# Proposed solution to Exercise 4

## Reinforcement learning

The main floor of a building is divided into 6 zones (see figure below). A cleaning robot needs to learn how to find its docking station in zone 6. That is a goal in “return mode”. Zone 3 has a staircase to the basement that pose a risk for the robot. It can fall off the edge. You model the floor plan as shown and assign a reward of +100 to the goal state. The zone with the staircase has been marked with a -10. Other zones are associated with no penalty or reward. Diagonal movements are not allowed.

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 -10 |
| 4 | 5 | 6 +100 |

To learn how to navigate to its docking station the robot is placed in arbitrary zones and allowed to roam in order to find its way. Yet, the memory of its search and rewards are maintained by means of Q-learning and a Q-matrix.

1. Construct the R-matrix and initial Q-matrix for this problem based on the figure above.
2. The robot is placed in state 4 first and consider a move to 5. Calculate the new Q-value for this state-action pair and update the Q matrix.
3. The robot is placed in state 5 and consider the move to 6. Calculate the new Q-value for this state-action pair and update the Q matrix.
4. The robot is placed in state 4 again and considers the move to 5. Calculate the new Q-value for this state-action pair and update the Q matrix.
5. The robot is placed in state 1 and consider the move to 4. Calculate the new Q-value for this state-action pair and update the Q matrix.
6. Based on the information entered, which path is the most likely choice of the robot at this stage when dropped in zone 2?

The discount factor is 0.9 and the learning factor is 1.0.

Q-value:

Q(s,a) 🡨 Q(s,a) + α( r + ϒ maxQ[s’,a’] – Q[s,a])

1. Construct the R-matrix and initial Q-matrix for this problem based on the figure above.

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 -10 |
| 4 | 5 | 6 +100 |

This yields the following R-matrix:

Each row indicates a state. Each column indicates an action taking the agent to a new state

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 |  | 0 | 0 | 0 |
| 0 | 0 | -10 |  | 0 |  |
|  | 0 | -10 |  |  | +100 |
| 0 |  |  | 0 | 0 |  |
|  | 0 |  | 0 | 0 | +100 |
|  |  |  |  | 0 | 0 |

The corresponding Q-matrix at the start:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 |  | 0 | 0 | 0 |
| 0 | 0 | 0 |  | 0 |  |
|  | 0 | 0 |  |  | 0 |
| 0 |  |  | 0 | 0 |  |
|  | 0 |  | 0 | 0 | 0 |
|  |  | 0 |  | 0 | 0 |

1. The robot is placed in state 4 first and consider a move to 5. Calculate the new Q-value for this state-action pair and update the Q matrix.

Using

Q(s,a) 🡨 Q(s,a) + α( r + ϒ maxQ[s’,a’] – Q[s,a])

Q(4,5) 🡨 Q(4,5) + α( r(5) + ϒ maxQ[(5,5), (5,4), (5,6)] – Q[4,5])

All Qs are = at this time α=1 and ϒ=0,9 this yields new Q(4,5) = 0

1. The robot is placed in state 5 and consider the move to 6. Calculate the new Q-value for this state-action pair and update the Q matrix.

Q(5,6) 🡨 Q(5,6) + ( r(6) + 0,9 maxQ[(6,5), (6,6), (6,3)] – Q[5,6])

New Q(5,6) = r(6) = 100

1. The robot is placed in state 4 again and considers the move to 5. Calculate the new Q-value for this state-action pair and update the Q matrix.

Q(4,5) 🡨 Q(4,5) + α( r(5) + ϒ maxQ[(5,5), (5,4), (5,6)] – Q[4,5])

Now Q(5,6) = 100 which means that Q(4,5) = 0,9 \* 100 = 90

1. The robot is placed in state 1 and consider the move to 4. Calculate the new Q-value for this state-action pair and update the Q matrix.

Q(1,4) 🡨 Q(1,4) + α( r(4) + ϒ maxQ[(4,4), (4,1), (4,,5)] – Q[4,5])

Now Q(4,5) = 90 which means that Q(1,4)) = 0,9 \* 90 = 81

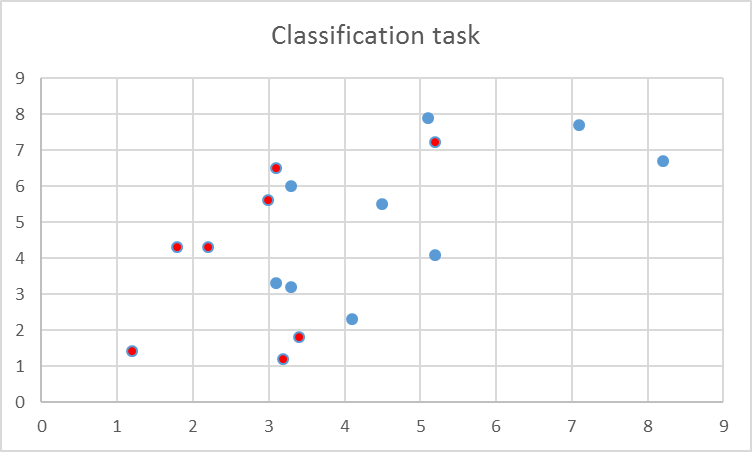
1. Q-matrix:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | 0 |  | 81 | 0 | 0 |
| 0 | 0 | 0 |  | 0 |  |
|  | 0 | 0 |  |  | 0 |
| 0 |  |  | 0 | 90 |  |
|  | 0 |  | 0 | 0 | 100 |
|  |  | 0 |  | 0 | 0 |

## 3 Classification

See the scatter plot below.

You will test 3 algorithms to be used for categorization of the observations in the plot below. The coordinates of these observations are listed in Table 1.



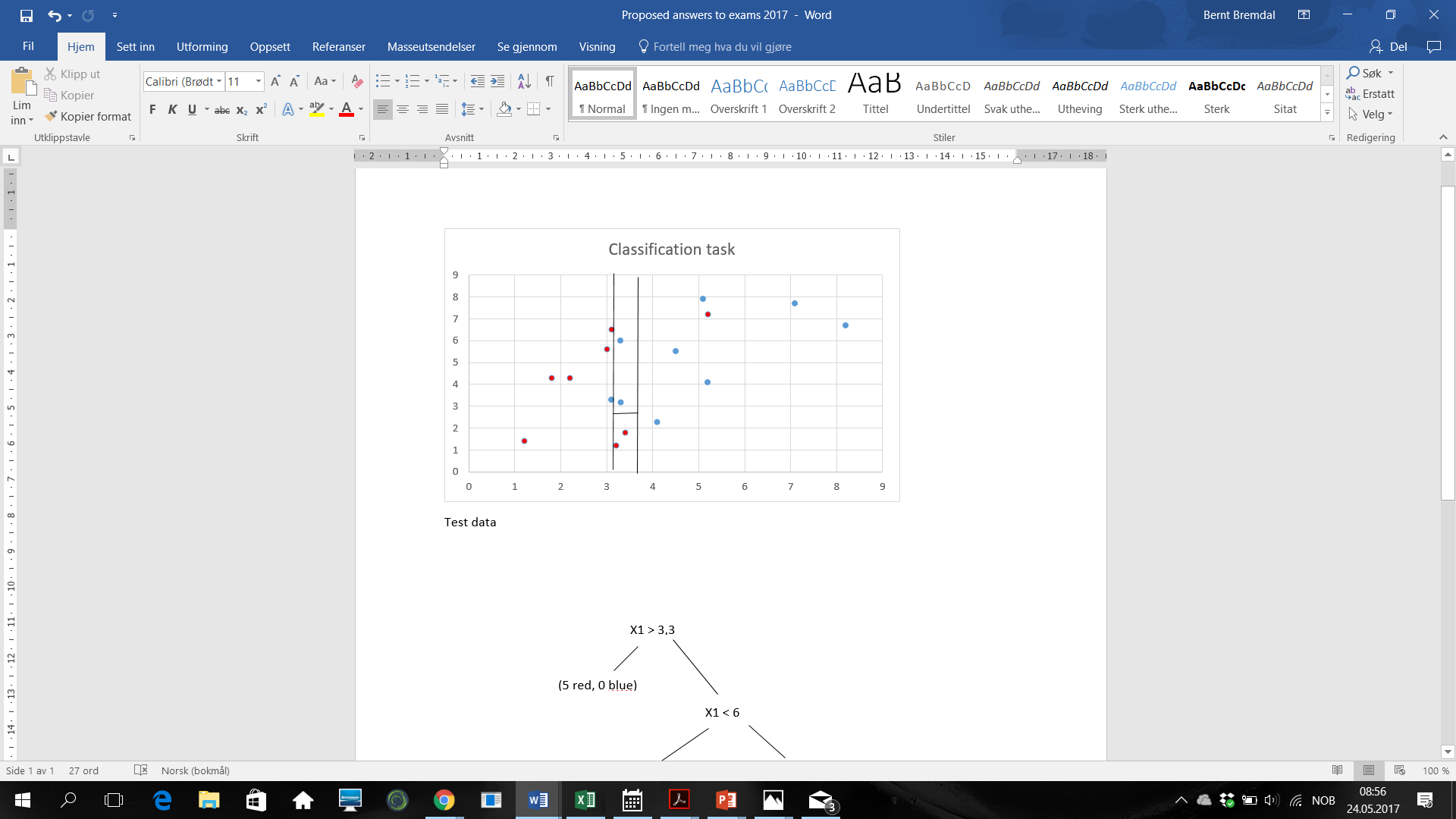
In addition there are two test observations, (X1,X2) ϵ S as blue or red that will not be part of the training.

(6,5 , 5,5), (4,1 , 6,5)

1. Create a CART classification tree with maximum 4 regions.
   1. Justify the splits.
   2. Use your classification tree to classify the test observations as blue or red.



With simple heuristics to guide the split there are multiple solutions, none really better than most others.



(0 red, 2 blue)

(2 red, 0 blue)

(1 red, 6 blue)

(5 red, 0 blue)

X2 > 2,8

X1 < 6

X1 > 3,3

Here the splits are minimized to reduce the overall entropy and increase the information gain. This is simply done graphically by sliding a ruler along the X1 and X2 axes and counting the ratio between red and blue dots.

By the majority vote rule.

(6,5 , 5,5) 🡪 Blue

(4,1 , 6,5) 🡪 Blue