Reinforcement Learning

Assignment-2 - Multi-Armed Bandit Problem using Epsilon greedy, UCB and Thompson Sampling

Student: Adisu Bezu Submitted to: Natnael Argaw(PhD)

ID: GSR/5572/15

Addis Ababa University

Addis Ababa Institute of Technology

Department of Artificial Intelligence

February 09, 2025

Introduction

Multi-Armed Bandit Problem

In Reinforcement Learning, we use the Multi-Armed Bandit Problem to formalize the notion of decision-making under uncertainty using k-armed bandits. A decision-maker or agent is present in a Multi-Armed Bandit Problem to choose between k-different actions and receives a reward based on the action it chooses. Bandit problem is used to describe fundamental concepts in reinforcement learning, such as rewards, timesteps, and values.

Action-Value and Action-Value Estimate

For an agent to decide which action yields the maximum reward, we must define the value of taking each action. We use the concept of probability to define these values using the action-value function.

The value of selecting an action is defined as the expected reward received when taking that action from a set of all possible actions. Since the value of selecting an action is not known to the agent, we use the 'sample-average' method to estimate the value of taking an action.

Q_t(a) = (sum of rewards when (a) taken prior to (t)) / (number of times (a) taken prior to (t))

Exploration vs Exploitation:

Greedy Action: When an agent chooses an action that currently has the largest estimated value. The agent exploits its current knowledge by choosing the greedy action.

Non-Greedy Action: When the agent does not choose the largest estimated value and sacrifices immediate reward hoping to gain more information about the other actions.

Exploration: It allows the agent to improve its knowledge about each action. Hopefully, leading to a long-term benefit.

Exploitation: It allows the agent to choose the greedy action to try to get the most reward for short-term benefit. A pure greedy action selection can lead to sub-optimal behaviour.

Algorithms

1. **Epsilon-Greedy Action Selection**

Epsilon-Greedy is a simple method to balance exploration and exploitation by choosing between exploration and exploitation randomly.

The epsilon-greedy, where epsilon refers to the probability of choosing to explore, exploits most of the time with a small chance of exploring.

A_t = max Q_t(a). With probability 1 - ε:
Any action(a) With probability ε:

2. **Upper Confidence Bound Action Selection:**

Upper-Confidence Bound action selection uses uncertainty in the action-value estimates for balancing exploration and exploitation. Since there is inherent uncertainty in the accuracy of the action-value estimates when we use a sampled set of rewards, UCB uses uncertainty in the estimates to drive exploration.

A_t = argmax_a (Q_t(a) + c * sqrt(ln(t) / N_t(a)))

1. Exploit Term: Q_t(a)

Represents the expected reward (value) of action ‘a’ at time ‘t’.

2. Explore Term: c * sqrt(ln(t) / N_t(a))

Encourages exploration:

- t: Total timesteps.
- N_t(a): Number of times action a has been taken.
- c: Constant controlling exploration.

We select the action that has the highest estimated action-value plus the upper-confidence bound exploration term.

3. Thompson Sampling for Bernoulli Bandits

In the Bernoulli bandit problem, there are K arms (actions), each with an unknown success probability p_i . At each time step t , the algorithm selects an arm a_t and observes a reward r_t , which is either 1 (success) or 0 (failure). The objective is to maximize the cumulative reward over T time steps.

Thompson Sampling uses a Bayesian approach to model the uncertainty in the success probabilities of the arms. Each arm's success probability is modeled as a Beta distribution, which is updated based on observed rewards. The algorithm selects arms by sampling from their posterior distributions and choosing the arm with the highest sampled value.

Result Analysis

1.1. Analysis of Epsilon-Greedy Algorithm Performance

The epsilon-greedy algorithm balances exploration and exploitation by selecting the best-known option most of the time while occasionally exploring other options. The results obtained from experimenting with different epsilon values (0.1, 0.2, and 0.5) reveal key insights into how this balance affects total rewards and arm selection distribution.

With $\varepsilon = 0.1$, the algorithm primarily exploits the best arm (908 selections for arm 3), leading to the highest total reward of 1912.76. The limited exploration (10% of the time) allows the algorithm to

identify and exploit the optimal arm effectively, demonstrating that a lower epsilon can be beneficial when the optimal arm is significantly better than others.
At ε = 0.2, exploration increases to 20%, leading to a more distributed selection across multiple arms (306 selections for arm 2 and 565 for arm 3). Consequently, the total reward drops to 1722.00, indicating that increased exploration prevents the algorithm from fully capitalizing on the best arm.

For ε = 0.5, where exploration and exploitation occur equally, arm selections are more evenly distributed across all arms (116, 117, 583, 86, and 98). The total reward further decreases to 1620.75, as the algorithm spends too much time exploring suboptimal arms instead of exploiting the best one.

These results indicate a trade-off between exploration and exploitation : a lower epsilon (0.1) yields better rewards in the short run by emphasizing exploitation, whereas a higher epsilon (0.5) ensures more exploration but at the cost of immediate rewards. The choice of epsilon should depend on the problem context—lower epsilon for stable environments and higher epsilon for dynamically changing reward distributions.

1.2. Analysis of the Upper Confidence Bound (UCB) algorithm Performance

The experiment evaluates the performance of the Upper Confidence Bound (UCB) algorithm by varying the exploration parameter c and analyzing its impact on the total reward and the arm selection distribution. Three values of c (1, 2, and 5) were tested.

When c = 1, the algorithm achieved the highest total reward of approximately 1950.45. The arm selection distribution was highly skewed, with arm 3 being selected 988 times, indicating heavy exploitation of the arm with the highest estimated value. Other arms were rarely explored, as seen from their low selection counts. This demonstrates that a lower cc prioritizes exploitation over exploration.

As c increased to 2, the total reward decreased slightly to approximately 1895.73. The arm selection distribution became more balanced, with arm 3 being selected 832 times, and other arms (especially arms 1 and 5) receiving more selections compared to when c = 1. This indicates that increasing c leads to more exploration, which can reduce short-term rewards but may potentially improve long-term decision-making.

For c = 5, the total reward further dropped to approximately 1723.75. The arm selection distribution was even more balanced, with arm 3 selected only 594 times and the remaining arms receiving significantly more selections compared to lower values of c. This suggests that a high exploration parameter shifts the algorithm's focus toward exploration, potentially at the cost of short-term rewards.

Overall, the results highlight the trade-off between exploration and exploitation in the UCB algorithm. Lower values of c result in higher short-term rewards by favoring exploitation, while higher values of c encourage exploration, which may help in discovering better actions but at the expense of immediate gains. In this experiment, $c = 1$ performed the best in terms of maximizing the total reward. However, the optimal choice of c depends on the specific application and the need to balance exploration and exploitation effectively.

1.3. Comparative Analysis of UCB and Epsilon-Greedy Strategies

This experiment compared the performance of the UCB algorithm and the epsilon-greedy strategy with varying exploration parameters. UCB consistently achieved higher total rewards than epsilon-greedy, demonstrating more efficient exploration and exploitation.

For UCB, the total reward decreased as the exploration parameter c increased, with $c = 1$ yielding the highest reward (1950.45). UCB's adaptive exploration mechanism allowed it to heavily exploit the best-performing arm while still exploring other arms selectively. In contrast, epsilon-greedy's performance declined as ϵ\epsilon increased, with rewards dropping from 1912.76 ($\epsilon=0.1$) to 1620.75 ($\epsilon=0.5$). Its fixed-probability exploration led to higher randomness, sacrificing immediate rewards.

At comparable exploration levels, UCB consistently outperformed epsilon-greedy. For example, at higher exploration ($c=5$ and $\epsilon=0.5$), UCB achieved a higher reward (1723.75 vs. 1620.75) while maintaining more focused arm selection. This highlights UCB's advantage in balancing exploration and exploitation effectively, making it the more efficient strategy in this experiment.

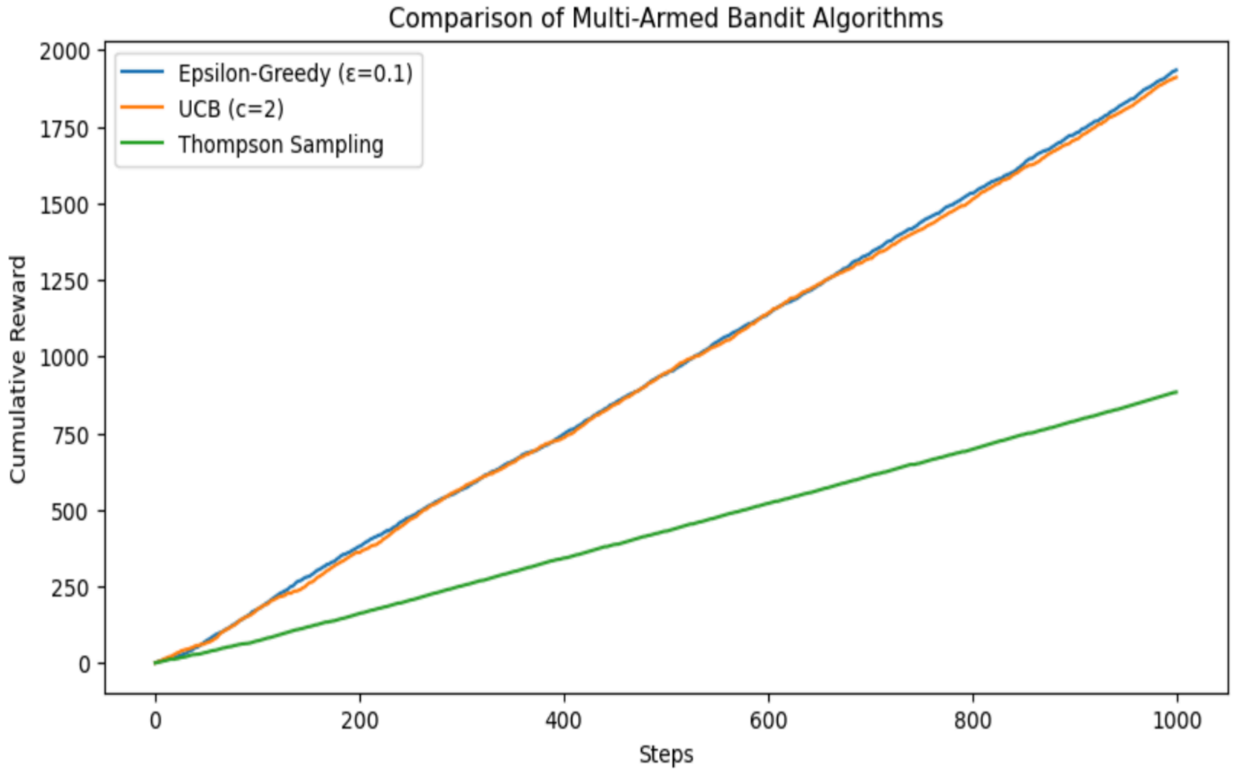
1.4. Analysis of the Thompson Sampling for Bernoulli Bandits algorithm Performance

The results indicate that Thompson Sampling effectively identified the best-performing arm. The selection counts show that one arm was chosen 966 times, while the others were chosen significantly fewer times (6, 13, 9, and 6 times, respectively). This suggests that the algorithm quickly converged to the optimal arm, demonstrating strong exploitation behavior. The total reward of 872 out of 1000 time steps indicates that the selected arm had a high success probability, validating the effectiveness of Thompson Sampling in maximizing rewards while minimizing unnecessary exploration.

1.5 Comparison of Three Algorithms

The analysis highlights key trade-offs in exploration efficiency and convergence speed among the algorithms. Epsilon-greedy relies heavily on the exploration parameter e, with lower values favoring exploitation and higher rewards.UCB adaptively balances exploration and exploitation, performing best with lower c values. Thompson Sampling excels, quickly converging to the optimal arm (966/1000 selections) and achieving high rewards (872/1000) with minimal unnecessary exploration, making it the most efficient and effective algorithm for Bernoulli bandits.

The plot compares the cumulative rewards of three multi-armed bandit algorithms: Epsilon-Greedy (ε=0.1), Upper Confidence Bound (UCB, c=2), and Thompson Sampling. Epsilon-Greedy and UCB exhibit similar performance, achieving the highest cumulative rewards, with Epsilon-Greedy slightly outperforming UCB in the long run. Thompson Sampling, however, lags behind, accumulating significantly fewer rewards over time. This suggests that in this particular setup, Epsilon-Greedy and UCB are more effective at identifying and exploiting the optimal arms compared to Thompson Sampling.


Comparison of Multi-Armed Bandit Algorithms
Epsilon-Greedy (ε=0.1)
UCB (c=2)
Thompson Sampling
Cumulative Reward
Steps