ECE 232E Project 3 Report

Ken Gu, 904-836-308

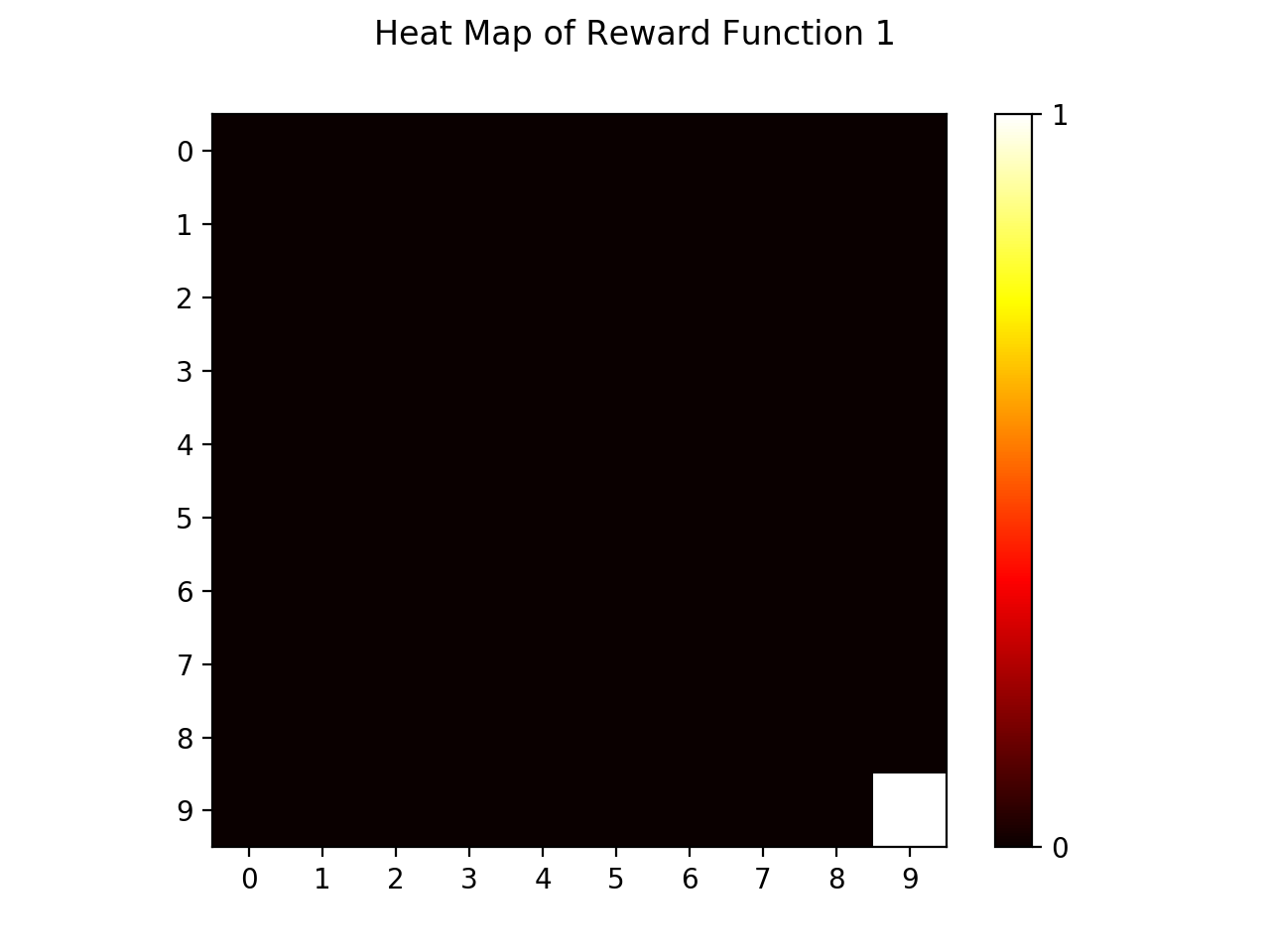
Yingbo (Max) Wang, 604-593-537

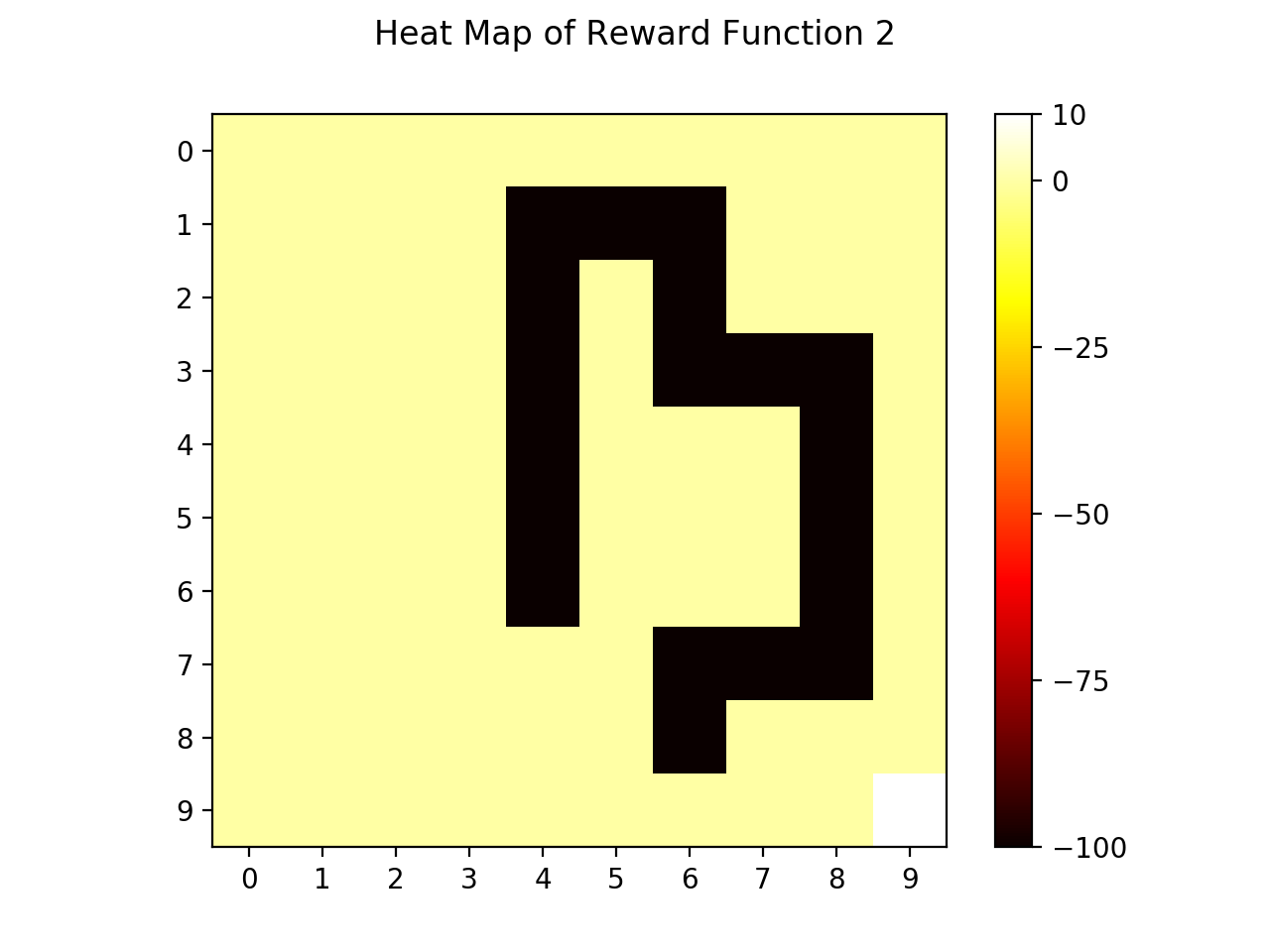
Ziying Feng, 404-395-859

Siqi Huang, 504-490-530

Part 1: Reinforcement Learning

Question 1



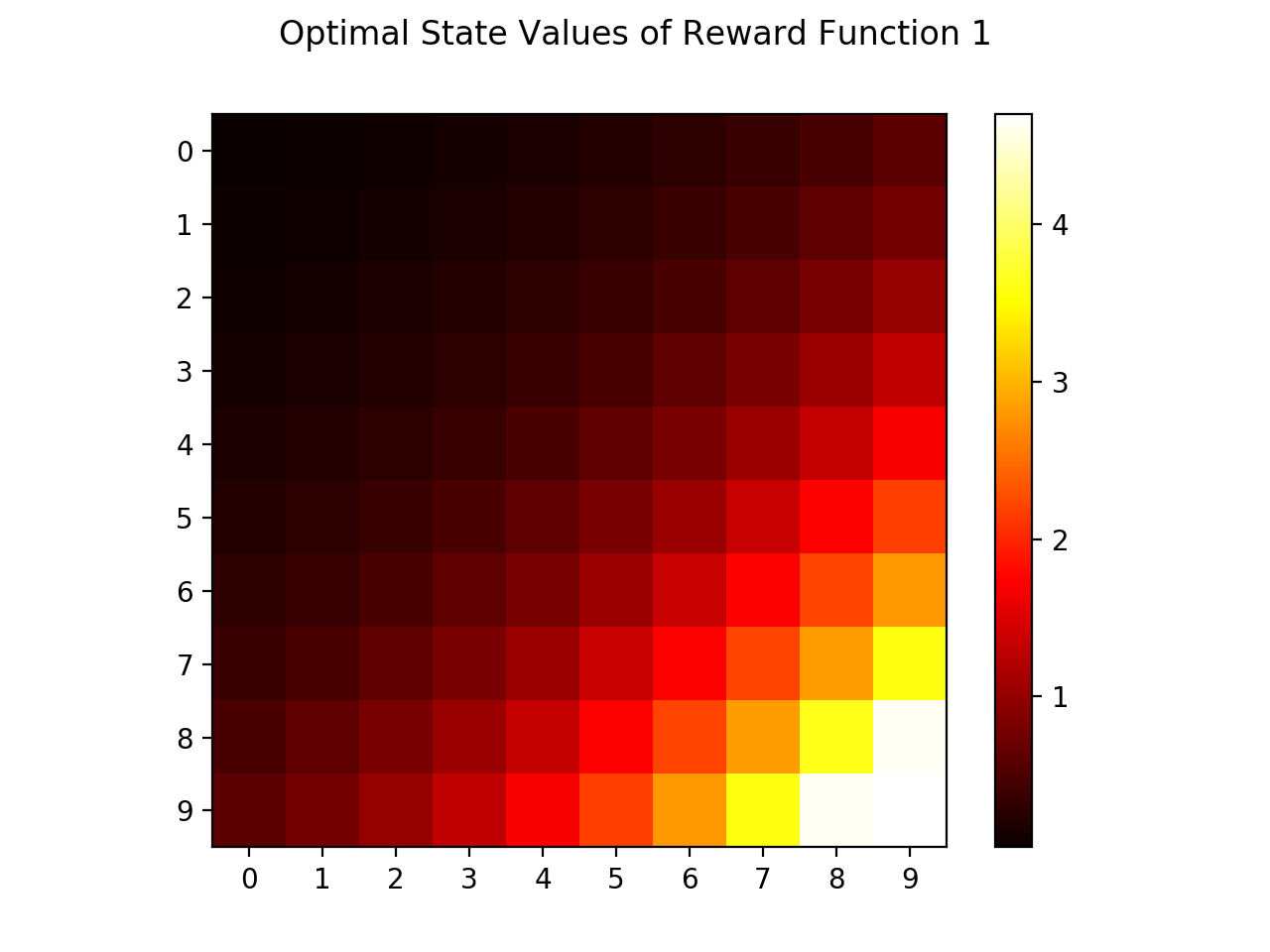


Question 2

Optimal State Values of Reward Function 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.042 | 0.063 | 0.09 | 0.124 | 0.167 | 0.222 | 0.291 | 0.379 | 0.491 | 0.61 |
| 0.063 | 0.088 | 0.122 | 0.165 | 0.219 | 0.289 | 0.378 | 0.491 | 0.633 | 0.787 |
| 0.09 | 0.122 | 0.164 | 0.219 | 0.289 | 0.378 | 0.491 | 0.635 | 0.817 | 1.019 |
| 0.124 | 0.165 | 0.219 | 0.289 | 0.378 | 0.491 | 0.636 | 0.82 | 1.052 | 1.315 |
| 0.167 | 0.219 | 0.289 | 0.378 | 0.491 | 0.636 | 0.82 | 1.054 | 1.352 | 1.695 |
| 0.222 | 0.289 | 0.378 | 0.491 | 0.636 | 0.82 | 1.054 | 1.353 | 1.733 | 2.182 |
| 0.291 | 0.378 | 0.491 | 0.636 | 0.82 | 1.054 | 1.353 | 1.734 | 2.22 | 2.807 |
| 0.379 | 0.491 | 0.635 | 0.82 | 1.054 | 1.353 | 1.734 | 2.22 | 2.839 | 3.608 |
| 0.491 | 0.633 | 0.817 | 1.052 | 1.352 | 1.733 | 2.22 | 2.839 | 3.629 | 4.635 |
| 0.61 | 0.787 | 1.019 | 1.315 | 1.695 | 2.182 | 2.807 | 3.608 | 4.635 | 4.702 |

Question 3



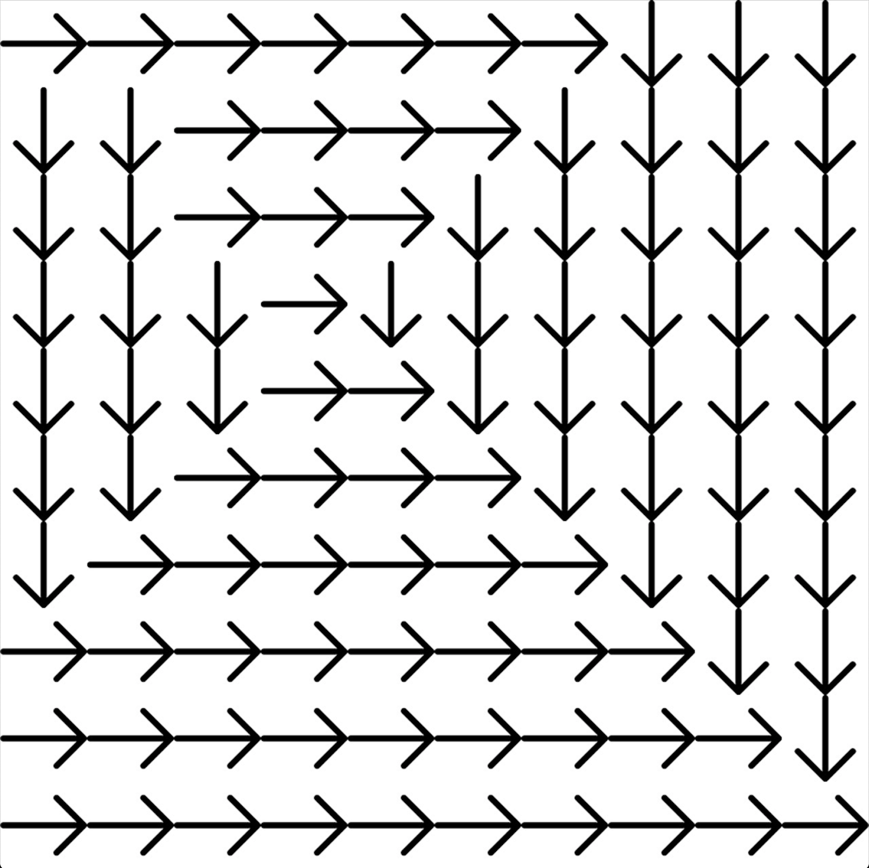
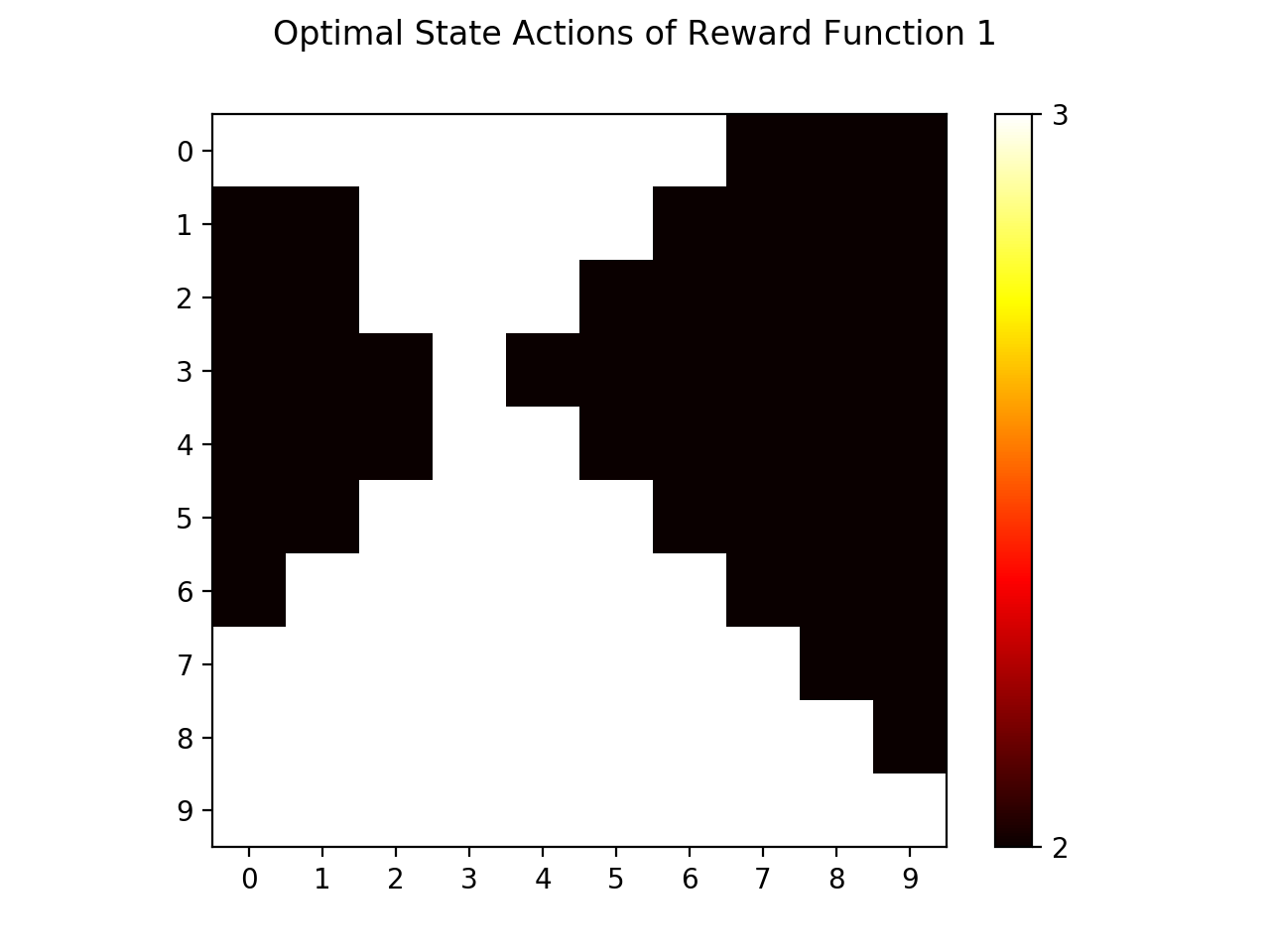
For this rewarding function, the maximum optimal state value equals 4.702 and the minimum optimal state value equals 0.042.

Question 4

The optimal value of states increases from left to right and from top to bottom. Consequently, the state on the top-left corner has the lowest optimal value and the state on the bottom-right corner has the highest value.

This result makes sense according to the reward function. It gives all but the bottom-right state a reward value of zero and the bottom-right state a positive value 10. The optimal value of a state represents the expected return starting from that state if following the optimal policy of action (s). Thus, intuitively it measures how “good” it is to be in each state. In this case, the best state is to be in is the lower-right state since it has the highest reward value among all states. All other states have the same reward value 0 (no reward and no penalty) but the discount rate is smaller than 1, which means reward values that take more state transitions to get are worth less. Thus, states that are farther away from the bottom-right state have less optimal value, and the state-value table should be symmetric by the diagonal. Our experimental result corresponds to these findings.

Question 5



The color of each grid represents the optimal action at that state with the following mapping: 0 - Going Up, 1 - Going Left, 2 (Black) - Going Down, 3 (White) - Going Right. Note that the actor should only go down or right according to this reward function.

The optimal policy of the agent at each state matches our intuition. First, since the only non-zero reward occurs at the bottom-right state with a positive value, the agent at all other states should be moving towards it, so the action can either be Right or Down. States that are closer to the grid’s bottom edge have the optimal action of moving Right and those closer to the grid’s right edge have the optimal action of moving Down. Also, all other states have reward value 0 so the optimal state actions are symmetric across the diagonal.

It is possible for the agent to compute the optimal action to take at each state by observing the optimal values of its neighboring states. Consider each iteration for one state in the Computation step of the value iteration algorithm: the optimal state action (s) is given by:



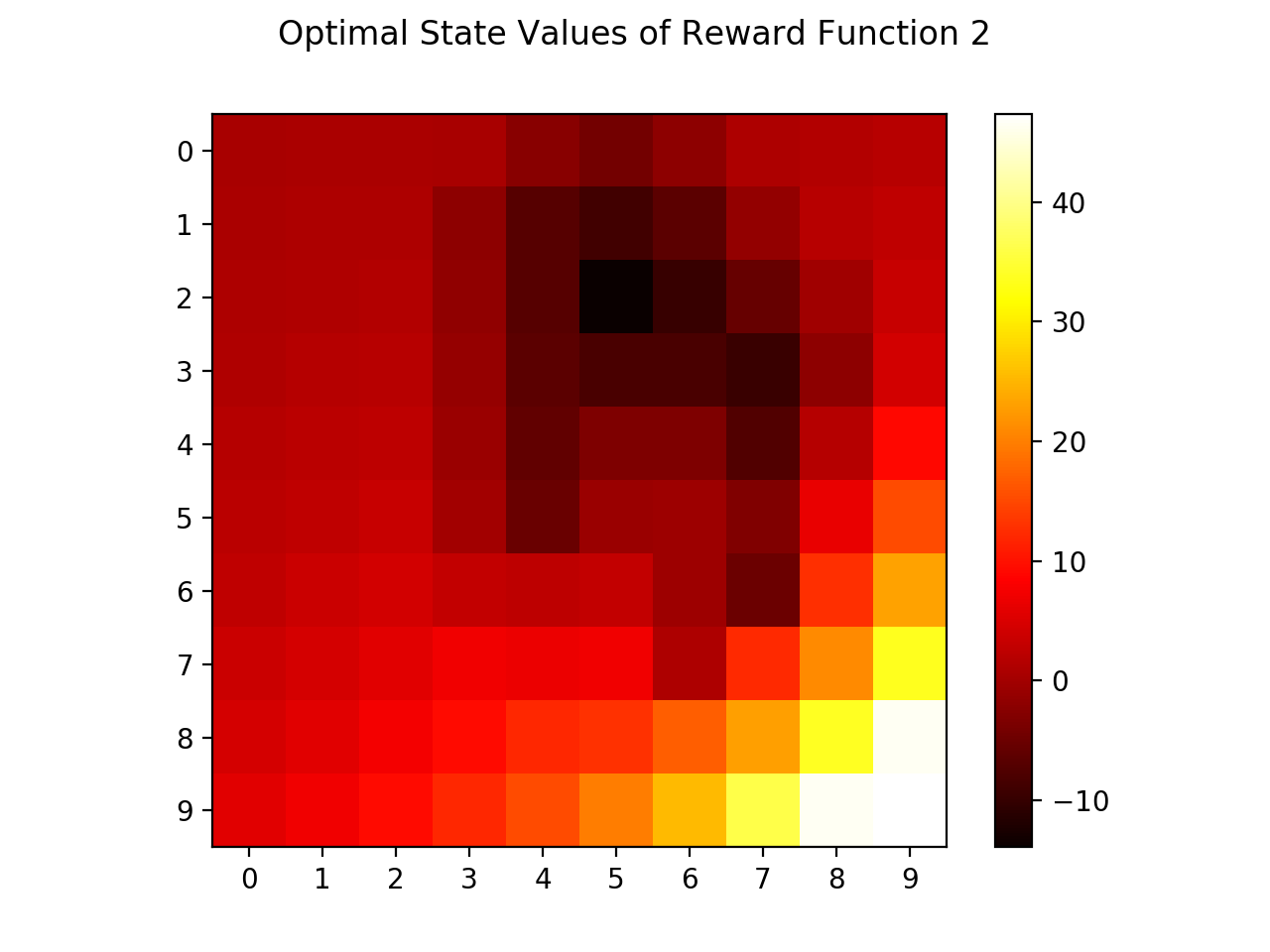
Although the value is summed over all states s', the transition probability to state s' given state sand action a equals 0 for all non-neighboring states. Thus, by knowing the optimum value of neighboring states V(s'), their reward values Rass'=R(s') and the transition probability Pass', it’s enough to calculate (s) at any state *s*.

Question 6

Optimal State Values of Reward Function 2

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.648 | 0.794 | 0.825 | 0.536 | -2.37 | -4.234 | -1.921 | 1.131 | 1.594 | 2.038 |
| 0.83 | 1.021 | 1.066 | -1.868 | -6.738 | -8.674 | -6.37 | -1.295 | 1.928 | 2.61 |
| 1.064 | 1.317 | 1.45 | -1.624 | -6.742 | -13.911 | -9.649 | -5.511 | -0.131 | 3.359 |
| 1.36 | 1.693 | 1.948 | -1.232 | -6.323 | -7.978 | -7.937 | -9.424 | -1.914 | 4.391 |
| 1.737 | 2.172 | 2.59 | -0.726 | -5.831 | -3.254 | -3.23 | -7.419 | 1.719 | 9.163 |
| 2.214 | 2.781 | 3.417 | -0.028 | -5.099 | -0.549 | -0.477 | -2.968 | 6.587 | 15.357 |
| 2.819 | 3.557 | 4.482 | 3.028 | 2.484 | 2.884 | -0.455 | -4.895 | 12.692 | 23.3 |
| 3.587 | 4.543 | 5.796 | 7.292 | 6.722 | 7.245 | 0.941 | 12.37 | 21.163 | 33.486 |
| 4.561 | 5.798 | 7.401 | 9.443 | 12.012 | 12.893 | 17.101 | 23.018 | 33.782 | 46.532 |
| 5.73 | 7.32 | 9.391 | 12.048 | 15.456 | 19.828 | 25.501 | 36.161 | 46.587 | 47.315 |

Question 7

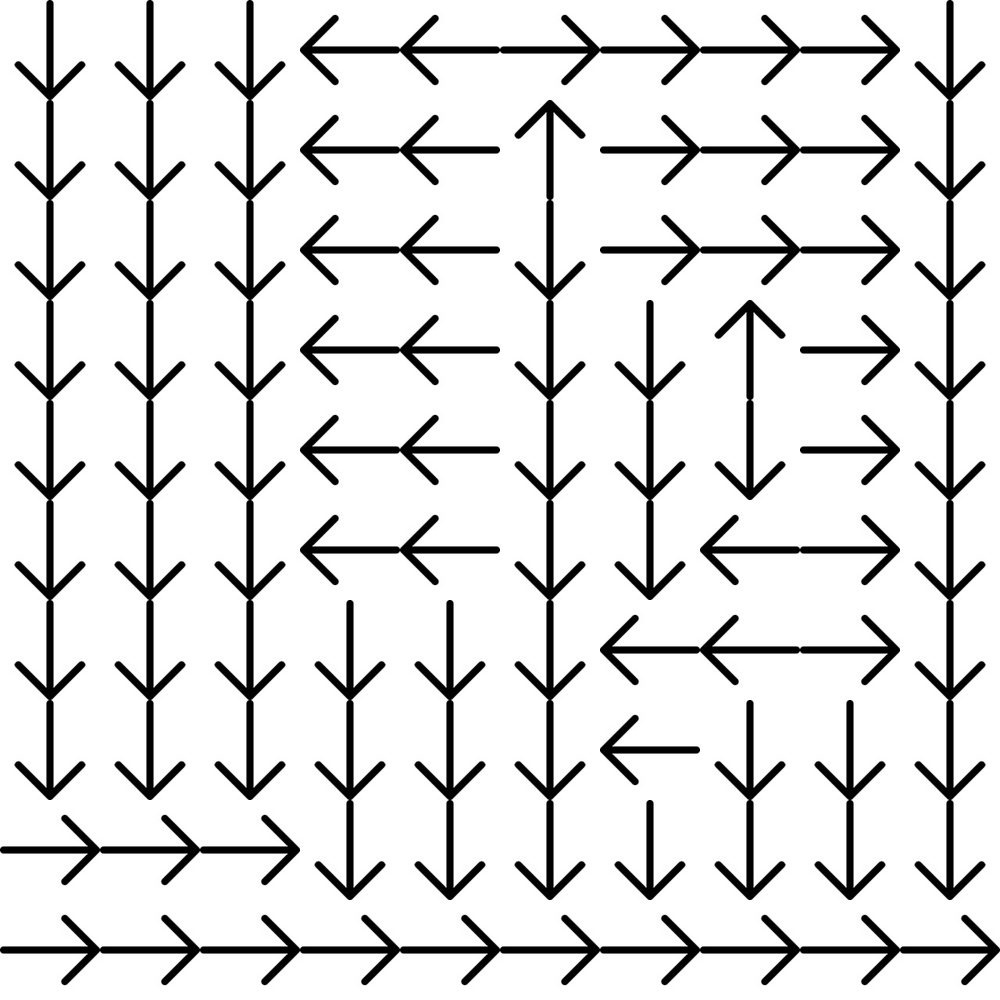
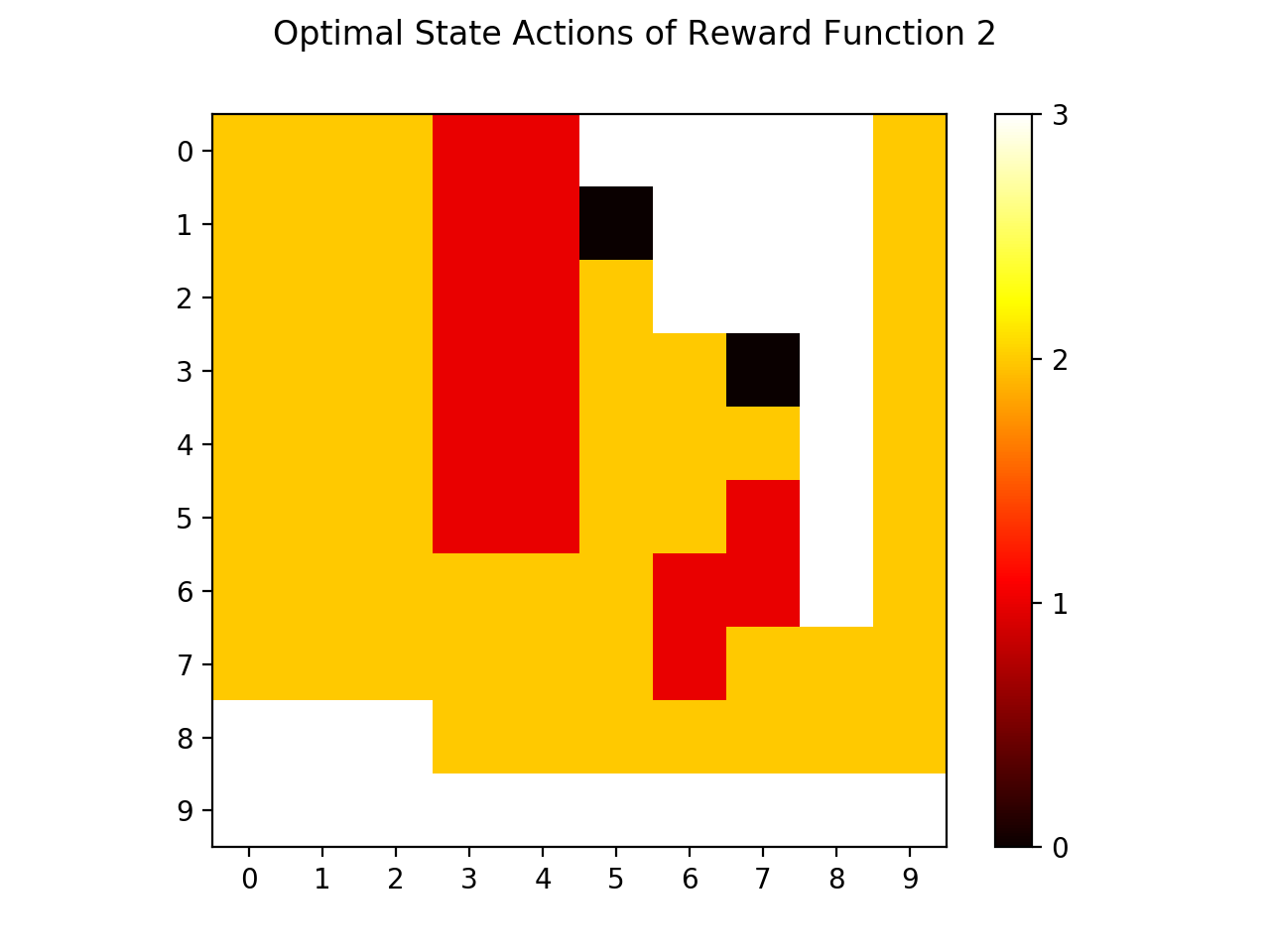


For this rewarding function, the maximum optimal state value equals 47.315 and the minimum optimal state value equals -13.911.

Question 8

The measured optimal state values match our intuition. First, states that are close to the bottom-right state (#99) have the highest values among all others: they’re close to the only state with positive reward, so the expected returns of starting from these states are high. Moving towards the left and top part of the grid, states begin to have decreasing optimal values due to the discount of that positive reward after taking an increasing amount of transitions to reach the reward state. Particularly, states near the upper central part of the grid have optimal values lower than their enclosing states and most of them have negative expected returns. That’s because they’re surrounded by states with high penalties (negative reward values). State #52 has the lowest expected return value as it has three neighboring states on the Left, Top and Right that have negative rewards. Starting at this state, the actor has high chance of reaching these “penalty states” due to wind and thus reducing the expected return.

Question 9



The color of each grid represents the optimal action at that state with the following mapping: 0 (Black) - Going Up, 1 (Red) - Going Left, 2 (Orange) - Going Down, 3 (White) - Going Right.

The optimal policy matches our intuition. Starting from states that are positionally outside those with negative rewards, the actor either moves Down or Right with the goal to reach the bottom-right “reward state” with positive reward. States that are close to the grid’s bottom edge have the optimal action to move Right and those close to the grid’s right edge have the optimal action to move Down. States that are either positionally enclosed by “penalty states” or close to these “penalty states” have the optimal action to move out and away of the enclosing rectangle (row 1 to 8, column 4 to 8). Note that the actor should stay away from any penalty state by at least 1 state due to winds potentially blowing it off to any adjacent state. This explains why states #30 - #35 have the optimal action to move Left rather than move Down towards the reward state.

Question 10

We assume that we are maximizing subject to as opposed to

which one of the TAs during discussion mentioned we can do. Also, here is the number of states. Also, our matrices already consider part of the equation

Let us define the vector as follows:

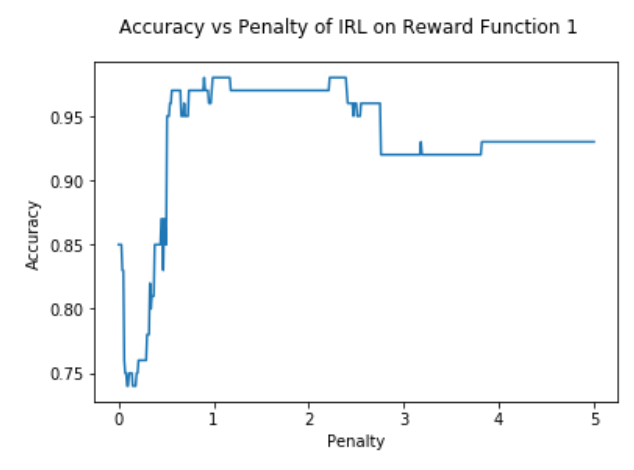
Then our vector is defined as such

Likewise, matrix is defined as such

where

Finally, is defined as

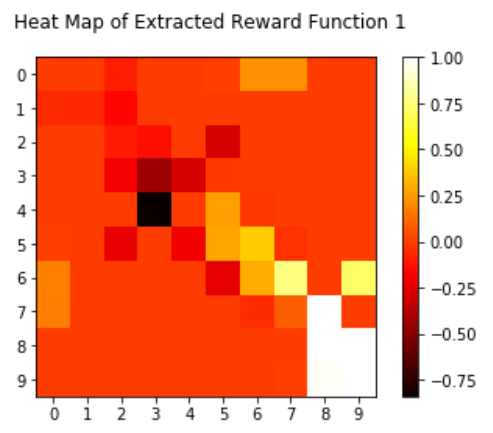
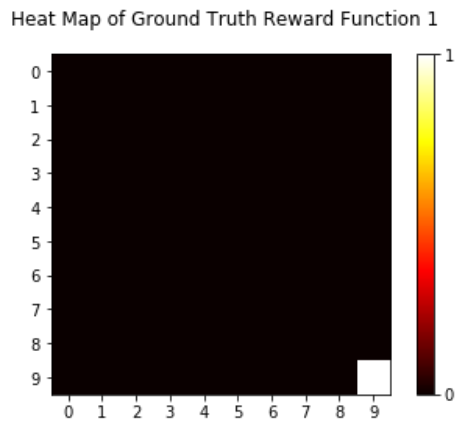
Question 11



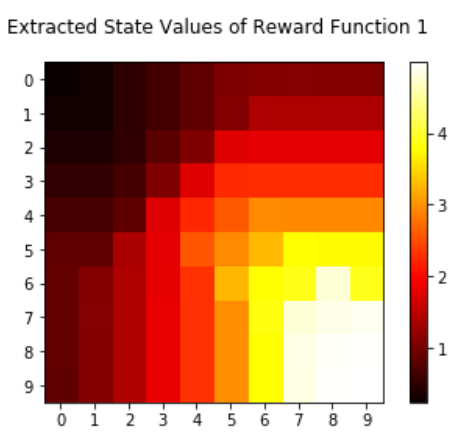
Question 12

The max accuracy for a given penalty in the range of 0 to 5 occurs at several places. The smallest penalty for which we get the max accuracy is 0.90. The accuracy at this penalty is 0.98. For the rest of the report

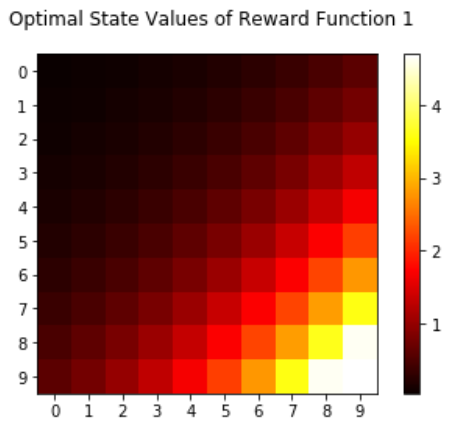
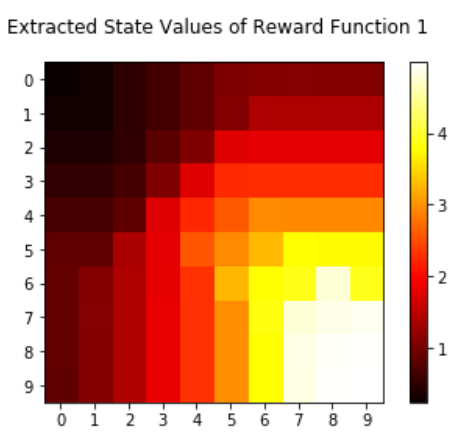
Question 13



These heatmap on the right is the result of the IRL optimization problem with

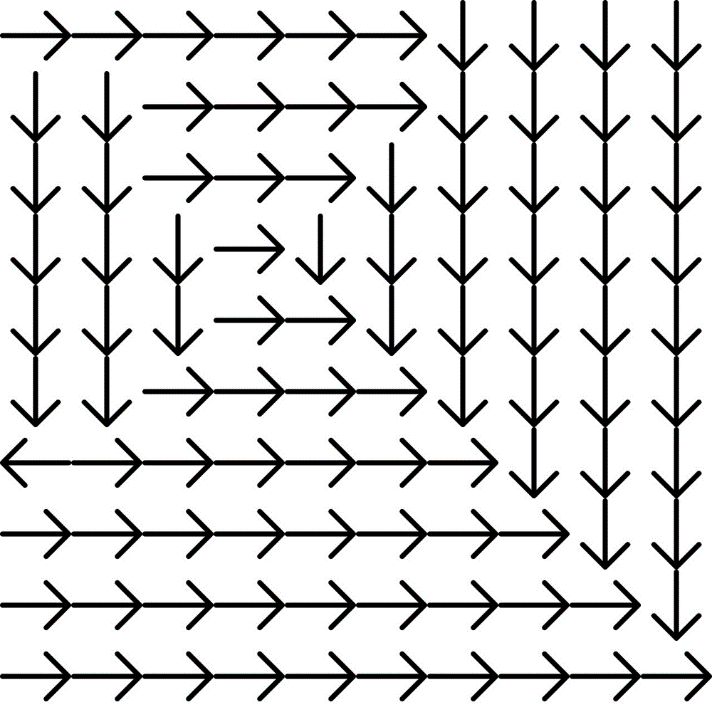
Question 14

Question 15



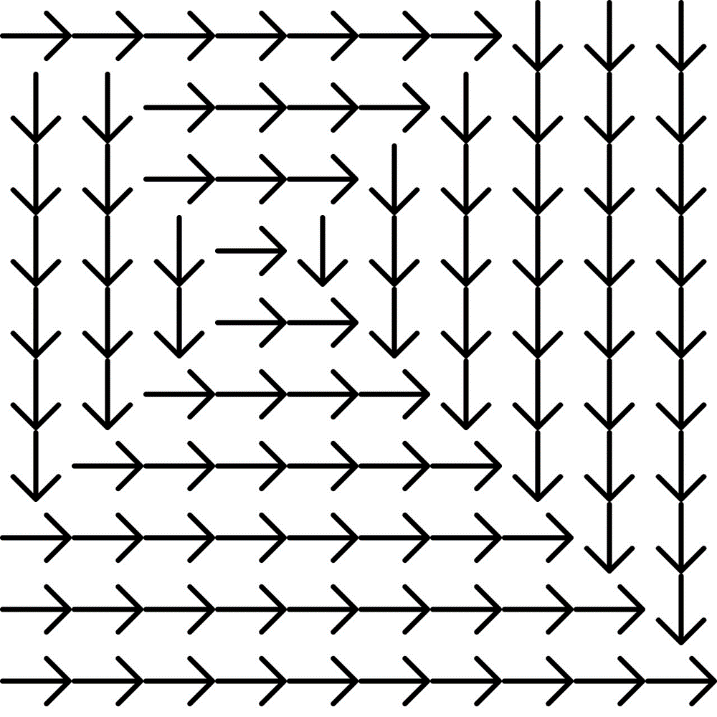
From the figure above comparing the optimal state values and the extracted state values we see that the general trend is the same. More specifically both heatmaps increase in value as we approach figure state 99 and decrease in value the closer, we are to state 0. This makes sense as the plot in question 13 showed the optimal reward function and the extracted reward function are also very similar. However, it must also be noted that the optimal state values the same value states are along the diagonals (bottom right to top left) while for the extracted function same state values generally have the pattern of around the borders of an increasing box starting from the bottom right corner. The diagonal pattern can be explained by how the reward function is the same everywhere except state 99. Thus, all equidistant states from state 99 should have the same state value function. Meanwhile the extracted reward function is not as uniform in every other state than state 99 but is still uniform outside of the main top left to bottom right diagonal (which has higher rewards than the surrounding non diagonal states).

Question 16

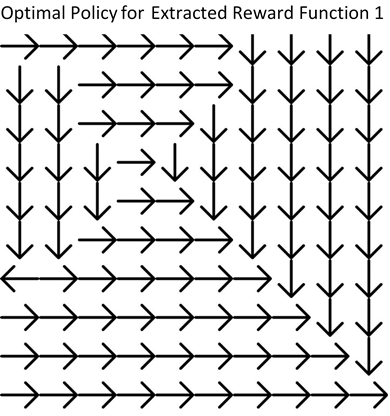


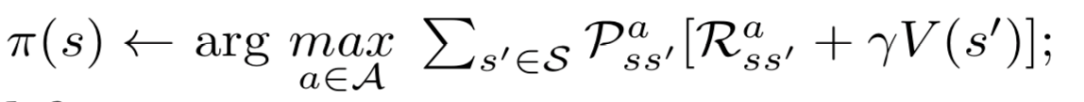
Optimal Policy for Extracted Reward Function 1

Question 17

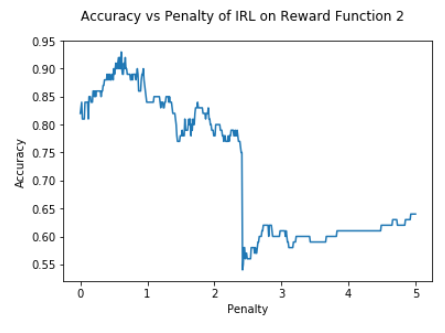


Optimal Policy for Ground Truth Reward Function 1



From the two figures above, we can see that the actions are almost the same except for states 6 and 60 as marked by the red circle. The states are very similar because the state value functions are similar, guiding the agent to the goal state. The difference in state 6 can be explained from the extracted reward function in question 13 where the reward is higher than its surroundings. From question 14 we see that at state 6 a higher state value is in state 16, however when we perform the computation of the value iteration algorithm as shown

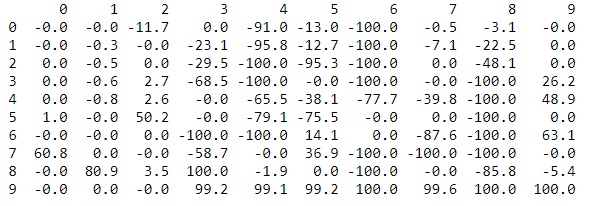
the difference between the reward at state 6 and its surrounding states is greater than that of the difference between the values at the surrounding states and state 6 such that the agent chooses to go off the grid to stay in this state and receive the award. Likewise, for state 60 we can see from the figure in question 14 and 15 that there is a greater difference between the extracted state value for state 61 and the extracted state value for state 70 than the difference between the ground truth state value for state 61 and the ground truth state value for state 70. This difference is what cause the IRL agent to choose to go down.

Question 18

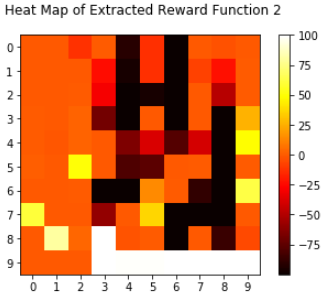
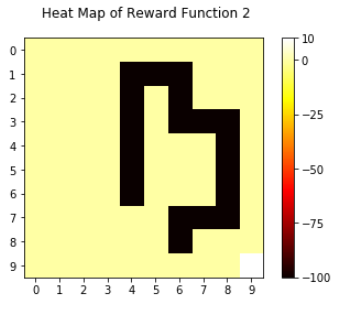
Question 19

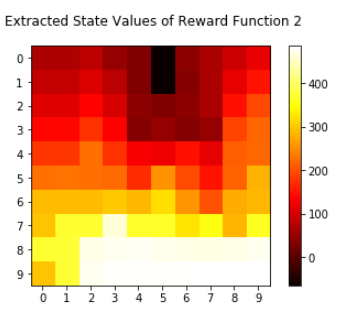
The max accuracy for a given penalty in the range of 0 to 5 occurs when the penalty is 0.61 in which the accuracy is 0.93. For the rest of the report

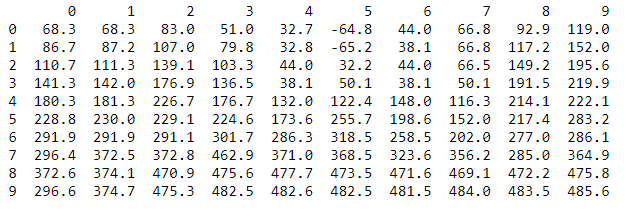
Question 20



Reward Values for Extracted Reward Function 2

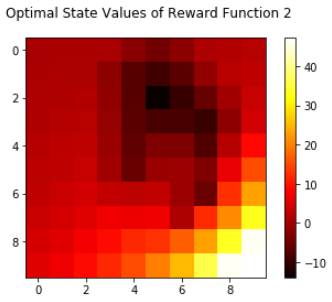
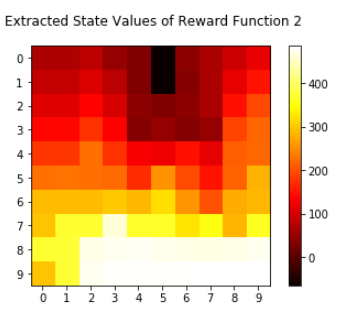


Question 21



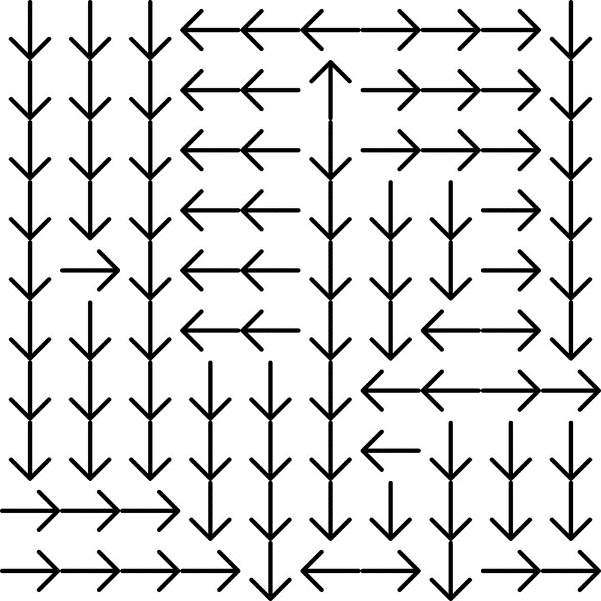
State Values for Extracted Reward Function 2

Question 22



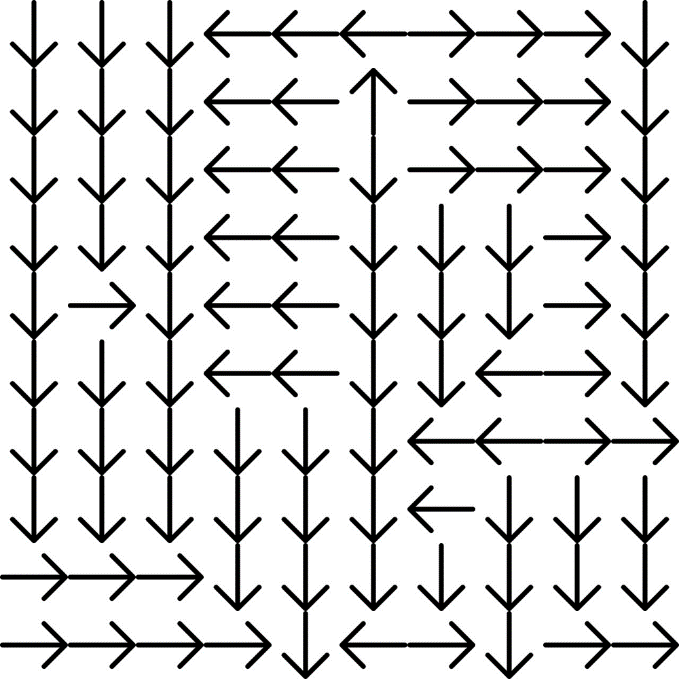
We can see what both functions generally have higher values towards the bottom right and lower values toward the top with an area near the top with significantly lower values (blackish area). They both offer the highest state values in states surrounding state 99. However, we also note that the scale is at a broader range for the extracted state values. This can be explained by the fact that the extracted reward function has a higher upper bound for values as the optimization problem was being constrained by the absolute value of the reward being less than 100. Likewise, for finer areas of the grid, the state value patterns are more different. The ground truth still has the diagonal pattern mentioned in question 15 while the extracted state values are generally equal along the same row. Moreover, the darkest or smallest state values do not match exactly with the smallest extracted state values shifted up compared to the ground truth.

Question 23

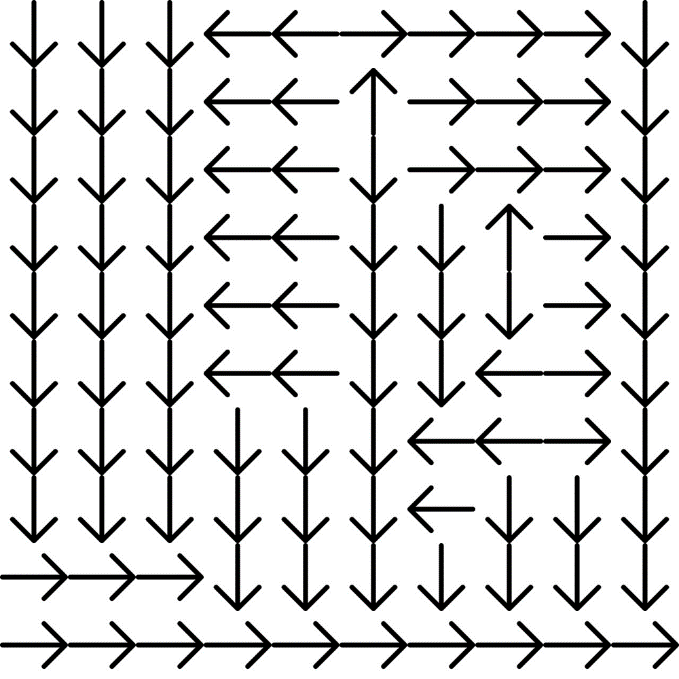


Optimal Policy for Extracted Reward Function 2

Question 24



Optimal Policy for Extracted Reward Function 2



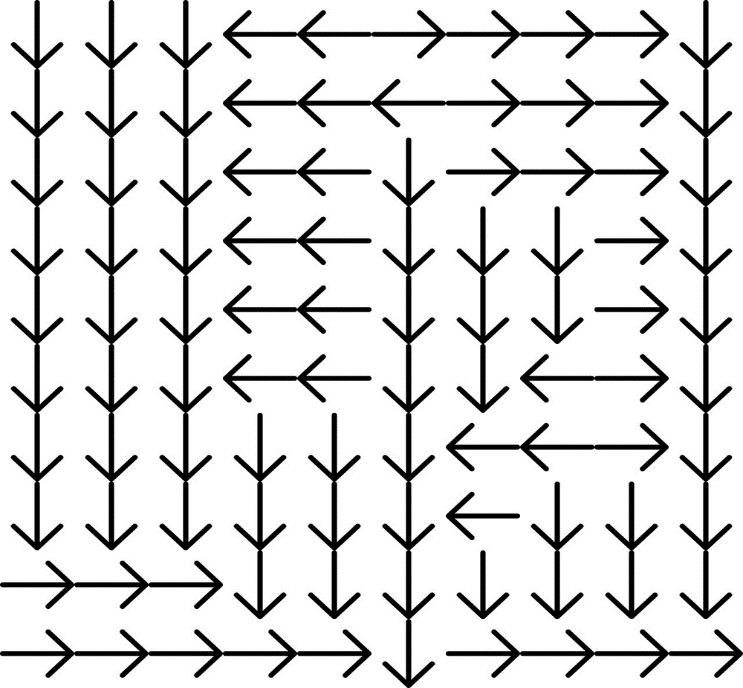
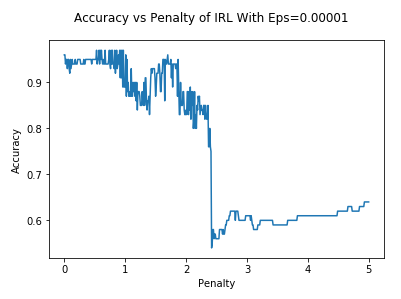
Optimal Policy for Ground Truth Reward Function 2

From the two figures above, we can see that the actions are almost the same except for states {14, 49, 50,59,73,79,96}, as marked by the red circle. The differences in some state such as {49, 79, 96} can be explained by observing the reward function as in question 21. Like the case mentioned in question 17, the reward at that value is higher than the surrounding states such that the agent is inclined to go out of the grid. Another reason for 49, 79, 59 behaving differently is because the state values and reward function along the bottom row is not very different from each other so it is almost arbitrary to pick a certain action. For state 50 we can see that there is less negative reward for going left, which is the reason why the extracted agent chooses to go this way.

Question 25

I think one of the major discrepancies is along the bottom row in which we can see from the extracted state values that the values at these states are very uniform which can make it hard to choose the optimal action. We can change the value iteration algorithm on the reward function to have a smaller epsilon such that the state values are ran more to completion.

We run the value iteration algorithm on the learned IRL reward functions with an epsilon of 0.00001 instead of 0.01.



New Optimal Policy for Extracted Reward Function 2

As we can see this improves some of the cases mentioned in question 24 as now, we achieve an accuracy of 0.97.

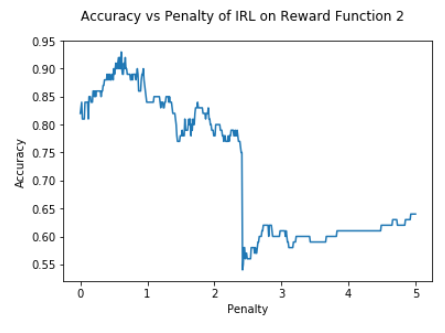
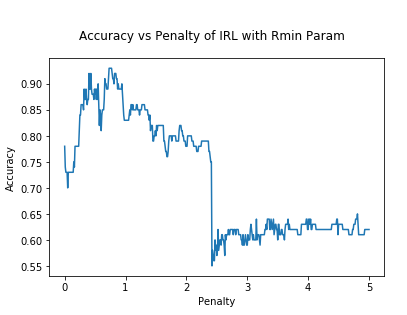
Another cause of the discrepancy is likely due to the larger range of allowable reward function values as the extracted reward function has a higher upper bound for values as the optimization problem was being constrained by the absolute value of the reward being less than 100. As a result, there is relatively larger differences between differing states which means a next state with a higher reward will more likely dictate the agent than a state value. For example, this can be used to describe the case for state 96.

To fix this we do the following change to the IRL optimization problem setup

The constraint is changed to.

Likewise, the vector becomes

We then re-run the modified IRL algorithm



The max accuracy is still 0.93 but we do see that there are more occurrences where the accuracy is equal or above 0.90. There are 19 such cases without the Rmin fix and 23 instances with the Rmin adjustment.