Reinforcement Learning HW1

Hadi Askari

April 2023

1 Implementation

The code is contained in the file Askari_Homework1.m and takes approximately 30 seconds to run. First, we run the entire flow for the epsilon greedy algorithm and then the entire flow for the UCB algorithm.

Initially we generate the 10 arm testbed rewards using a normal distribution of mean 0 and variance 1 with arms 1 to 10. These values are different each time you run the algorithm hence will get slightly different average reward values than the book.

Next for 2000 iterations (number of experiments) we run multiple bandit experiments to calculate the rewards. We initialize q is the state-action space and n is the number of times that action was taken. Both are initialized to zero and our rewards for each time step (1000 in our case) are also initialized to 0.

Next we calculate which action to take by the epsilon greedy formula with epsilon as 0.1:

- `randn(1, 10)` with probability ϵ (exploration)
- `argmax(q)` with probability $1 - \epsilon$ (exploitation)

We update the reward, q and n. Q is updated by the following formula:

$$q(A) = q(A) + \frac{1}{n(A)} * \big(Reward - q(A)\big);$$

Finally the total rewards collected in each of the 2000 iterations are collected and the first 1000 of them are plotted.

The flow for UCB was also very similar to this. The algorithm was implemented as follows. Initially take all actions once, then compute the upper confidence bound of all actions using the following formula:

$$\text{UCB}(A) = q(A) + c * \sqrt{\frac{\ln(t)}{n(A)}}$$

where t is the number of steps taken so far. $n(A)$ is the number of times that action was taken and c is a hyperparameter (2 in our case). Then return the action that corresponds to the highest UCB value.

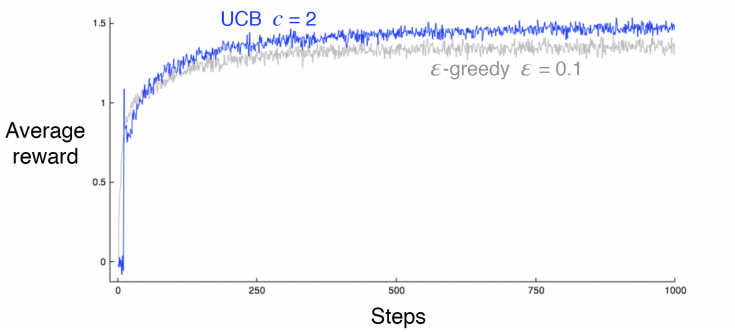

Figure 1: Books Image

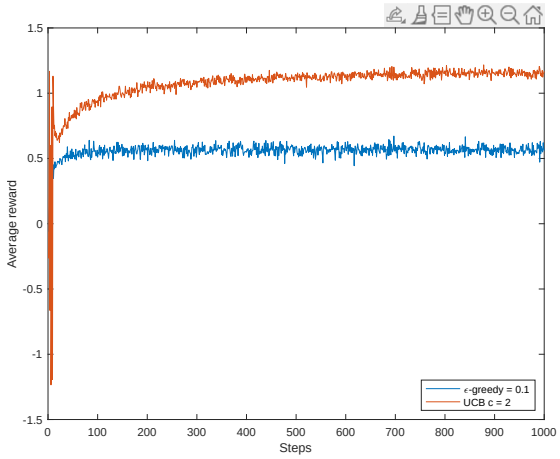

Figure 2: Image Generated

2 Figures

The final plots can be seen as following:

As you can see both the UCB and the epsilon greedy algorithms follow the similar trajectories as the book. The average reward is a little different due to the different initialized of true reward values. There is a spike in UCB after the 10th iteration since it has explored all of the actions atleast once.