**Machine Learning – Q-Learning Report**
**CSC3022H**
**19/10/2015**

**Alon Bresler (BRSALO001)**

---

**Method Pseudo-Code:**

Reward Function:

```
bool checkObj = false

for each object on the grid:

    get the object x and y position

    if objects x,y position is equal to the sweepers x,y position:

        if the object is a mine:
            clearState(x, y, sweeper_no)
            return mineReward

        if the object is a rock:
            return rockReward

        if the object is a superMine:
            clearState(x, y, sweeper_no)
            return supermineReward

        checkObj = true
```

clearState Function:

```
clear all the sounding state of a block where a mine or super mine was found
```

highestHistoricReturn Function:

```
generate a random probability = prob

if prob <= epsilon and not findMax call:
    choose an action at random → 0-3

else:
    if all possible actions have same value:
        choose one at random

    else:
        choose action with the highest value
```

Update Function:

If all sweepers are dead:
End the current iteration

Every 1000 iteration – reduce the epsilon value by 0.01

For each sweeper:
Get the x and y position of the sweeper on the grid
Get the current state at the position of the sweeper

Get the next action to be performed by the sweeper with the highest historic return.
Set the rotation of the sweeper with the selected action.

Update the parent object

For each sweeper:
Get the sweeper's state's x and y position.
Get the sweeper's current x and y position.

Get the sweepers next state from the qTable
Select the sweepers next best move with the highest historic return

Get the reward at the sweeper's position.

Get Q value using: Q(s,a) = Q(s,a) + (learning rate * (Reward + discount * (Q(s',a')) - Q(s,a)))

**Results**

The results are written to a text file each iteration to be examined after running the simulation.

I allowed the sweepers to run for 50 iterations to learn the desired behaviour, then an average was taken over the next 50 iterations which were used for the results.

Results for testing for the best learning rate:

| Learning Rate | Average Mines Gathered | Most Mines Gathered | Number of Deaths |
|---|---|---|---|
| **0.25** | 0.9027 | 3.3 | 10.74 |
| **0.5** | 0.9740 | 3.5 | 8.66 |
| **0.75** | 0.9860 | 3.52 | 8.1 |

Results for testing for the best discount factor:

| Discount Factor | Average Mines Gathered | Most Mines Gathered | Number of Deaths |
|---|---|---|---|
| **0.3** | 0.9880 | 3.84 | 10.28 |
| **0.6** | 1.0647 | 3.92 | 7.16 |
| **0.9** | 0.9887 | 3.54 | 10.98 |

Results for the different test environments:

| | **Average Mines Gathered** | **Most Mines Gathered** | **Number of Deaths** |
|---|---|---|---|
| **Environment 1** | 1.0647 | 3.92 | 7.16 |
| **Environment 2** | 0.6540 | 2.58 | 13.08 |
| **Environment 3** | 0.1467 | 1.3 | 19.44 |