CPS841 Reinforcement Learning - Assignment 1

Nariman Saftarli - nsaftarli@ryerson.ca

January 31, 2019

Question 1. Both the UCB algorithm as well as the ϵ-greedy algorithm were implemented, the latter as a baseline. Running each algorithm for $100,000$ rounds, the following observations were made:

1. For both the UCB algorithm as well as the ϵ-greedy algorithm, the real probability distribution and estimated probability distribution were plotted. The results were as expected. As the focus is to select the option with the highest chance of receiving a reward, the estimated probability distribution does not generally accurately represent the true probability distribution for any more than 1 of 10 values where 10 represents the number of actions that can be taken in the 10-armed bandit problem. This is because there's no motivation other than maximizing the reward signal over time, which makes low-valued actions irrelevant over time, and predicting their actual probabilities is unnecessary; all that's necessary is to confirm that they are not optimal. Figure 1 illustrates this. There are two things important to note here. The first is that the estimated probability is acceptably accurate for a *single* value in the distribution (*i.e.* the peak of the estimated probability distribution is reasonably close to the actual value at that index in the true probability distribution). The second is that the ϵ-greedy method failed at finding the optimal move.

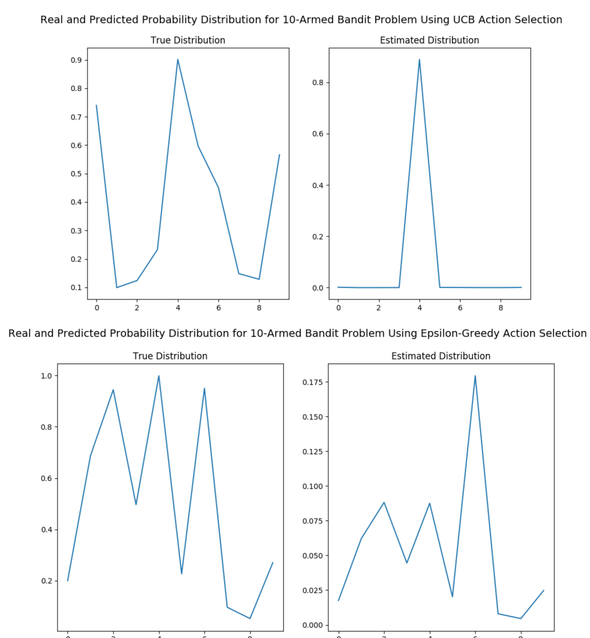

Figure 1: Both the UCB and ϵ-greedy algorithms predict the entire distribution incorrectly, but fairly accurately approximate a single probability.

2. The UCB algorithm converged on the optimal action much more reliably

in the 10-armed bandit problem than did the ϵ-greedy algorithm. This also makes sense since the ϵ-greedy algorithm fails to have some measure of uncertainty. That is, if $\epsilon = 0.9$, and some action produces a positive reward, the ϵ-greedy algorithm is much more likely to choose that value and get stuck at a local optimum, because there's nothing to specify its uncertainty about the potential of other values to yield a high reward. This is shown in figure 2, as the algorithm starts off making the optimal action 10% of the time (as expected), but which quickly drops down to 1% as the non-optimal action is chosen more often. The UCB algorithm however increases the rate at which it makes the optimal action. This is because of two major factors. The first is that the UCB algorithm (*i.e.* action selection specified by $A_t = \text{argmax}_x[Q_t(a) + c\sqrt{\frac{lnt}{N_t(a)}}]$ has some notion of uncertainty, while the second makes it such that the degree of uncertainty is decreased over time.

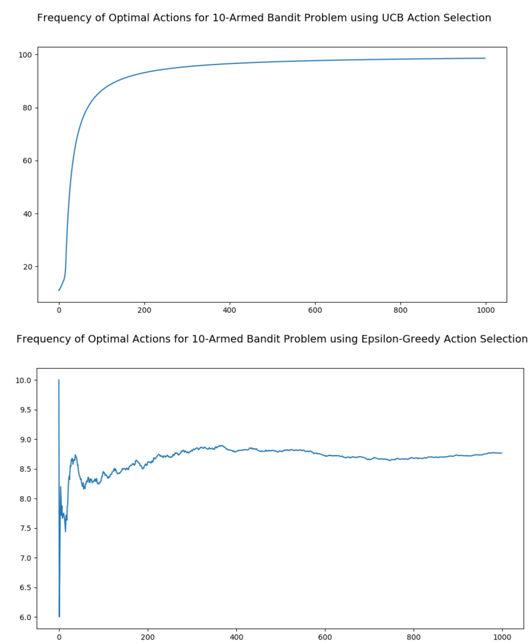

Figure 2: The UCB algorithm quickly converges on making the optimal move, whereas the ϵ-greedy algorithm gets stuck at a local peak.

3. The UCB algorithm increases average reward over time, getting converging on the highest average reward possible whereas the ϵ-greedy algorithm converges on a lower average reward. This is again due to the fact that the ϵ-greedy algorithm is more likely to pick actions that are local optima. Figure 3 illustrates the convergence of the average reward over time.

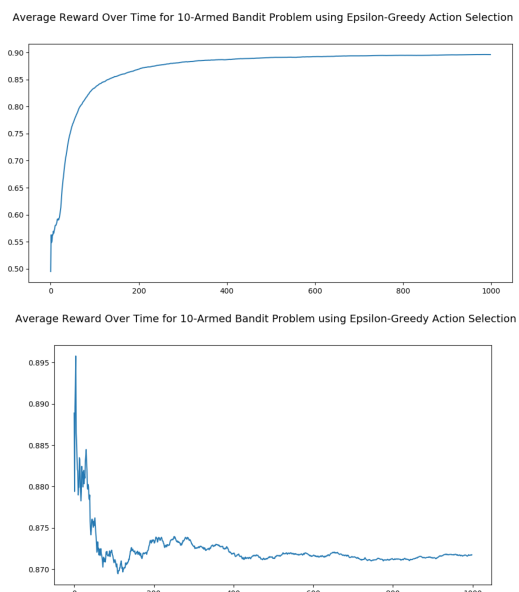

Figure 3: The UCB algorithm converges to a higher average reward than the ϵ-greedy algorithm due to its increased robustness to local optima (making the global action more likely to be chosen over time).