Assignment 1A: Exploration vs. Exploitation in Bandit Problems

Adrien Im (s3984389) & Bence Válint (s3796426)
Group Number: 4

1 INTRODUCTION

Reinforcement Learning is a fundamental part of machine learning. It consists of agents learning through interactions with their environment. One of the key difficulties of reinforcement learning is the question of exploration and exploitation. As the agent learns about actions and their rewards through exploring its environment, it is important to determine at which point the agent should stop exploration, and start exploiting actions. One framework for this exploration vs. exploitation trade-off problem is the multi-armed bandit problem, which allows us to study this in more detail, as it is one of the simplest models of problems. In the context of reinforcement learning, the term bandit refers to a problems where the agent has to take a series of actions to maximize its reward in an unknown environment. Efficient exploration is important, to maximize the actions that yield the highest reward. [1].

In this report, we will analyze different exploration strategies such as ϵ-greedy, Optimistic Initialization, and Upper Confidence Bound in the context of multi-armed bandit problems. We will explore their differences and compare their performances in finding the highest reward optimally.

2 EPSILON-GREEDY EXPLORATION

2.1 Methodology

One of the main challenges of reinforcement learning is balancing exploration and exploitation. The ϵ-greedy policy addresses this challenge by combining the naive agent and random agents to form a policy that balances exploration and exploitation. The ϵ-greedy agent allows us to control the exploration and exploitation of the agent. ϵ determines the probability of the agent choosing a random action, and $1 - \epsilon$ represents the probability that the agent selects the highest reward action that is known by the agent.

The ϵ-greedy policy can be formalized as follows:

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

where: ϵ is the exploration probability, $\mathcal{A}$ is the set of all possible actions, $|\mathcal{A}|$ is the number of available actions, and $Q(b)$ is the estimated value of action b.

This means that the agent chooses the action with the highest reward with a probability of $1 - \epsilon$ and other actions are randomly chosen with a probability of ϵ. The best-known action is selected most of the time, and there is still a small chance to select another action for exploration.

In our implementation of the ϵ-greedy agent, the estimated reward for each action is stored in an array Q that is initialized to 0. After an action is selected, the agent updates the reward estimate for that action using the following formula:

$$Q_{t+1} = Q_t + \frac{1}{N_t}(R_t - Q_t) \tag{2}$$

where: Q_t is the current estimate of the action's value, R_t is the reward received after selecting the action at time t, N_t is the number of times the action has been selected.

This is the update rule that we use which allows the agent to calculate the average of the rewards for each action without storing the history of past rewards. This rule allows us to put more weight on rewards that were received early in the learning process compared to those earned later.

In summary, the ϵ-greedy policy takes care of the exploration-exploitation trade-off by using probability to select either random actions (exploration) or actions with the highest known reward (exploitation). This, combined with the update rule that allows the agent to be more accurate over time allows the ϵ-greedy policy to manage this trade-off issue.

2.2 Results

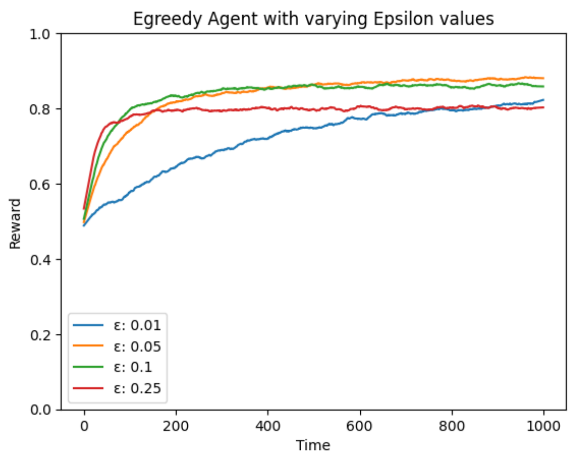

Figure 1: Average reward for ϵ-greedy across different ϵ values

Figure 1 shows the average rewards over time for ϵ-greedy agent graph with varying values of ϵ. The values ϵ = 0.01, 0.05, 0.1, and 0.25 were used to analyze trade-off between exploration and exploitation. To ensure stable and reliable results, all experiments were averaged over 500 runs.

Figure 1 shows that while higher epsilon values (ϵ = 0.25 for example) get higher reward on the short-term, they plateau earlier and at lower values than lower ϵ values. Lower ϵ values (such as ϵ = 0.01) converge more slowly due to less exploration but achieve higher long-term rewards by focusing on exploitation.

This behavior illustrates that there exists a trade-off in ϵ-greedy exploration. A higher epsilon increases exploration, and over time, too much exploration is harmful because the agent keeps selecting suboptimal actions, even though it knows the optimal action. Also, too little exploration means that the agent does not discover the best action, and too quickly settles for a suboptimal one.

Figure 1 illustrates this trade-off, showing that higher ϵ values lead to longer exploration, delaying the convergence due to time spent on suboptimal actions.

From the results of the experiment, $\epsilon = 0.1$ can be recommended, as it is a middle value that balances exploration and exploitation. When ϵ is 0.1, it quickly reaches maximal reward value, and maintains one of the highest rewards in the long-term. We can say that it is therefore one optimal value for ϵ in this setting.

3 OPTIMISTIC INITIALIZATION (OI)

3.1 Methodology

In addition to ϵ-greedy, there are other policies that allow us to address the challenge of balancing exploration and exploitation. We will be looking at two of these methods: Optimistic Initialization (OI) and Upper Confidence Bounds (UCB).

OI consists of exploring by initializing the estimated Q -values to a very large value, such as the maximal reward for an action. In other agents such as ϵ-greedy, Q-values are initialized to zero. However in OI, we initialize the Q-values to a high value such as 1.0 so that the agent expects that all actions will give a high reward.

The agent then follows a greedy policy, and will always select the action with the highest reward. Since all actions are initialized to a high Q-value, the agent has the tendency to try all the actions. As the agent explores the different actions, the Q-values of each action are updated. This allows the agent to determine which actions give a higher reward without having to randomly explore all actions [1].

Similarly to ϵ-greedy, in OI as well, the Q value can be updated using the update rule as shown in Equation (2). An alternative update rule using a constant learning rate α has been tested.

$$Q_{t+1} = Q_t + \alpha(R_t - Q_t) \tag{3}$$

where α is a small positive constant. The usage of this update rule resulted in a constant learning rate for corrections to the Q-values. For the experiement conducted in this paper the alternative rule that was used for our experiment of OI.

3.2 Results

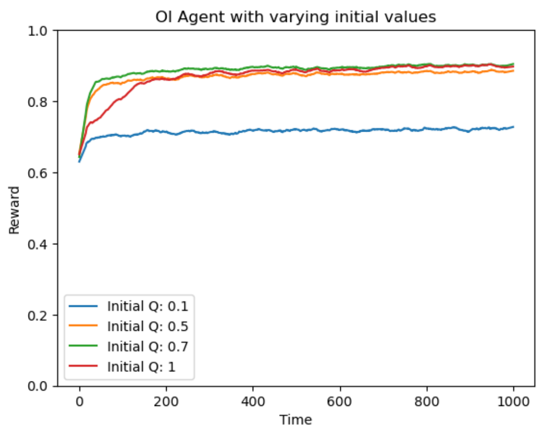

Figure 2: Average reward over time for the Optimistic Initialization agent with varying initial Q-values

Figure 2 shows the average rewards over time for the OI agent using different initial Q-values. We can observe that higher initial values tend to show better results in long-term performance. Among the different initial Q-values used, we can notice that $Q = 0.7$ performed the best, by yielding a higher reward than other values. Values of $Q = 0.5$ and 1 also perform reasonably well, in contrast with $Q = 0.1$.

The results above show that the choice of Q is crucial for OI agents. Higher values of Q, have been tested, but have not shown any significant performance increase compared to $Q = 1$, as the maximum reward is 1. We can conclude that for environments with rewards ranging from 0 to 1, initializing the Q-value to 0.7 provides a good balance between exploration and exploitation.

4 UPPER CONFIDENCE BOUND (UCB)

4.1 Methodology

The third strategy for balancing exploration and exploitation that we will discuss is Upper Confidence Bound (UCB). UCB differs from the ϵ-greedy and Optimistic Initialization in the way it balances exploration and exploitation. UCB keeps track of how many times a certain action has been taken in order to calculate the uncertainty term, to find the potential highest reward for an action.

At every time step t, UCB selects the action a according to the following policy:

$$\pi_{UCB}(a) = \begin{cases} 1, & \text{if } a = \arg\max_{b \in \mathcal{A}} \left[Q(b) + c\sqrt{\frac{\ln t}{N(b)}} \right] \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Importantly, this policy favors exploration of actions that have been selected fewer times in the past since they have a higher potential for the highest reward. We must note that depending on the environment, the UCB method requires to carefully tune the c value. This is because a c value that is too high is susceptible of over-exploration, while a c value that is too low will miss on higher reward actions [1].

4.2 Results

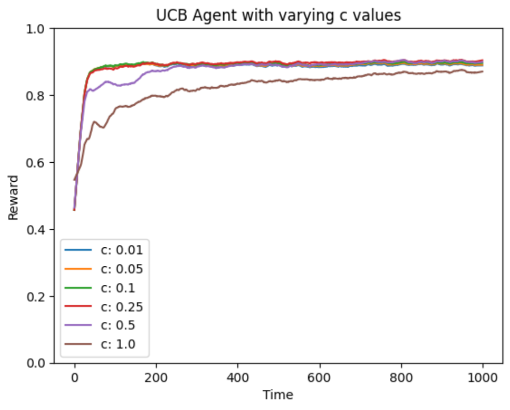

Figure 3: Average reward over time for the UCB agent with varying c values

Figure 3 shows the performance of the UCB agent with different c values. Higher c values clearly perform worse initially, where as smaller values seem to converge more quickly to the maximum reward, such as when $c = 0.01$ and 0.1.

We can explain this difference by looking at how the parameter c (learning rate) in the formula affects the balance between exploration and exploitation: higher c values place greater emphasis on exploring actions with the highest potential for a high reward, which delays convergence, whereas lower c values focus more on exploiting known good actions, leading to faster convergence. For our specific experiment, we can recommend the use of $c = 0.1$.

5 DISCUSSION

After experimenting with three different exploration methods, we can see that they have key differences in the way they balance exploration and exploitation. ϵ-greedy introduces exploration through randomly selecting actions, regardless of how often a specific action has been tried in the past. Optimistic Initialization tries to explore by artificially setting Q-values very high. This can be an effective method, under the important condition that the initial Q-value is correctly selected. Finally, Upper Confidence Bound explores in a smarter way by exploring each action's maximum potential reward.

In order to compare empirically the different methods, we conducted an experiment using each agent's near-optimal parameter settings: $\epsilon = 0.1$ for ϵ-greedy, an initial Q-value of 0.7 for Optimistic Initialization, and $c = 0.1$ for UCB.

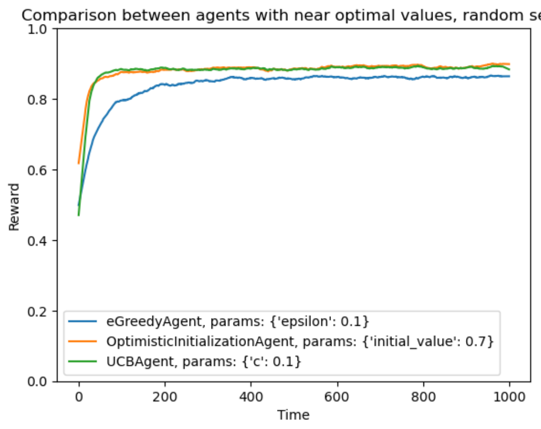

Figure 4: Comparison of the average reward over time using near-optimal parameter values

Figure 4 illustrates the overall performance of each method, highlighting that Optimistic Initialization and UCB achieve faster convergence and higher rewards compared to ϵ-greedy.

From the figure, we can observe that OI and UCB both initially reach a high reward at a faster rate than ϵ-greedy. This is because the exploration rate for the UCB and IO decreases with steps taken, leading to better performance in the short and long term. We can also point out that the careful selection of an optimal parameter is paramount for different agents.

In case the computational power is limited, ϵ-greedy can be an appropriate method for balancing between exploration and exploitation, as it requires the least amount of mathematical calculations. However, the ϵ-greedy agent is outperformed by the UCB and IO agents, given unlimited resources.

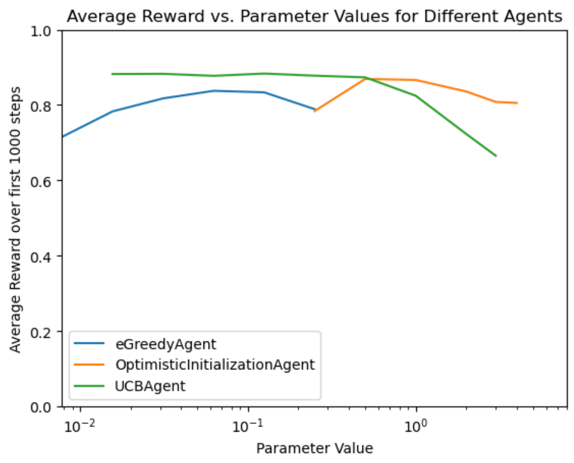

Figure 5: Average reward over the first 1000 steps for each agent across different parameter values

Figure 5 demonstrates the importance of parameter tuning as it shows that each type of agent has a specific parameter at which it performs best in a given environment. It is crucial to use this optimized parameter to achieve the highest results.

An interesting future step that can be taken could be the combination of different strategies seen in this report. Furthermore, changing the parameter over time instead of having it fixed could be an interesting development.

6 CONCLUSION

In conclusion, we have looked at the challenge of balancing exploration and exploitation in reinforcement learning through multi-armed bandit problems.

Through our experiments, we observed that while ϵ-greedy provides a simple and robust solution, it can be less efficient compared to other methods like Optimistic Initialization and UCB in terms of performance. UCB and IO seem to have the same long-term performance with slight differences in the begging of the learning curve. Careful selection of parameters plays a crucial role in maximizing performance in all strategies. Future work could explore hybrid methods or adaptive parameterization to further optimize exploration strategies.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press, 2018.