Jeremy Coupland

Machine Learning: Q-Learning Assignment Report

October 2015

Method Pseudo-code:

This is the pseudo-code for the most significant methods used in this assignment.

Initializing The Learning Algorithm: Fills the Q Table of each sweeper.

Fill a 2D vector of integers with the positions of the mines, which will serve as a map.

Initialize the Q table to be a 3D vector of doubles, where the first index is the sweeper number, the 2nd is the x coordinate, and the third the y coordinate: QTable[sw][x][y].

Loop through each x and y coordinate for each sweeper, setting the value of the QTable equal to the map where there is a mine, and 0 for every other index.

R: returns the reward of the current position QTables[sw][x][y]

Update: Moves and updates the values of each sweeper and their QTables.

Loop through each sweeper in the Qtable

For each sweeper, calculate the values of each of the states around them (up, down, left, right).

Once all values are calculated, find the max and set the rotation of the sweeper to face it.

Once all sweepers are rotated and ready to move, call the parents update() method, which moves the sweepers.

Loop through each sweeper again:

If the sweeper is dead, set the sweepers current QTable value to equal -100, as they must have moved onto a rock or supermine. Update the previous states value to equal: Qtables[sw][prevXCord][prevYCord] + learningRate * (R(prevXCord, prevYCord, sw) + (discountFactor * Qtables[sw][xCord][yCord]) - abs(Qtables[sw][prevXCord][prevYCord]));

If the sweeper is not dead:

Just update the previous states value using the same formula above.

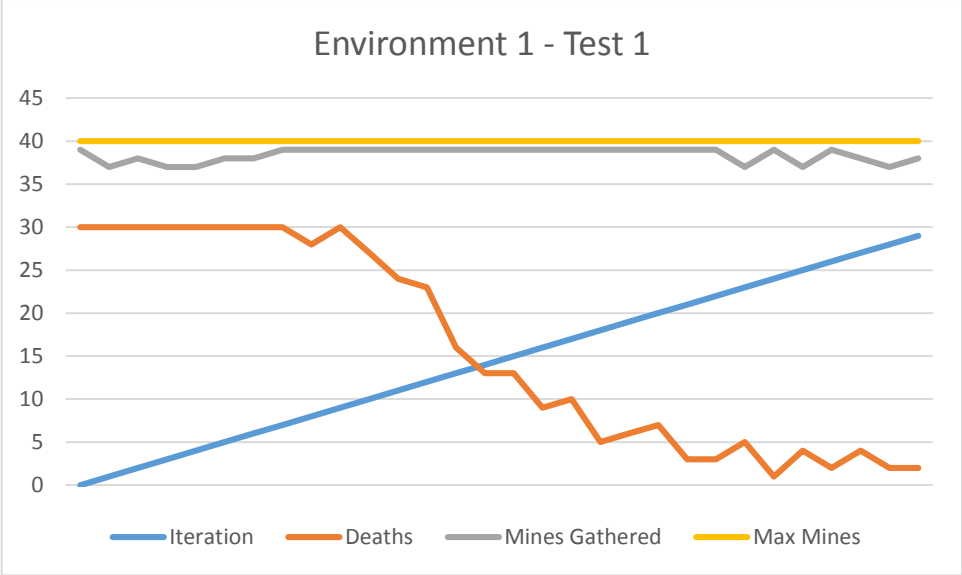
Testing Results:

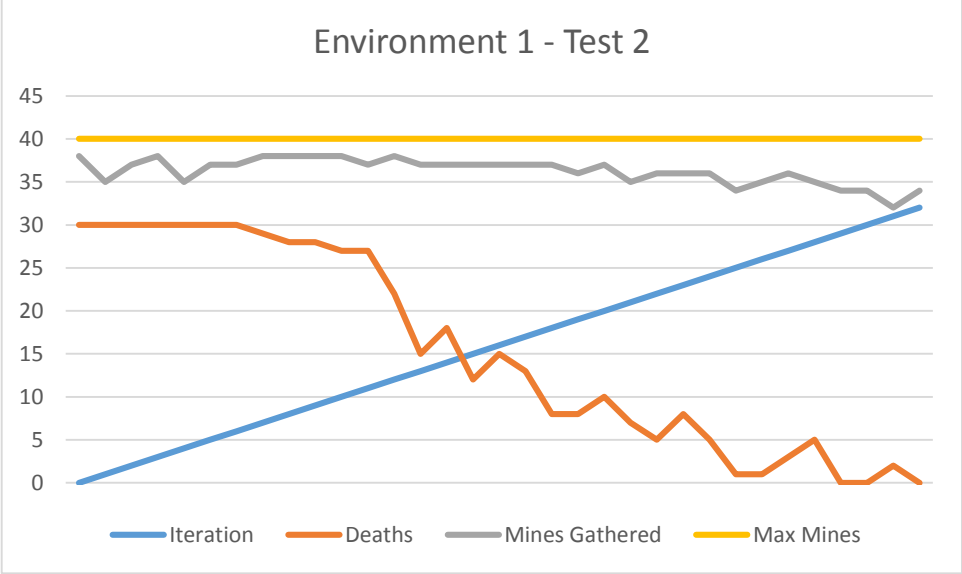
The following test results show how my sweepers have learnt reliable paths to the mines over multiple iterations. Three different testing environments were used, as well as two different test cases for each environment were used. Individual results can be found in the attached spreadsheet.

In each of the following graphs, four different lines are shown. The yellow line is the number of mines present in each iteration, and can be used as a baseline to be compared with the grey line, number of mines gathered in each iteration. The red line is the number of deaths per iteration. Finally the blue line is the current iteration number, which changes in increments in certain test cases.

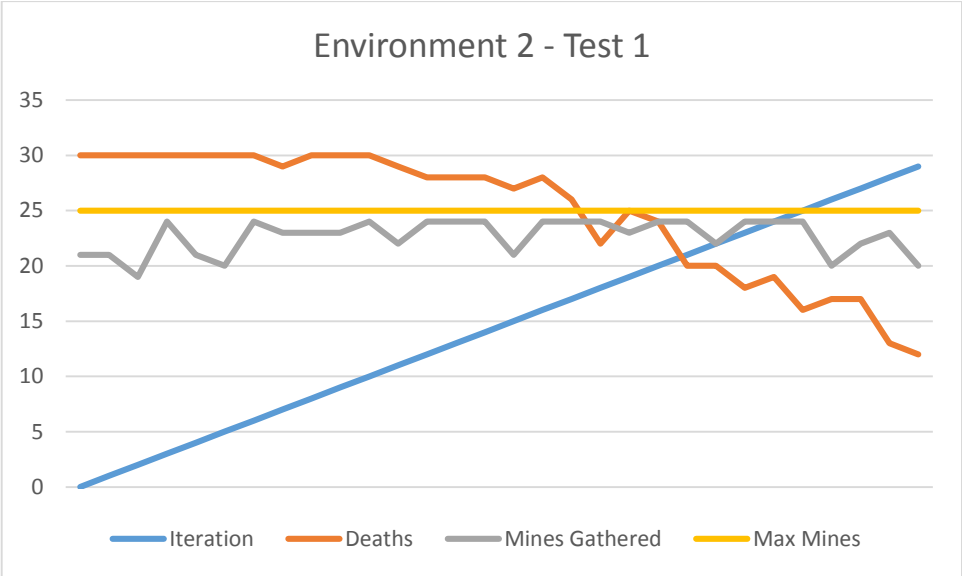
In each of the test cases, we can see that the number of deaths per iteration slowly decreases and converges towards zero, while the number of mines gathered remains close to the maximum number of mines present.

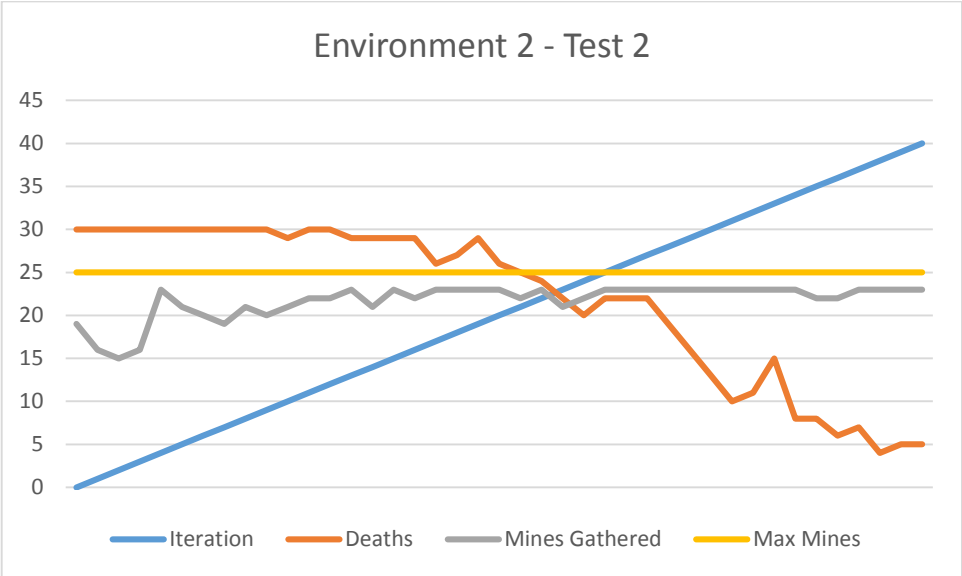
Environment 1:



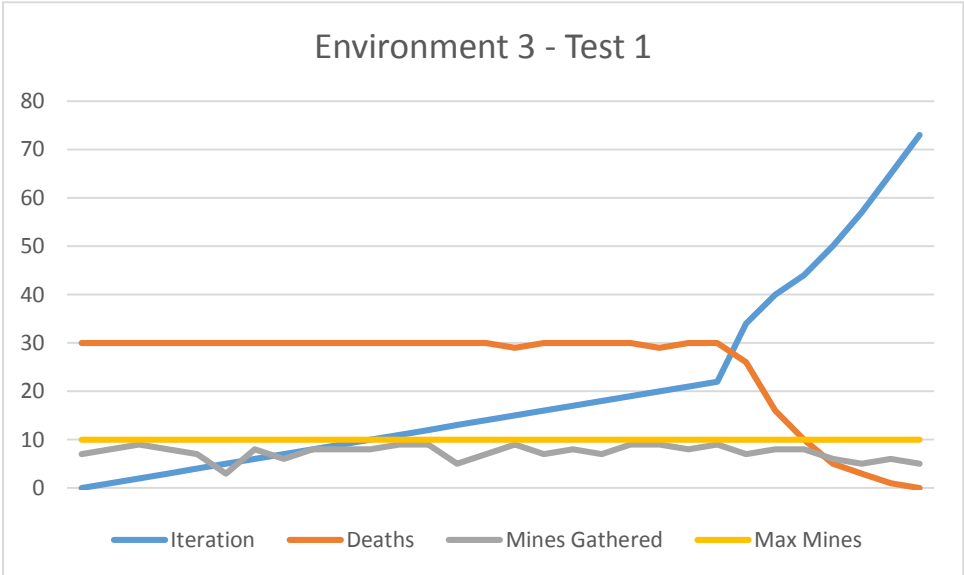


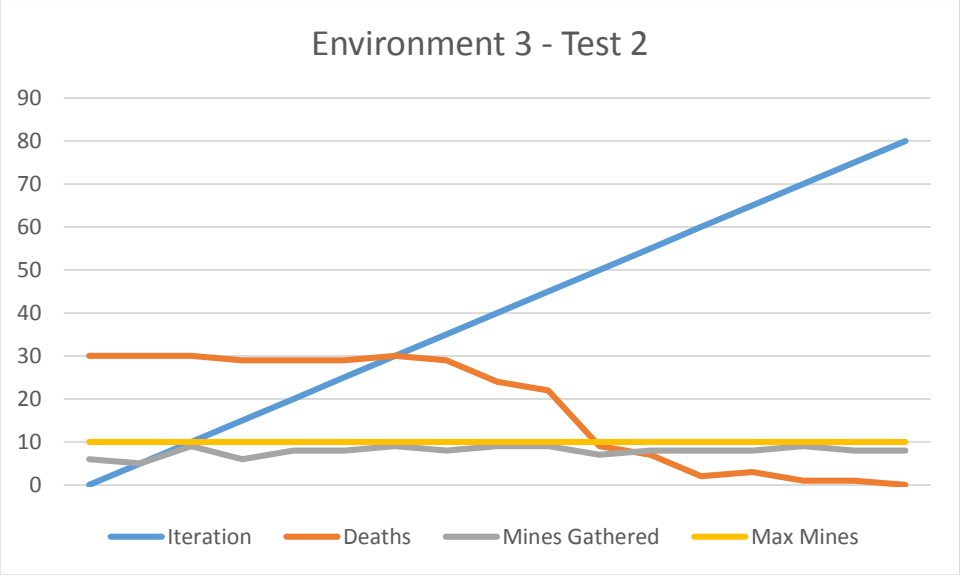
Environment 2:





Environment 3:





Conclusion:

I found that while testing, in all environments the number of deaths slowly decreased while the number of mines gathered also either decreased, increased or remained more or less the same. This may be because, in some environments, the sweepers are finding some paths to be too dangerous to move around, and skipping mines contained within, whereas in other environments, the mines may be easily accessible. However, the rate at which the number of mines gathered decreases is so low that it is almost negligible.