. Another for the extracted reward which is calculated using $\lambda_{max}^{(2)}$ from above.





Question 21:

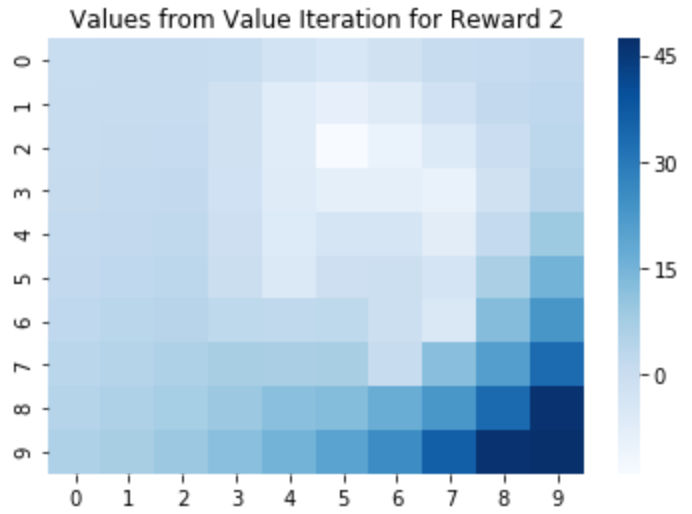
The following is the heat map plot of the values of the state obtained after performing value iteration on the predicted reward function. The reward function is predicted using Inverse Reinforcement Learning on the optimal policy of the expert agent. The policy of the expert agent is the policy obtained as a result of running value iteration on the provided reward reward2.

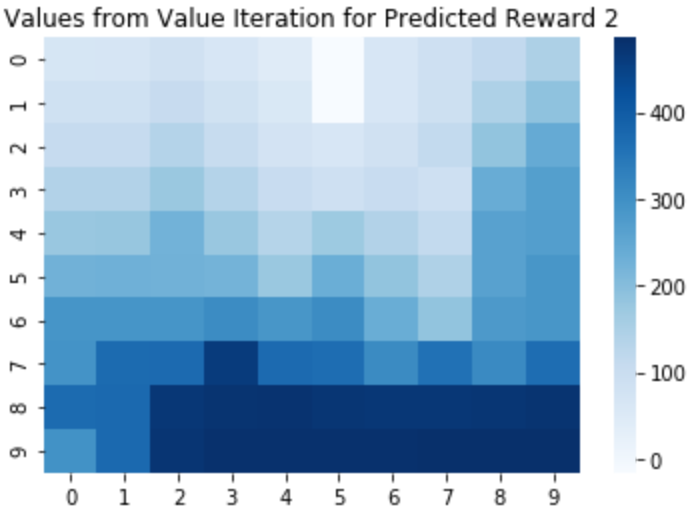


Question 22:

We compare the heat maps of

1. The value of states obtained from value iteration using the reward reward2.
2. The value of states obtained from value iteration using the predicted reward. The predicted reward was got using Inverse Reinforcement Learning on the optimal policy which was in turn found using value iteration algorithms using reward2.





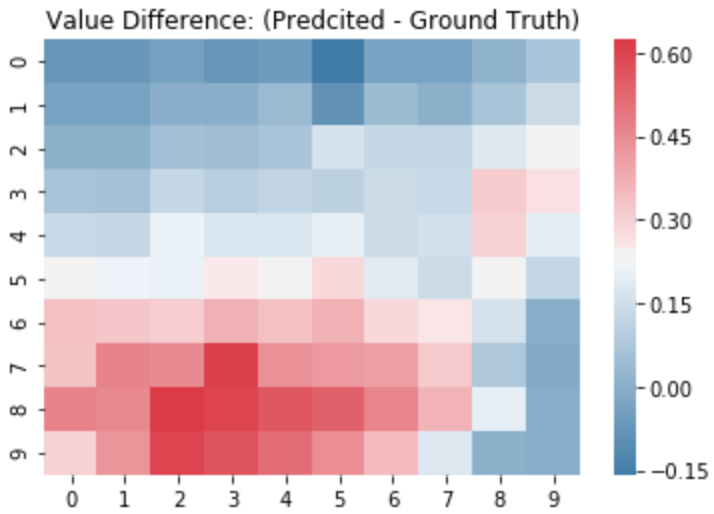
The scale of the values of the states got using value iteration using the predicted rewards is on a different scale compared to the values for using value iteration using the reward2. Inverse Reinforcement Learning is used to predict the reward of states based on the given optimal policy followed by an expert. Using just the policy, the algorithm can only predict the relative values of the values of the states and rewards of the states, and not the magnitude.

The original reward2 has a maximum state reward of 10 and a minimum state reward of -100. Given this, we can see that the converged value magnitudes derived using the value iteration are in a similar scale from -13.91 to 47.31.

The IRL algorithm has only the policy as the input, and the magnitude of the derived reward function is controlled using the penalty terms during constraint satisfaction. Because of this, the scale of the predicted rewards and the values of the states got using the predicted reward are different and larger to prevent insignificantly small rewards which satisfy the inequalities. The predicted values of the states are in the range 485.797735663887 -16.029960271859125.

This does not mean that the predicted values are wrong, as only the relative magnitude of values and rewards matters while deciding which policy to adopt. Using the same color map for the heat map shows the relative values of the value of the states, and the scale of the value does not matter.

The following heat map shows actual value of the states and the predicted value of the states after scaling both down to the range [0.0,1.0] and then taking their difference. We use a divergent color map to show the areas where the predicted values were larger than the ground truth (positive) and where they were lesser than the ground truth values (negative).



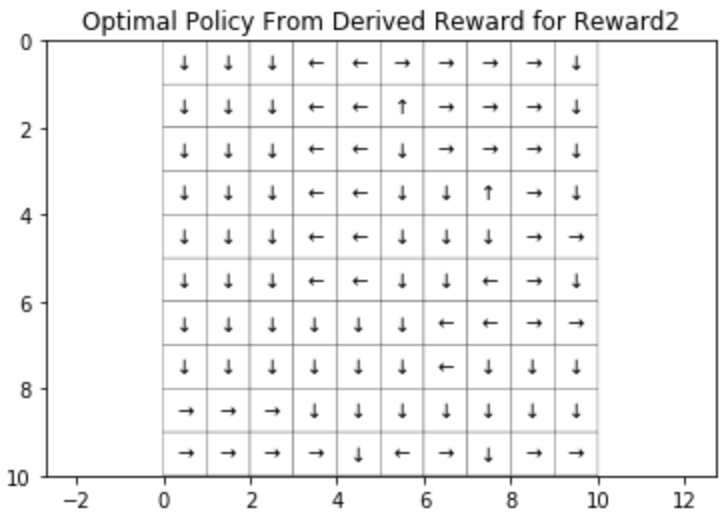
It can be seen that the prediction of the reward and values was more optimistic closer to the state with the highest reward. That is why the bottom of the plot, which is closer to (9,9) is positive. The predicted values were more than the ground truth here.

Further away from the state with high reward, near the (0,0) corner, the prediction was pessimistic, with the values assigned to state were lesser than the ground truth.

In the areas where the ground truth reward was highly negative (-100) the predicted values are similar to the actual values of the states from value iteration on reward2.

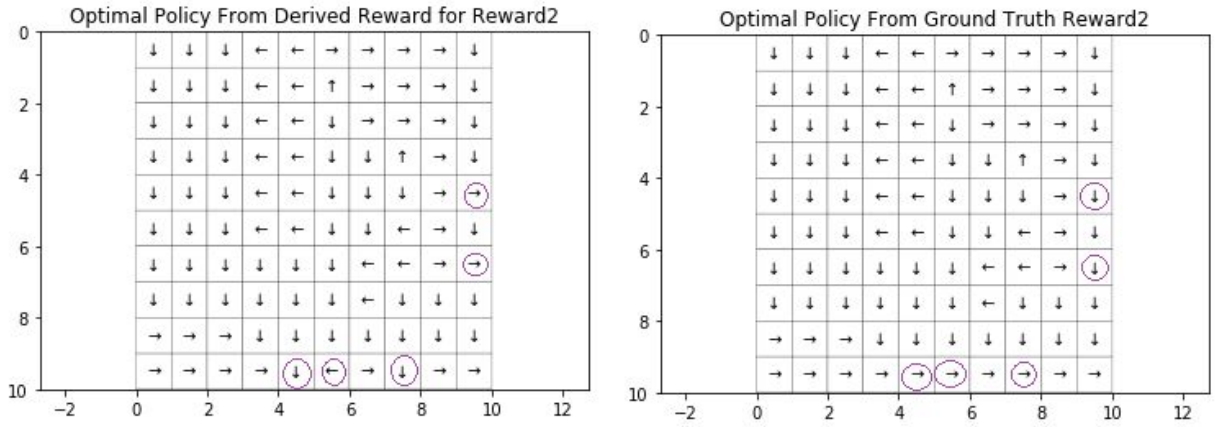
Question 23:

Optimal Policy in this part is calculated by running the value iteration algorithm on the reward obtained using the Inverse Reinforcement Learning on the policy of the expert agent. The expert agent's policy is the one obtained from value iteration using the provided reward reward2.

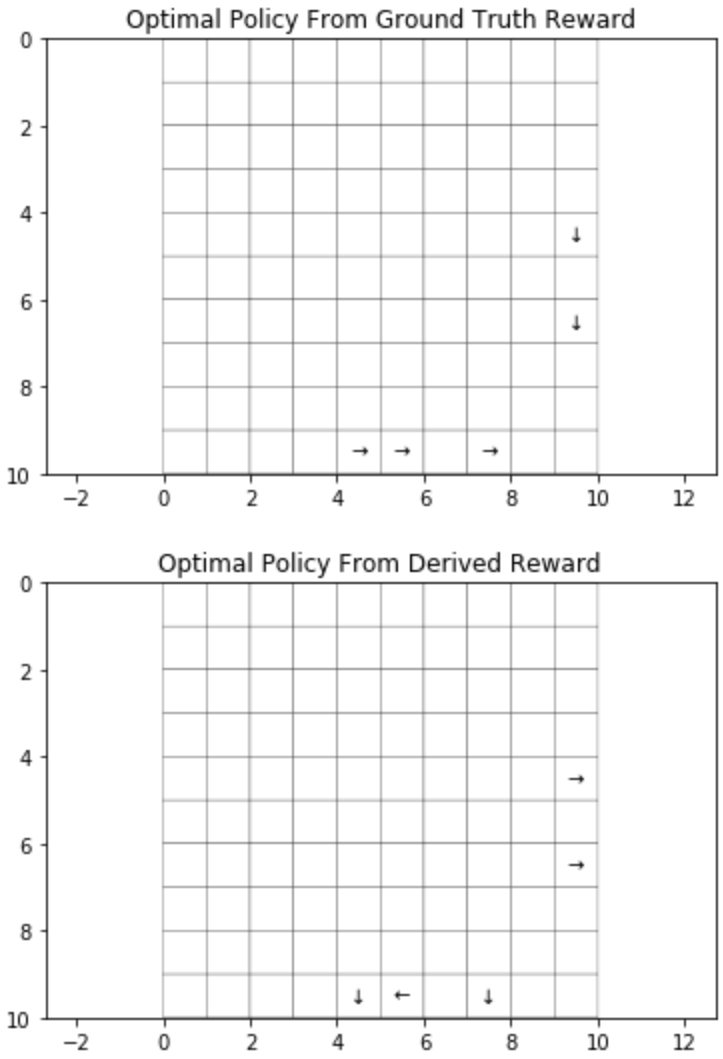


Question 24:

We now compare the policy obtained using the derived reward (Question 23), with the expert policy provided to the IRL algorithm (Question 9).



The following figure only shows the places where the expert policy and the policy obtained by value iteration on the predicted reward differs:



There are 5 places where the policy from the derived reward differs from the policy of the optimal agent from question 9.

In general, the discrepancies can be explained based on the differences in the predicted and ground truth value of the states. The arrows in general will point toward the states which have a higher value.

Question 25:

The discrepancies can arise when we terminate the value iteration algorithm before the values of the state have converged sufficiently to their true values. When the algorithm is terminated too early, the wrong unconverged values of the states may cause some differences in the policy, which leads to a wrong ground truth expert policy for the Inverse Reinforcement Learning algorithm.

Reducing the stopping threshold from 0.01 to 0.0001 gradually reduces the number of discrepancies to 0.

Threshold (e)	**Discrepancies**
0.01	5
0.001	4
0.0001	2
0.00001	0