TBMI26 – Computer Assignment Reports  
Reinforcement Learning

Deadline – March 15 2020

Fredrik Johansson

In order to pass the assignment you will need to answer the following questions and upload the document to LISAM. Please upload the document in PDF format. **You will also need to upload all code in .m-file format**. We will correct the reports continuously so feel free to send them as soon as possible. If you meet the deadline you will have the lab part of the course reported in LADOK together with the exam. If not, you’ll get the lab part reported during the re-exam period.

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

**V-function:**

Where is the optimal policy and is a discount factor. This function describes the value of being in a certain state given the optimal policy.

**Q-function:**

Which can be rewritten as:

Where is the optimal policy for the action and is a discount factor as previously mentioned. The Q-function described the expected future reward of doing action a with the requirement that the optimal policy is followed afterwards.

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

Where is the learning rate, is the reward from the current state and is a discount factor. The Q-function for the current state and action is updated using two different estimates. The first term includes the previous estimate and the second term includes a better estimate. This makes the updated Q-function retain previously attained knowledge.

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

I begin with initializing the world, the Q-table and the hyperparameters. The positions in the Q-table which corresponds to the direction of an action which would make the robot exit the world are set to minus infinity. This will cause the robot to never choose a direction which would make it leave the world under the assumptions that the world is always a square or a rectangle.

To train the robot I update the Q-table using the function from question 2. This is done by repeatedly choosing an action and evaluating the reward of the new state until the robot reaches its goal. When choosing an action there is a specified chance that the action will be suboptimal. This is done to allow for exploration of actions which may result in the discovery of better policies. Once the robot has reached its goal the whole process is repeated for a specified number of episodes. Lastly, the robot is tested by only choosing the optimal actions according to the Q-table.

1. **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.**

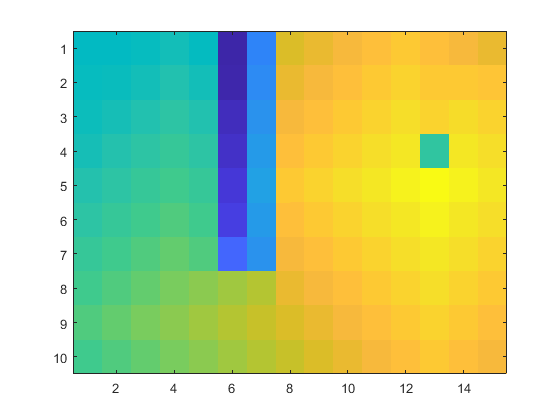
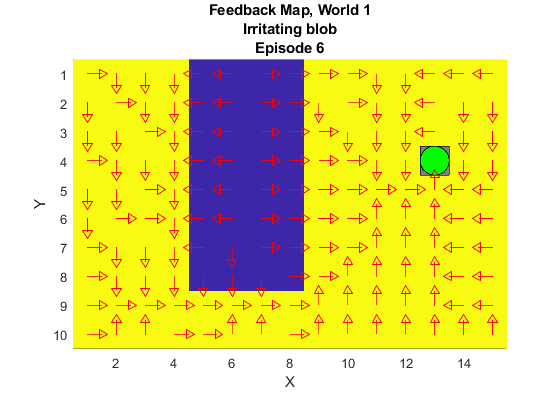
The goal of this world is to reach the green circle while avoiding the dark areas since they give a high negative reward.

Learning rate = 0.2

Discount factor = 0.99

Number of episodes = 1000

Exploration factor = 1 – (current episode / (number of episodes \* 1.1))

****

1. **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.**

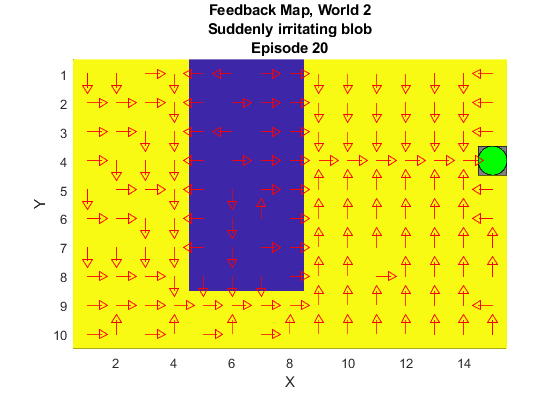
The goal of this world is to reach the green circle which in this case is always at an edge which may cause problems if we are not careful about how we handle the edge cases. There is also a hidden trick where the world has two separate cases. In the first case the world is equal to world 1. In the second case all the rewards are set to the same value which is some value between the dark area and the light area from the first case. By using reinforcement learning the robot can learn how to handle both cases. For example, if the probability of there being a dark area is high enough, it would be worth to travel around that area even if in the case where there is none. This behavior can be achieved with reinforcement learning.

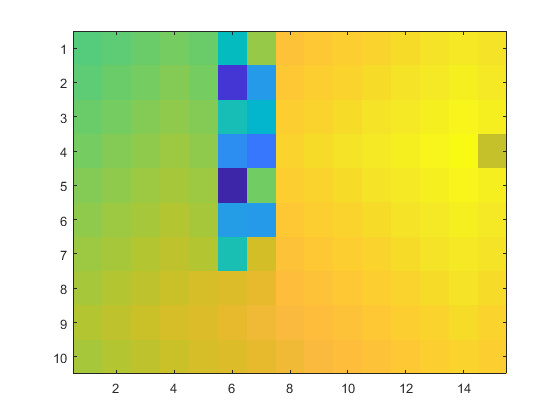
Learning rate = 0.2

Discount factor = 0.99

Number of episodes = 2000

Exploration factor = 1 – (current episode / (number of episodes \* 1.1))

**En bild som visar text

Automatiskt genererad beskrivning**

1. **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 in this world is to make the robot traverse in the gap between the dark areas while having a static starting position for each episode. To achieve this the algorithm must explore different sub optional actions, we cannot rely on different starting positions to explore. Therefore, we must have a high exploration factor.

It is possible to get a good policy for every state in this world by choosing a good value for the exploration factor. By using an exploration factor of 0.1 the robot will make the optimal policy to traverse above the dark spots instead of between them. By using a larger value for the exploration factor the robot will traverse between them.

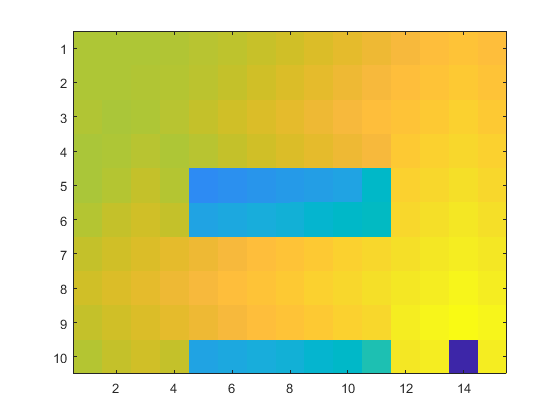
Learning rate = 0.2

Discount factor = 0.99

Number of episodes = 2000

Exploration factor = 1 – (current episode / (number of episodes \* 1.1))

**En bild som visar objekt

Automatiskt genererad beskrivning**

1. **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 goal in this world is to handle outside interference which we have no influence over. In this world there is a risk of the robot being pushed one tile in any direction except in the movement direction. The push occur after a move has been made and the robot can never be pushed outside of the world.

Similar to world 3 the best path would be between the dark areas. However, since the random occurrence of pushes is frequent enough for it to most likely push the robot into a dark area while traversing the gap, the optimal policy differs. To avoid being pushed into a dark area, the robot must keep at least one tile distance to the dark areas which is impossible to do in the gap. In order to keep at least one tile distance, the robot must move above the dark areas.

The optimal policy found suggests that the robot should traverse at the edge of the world which results in two tiles between the robot and the dark areas. This may seem unnecessary since I previously mentioned that there must only be one tile between them. However, this is actually the correct strategy which exploits the fact that the robot can never be pushed outside of the world or in the movement direction. Since the robot cannot be pushed in the movement direction, there are three different cases of pushes where at least two of them are always negative. By pathing along the edge one of the bad cases can be removed resulting in better rewards. This strategy is further optimized and demonstrated in the upper corners where the path differentiates from the edge since one of the negative cases has been changed to a positive one.

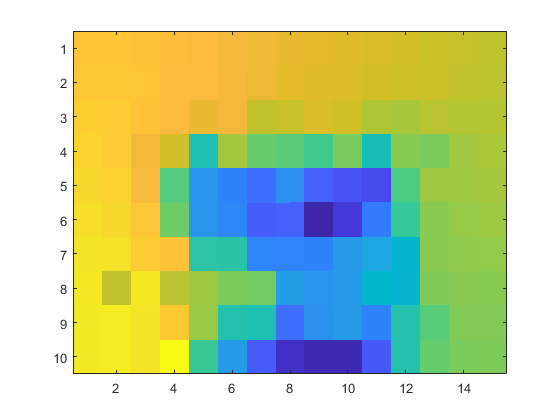
Learning rate = 0.2

Discount factor = 0.99

Number of episodes = 3000

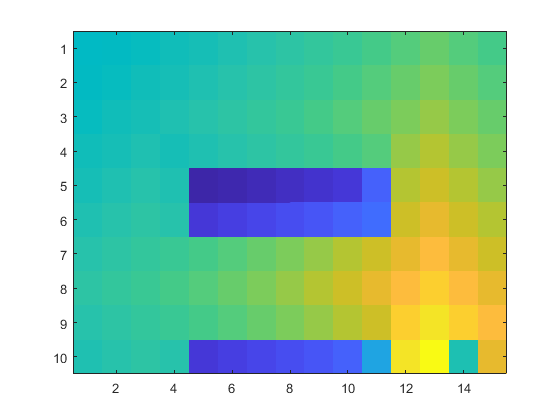
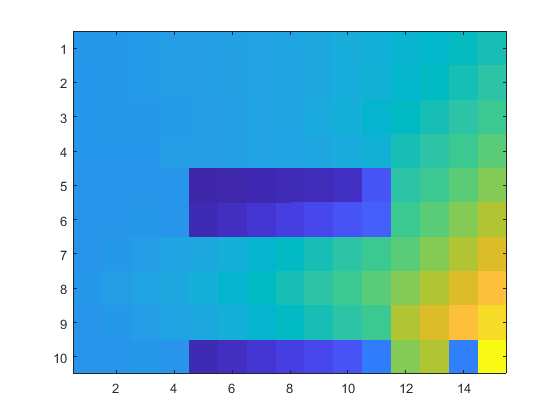
Exploration factor = 1 – (current episode / (number of episodes \* 1.1))

En bild som visar objekt

Automatiskt genererad beskrivning

1. **Explain how the learning rate α influences the policy and V-function. Use figures to make your point.**

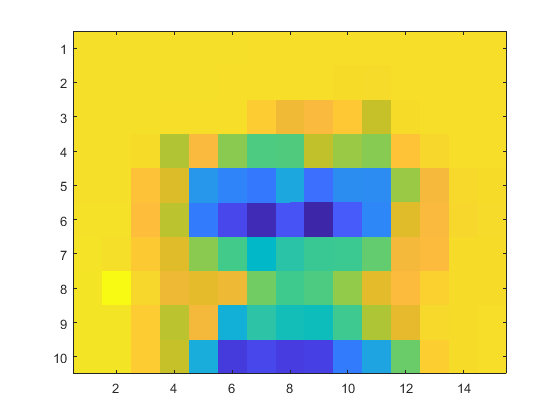
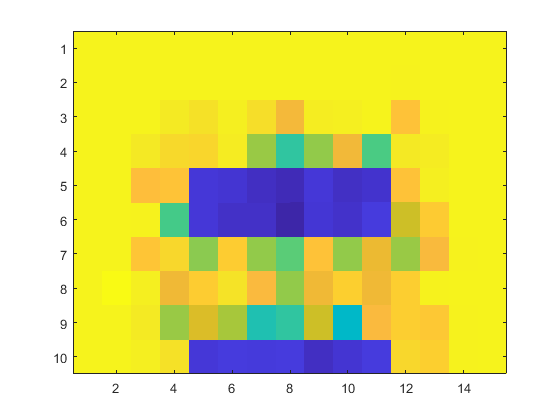
The learning rate influences what to put emphasis on. A value close to 0 puts more emphasis on already learned experience while a value close to 1 does the opposite. For example, with a large learning rate the Q-function will converge to values highly dependent on the last episodes. The concept is illustrated below by plotting the V-functions for world 3 with γ = 0.9 and with the different learning rates of α = 0.1 and α = 0.9 respectively.

****

In the images we can see that there are some slight variations of the V-function due to different learning rates.

1. **Explain how the discount factor γ influences the policy and V-function. Use figures to make your point.**

The discount factor determines if the algorithm should seek to maximize the short-term rewards or the long-term rewards. A value closer to 1 will focus the learning on long-term rewards. By using a low discount factor the robot tends to get stuck in loops since the immediate rewards are evaluated highly. The images illustrate the V-functions for world 4 with α = 0.2 and with the different discount factors of γ = 0.2 and γ = 0.9 respectively.



In the first image we can see that the difference between the tiles are greater than in the second image. That is because in the first image the algorithm mostly cares about the short-term reward. By caring more about the long-term reward, the algorithm gets more robust and reliable. However, by focusing on the short-term reward the algorithm can give better result in some cases but most likely worse overall.

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

The exploration rate influences how probable it is that the algorithm will choose a suboptimal action causing exploration which may result in a better strategy. An example on how the exploration rate affects the result is discussed in question 6.

I used the exploration factor = 1 – (current episode / (number of episodes \* 1.1)) which decreases the exploration in relation to the episodes. This makes the algorithm explore its possibilities early on when there is no prior information about the optimal policy. For each episode the probability to explore decreases which will converge the policy to the optimal solution. The multiplication by 1.1 in the denominator prevents the exploration from reaching 0. This ensures that the training phase can’t get stuck in a loop for certain episodes.

En bild som visar objekt

Automatiskt genererad beskrivningEn bild som visar objekt

Automatiskt genererad beskrivning

The images illustrate the optimal policy from world 3 with the exploration rates ε = 0.1 and ε = 1 – (current episode / (number of episodes \* 1.1)) respectively. The first case resulted in a path above the dark areas using 28 moves. The second case resulted in a path between the dark areas using 14 moves.

1. **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?**

**Suddenly irritating blob:**

The algorithm would always find the route with the least resistance. However, since the world changes the result would be different for the two cases. For the first case with a dark area the path would go around it. For the other case the algorithm would just find the shortest path.

**Irritating blob:**

The route would always go around the dark area similar to what happens with our reinforcement learning approach.

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

In any case where there is a benefit to retain prior knowledge or to discover new unseen strategies. It could for example be applied in robotics to teach robots new tasks while retaining their prior knowledge.

1. **(Optional) Try your implementation in the other available worlds 5-12. Does it work in all of them, or did you encounter any problems, and in that case how would you solve them?**