COMP532 Assignment 1: Reinforcement Learning

Frances Crouch and Rodrigo Gonzalez

Question 1 - see egreedy.py for code

See Figure 1 for the output of our code. The top graph plots the average reward at each play of the game, averaged over 2000 tasks. We see that the greedy method ($\epsilon = 0$) plateaus at an average reward of 1, a lower value than the other methods. This is because the greedy method fails to find the optimal value across all the bandits but instead sticks to one of the first bandits it chooses. Both of the ϵ-greedy methods obtain a higher average of around 1.4 after 1000 plays, but with more exploration ($\epsilon = 0.1$), the optimal value is found more quickly.

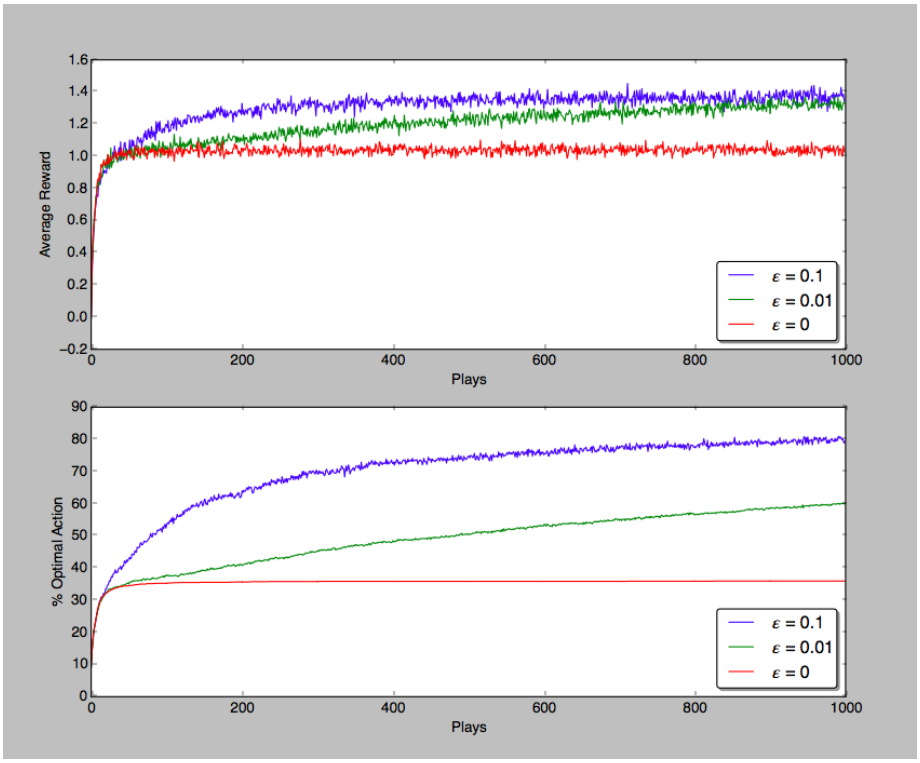

Figure 1: Average performance of ϵ-greedy action-value methods on the 10-armed testbed (as per Sutton & Barto).

The lower graph shows the percentage of times the optimal action was taken across the whole game, averaged across all 2000 tasks. The greedy method chooses the optimal value around a third of the time, while the ϵ-greedy methods improve this percentage as they find the true optimal value with more exploration. We also noted that the greedy method line is very smooth, while the other lines are jagged, due to the element of randomness in the ϵ-greedy methods.

We further experimented with this game by changing some of the parameters. Firstly, we investigated what would happen if the game was extended to 2000 plays (instead of just 1000). Figure 2 shows the output of this experiment. This shows that $\epsilon = 0.01$ actually overtakes $\epsilon = 0.1$ in terms of the average reward received. This is because both methods find the optimal value, but with less exploration the $\epsilon = 0.01$ method chooses the optimal value more often. Indicating that, initially, it is better to have a higher exploration rate, but assuming the action values remain constant, it could be beneficial to reduce the exploration rate over time.

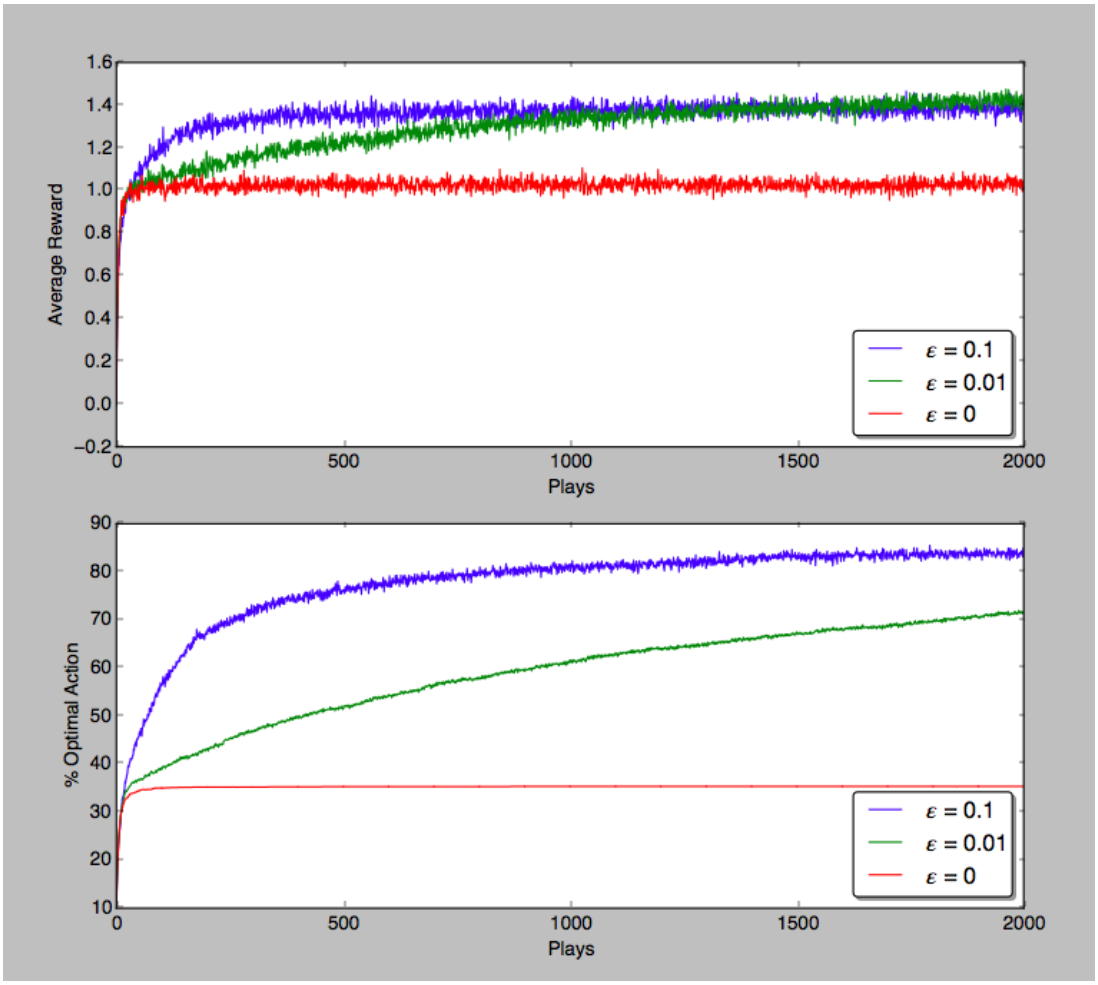

Figure 2: Extending the game to 2000 plays

Next, we changed the reward variance from 1 to 5, e.g. making the data more noisy. The lines on upper graph are a lot more indistinct, which reflects the fact that the reward values have additional variance. Initially, the greedy method appears to achieve a higher average reward, but after 1000 plays, it is clear that the $\epsilon = 0.1$ method achieves a higher average reward value.

The lower graph shows that the greedy method only picks the optimal value around a third of the time, consist with our results from the original experiment, but it takes more time to reach this percentage. In contrast, neither of the ϵ-greedy methods achieve the same results as the original (less noisy) experiment. This is because the additional noise makes it more difficult to determine what the action values are, and hence makes it difficult to determine which is the optimum. It is also interesting to note that, initially, the greedy method out performs the ϵ-greedy methods.

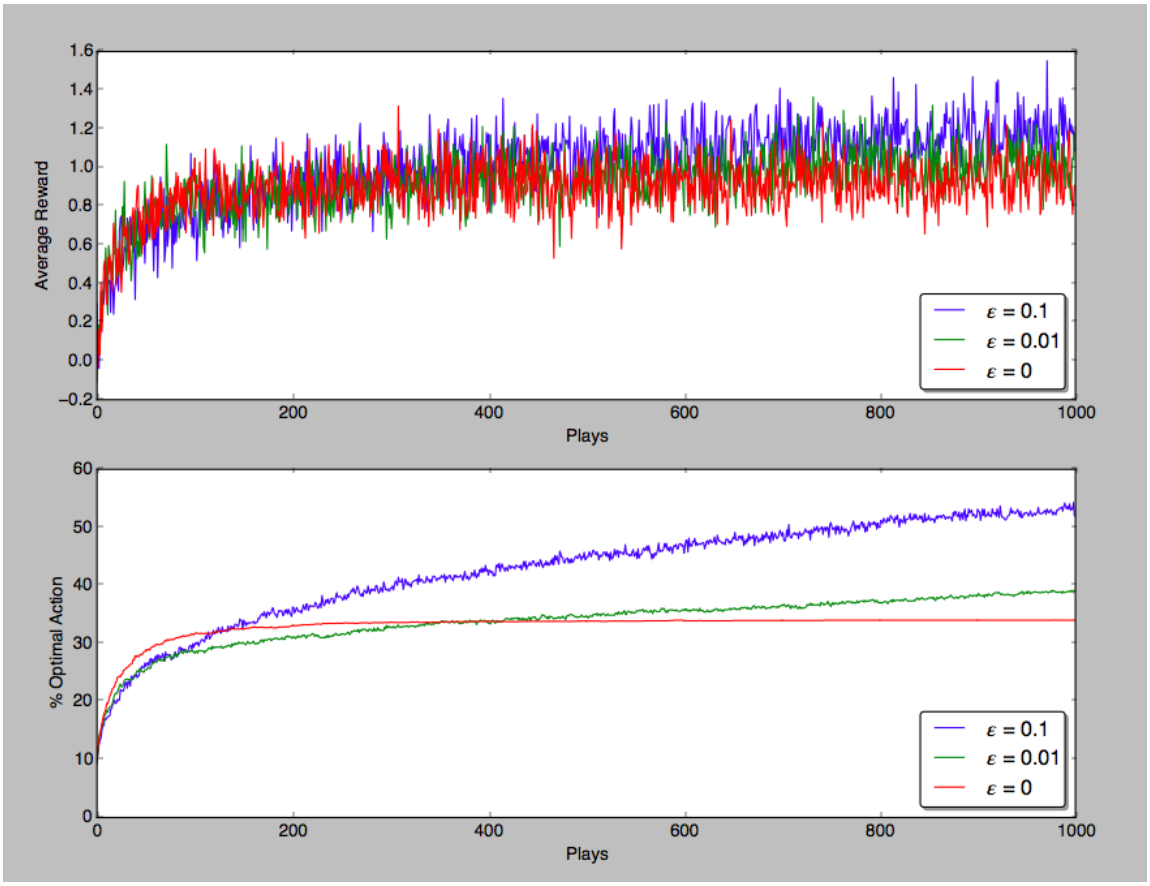

Figure 3: Adding more variance to the reward.

Finally, we extended the previous experiment to a total of 5000 plays, see Figure 4. These results resemble our original output; both graphs have a similar shape to Figure 1. However, the average reward graph has a "thicker" line due to the additional variance and the optimal action graph shows that the ϵ-greedy methods do not achieve as high as percentage as the original experiment, as it take longer to determine the optimal action.

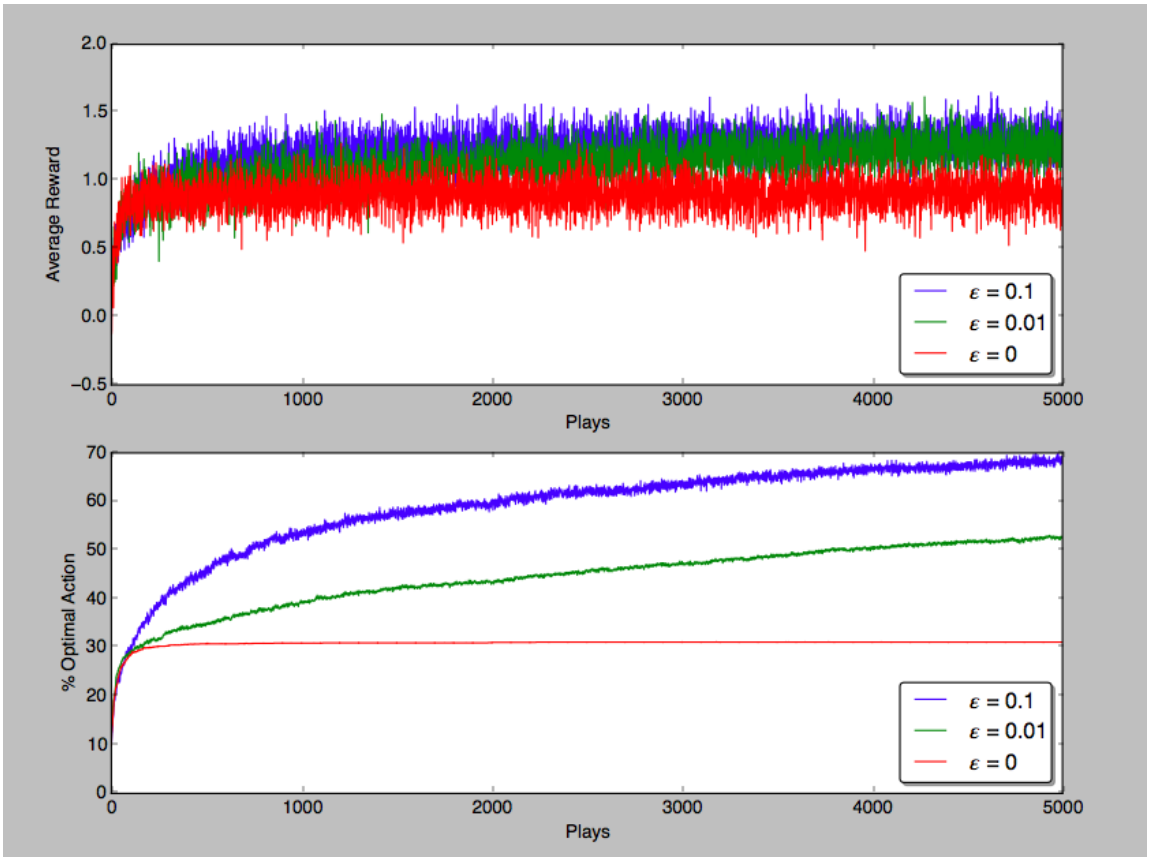

Figure 4: Extending the game with additional variance