Lab A4 Reinforcement Learning

Tore Andersson, Keshav Padiyar Manuru

April 5, 2021

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

V-Function: $\hat{V}(S_k) = (1-\eta)\hat{V}(S_k) + \eta(r_k + \gamma\hat{V}(S_{k+1}))$ – 1

Q-Function: $\hat{Q}(S_k, a_j) = (1-\eta)\hat{Q}(S_k, a_j) + \eta(r_k + \gamma\hat{V}(s_{k+1})$ – 2
where $\hat{V}(s_{k+1}) = max_a(\hat{Q}(S_{k+1}, a)$ in equation 2
η = Learning Rate
γ = Discount Factor
r_k = Potential Reward at state S_k
S_k = Current State
S_{k+1} = Next State
$\hat{V}(S_{k+1})$ = Optimal Policy value from next state
$\hat{Q}(S_k, a_j)$ = The value obtained in state S_k by performing task a_j

$V(\hat{S}_k)$ is a function that tells us the value of being in the state given a policy. This means that the value function tells us the expected amount of reward/future reward we get from state S_k by following a policy. Using the above V function (equation 1) we can update the value of current state by taking the current state value and then adding with the value obtained from the next step and associated reward. Finally the optimal policy has the maximum $V(S)$ for all states.

The Q function is a function of both state and action - $Q(S_k, a_j)$. Which is used to compute the expected future reward obtained by performing some action a_j at state S_k and optimal policy value in the next state. This Q function uses the V function implicitly. In Equation 2, $\hat{V}(S_{k+1})$ is nothing but the maximum Q value obtained out of all the action that the agent had performed in a 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 the 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): This parameter is a value between 0 and 1 which control's the step lengths towards the optimal policy .When close to 0 algorithm puts more emphasis on already learned experiences. And values closer to 1 will overwrite the previous experiences with new information. Good value to start with is : 0.1 to 0.5.
Discount Factor (γ): It is a value between 0 and 1 which controls the trade-off between between immediate and Long term reward optimization for the agent. A value close to 0 will seek to maximize short term reward where as value closer to 1 will focus the learning on long term rewards. Good value to start with is around 0.9

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 area which could be seen as a shallow lake. The goal of this map is for the agent to learn to avoid the blue area. In case the agent gets deployed in blue area, the agent should ideally exit in the path closest to the target. So the agent has to move towards the goal using the shortest or optimal path and by avoid the blue area which has lower rewards.

Parameter Setting:
1. learning rate (η): 1
2. discount factor (γ): 0.9

Since the world is static, by using the above parameter setting the agent could learn to reach the target within 500 episodes.
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

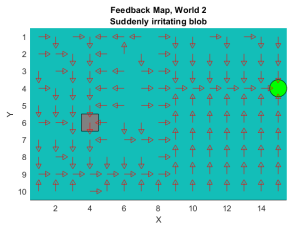
Q5. Describe World 2. What is the goal of the reinforcement learning in this world? This world has a hidden trick. Describe the trick and why this can be solved with reinforcement learning. What parameters did you use to solve this world? Plot the policy and the V-function.

In world-2 the hidden trick is that the environment will be changed at random instances, there will be a blue area in the environment for some episodes and in some episodes the agent would not find any blue area.
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. When the agent passes over the blue area it receives very low rewards. We have set a learning rate = 0.1 as a result the agent will take more episodes to learn its environment hence we increased the number of episodes to 5000. The updated rewards when the blue area is not there, will not diminish the results of it being there. This was possible by lowering the learning rate as the agent could store more of the previously learned information for a longer time (using more episodes). Thus the agent will eventually avoid the blue area and exit in the path closest to the target if it's the starting position. The ad-hoc feedback from the environment in the form of rewards along with the rest of the parameter settings makes the agent robust to the changing environment, this sort of learning is only possible using reinforcement learning.

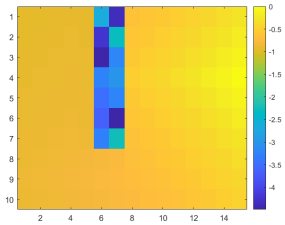
Parameter Setting:

1. Number of Episodes: 5000
2. learning rate (η): 0.1
3. discount factor (γ): 0.9

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 reinforcement learning in this world is similar to the one in world 1. The difference is that there are two blue regions with low rewards and the agent has to find the best policy that makes the agent reach the target with through the optimal path. i.e. the agent will either take the route around or between the blue areas depending on the shortest path from the its starting point. As the environment is static the learning rate used was 1. It's possible to have a good policy from every starting state in this world as is shown in figure 3 (a) every action will move the agent closer to the goal. Also, if we see the arrows in the blue region since the rewards in those region is very low the agent tries exit the region towards the nearest yellow region it could find.

Parameter Setting:
1. Number of Episodes: 1000
2. learning rate (η): 1
3. discount factor (γ): 0.9

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 appearance of world 4 is same as world 3 with the exception that at some instances the agent is forced to move to random directions as the rewards in those states are random and are not allowing the agent to take the optimal step this could be viewed from the direction of arrows in the figure 4 (a) and the random color contrast around the blue region 4 (b),In addition, we observed in

the test loop that the agent will change its action at random: For example let the agent going in the upward direction in a path, suddenly it will change its path to another policy or it will go some steps down even if the optimal policy say to go up. Also, from the figure 4 (b) we could view that at states between- **(8,8) and (10,8)** has low rewards or there is a high chance that the agent end up at blue regions more often in those states, therefore the agent avoids to take action towards those states instead takes the longer path. The agent requires more number of episodes and smaller learning steps to understand these uncertainties in the environment, therefore we reduced the learning rate to 0.01 and increased the number of episodes to 10000.

The environmental behavior problem such as which ever position the agent started and tries to take a deviation form the optimal policy the agent will later try to rejoin the optimal policy and reach the target. These sort of randomness only can be solved using by reinforcement learning.

Parameter Setting:

1. Number of Episodes: 10000
2. learning rate (η): 0.01
3. discount factor (γ): 0.9

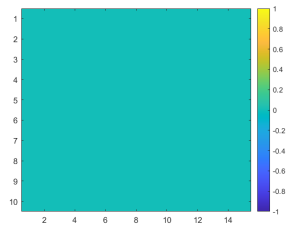
Figure 4 are the plots for best Policy and V-function for world 4.

(a) Policy for World 4 (b) V-function for World 4

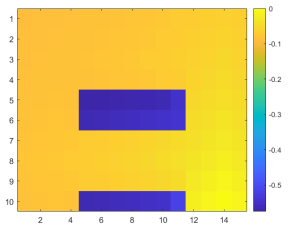
Figure 4: Policy And V-function: World 4

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

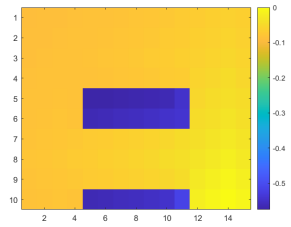
Learning rate impacts the rate of which new information related to the environment is stored in the algorithm by which it controls the step length towards achieving the optimal policy. The learning rate can take values for $0 \leq \eta \leq 1$. If the value is close to zero it means that the algorithm 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.



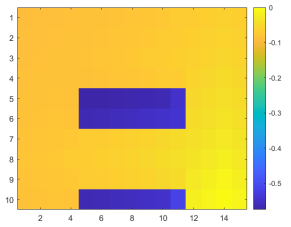
(a) V-Function of World 3 When $\eta = 0$



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

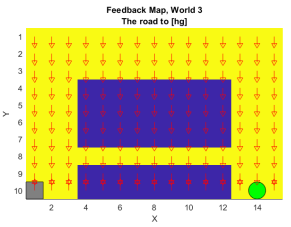


(c) V-Function for World 3 When $\eta = 0.5$

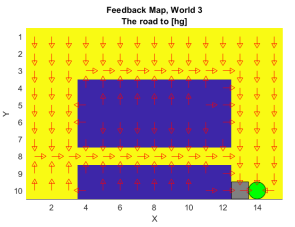


(d) V-Function of World 3 When $\eta = 1$

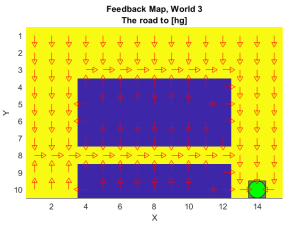
Figure 5: V-Function of World 3 for different Learning Rate η



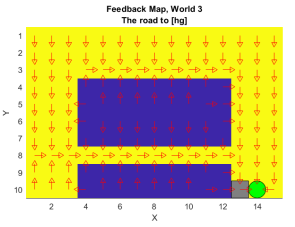
(a) Policy of World 3 When $\eta = 0$



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



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



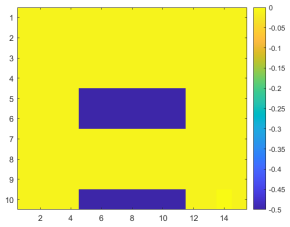
(d) Policy of World 3 When $\eta = 1$

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

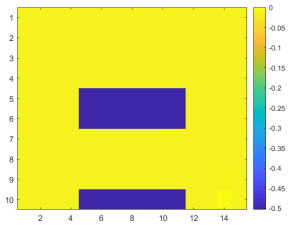
In Figure 5 we could see that, when the $\eta = 0$ the algorithm doesn't update any new information from the environment instead the initial values of each

state from the Q table is passed on in every episode as a result the agent has not identified the blue region in the environment. And the rate at which the information is stored is increased with the η value, while in world 3 figure 6 we could not actually view the impact of increase in learning rate as the world is static and there is no randomness. The smaller learning rate the agent takes slightly more number of episodes to converge, therefore we could directly use learning rate = 1 and increase the efficiency of the agent to find the optimal policy.

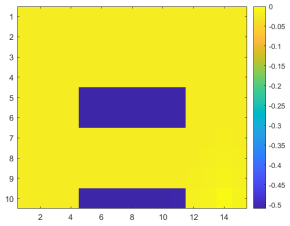
Q9. Explain how the discount factor g influences the policy and V-function. Use figures to make your point.



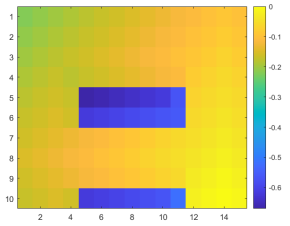
(a) V-Function of World 3 When $\gamma = 0$



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



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

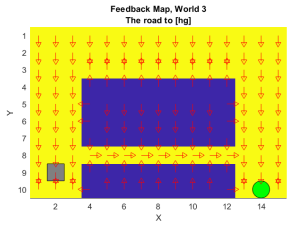


(d) V-Function of World 3 When $\gamma = 1$

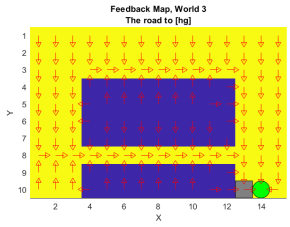
Figure 7: V-Function of World 3 for different Discount Factor γ

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.

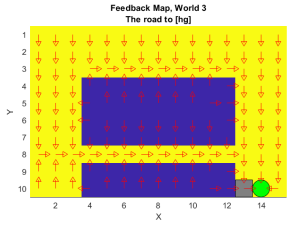
For $\gamma = 0$ the agent gets only the actual reward as a result at the boundary or near the edges of blue region when the agent tries to avoid that and find the optimal action, now as discount factor is zero the action that agent takes might result in infinite loop condition - figure 8 (a) . As the 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.



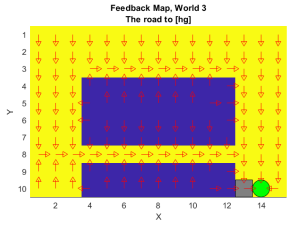
(a) Policy of World 3 When $\gamma = 0$



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



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



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

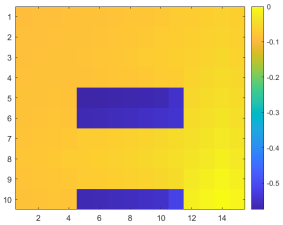
Figure 8: Policy of World 3 for different Discount Factor γ

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?

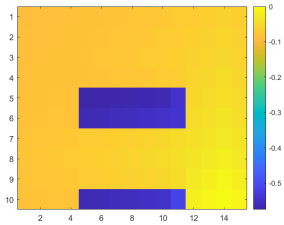
For deciding the exploration rate the following formula were used:

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

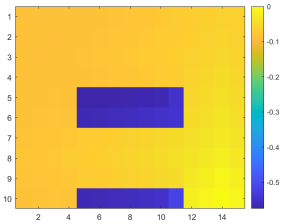
Where in the beginning the algorithm will explore more and as for each iteration the probability for a random walk is reduced until it becomes 0 in the last iteration



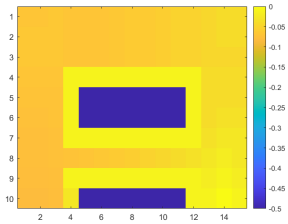
(a) V-Function of World 3 When $\epsilon = 1$



(b) V-Function of World 3 When $\epsilon = 0.75$



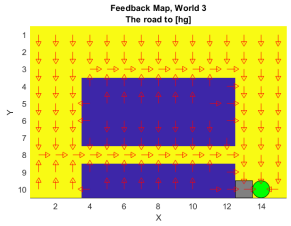
(c) Policy for World 3 When $\epsilon = 0.5$



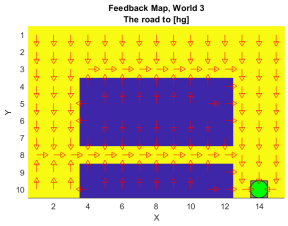
(d) V-Function of World 3 When $\epsilon = 0$

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

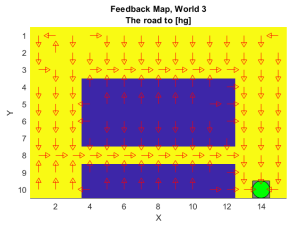
Figure 9 and 10 depicts the V-function and Best policy of World 3 for different Exploration Rates. In the figure 10 (d) we could see that the arrows are not at all focused at the optimal direction, that means the agent was not actually trying different move at a state instead it was trying to follow the move that it has already done previously. At high ϵ values the agent explores more actions ending up with the optimal ones hence the arrows start to align focusing the optimal policy this phenomenon can be viewed in 10(c),(b),(a).



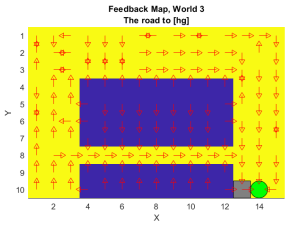
(a) Policy of World 3 When $\epsilon = 1$



(b) Policy of World 3 When $\epsilon = 0.75$



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



(d) Policy of World 3 When $\epsilon = 0$

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, while the Dijkstra's algorithm also work in the same way, it also would have given the same result in this world. While for "Suddenly Irritating Blob" it would depend on the map we get, as in difference instances the map changes Dijkstra's model is not fit when there is randomness in the map itself. Considering even if train the Dijkstra's model with 2 different maps and find the optimal policy while training, while testing we cannot surely say at what instance which optimal policy the agent has to consider. Therefore we cannot use Dijkstra's algorithm for the suddenly irritating blob.

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.