TBMI26 – Computer Assignment Reports

Reinforcement Learning

Deadline – Mars 12 2018

Author/-s:

David Tran **davtr766**
Jakob Bertlin **jakbe457**

1. **Define the V- and Q-function given an arbitrary policy as well as a given optimal policy (See lectures/classes).**

V-function defines the expected reward by following a certain policy for the state.
Definition:

$$\text{Alt } 2: V(s_t) = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \text{ where } 0 \le \gamma \le 1$$

Makes immediate rewards more important than distant rewards

However, for an optimal policy, we choose the state with maximum V(s). And to find this we can choose to use Monte Carlo approach or Temporal Difference method.

Q-function: Expected future reward of doing action a in state s and then following the optimal policy.
Definition:

$$Q(s_k, a) = r(s_k, a) + \gamma V^*(s_{k+1})$$

V*(s,a) encodes the optimal policy and its value.

2. **Define a learning rule for the Q-function (Theory, see lectures/classes).**

Q-learning rule is defined below:

$$\hat{Q}(s_k, a_j) \leftarrow (1-\alpha)\hat{Q}(s_k, a_j) + \alpha\left(r + \gamma \max_a \hat{Q}(s_{k+1}, a)\right)$$

Our learning strategy for the Q-function is that we have an relatively high exploration factor which gradually decreases as the number of episodes increases. This is done in order to let the agent learn as much as possible in the beginning and then focus on good policies only as the exploration factor decreases.

3. **Describe your implementation, especially how you hinder the robot from exiting through the borders of a world.**

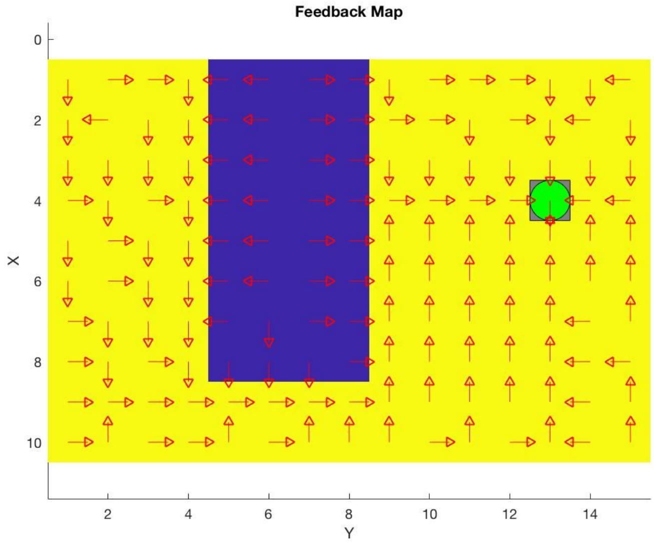
The implementation is done by using Q-learning algorithm to find the most optimal action for given state which maximizes the reward. By executing an action which takes the agent to a bad position, i.e. the borders of the world, we penalize it with -Infinity.

4. Describe the differences between the worlds explored by the robot. Any surprises?

World 1 and World 2 do not have significant difference other than world 2 which introduces some noise to the world. In World 2 there is 2/10 probability to spawn a negative feedback area.World 3 and World 4 have a narrow path between water that is optimal. The major difference between world 3 and world 4 is the spawning position and world 4 sometimes executes the wrong action. More detailed explanation in question 8.

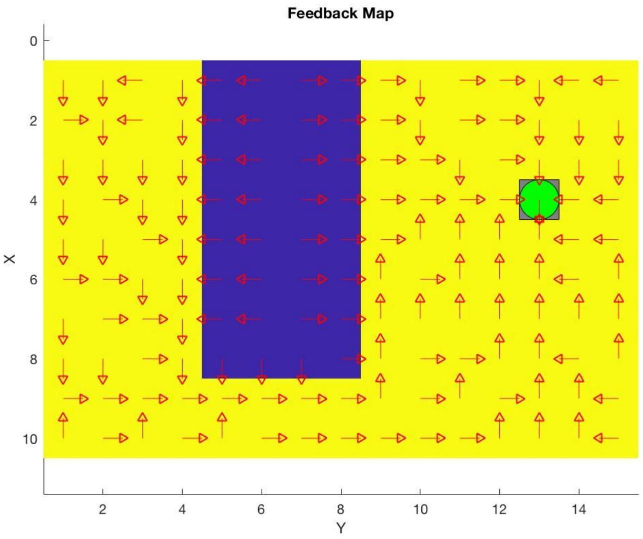
5. For each world: Plot the V-function, i.e. how do you get to the goal from each position.

Plot policy on top of feedback map for Irritating Blob (World 1) below:

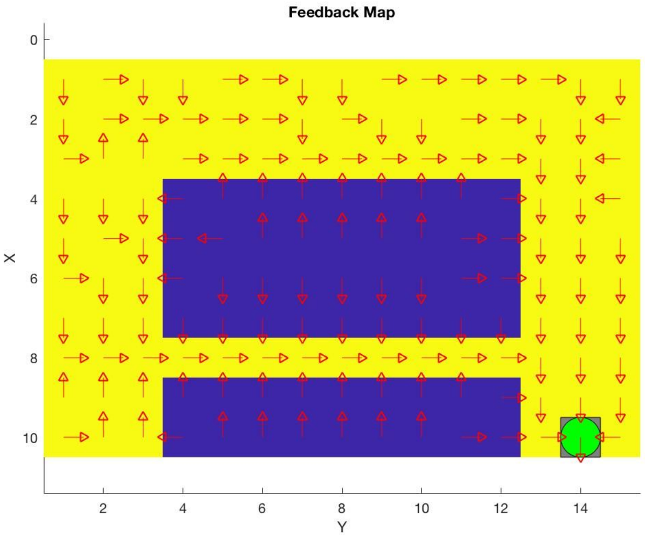


David Tran **davtr766**
Jakob Bertlin **jakbe457**

Plot policy on top of feedback map for Suddenly Irritating Blob (World 2) below:

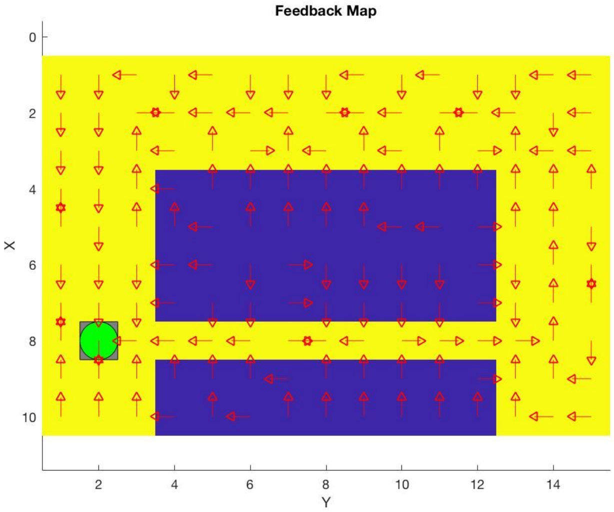


Plot policy on top of feedback map for Road to HG (World 3) below:



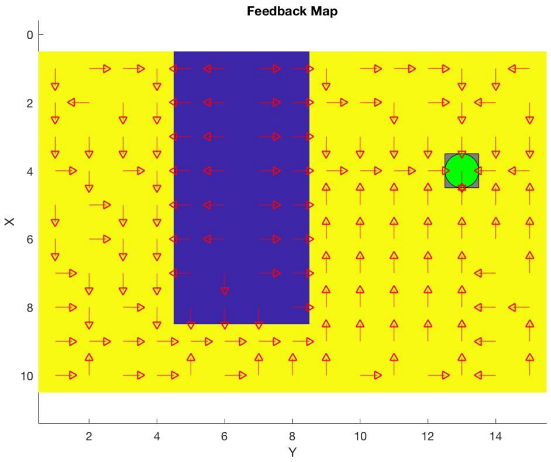
David Tran **davtr766**
Jakob Bertlin **jakbe457**

Plot policy on top of feedback map for Road home from HG (World 4) below:

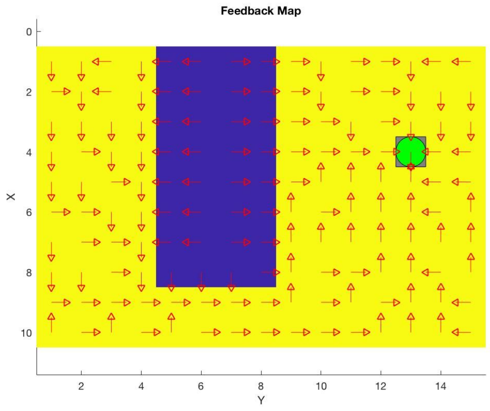


6. **For each world: describe the key observations you have made with respect to parameter choices. Provide documentation of the parameters you have used for each figure! A good rule is to provide each figure with a caption. Plot policies and the V-function for appropriate worlds to the extent you find appropriate in order to explain what you have done and learned during the assignment.**

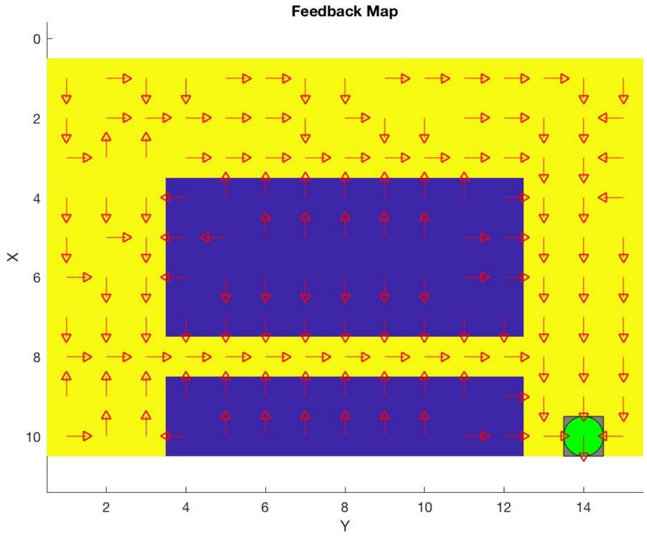
World 1 - Irritating Blob.
Learning rate: 0.8
Discount factor: 0.9
Number of episodes: 500
Motivation of parameters: The first world is a static world, therefore the learning rate can be relatively high in order to learn new information since the world won't change. The number of episodes is quite low because the world is static.



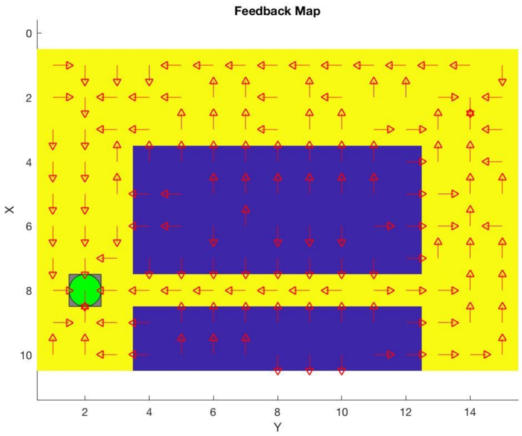
World 2 - Suddenly Irritating Blob.
Learning rate: 0.3
Discount factor: 0.9
Number of episodes: 1000
Motivation of parameters: This world introduces some noise and therefore we have a relatively low learning rate compared to world 1. The number of episodes has also increased.



World 3 - The road to HG.
Learning rate: 0.6
Discount factor: 0.9
Number of episodes: 3000
Motivation of parameters: Increasing the learning rate because the world is static. We want to make each step learn us as much as possible. High amount of episodes furter secures that choose optimal paths to the goal.



World 4 - The road home from HG.
Learning rate: 0.1
Discount factor: 0.9
Number of episodes: 10000
Motivation of parameters: Since the robot can make actions that was not intended it means that individual steps should not affect the total learning too much since the might be completely wrong. And to counter the fact that we decrease learning rates we need to increase episodes to continue to converge to a more optimal solution.The optimal solution is to hug the wall.



7. **What would happen if we were to only use Dijkstra's shortest path finding algorithm in the "Suddenly Irritating blob" world? What about in the static "Irritating blob" world?**

Given a graph or in this case, a grid-like world, Dijkstra's shortest path is an algorithm that finds the shortest path from starting position to the goal node. In the "Suddenly Irritating Blob" it occurs some randomness in the world which may affect the algorithm negatively. Every path from one node to another has a weight and because of the noise, the agent may pick a worse path with Dijkstra's.
However, with a static world, we believe that the agent would perform much better.

8. **Include an in-depth description of the to/from HG worlds (world 3 and 4). What happens on the way from HG? How and why can this problem be solved with Q-learning? Which path does the robot prefer, and why?**

World 3 and world 4 is indeed very similar. The major difference is that the starting position has switched spawn positions. That is why the agent's goal position in world 3 is now the starting position in world 4, thus the title "Road home from HG".

However, a crucial underlying information has been presented in world 4. In the algorithm we choose an action based on our position, exploration rate and our current state. But one may not know that the actual action we want to perform has a ⅓ probability chance that the action executed is not the intended one.

We solved this problem by looking if we hit a wall, then finding out what action has actually been executed and compare it with our intended action. If the action executed was not intended then we penalize it by an arbitrary number not less than 10. However, if the bad action executed was intended then we penalize the agent with -Infinity. This makes the robot avoid positions which might put it in bad states due to the random action, such as walking into a wall or into the water even if you didn't intend to.

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

There are a lot of interesting practical uses with reinforcement learning. Probably the most dominant one is in the field of robotics and industrial automation.

Reinforcement learning is mostly needed when you simply can't use examples to train a machine learning algorithm since reinforcement learning learns it self on the fly.

Examples of this could be autonomous cars. Since reinforcement learning can "see" rewards and penalties in the future, driving should be a great task to apply reinforcement learning to since driving is all about looking ahead and avoiding future dangers.

10. **How does the different parameters influence learning and appearance of the Q- and V-functions?**

- Q(s,a): Mapping between the expected utility associated with executing an action in a given state s.

- Alpha: is the learning rate which essentially is the step size that is taken towards the solution which should converge to a solution. The time it takes to find a solution is depended on the value of the learning rate. A value close to 0 puts more emphasis on already learned experience and a value close to 1 will overwrite previous experience with new information.

- R: is the reward value which is observed in the current state. How good or bad is it that we are this current state?

- Gamma: is the discount factor which determines the importance of future rewards. Depending on the value of the discount factor, the agent can consider current rewards over future rewards instead.

- Eps: Exploration factor which is the probability of taking a random action and “explore” new paths instead of current known paths.