TBMI26 – Computer Assignment Reports  
Reinforcement Learning

Deadline – Mars 12 2018

Author/-s:

In order to pass the assignment you will need to answer the following questions and upload the document to LISAM. If you meet the deadline we correct the report within one week after the deadline. Otherwise we give no guarantees when we have time.

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

Q-function:

V-function:

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

In the beginning we explore more (we use a large ɛ) and then we lower the ɛ and exploit instead.

We can also change the learning rate.

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

We initialize a look-up table for the size of the world (10 \*15) and then depth 4 ( dimension that describes the action). The we let the robot run and learn it’s way. For every 1000th epoch we lover the epsilon so that the robot in the beginning explore a lot and later on it exploits instead.

To avoid the robot bumping into the wall we use a while loop. If the action is invalid a new action is choosen until the action is valid. See following code:

while act.isvalid ~= 1

[a, oa] = chooseaction(look\_up, pos\_prev(1), pos\_prev(2), [1 2 3 4], [1 1 1 1], eps);

act = gwaction(a)

end

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

The obstacles are different in the different worlds. The robot hade some issues with the obstacles and keep going back and forth next to the obstacle.

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

|  |  |
| --- | --- |
|  |  |
| **World 1** | **World 2** |
|  |  |
| **World 3** | **World 4** |

**For world 3 and 4 the algorithm doesn’t converge. It never find a good path.**

1. **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.**

|  |  |
| --- | --- |
| 1. eps = 0.9; 2. alpha = 0.5; 3. gamma = 0.9; | 1. eps = 0.9; 2. alpha = 0.5; 3. gamma = 0.9; |
| **World 1** | **World 2** |
| 1. eps = 0.9; 2. alpha = 0.4; 3. gamma = 0.9; | 1. eps = 0.9; 2. alpha = 0.25; 3. gamma = 0.9; |
| **World 3** | **World 4** |

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

Then the look-up table would be initialized from us in the beginning (the rewards would already be there), which mean that the robot doesn’t learn by itself, thus the robot can not be better than the teacher.

1. **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?**

Since it starts on the same position every time the training initializes, it will always try to explore in the beginning and try to find a way to the goal. But since the world is a bit tricky, it will take some time for it to find the goal. Closer to the end of the learning, the algorithm should have mapped a optimal path to the goal and will therefore start to use a more exploitation method to find the closest part to the goal. This is the exploration-exploitation dilemma. When do we decide that we have explored enough? The robot doesn’t find a good path and it still takes a huge amount of time to reach the goal.

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

In manufacturing we can train a robot to optimize the speed of, for example, packing a box for shipping. This robot can learn by failing and then optimize on many different situations of packing.

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

The different parameters influence in terms of how the algorithm learns the optimal path to the goal. A lower learning rate puts more focus on already learned experience and a higher rate will overwrite previous learned experience.   
The discount factor regulates the long-term and short-term rewards. A values close to zero will maximize a short term reward. A high value will maximize long-term reward.

The exploration factor regulates how much the algorithm will explore. A high value will explore a lot and a low value will exploit a lot.