EXPLORATION/EXPLOITATION ON BANDITS

Razo van Berkel, Liva van der Velden

1 INTRODUCTION

For this assignment we investigate different methods to solve the multi-armed bandit problem. In this problem the agent chooses which arm of a bandit to pull with the aim to maximize the reward, but the agent initially does not know the average reward for each arm. Therefore it has to find a balance between exploration and exploitation. If it explores too much, it will miss out on high rewards because it will not choose a high reward when it can. If it exploits too much it will miss out on high rewards because it will choose the highest reward found so far, but there could be a higher reward at an arm it has not tried yet.
We consider three methods to balance exploration and exploitation and compare them. These methods are ϵ-greedy, optimistic initialization and upper confidence bounds. For each we compare multiple parameter values to find the best settings.
To compare the three methods we compute the average rewards for multiple settings and decide which algorithm works best based on these results.

2 ϵ-GREEDY BANDIT

2.1 METHODS

A bandit chooses an action based on its policy. The first policy we explore is the ϵ-greedy policy. This policy is:

$$\pi_{\epsilon-greedy}(a) = \begin{cases} 1-\epsilon, & \text{if } a = argmax_{b \in \mathcal{A}} Q(b) \\ \frac{\epsilon}{(|\mathcal{A}|-1)}, & \text{otherwise,} \end{cases}$$

so we choose the action that leads to the average highest reward so far with chance $1-\epsilon$, and choose a random other action with chance ϵ.
We initialize the means $Q(a)$ and counts $n(a)$ for each action to 0, and then update these values every time an action is chosen. $n(a)$ is updated by adding 1, since it counts the number of times action a was chosen. $n(a) \leftarrow n(a) + 1$. $Q(a)$ is updated by altering the average reward of the chosen action by adding the new reward.

$$Q(a) \leftarrow Q(a) + \frac{1}{n(a)}(r(a) - Q(a))$$

2.2 RESULTS

We run the loop of choosing an action and updating the values 1000 times, which shows the learning curve of the agent. To make sure the learning curve is accurate we take the average over 500 runs of this experiment. We compare the learning curves with different values of ϵ. The resulting plot is shown in figure 1.

2.3 DISCUSSION

In figure 1 we see that on average the ϵ-greedy policy converges quickly to a high reward if $\epsilon = 0.05$ or $\epsilon = 0.1$. With a low value for ϵ the greedy option will be chosen more often, so it will take longer to explore the whole search space and find the action that leads to the highest reward. The learning curve for $\epsilon = 0.01$ converges a lot slower, but eventually it reaches approximately the same reward. $\epsilon = 0.25$ appears to be too high, it converges to a lower reward. In this case it will explore too much and exploit too little. Even after the action that gives the highest reward is found, it still will choose a sub optimal action in 25% of the cases.

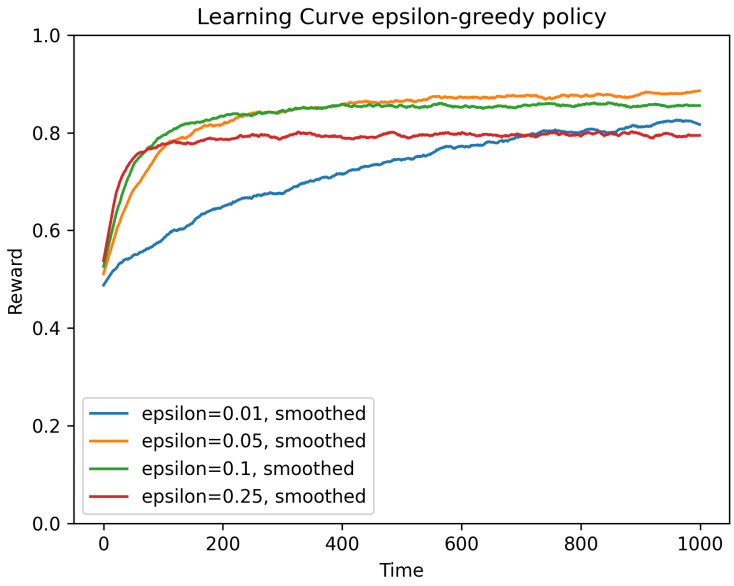

Figure 1: The average learning curve of a bandit using an ϵ-greedy policy, varying the values of ϵ. The average is taken of 500 repetitions and the learning curves are smoothed with `smoothing_window=31`.

3 Optimistic Initialization Bandit

3.1 Methods

The optimistic initialization policy for our bandit is similar to the ϵ-greedy policy, but differs in the following way. Instead of choosing the seemingly most optimal action with a chance of $1 - \epsilon$, it chooses this action by default. This means our bandit *always* chooses the most optimal action. This is the action with that leads to the average highest reward currently known. This leads us to the following formula for the policy of this bandit.

$$\pi_{\epsilon-greedy}(a) = \begin{cases} 1, & \text{if } a = argmax_{b \in \mathcal{A}} Q(b) \\ 0, & \text{otherwise} \end{cases}$$

What also sets this new policy apart from the ϵ-greedy policy is the new update rule. Here α is the *learning rate*, but it is fixed to $\alpha = 0.1$ in these experiments.

$$Q(a) \leftarrow Q(a) + \alpha[r - Q(a)]$$

3.2 Results

Figure 2 is a plot containing four learning curves. Each curve is the average of one of four experiments, each ran 500 times with a thousand timesteps. Each experiment uses a different one of four different values for the `initial_value` variable. The tested values are $[0.1, 0.5, 1.0, 2.0]$.

3.3 Discussion

It immediately becomes apparent that the lowest initial value, of 0.1, returns the worst learning curve. The reward remains around 0.7, instead of converging towards $reward \approx 0.9$, like the other three initial values. This suggests that the this value is simply too low for optimal results. Ironically, we see a slower rise in reward as the initial values climb, with the $initial\_value = 2.0$ showing

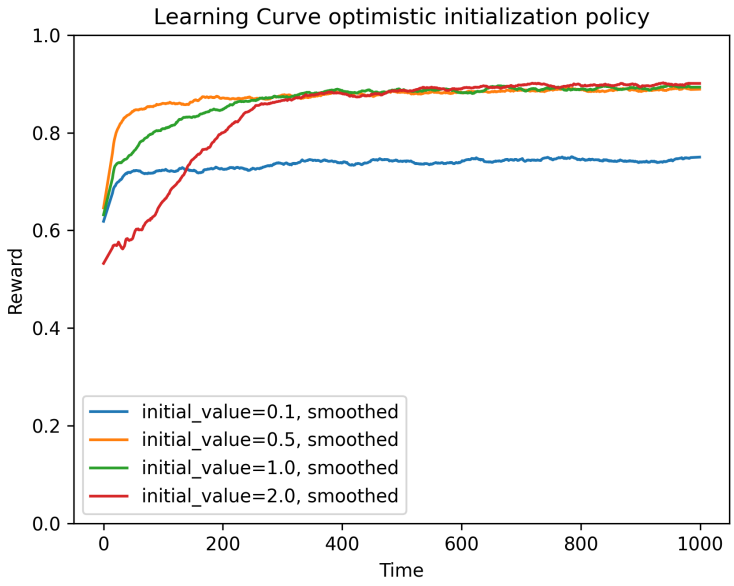

Figure 2: The average learning curve of a bandit using an *optimistic initialization* policy. The average is taken of 500 repetitions and the learning curves are smoothed with `smoothing_window=31`.

the slowest reward climb. Ironically, it also has the lowest starting $reward \approx 0.5$, compared to the other three initial values of having a starting $reward \approx 0.62$.

4 Upper Confidence Bounds bandit

4.1 Methods

The UCB policy always chooses the action with the highest upper bound of its confidence interval. This is formulates in an equation like this:

$$\pi_{UCB}(a) = \begin{cases} 1, & \text{if } a = argmax_{b \in \mathcal{A}}(Q(b) + c \cdot \sqrt{\frac{\ln(t)}{n(b)}}) \\ 0, & \text{otherwise,} \end{cases}$$

We initialize the means $Q(a)$ and counts $n(a)$ for each action to 0, and then update these values every time an action is chosen. $n(a)$ and $Q(a)$ are updated the same way as for the ϵ-greedy method.

4.2 Results

We plotted four learning curves for different values of the parameter c. For each of those values we ran the experiment of a 1000 time steps 500 times. The learning curve in the plot is the average of the 500 runs. The plot is shown in figure 3.

4.3 Discussion

Figure 3 shows that the learning curve converges to the same value, $reward \approx 0.82$, for all values of c. However, for c=0.5 and c=1.0 the curve converges a lot slower than for the other values of c. If c decreases, the choice for an action will depend more on the average reward and less on the confidence interval.

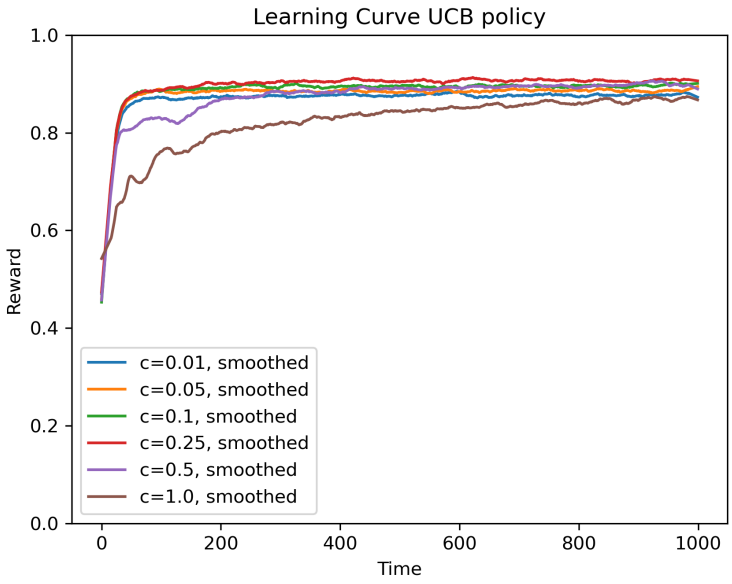

Figure 3: The average learning curve of a bandit using an UCB policy, varying the values of c. The average is taken of 500 repetitions and the learning curves are smoothed with `smoothing_window=31`.

5 Comparison

5.1 Methods

In the previous paragraphs we have tested three different policies for bandits. These policies are an ϵ-greedy policy, an optimistic initialization policy and an upper confidence bounds policy. Now, we want to compare these three approaches. To properly compare them, we will compare the average reward over all runs for all values we tested them on. For example, this means testing all epsilon values in the ϵ-greedy policy. The tested values are:

- `epsilon: [0.01,0.05,0.1,0.25]`
- `initial_value: [0.1,0.5,1.0,2.0]`
- `c: [0.01,0.05,0.1,0.25,0.5,1.0]`

The average rewards are calculated using this formula:

$$\bar{r} = \frac{1}{(N \cdot T)} \sum_{n=1}^{N} \sum_{t=1}^{T} r_{t,n}$$

Secondly, we want to take a closer look at the reward values over time, for each of the best of the policy's settings. From figure 4, we derive the following seemingly optimal values: `epsilon=0.05`, `initial_value=0.5` and `c=0.25`. These values are used for creating *Learning Curves* as we have seen before. These curves, for each of the optimal values, are put into figure 5.

5.2 Results

In figure 4 we see the average reward plotted against the parameter values. The three lines all indicate one of the three bandit policies. Figure 5 shows three learning curves, generated similar to the plots seen earlier in the report. But now each line corresponds to another method, with its optimal value.

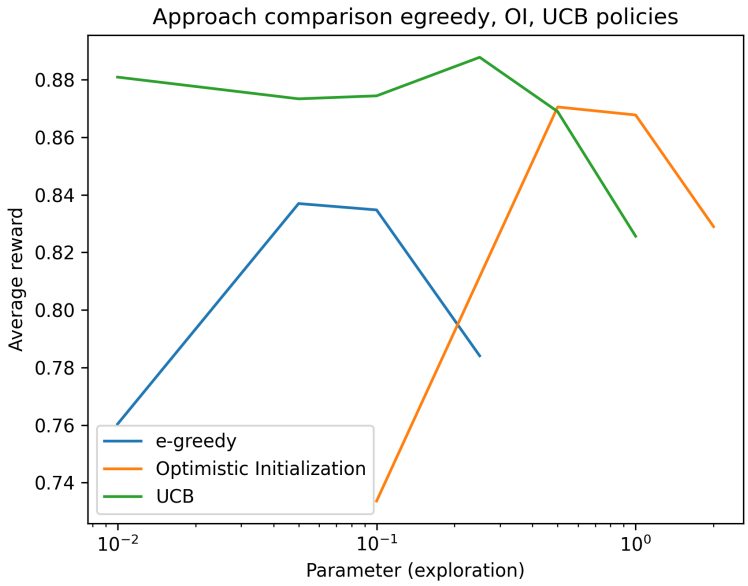

Figure 4: Comparing different approaches with different parameter settings. Compared policies: ϵ-greedy, Optimistic Initialization and Upper Confidence Bounds.

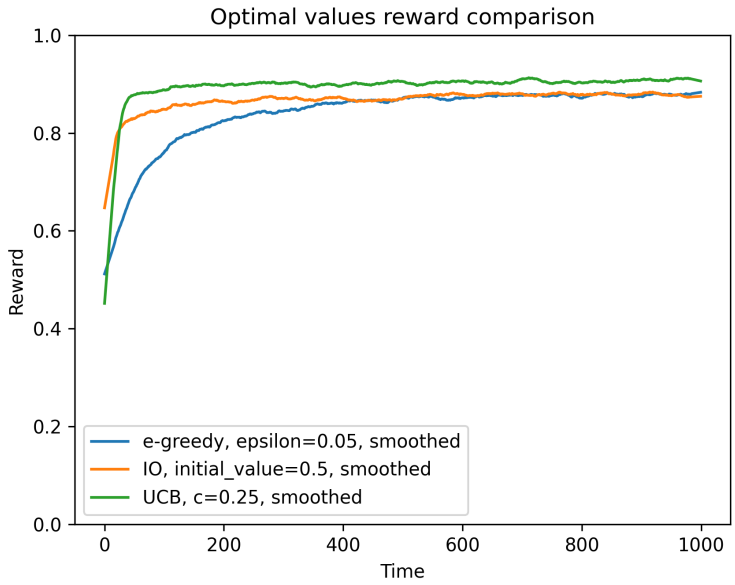

Figure 5: Learning curves for three policies with their optimal values. Optimal values are: `epsilon=0.05`, `initial_value=0.5` and `c=0.25`.

5.3 Discussion

Something we immediately notice in figure 4 is that the average reward is the highest throughout the setting range for the upper confidence bounds approach. This suggests that the UCB approach is superior to the other two approaches, for the values tested. The ϵ-greedy approach comes in last, with the lowest average rewards. We can also deduce that the better values are the values in the

middle of the value range as shown in section 5.1. The lowest and highest values perform worse than the median(s). Figure 5 shows that even for their best value, the observed performance rank still holds. The upper confidence bounds approach comes in first, and ϵ-greedy last. It is interesting to see ϵ-greedy and optimistic initialisation converge after $Time \approx 500$, and we also see that the UCB approach starts at the lowest reward, at $Time = 0$. These two approaches could definitely benefit from more fine tuning of the epsilon and initial values. Figure 4 shows a clear plateau in the middle, instead of a pointy top, as we see with the UCB approach. This suggests that we could increase the average reward by optimizing the epsilon and initial values further. There should be more optimal values in the ranges $(0.05, 0.1)$ for epsilon and $(0.5, 1.0)$ for initial_value.