ECE 232E Project 3
Reinforcement Learning and Inverse Reinforcement Learning
Spring 2018

Cyrus Tabatabai-Yazdi (405029242)

Zongheng Ma (905027293)

Akshay Shetty (905028886)

Bakari Hassan (705035029)

21 May 2018

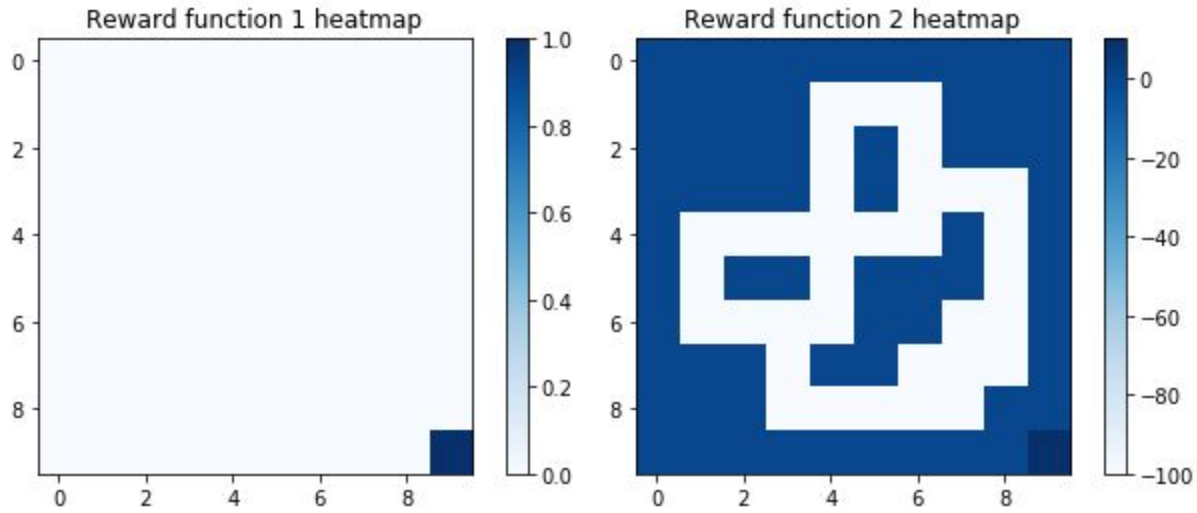
Table of Contents

PART 1: REINFORCEMENT LEARNING

The goal of Part 1 was to train an agent to navigate in a gridworld using an optimal policy derived using the value iteration algorithm. The agent starts from the top left of a grid and has to navigate to the bottom right. Each state has a reward associated with it and our goal is to calculate the values of being in each possible state and the optimal action to take at each state.

Question 1:

In this question, we were asked to visualize the reward functions through a heat map. The heat map for reward function 1 is shown on the left and the heat map for reward function 2 is on the right.



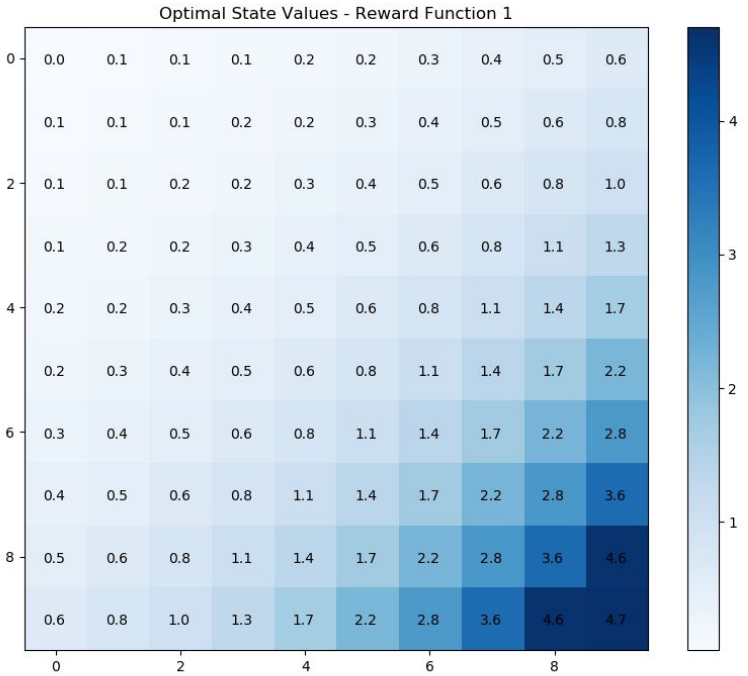
Question 2:

Below is the optimal state-value for Reward function 1 implemented using the Value-Iteration algorithm. As we can see value of states increases in the vicinity of state 99. This means that the states closer to the goal state are essentially better as they represent progress towards the goal.

0.042	0.063	0.09	0.124	0.167	0.222	0.291	0.379	0.491	0.61
0.06	0.088	0.122	0.165	0.219	0.289	0.378	0.491	0.633	0.787
0.086	0.121	0.164	0.219	0.289	0.378	0.491	0.635	0.817	1.019
0.119	0.164	0.219	0.289	0.378	0.491	0.636	0.82	1.052	1.315
0.161	0.219	0.289	0.378	0.491	0.636	0.82	1.054	1.352	1.695
0.214	0.289	0.378	0.491	0.636	0.82	1.054	1.353	1.733	2.182
0.281	0.378	0.491	0.636	0.82	1.054	1.353	1.734	2.22	2.807
0.366	0.491	0.635	0.82	1.054	1.353	1.734	2.22	2.839	3.608
0.473	0.633	0.817	1.052	1.352	1.733	2.22	2.839	3.629	4.635
0.609	0.787	1.019	1.315	1.695	2.182	2.807	3.608	4.635	4.702

Question 3:

Below is the heat map for the optimal state values derived in question 2. This heat map agrees with results for the numerical state values shown above: state values increase downwards and to the right with hotter states congregated near the goal state.



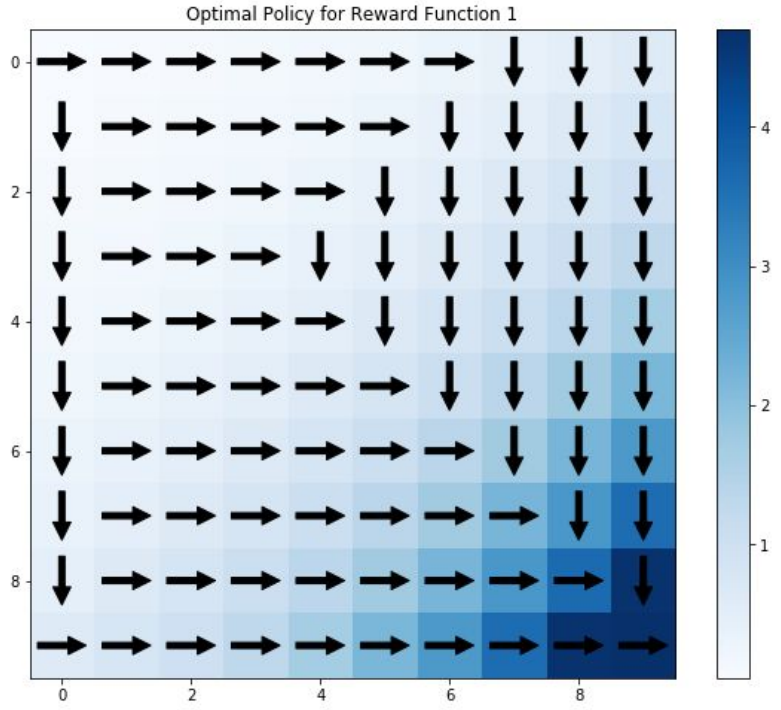
Question 4:

The above heat map for optimal-state value for reward function 1 shows the high rewards as the agent moves downward towards the goal state (State 99), and is least when we start from

state 0. It looks as if the optimal rewards is inversely proportional to the distance from the goal and looks ideal for reinforcement rewards. This plot supports our expectations.

Question 5:

Below is the optimal action-value for reward function 1. The figure corresponds directly to the state-value map for reward function 1 from question 2. The intuition is that the agent should be moving closer and closer to the goal after each action. An agent starting from state 0 will start looking for neighbours with better state-value (lighter to darker colored values), discounted for future rewards.



A brief sanity check can be used to find intuitive meaning in the graph. By randomly choosing a cell on the grid as a starting location, one can reliably follow the arrows to the goal state. While this may not be guaranteed to be true, it should be verifiable in general.

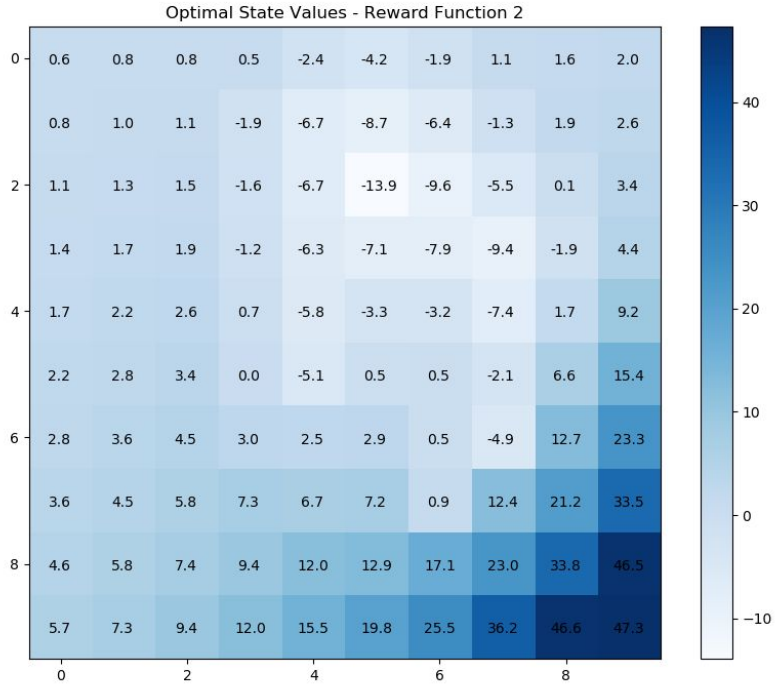
Question 6:

Below are the optimal state values using Reward function 2. By using the optimal state-value function, we derived the value for each state as shown below. As you can see, states closer to the goal have a higher value and the states that are in the region where the -100 reward values are located are very low or negative. This makes intuitive sense as the agent wants to avoid states where a low negative reward is possible in order to maximize its total reward at the end.

0.64	0.79	0.83	0.54	-2.37	-4.23	-1.92	1.13	1.59	2.04
0.82	1.02	1.07	-1.87	-6.74	-8.67	-6.37	-1.29	1.93	2.61
1.05	1.32	1.45	-1.62	-6.74	-13.91	-9.65	-5.51	-0.13	3.36
1.35	1.69	1.95	-1.23	-6.32	-7.98	-7.94	-9.42	-1.91	4.39
1.72	2.17	2.59	-0.73	-5.83	-3.25	-3.23	-7.42	1.72	9.16
2.19	2.78	3.42	-0.03	-5.1	-0.55	-0.48	-2.97	6.59	15.36
2.79	3.55	4.48	3.03	2.48	2.88	-0.45	-4.89	12.69	23.3
3.55	4.54	5.8	7.29	6.72	7.24	0.94	12.37	21.16	33.49
4.51	5.8	7.4	9.44	12.01	12.89	17.1	23.02	33.78	46.53
5.73	7.32	9.39	12.05	15.46	19.83	25.5	36.16	46.59	47.32

Question 7:

Below is the heat map for the Reward function 2. While the hotter states are nearer to the goal state, the colder ones are located in the region where the -100 reward values are concentrated. Therefore, this plot supports our expectations.

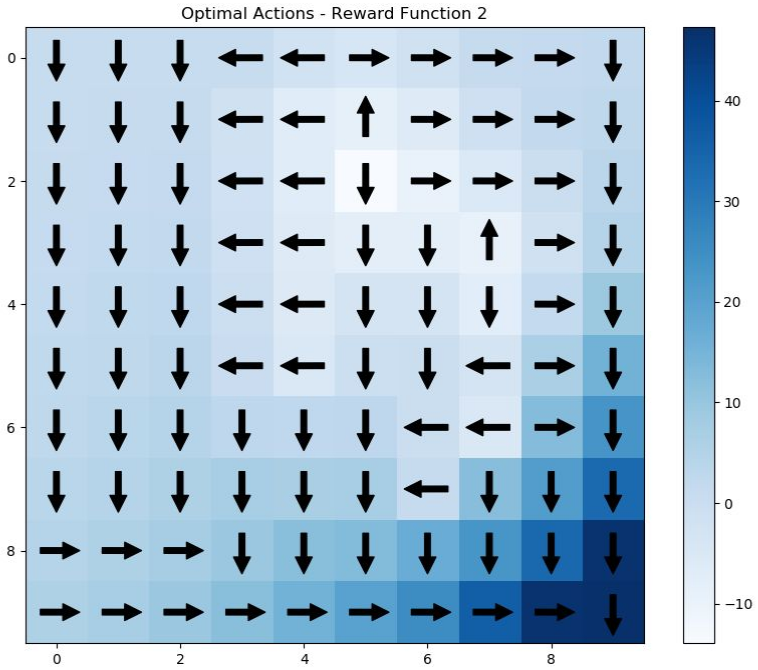


Question 8:

The heatmap for the optimal values for Reward Function 2 makes sense in that states closer to the goal (bottom right region) have high values while the region where negative rewards (-100) are located is coolest. The initial guess was that the lowest state value would coincide with one of the states with -100 reward. However, the “intelligence” in the reinforcement learning process shows here, indicating that worst state to occupy is the state that is surrounded by the largest number of -100 reward states. This means that the degrees of freedom available to avoid a low-reward state are highly limited.

Question 9:

Below are the optimal actions for each state according to the policy derived using the optimal action-value function for Reward Function 2. As you can see, the agent sticks along the left edge and bottom edge of the grid, intelligently avoiding the areas where negative rewards are possible (the cold region from the previous question’s heatmap).



A brief sanity check can be used to find intuitive meaning in the graph. By randomly choosing a cell on the grid as a starting location, one can reliably follow the arrows to the goal state. While this may not be guaranteed to be true, it should be verifiable in general.

PART 2: INVERSE REINFORCEMENT LEARNING

The goal of part 2 is to infer a reward function by learning from an expert. In our case, the "expert" is the agent following the optimal policy for reward function's 1 and 2 respectively. Thus, we want the agent to learn a good reward function that can lead to as best of an optimal policy as possible.

Question 10:

We define vectors T, U, and R as follows:

$$T = \left[t_1, t_2, ..., t_{|S|} \right]^T$$

$$U = \left[u_1, u_2, ..., u_{|S|} \right]^T$$

$$R = \left[R(s_1), R(s_2), ..., R(s_{|S|}) \right]^T$$

With this representation, we can represent the cost function as a function of T, U, and R:

$$\sum_{i=1}^{|s|} (t_i - \lambda u_i) = \begin{bmatrix} 0 \\ I \\ -\lambda \end{bmatrix}^T \begin{bmatrix} R \\ T \\ U \end{bmatrix}$$

Define c and x as follows:

$$c = \begin{bmatrix} 0 \\ I \\ -\lambda \end{bmatrix} \quad x = \begin{bmatrix} R \\ T \\ U \end{bmatrix}$$

We can represent the original function in $c^T x$ format.
The sizes of the 0 block, I block, and $-\lambda$ block are 100 x 1 to match the size of R, T, and U. Therefore, the sizes of c and x are both 300 x 1.

We have four constraints.
Constraint 1:

$$\left[(P_{a1}(i) - P_a(i)) (I - \gamma P_{a1})^{-1} R \right] \geq t_i$$

Constraint 2:

$$(P_{a1} - P_a)(I - \gamma P_{a1})^{-1} R >= 0$$

Constraint 3:

$-u <= R <= u$

Constraint 4:

$|R_i| \leq R_{max}$

Thus, using block matrix, D becomes

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

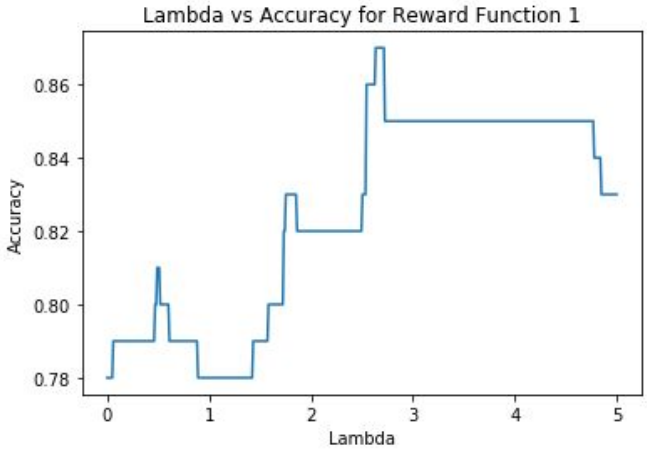
Note that we included R_{max} in the b vector passed to *solvers.lp*.

The $(P_{a1} - P_a)(I - \gamma P_{a1})^{-1}$ block's dimensions are 300 rows by 100 columns, since we repeat the expression for every action (except the optimal one) and for every position/state. There are 4-1=3 actions and 100 positions. The I blocks and 0 blocks in the first two rows are the same size as the $(P_{a1} - P_a)(I - \gamma P_{a1})^{-1}$ blocks.

The I blocks and 0 blocks in the last four rows are in the dimensions of 100 rows by 100 columns, because the sizes of T and U is 100.

Therefore, the dimension of D matrix is (300 + 300 + 100 + 100 + 100 + 100) rows and (100 + 100 + 100) columns, which is 1000 rows and 300 columns.

Question 11:

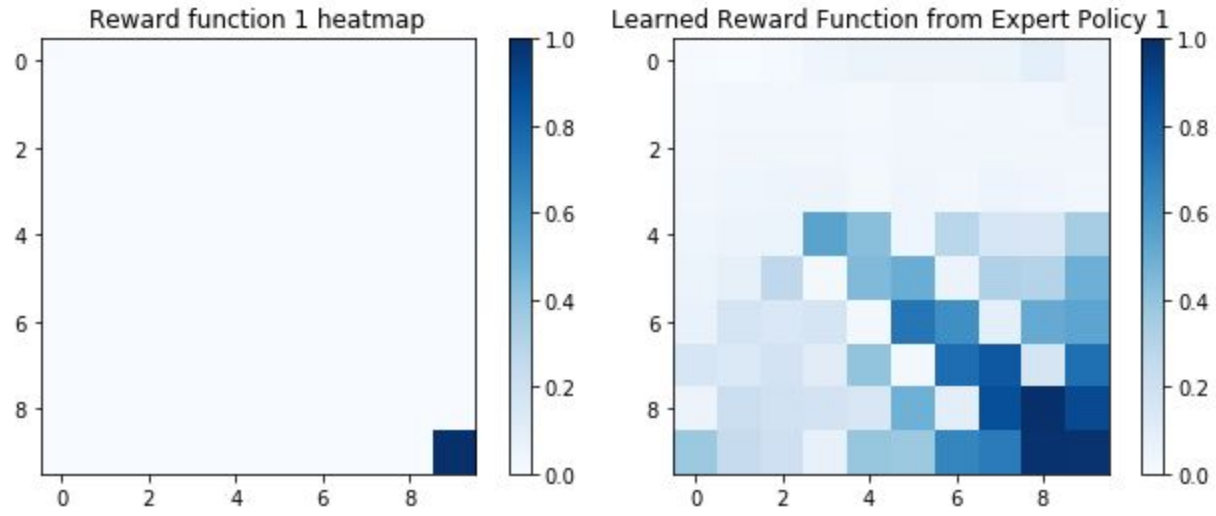


The trend seems to be that the accuracy steadily increases until around $\lambda = 2.6$ and then plateaus until 4.75 and decreases.

Question 12:

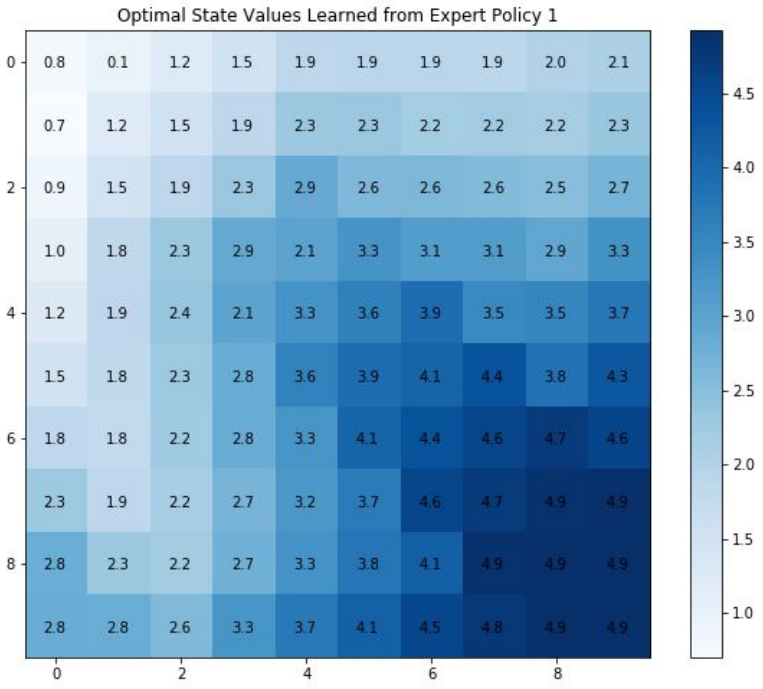
$\lambda_{max}^{(1)}$ = 2.63 with an accuracy of 88%. We believe this is high because the environment of the agent is not very complicated. All the states have reward 0 except the goal state. Thus, it seems natural that the agent will be able to learn a good reward function and an optimal policy very similar to the expert policy from part 1.

Question 13:



Question 14:

The state values shown below agree with reward function 1 and the learned reward function from expert policy 1. There's an overall downward and rightward trend with increasing state values. Additionally, the anti-diagonal band of high rewards in the learned reward function is replicated in the computed optimal state values. This indicates agreement within our problem setup and results.

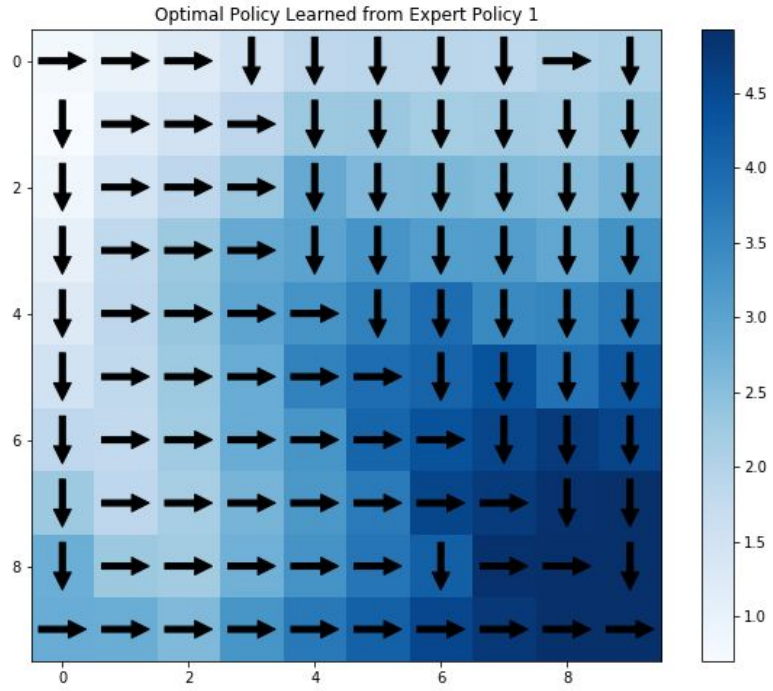


Question 15:

The heat maps of question 3 and question 14 are very similar. The darkest states are congregated near the goal state and the state values seem to be inversely proportional to the distance to the goal state. The differences are that the extracted reward optimal state values seem to be different among the edges. This is fine as we have seen that the extracted reward policy is 88% accurate compared to the expert version. In addition, the bottom left corner has higher values than the top right corner whereas they are equivalent in question 3. We believe this is a shortcoming of the linear programming formulation approach to learning the reward function.

Question 16:

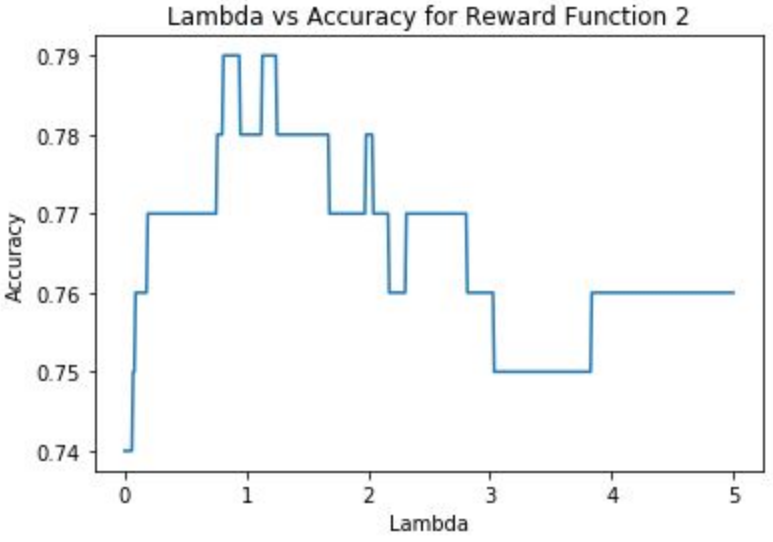
By randomly choosing a cell on this grid as a starting location, one can reliably follow the arrows to the reward state. While this may not be guaranteed to be true, it should be verifiable in general. In this case, it is true and the optimal policy makes sense in general. Find below the plot



Question 17:

It seems that the policies are very similar although the learned policy seems to take a more staggered approach rather than going all the way to the far right before proceeding downwards. Otherwise, the trend in each state seems to move as close to the bottom right as possible, whether moving downwards or right.

Question 18:



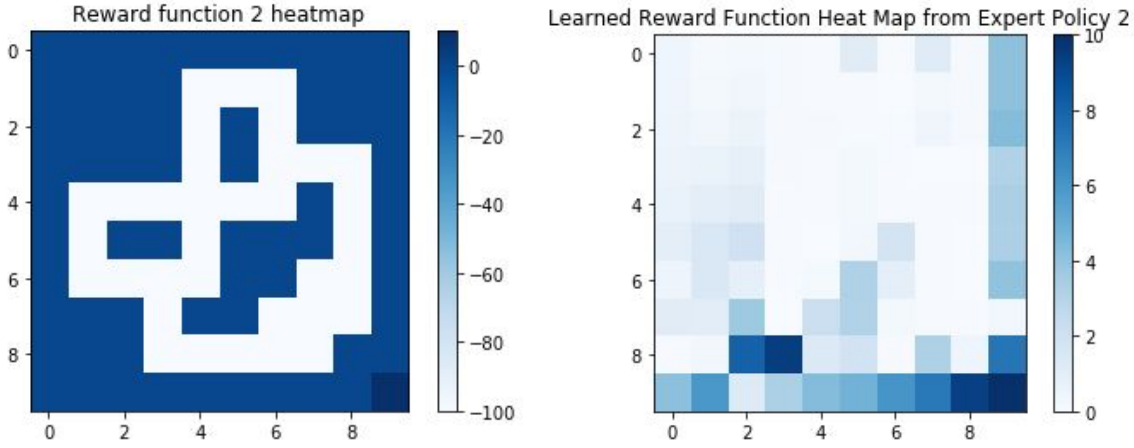
The accuracy tends to increases up to $\lambda = 1.5$ and decreases for larger values of lambda. We can see a similar pattern as question 11 that the accuracy increases at first and starts to decrease after it hits the maximum value.

Question 19:

$\lambda_{max}^{(2)}$ = 0.81 with an accuracy 0.77. It seems intuitive that the accuracy will be lower for the second policy versus the first one. The environment of the agent is more complex with states with low negative values interspersed with neutral states. Thus, it is to be expected that the agent will have a harder time matching the expert policy perfectly. However, it could be more complicated in that the distribution of low reward states could be throughout the grid rather than concentrated in one region. Thus, the agent is still able to learn a good policy.

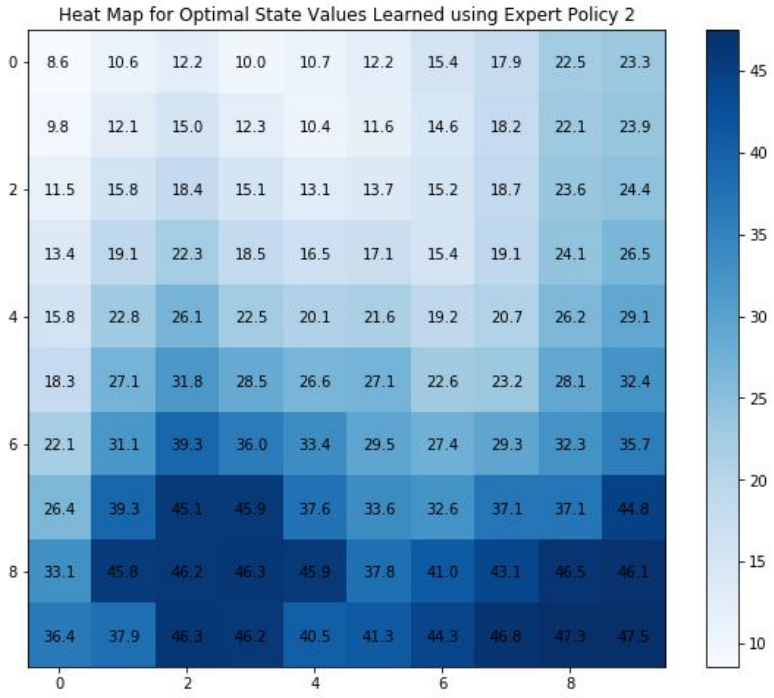
Question 20:

Below, we visualize the reward functions through heat maps. Reward function 2 is on the left and the heat map for learned reward function from expert policy 2 is on the right.



Question 21:

The heat map for learned optimal state values from expert policy 2 is shown below, and is detailed in the paragraph that follows:

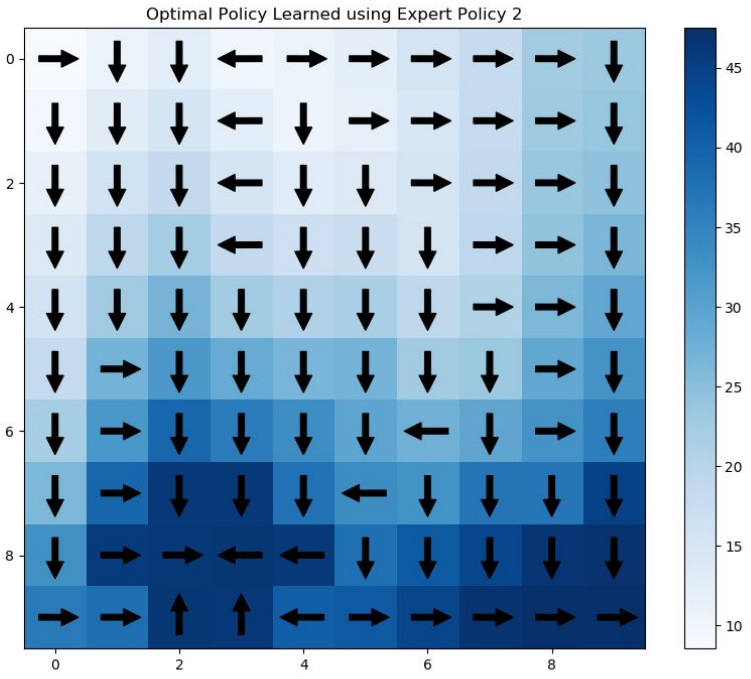


Question 22:

The heat maps from question 21 and question 7 are similar in that the region with concentration of -100 rewards are the lightest in both. In addition, states closer to the goal state tend to be darker. However, in the the plot from question 21, there is a region in the bottom left with very dark values unlike the plot in question 7. Ideally, based on the expert policy, the agent should move right but it has the potential to keep going in loops. We believe this is to avoid the chance of hitting state 68 which has a -100 reward.

Question 23:

As discussed throughout the report, although an informal test can be used to check the optimal policy, all possible starting points may not necessarily lead to the end point. By randomly choosing a cell on this grid as a starting location, one can reliably follow the arrows to the final state for most states in the case. Neighboring states (28 and 38) are great examples of this case. If the agent begins in state 28, the optimal action is to move right to state 38. Once it reaches state 38, the optimal action is the go left and return to 28. This process would repeat itself if the agent only follows the optimal policy.

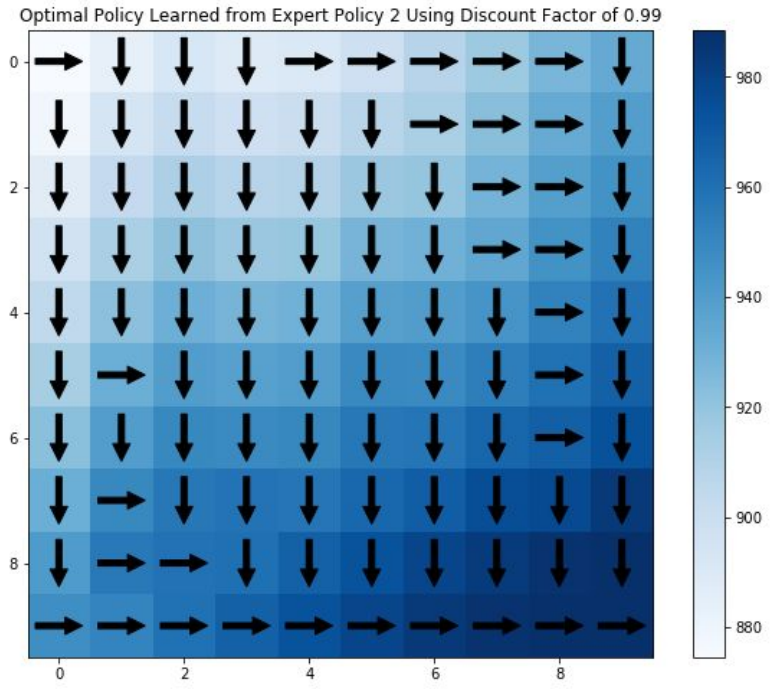


Question 24:

The figure in Question 23 goes right at the beginning instead of down like in Question 9 and in row 8, columns 3-4, it moves left instead of moving toward the target. We believe this is to avoid the low valued states likes 67,68,77,and 58. Likewise, in the last row, in columns 2 and 3, it goes up instead of to the right. Otherwise, it seems to do the same thing in trying to avoid the low value areas where bad rewards are possible.

Question 25:

The first major discrepancy is in row 8, columns 3-4 and row 9, columns 2-3 (All those numbers are zero-indexed). In these locations, the agent does not go, as expected, right. We believe this is because there are low rewards in state 67,68,77, and 58. so maybe the agent has learned to avoid that area. By changing the discount factor to 0.99, we are taking into account more the possible rewards in the future so the problem in row 8, columns 2-4 where the agent avoids going right because of the possibility of bad rewards is skirted because it knows that there is a greater reward in the long term. The second discrepancy is that the agent goes right instead of down at the beginning. This goes against the idea that going right moves it closer to the negative rewards so ideally the agent should move down to avoid the region with negative rewards as much as possible. As you can see from the heat map for the reward function, in row 8, the states closer to the goal are hotter unlike before where, due to the -100, those states were not the most attractive for the agent to move to.



The max accuracy after using a discounting factor of 0.99 becomes 0.78, leading to minimal change from the original of 0.77. We believe this is for a few reasons. Despite, fixing the discrepancy at at the bottom left, in the region where the negative rewards are concentrated, the agent becomes more bold and does not try to steer away from those areas if it finds itself there. This is because the agent is not discounting future rewards and the future reward of the goal state motivates the agent to earn a negative reward for the promise of the goal state. However, the agent is unlikely to find itself in one of these states if it follows the optimal policy from the start state. Thus, the lesson here is that tradeoffs are an inherent part of reinforcement learning and there is no prescribed remedy to mitigate issues stemming from these tradeoffs. Different problems might require different tradeoffs.