Lab A4 Reinforcement Learning

Tore Andersson, Keshav Padiyar Manuru

March 10, 2021

Q1. Define the V- and Q-function given an optimal policy. Use equations and describe what they represent. (See lectures/classes)

V-Function: $\hat{V}(S_k) = (1-\eta)\ \hat{V}(S_k) + \eta\ (\mathrm{r}_k + \gamma\hat{V}(S_{k+1}))$

Q-Function: $\mathrm{Q}(\mathrm{S}_k, a_j) = r(S_k, a_j) + \gamma\ \hat{V}(S_{k+1})$

η = Learning Rate
γ = Discount Factor
r_k = Potential Reward at state S_k
S_k = Current State
S_{k+1} = Next State
$r(S_k, a_j)$ = Reward in State S_k for performing action a_j
$\hat{V}(S_{k+1})$ = Optimal Policy value from next state

Q2. Define a learning rule (equation) for the Q-function and describe how it works. (Theory, see lectures/classes)

Q-Function: $\hat{Q}(S_k, a_j) = (1-\eta)\ \hat{Q}(S_k, a_j) + \eta\ (\mathrm{r}_k + \gamma\ \max_a(\hat{Q}(S_{k+1}, a))$

The Q function is an instrument for exploration around the best policies during learning. Q function updates the values obtained from above equation when the agent performs any action. Because of this functionality we could view the values for different actions and can choose to explore actions that could lead to high rewards in the future.

Now, let us consider that the Q table is initialized with zeros. In an episode we initialize the state (S_k) of the agent. In that state agent performs an action (a_j) to transit to next state S_{k+1}. By performing that action the agent receives feedback in the form of reward (r_k) from the environment. While updating the Q table, we also consider the optimal policy of its next state ($max_a\hat{Q}(S_{k+1}, a)$). The other parameters - Learning Rate (η): Controls how often should an agent learn its environment and Discount Factor (γ) controls the trade-off between between immediate and Long term reward optimization for the agent.

Q3. Briefly describe your implementation, especially how you hinder the robot from exiting through the borders of a world.

To hinder the robot from leaving the borders of the world, we have set the value to $-\infty$ along the border positions in the Q table. Whenever agent tries to move towards the border since the Q values in those region $-\infty$ no action will be chosen for those states.

Q4. Describe World 1. What is the goal of the reinforcement learning in this world? What parameters did you use to solve this world? Plot the policy and the V-function.

In the world 1 we see that the agent starts at random position every time, and there is a static blue wall in between in the map. So the agent has to move towards the goal surpassing the wall.

Parameter Setting:
1. Number of Episodes: 5000
2. learning rate (η): 0.5
3. discount factor (γ): 0.9

We tested our agent for 5000 episodes out of which all 5000 times it could reach goal successfully.

Figure 1 are the plots for best Policy and V-function for world 1.

(a) Policy for World 1

(b) V-function for World 1

Figure 1: Policy And V-function: World 1

Q5. Describe World 2. What is the goal of the reinforcement learning in this world? This world has a hidden trick. Describe the trick and why this can be solved with reinforcement learning. What parameters did you use to solve this world? Plot the policy and the V-function.

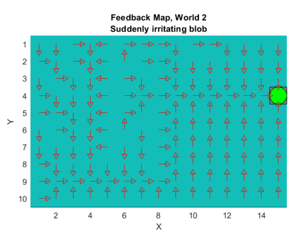
In world-2 the hidden trick is that the environment will be changed at random instances, there will be a wall in the environment for some episodes and in some episodes the agent wont find any wall.

The agent will be in the exploration phase during initial episodes, even with the changing environment the agent learns about the obstacles and chooses the best policy based on the rewards it receives from the environment. Since there is total uncertainty in the environment itself the model has to have some sort of a ad-hoc feedback system to achieve the objective. Therefore the reinforcement learning (RL) can solve this problem.

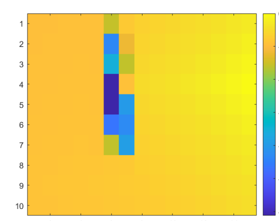
Parameter Setting:
1. Number of Episodes: 5000
2. learning rate (η): 0.5
3. discount factor (γ): 0.9

We tested our agent for 5000 episodes out of which all 5000 times it could reach goal successfully.

Figure 2 are the plots for Policy and V-function for world 2.



(a) Policy for World 2



(b) V-function for World 2

Figure 2: Policy And V-function: World 2

Q6. Describe World 3. What is the goal of the reinforcement learning in this world? Is it possible to get a good policy from every state in this world, and if so how? What parameters did you use to solve this world? Plot the policy and the V-function.

The goal of the reinforcement learning in world 3 is to reach the goal without touching the invalid borders of the world. Also expect the agent to move through the passage between the 2 walls rather taking a longer route. In world 3 it's not uncommon of the agent to spawn directly on the goal, we observed this more often. As algorithm is designed such that when ever the agent encounters the goal the learning ends. So we increased the learning rate is high enough so that the agent gets enough learning from the other episodes where it doesn't get spawn over the target.

Parameter Setting:
1. Number of Episodes: 5000
2. learning rate (η): 0.75
3. discount factor (γ): 0.9

We tested our agent for 5000 episodes out of which all 5000 times it could reach goal successfully.

Figure 3 are the plots for best Policy and V-function for world 3.

(a) Policy for World 3 (b) V-function for World 3

Figure 3: Policy And V-function: World 3

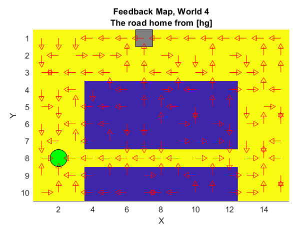
Q7. Describe World 4. What is the goal of the reinforcement learning in this world? This world has a hidden trick. How is it different from world 3, and why can this be solved using reinforcement learning? What parameters did you use to solve this world? Plot the policy and the V-function.

The difference from world 3 to 4 is that the agent will not spawn on the on the goal frequently. It also is spawned more frequently on the opposite side of the blue invalid borders and if we observe the V-function image in Figure 4 we see that the environment returns a smaller reward at states between- **(8,8) and (10,8)** or there is a high chance that the agent end up at invalid state more often in those states, therefore the agent avoids to take action towards those states instead takes the longer path. These environmental behavior problem can be solved using by reinforcement learning because which ever position the agent is spawned it will learn from previous information and chooses the optimal and best action.

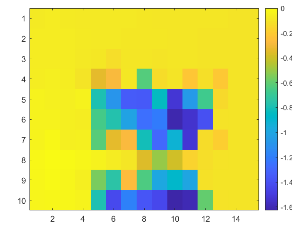
Parameter Setting:
1. Number of Episodes: 5000
2. learning rate (η): 0.75
3. discount factor (γ): 0.9

We tested our agent for 5000 episodes out of which all 89 times it could reach goal successfully.

Figure 4 are the plots for best Policy and V-function for world 4.



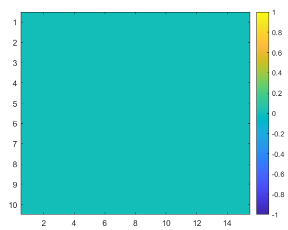
(a) Policy for World 4



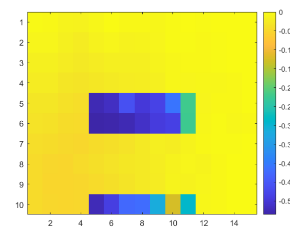
(b) V-function for World 4

Figure 4: Policy And V-function: World 4

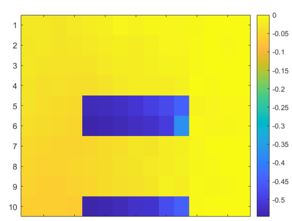
Q8. Explain how the learning rate a influences the policy and V-function. Use figures to make your point.



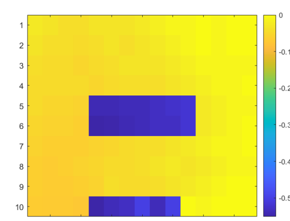
(a) V-Function of World 3 When $\eta = 0$



(b) V-Function of World 3 When $\eta = 0.25$



(c) V-Function for World 3 When $\eta = 0.5$

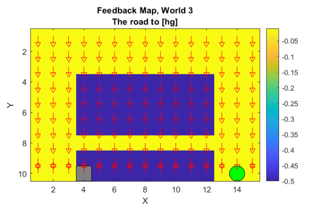


(d) V-Function of World 3 When $\eta = 1$

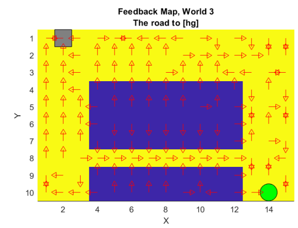
Figure 5: V-Function of World 3 for different Learning Rate η at Episode = 500

Learning rate impacts the rate of which information is stored in the algorithm. The learning rate can take values for $0 \leq \eta \leq 1$. If the value is close to zero it means that the algorithm will not take into account the current information to the same extent. Increasing the η will increase the impact of each iteration to the previous.

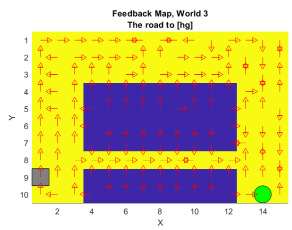
In Figure 5 we could see that, when the $\eta = 0$ the algorithm doesn't update any new information from the environment instead the initial values of each state



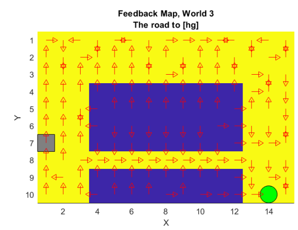
(a) Policy of World 3 When $\eta = 0$



(b) Policy of World 3 When $\eta = 0.25$



(c) Policy for World 3 When $\eta = 0.5$



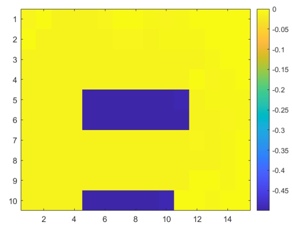
(d) Policy of World 3 When $\eta = 1$

Figure 6: Policy of World 3 for different Learning Rate η at Episode = 500

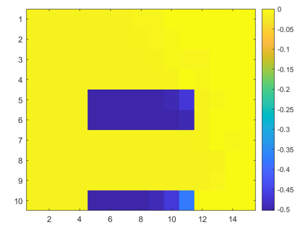
from the Q table is passed on in every episode. And the rate at which the information is stored is increased with the η value which is evident from the figure.

In Figure 6 we could see the improvement in the path chosen by the agent to reach the goal i.e as learning rate increases the agent learns more new formation and optimizes its route.

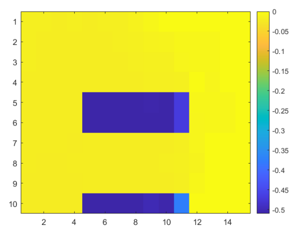
Q9. Explain how the discount factor g influences the policy and V-function. Use figures to make your point.



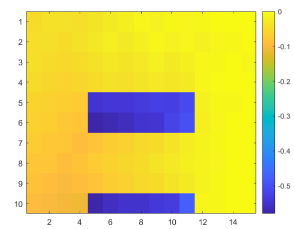
(a) V-Function of World 3 When $\gamma = 0$



(b) V-Function of World 3 When $\gamma = 0.25$



(c) V-Function for World 3 When $\gamma = 0.5$

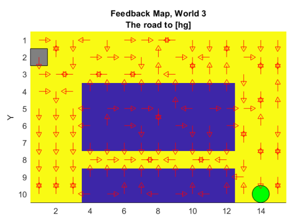


(d) V-Function of World 3 When $\gamma = 1$

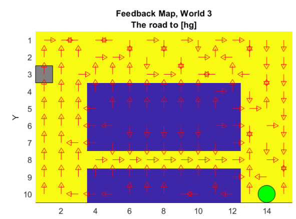
Figure 7: V-Function of World 3 for different Discount Factor γ at Episode = 500

The discount factor γ takes on values between $0 < \gamma < 1$ where a lower value will maximize the short term rewards for learning and a higher value will focus more on the long term rewards in the system.

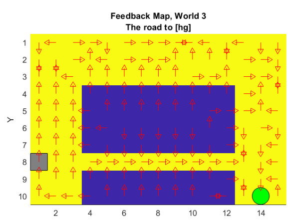
For $\gamma = 0$ the agent gets only the actual reward but as the γ value increases the algorithm adds the best value from the Q table along with the reward which is useful to the agent in long run, this difference in reward value updating can be visualized in Figure 7. In figure 8 we find the policy for different values of γ



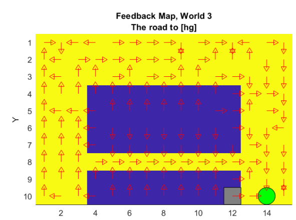
(a) Policy of World 3 When $\gamma = 0$



(b) Policy of World 3 When $\gamma = 0.25$



(c) Policy for World 3 When $\gamma = 0.5$



(d) Policy of World 3 When $\gamma = 1$

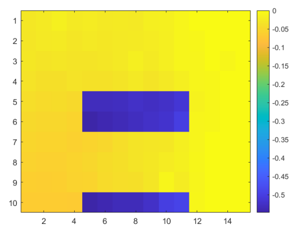
Figure 8: Policy of World 3 for different Discount Factor γ at Episode = 500

Q10. Explain how the exploration rate e influences the policy and V-function. Use figures to make your point. Did you use any strategy for changing e during training?

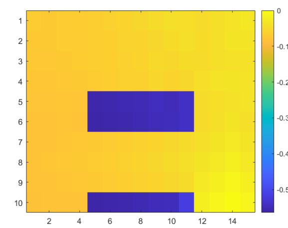
For deciding the exploration rate the following formula were used:

$$\epsilon = \frac{maxiteration - currentiteration}{maxiteration}$$

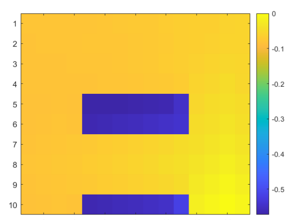
Where in the beginning the algorithm will explore more and as for each iteration the probability for a random walk is reduced until it becomes 0 in the last iteration



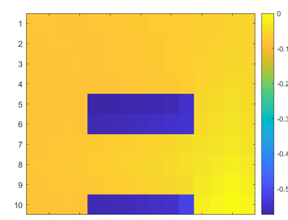
(a) V-Function of World 3 When $\epsilon = 0.9$



(b) V-Function of World 3 When $\epsilon = 0.7$



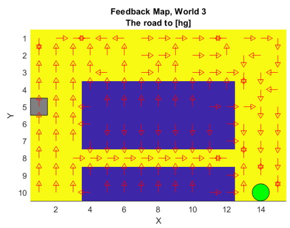
(c) Policy for World 3 When $\epsilon = 0.5$



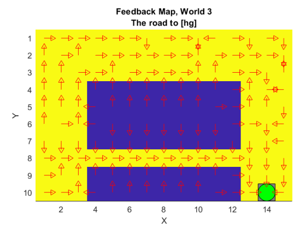
(d) V-Function of World 3 When $\epsilon = 0.2$

Figure 9: V-Function of World 3 for different Exploration Rate ϵ

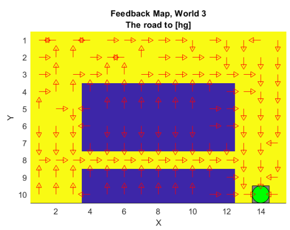
Figure 9 and 10 depicts the V-function and Best policy of World 3 for different Exploration Rates. There is a change in color contrast in the Figure hen $\epsilon = 0.9$ and $\epsilon = 0.7$ that means the agent was exploring the map and updating the Q table based on its encounter with the environment. But we see a constant color contrast in the later ϵ values where the agent was optimizing its best path.



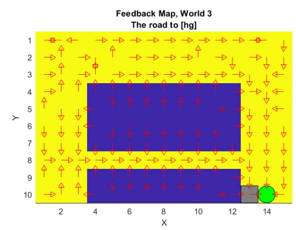
(a) Policy of World 3 When $\epsilon = 0.9$



(b) Policy of World 3 When $\epsilon = 0.7$



(c) Policy of World 3 When $\epsilon = 0.5$



(d) Policy of World 3 When $\epsilon = 0.2$

Figure 10: Policy of World 3 for different Exploration Rate ϵ

Q11. What would happen if we instead of reinforcement learning were to use Dijkstra's cheapest path finding algorithm in the "Suddenly irritating blob" world? What about in the static "Irritating blob" world?

In the static irritating blob world it would have to calculate the shortest path for every different starting position on the map. As for each new starting position there would be a new shortest path. For the suddenly irritating blob world it would have to do more than twice the amount of calculations as now it could start within the invalid regions. Which could quickly become very computationally expensive for more complex cases.

Q12. Can you think of any application where reinforcement learning could be of practical use? A hint is to use the Internet.

1. Application areas for reinforcement learning could be for example to autonomous robots to learn different actions, in the lecture a video of a robot flipping a pancake was presented. In more real world applications it could be a robot arm used in manufacturing to improve it's gripping performance.

2. Reinforcement learning could be used for implementation of autonomous driving vehicles, from using sensors to describe the world around it. It can then define a Q-learning function to keep the car on the road and between lanes.
3. It has wide range of application in game simulators such as - Board Games like chess.