CS 440
MP 1 Part 2

Nathaniel Cook nvcook2

Sep 29 2014

Problem 1.

(a) The Q learning agent quickly, after 5 episodes, finds a solution that provides a 0 reward which is the best it has seen and then for the rest of the 1000 episodes follows the same policy. It never tries anything new.

(b) The learning agent now tries things it hasn't ever tried before and so explores the world better. The reward it achieves is significantly higher at anywhere from 100-200. Instead of just sticking to what it has already seen the learner is willing to try different actions that may result in better or worse rewards. As a result it is able to find better solutions. Charactarizing the problem a function the behavior in part (a) found a local minimum and was content to stay there. In this part the randomess associated with ϵ allows the learner to climb out of local minima and find either the global minimum or at least a better local minimum.

(c) With an higher ϵ the learner is to agressive at trying new unknown states. The results is that it rarely if ever achives a positive reward. The learner is too optomistic that trying something that failed before would eventually provide a positive reward and so it continues to attempt states that penalize it.

Problem 2.

(a) With these settings the learner fails to achieve a positive reward. Since it does not know where the thief is it cannot learn to avoid it. The penalty of being caught and the chance of being caught combine to form a world where its difficult to achieve a positive reward.

(b) When the learner is aware of the thief it is able to learn how to avoid the thief achieving rewards of 300+. Since 'knowing' where the thief is, is represented by new states in the world the learner learns to avoid states where the thief will it will get caught.

(c) The best learning rate was 0.04 with an ϵ of 0.01. I searched for these values by simple scanning the two dimensional space incrementing by 0.01 starting from 0 to 0.2 for each value(using a simple script). I scored each run by the average of the last 100 episodes. With this rate and ϵ I was able to achieve an average reward of 332 over the last 100 episodes. I also used some simple minimization alogithms in the scipy package, but these methods were unsuccessful, they either changed the values too slowly and never improved the solution or changed the values too quickly and got lost. I decided a simple grid search would be better and it was.

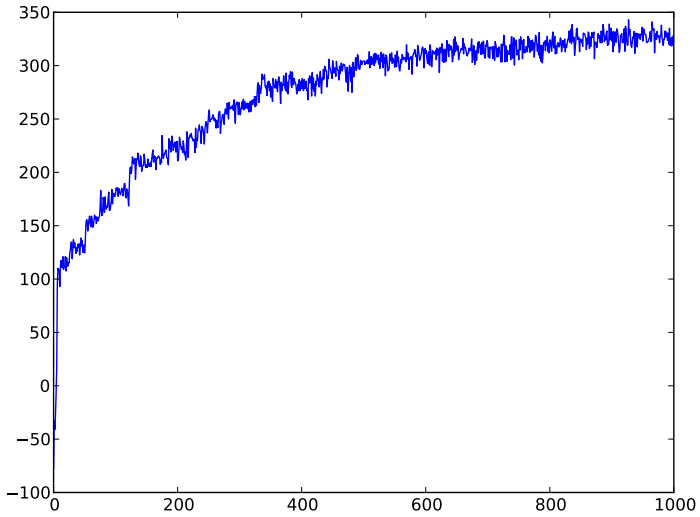

Figure 1: The first run using a learning rate of 0.04 and ϵ of 0.01

Problem 3.

As can be seen in fig 1 the learner very quickly learns basics of the world. It is able to go from a negative 50 reward to about a positive 110 reward in the firt dozen episodes. It then spends the rest of the episodes approaching a greater reward with diminishing returns. There are episodes where the learner drops significantly for a given episode but on the whole it improves. For the average of 10 runs (see fig 2) the function is smoother. There is less variance and doesn't suffer from episodes of significant drops in rewards. The last figure fig 3 plot both the initial and averages so that can easily be compared.

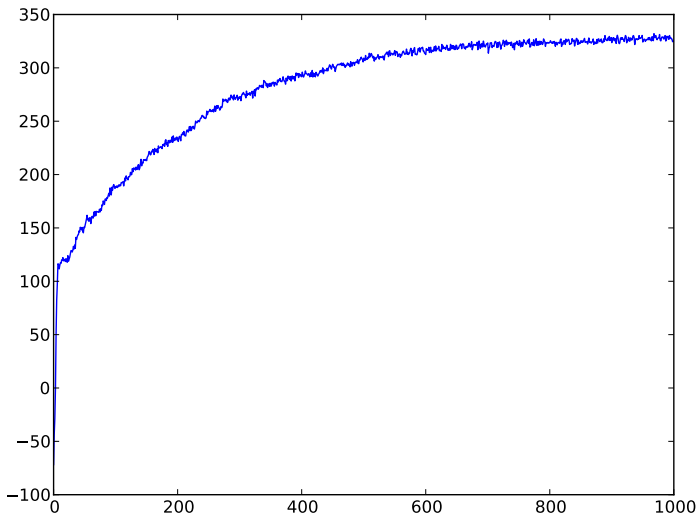

Figure 2: The average of 10 runs using the same parameters:

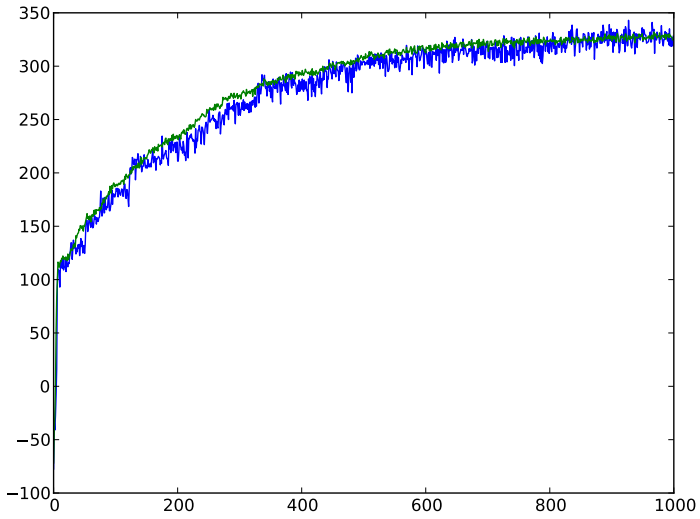

Figure 3: The two plots on the same graph. Blue is the initial run and green is the average: