FRA503 Deep Reinforcement Learning for Robotics

Homework 1: Multi-Armed Bandit

Name	Student ID
Chantouch Orungrote	6340500011
Sasish Kaewsing	6340500076

1 Abstract

Textbook Perspective. An Introduction by Sutton and Barto [1], analyze multi-armed bandits using continuous rewards to illustrate the exploration–exploitation trade-off. Because algorithm update mechanisms rely on expected rewards, they remain robust across various reward distributions.

This homework adapts that framework to a **binary reward** (success–failure) setting. This allows for the evaluation of bandit algorithms under a Bernoulli outcome model while maintaining the core objective of maximizing cumulative return.

2 Multi-Armed Bandit Problem

The Multi-Armed Bandit Problem is a single state problem formulation with sequential structure and a finite action set $A = \{1, \ldots, K\}$. At each time step t, an agent selects an action $A_t \in A$ and receives a stochastic reward R_t with an unknown reward distribution for the selected action $p(r|a)$. The objective is to maximize return over the lifetime T.

2.1 Mathematical Formulation

Concept	Form	Definition
Stochastic Reward	$R_t \in \{0, 1\}$	The reward is observed after selecting A_t
Action Value	$q^*(a) = \mathbb{E}[R_t \mid A_t = a]$	The true value of action by the expected reward
Optimal Action	$a^* = \arg\max_{a \in A} q^*(a)$	The action with the highest expected reward
Optimal Value	$q^*(a^*) = \max_{a \in A} q^*(a)$	The maximum achievable expected reward per timestep
Value Estimation	$Q_{t+1}(a) = Q_t(a) + \frac{1}{N_t(a)}\big(R_t - Q_t(a)\big)$	Sample-average update rule used to estimate action values, where $N_t(a)$ is the number of times the action a has been selected.
Regret	$L(T) = \sum_{t=1}^{T}\big(q^*(a^*) - q^*(a)\big)$	Measures the total performance loss compared to always selecting the optimal action.

Table 1: Mathematical formulation of the multi-armed bandit problem.

2.2 Understanding the Algorithms

The Epsilon-Greedy alternates between exploration and exploitation by randomizing action selection. Which exploits the current value estimates $Q_t(a)$ with probability $1-\epsilon$, explores by selecting a random action with ϵ.

$$A_t = \begin{cases} \text{a random action}, & \text{with probability } \epsilon, \\ \arg\max_a Q_t(a), & \text{with probability } 1-\epsilon, \end{cases} \tag{1}$$

The Upper Confidence Bound selects actions by augmenting the value estimate with an exploration bonus. Select actions by combining $Q_t(a)$ with an uncertainty bonus term $c\sqrt{\frac{\log t}{N_t(a)}}$, which is larger for less-sampled actions $N_t(a)$ and decreases over time, resulting in directed exploration.

$$A_t = \arg\max_a \left(Q_t(a) + c\sqrt{\frac{\log t}{N_t(a)}} \right). \tag{2}$$

3 Experiment Design

3.1 Bernoulli Reward Setting

A Bernoulli reward model is used when each action yields a binary outcome, representing success or failure, By fixing (p_a) the multi-armed bandit, where each arm produces a reward $\{0, 1\}$ with an arm-specific success probability. The reward distribution of each arm is fixed throughout all experiments and does not change across runs.

This configuration includes near-optimal arms (e.g., $p = 0.75$ versus $p^* = 0.80$), making effective exploration necessary to reliably identify the optimal action.

Arm (a)	1	2	3	**4**	5	6	7	8	9	10
Success probability (p_a)	0.10	0.50	0.60	**0.80**	0.10	0.25	0.60	0.45	0.75	0.65

Table 2: Fixed Bernoulli reward probabilities for each arm. The optimal arm is $a^* = 4$ with $p^* = 0.80$.

Note: In Bernoulli setting, each arm's action value is the probability $(q^*(a) = p_a)$

3.2 Configuration

Paramater	Value
Number of arms (k)	10
Timesteps (T)	500
Independent runs (n_{runs})	10,000
Reward type	Bernoulli(p_a)
ϵ values	$\{0, 0.01, 0.05, 0.1, 0.5\}$
c values	$\{0.5, 1, 2, 3, 5\}$
Seed	42

Table 3: Experiment parameter setup.

3.3 Evaluation Metrics

The performance is evaluated using the following metrics, averaged over 10,000 independent runs.

Metric	Description
Average Reward ($\bar{r}_t$)	The mean reward obtained at timestep t, averaged across runs. This metric reflects
Optimal Action %	The proportion of runs in which the optimal arm a^* is selected at timestep t
Cumulative Regret	The accumulated difference between the optimal expected reward and the reward obtained by the selected actions up to time t
Convergence Step	The timestep at which the agent shows the optimal reward performance.

Table 4: Evaluation metrics.

4 Analysis

Following the methodology in Section 3., this section presents a comparative analysis of the performance of the ϵ-greedy and UCB agents. Based on graphical representations that illustrate the reward, optimal action selection, and regret over each timestep.

4.1 Epsilon-Greedy Performance

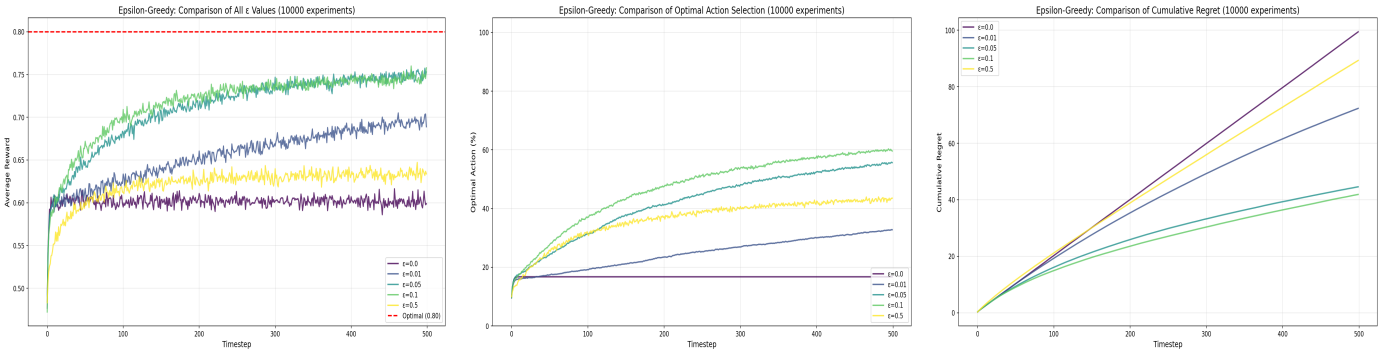

Figure 1: Epsilon-greedy reward, optimal action, and regret graphs.

Pure Exploitation Problem, in reinforcement learning, a **local maxima** occurs when an agent settles on a suboptimal action. From the reward graph, we can see that with $\epsilon = 0.0$ (no exploration), the agent remains stuck at this local maximum of 0.60, never reaching the optimal reward of 0.80.

The Optimality Limit is a limit on the maximum average reward an agent can achieve. Even if the agent knows exactly which arm is the best, the ϵ-greedy algorithm forces it to explore random arms ϵ percent of the time. According to the linear regret that increasing cumulative regret over time.

To find $\mathbb{E}[R_t]$ for an ϵ-greedy agent after it has finished learning, we must look at the root questions for each part of the decision.

- **Exploitation Reward:** The probability of picking the optimal arm $(1-\epsilon)$ times its action value $(q^*(a))$.
- **Random Luck:** The probability of picking the optimal arm during a random exploration step $(\frac{\epsilon}{k})$.
- **Exploration Loss:** The probability of picking a other arms $(\epsilon - \frac{\epsilon}{k})$ times the average success rate of bad arms $(\bar{q}_{sub})$.

$$\mathbb{E}[R_t] = \underbrace{(1 - \epsilon + \frac{\epsilon}{k}) \cdot q^*(a)}_{\text{Selecting the Best Arm}} + \underbrace{(\epsilon - \frac{\epsilon}{k}) \cdot \bar{q}_{sub}}_{\text{Selecting a Bad Arm}} \tag{3}$$

Lets $k = 10, \epsilon = 0.1, q^*(a) = 0.80$. We find $\bar{q}_{sub}$

$$\bar{q}_{sub} = \frac{0.10 + 0.50 + 0.60 + 0.10 + 0.25 + 0.60 + 0.45 + 0.75 + 0.65}{9} = \frac{4.0}{9} \approx \mathbf{0.444} \tag{4}$$

Now, calculate the final expected reward

$$\mathbb{E}[R_t] = (0.9 + 0.01) \cdot 0.80 + (0.1 - 0.01) \cdot 0.444 = \mathbf{0.767} \tag{5}$$

This shows explains why our $\epsilon = 0.1$ graph levels off at approximately 0.77, failing to reach the 0.80 optimal line.

The Stationary Problem. Bernoulli was setup as an experiment, which makes the fixed rewards and probabilities, means the ϵ value uniformly random the exploration through the entire episode. However, we can solve this problem by create a **decay function for ϵ.** In the early stages, a high ϵ value encourages broad sampling of all available actions to build initial knowledge about the environment's reward structure. However, as the agent gains experience and its value estimates $Q_t(a)$ become more accurate, the ϵ value is gradually reduced.

Linear Regret. From the graph, it is evident that the different agent types generate distinct errors, particularly in the group with a fixed ϵ. This results in linear regret, **as there is no reduction in exploration even after the best arm is identified.** Confirms that neither the greedy strategy nor the traditional ϵ-greedy strategy can achieve logarithmic regret in this experiment.

ϵ **value**	**Convergence**	**Interpretation**
0.0	No Convergence	The agent selected the optimal arm that it believes
0.01	Slow	Less explored, take long time to identify the optimal arm, or it might get trapped
$0.05, 0.1$	Fast	Balance the exploring and effectively identify the best approaches
0.5	Very Fast	Fast explores and identify true values, but less exploit from fixed $\epsilon = 0.5$
Decaying ϵ	Best	Early exploration and more exploration in the left steps.

Table 5: Comparison of convergence's behavior by ϵ-greedy

4.2 Upper Confidence Bound Performance

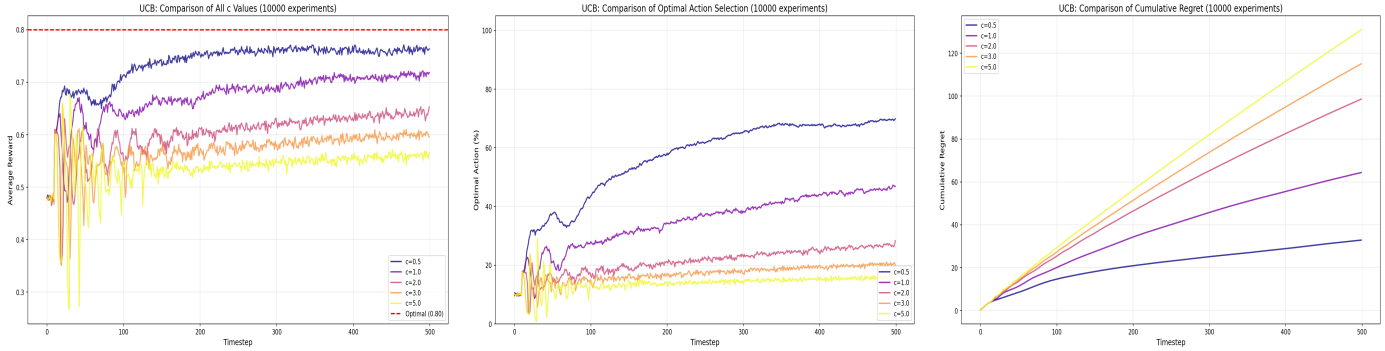

Figure 2: UCB reward, optimal action, and regret graphs.

Directed Exploration is a strategy that UCB uses to avoids local maxima. Instead of picking a random arm, it calculates an uncertainty bonus for each arm. If an arm has not been pulled many times, the bonus becomes dominant, forcing the agent to try it and see if it is better than the current best. This is clearly illustrated in the reward and optimal action graph for the first 100 steps.

Optimal Actions Convergence. The value c also dictates how quickly the agent stops searching and starts exploiting the best-known arm. With a low value such as $c = 0.5$, the agent gives more weight to the actual rewards it has already seen. This allows the optimal action curve to rise quickly, reaching approximately 40% in the first 50 steps.

Non-Convergence Scenario. In theory, a high value c in the UCB may not *prevent* convergence to the optimal point in the long run $t \to \infty$. However, in practice, **a high value of c leads to significant performance degradation**, causing the algorithm to reach optimal rewards more slowly, which may ultimately be inefficient and even become non-converge.

To prove *how a large value of c causes the agent to fail in selecting the optimal arm*, it is necessary to make the assumption which the UCB value of suboptimal arm a_{sub} surpasses that of the optimal arm.

$$Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} > Q_t(a^*) + c\sqrt{\frac{\log t}{N_t(a^*)}} \tag{6}$$

Let $\Delta = Q_t(a^*) - Q_t(a)$ represent the value gap, and assuming the optimal arm has been sampled sufficiently $N_t(a^*) \gg 1$, its uncertainty term becomes negligible.

$$c\sqrt{\frac{\log t}{N_t(a)}} > \Delta \tag{7}$$

Now, square and rearrange it.

$$N_t(a) < \frac{c^2 \log t}{\Delta^2} \tag{8}$$

From this inequality, we able to tanslate it into 2 main causes.

- **Quadratic Parameter:** c is squared (c^2), which increases the $N_t(a)$ exponentially. The agent is mathematically obligated to pull a suboptimal arm a lot of times before the uncertainty bonus finally drops enough to allow the agent to pick the optimal arm.
- **Time Run Out:** The time ends before the agent finishes these exploration pulls. The agent fails because it runs out of time while still trying to prove that a suboptimal arm is actually suboptimal.

Dominant of C. By default the UCB regret is logarithmic, but in this experiment it is linear when $c = 3, 5$. Because the c value dictates the agent uncertainty term. When $c > 3$, the exploration bonus remains dominant even after substantial evidence has been gathered.

This leads to a scenario where the confidence bound for suboptimal arms stay artificially inflated, overcompensating for their low empirical means. Consequently, the agent continues to pull inferior arms at a constant frequency relative to T, preventing the convergence of the policy and causing the regret to grow linearly.

Essentially, the agent becomes overly optimistic, failing to transition from exploration to exploitation within the given time.

5 Conclusion

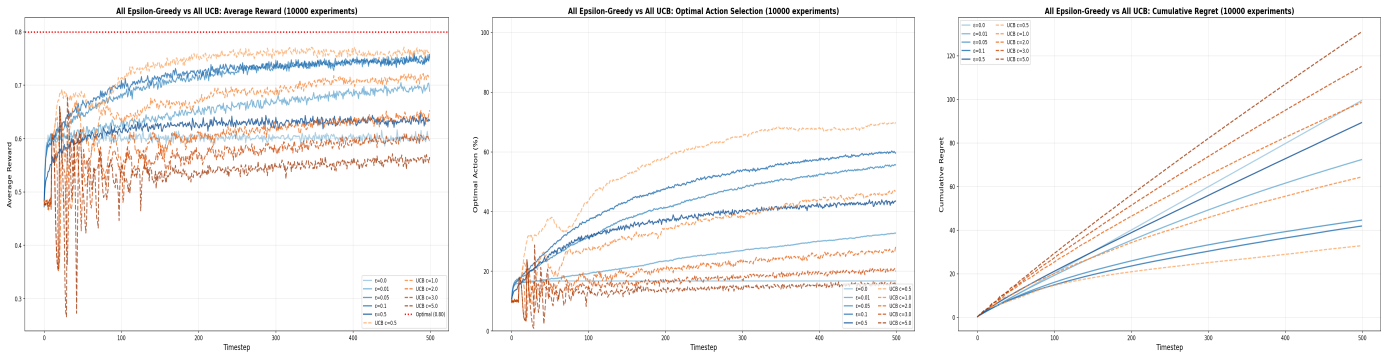

Figure 3: Both algorithm reward, optimal action, and regret graphs.

5.1 Why is UCB more effective than ϵ-greedy in Multi-Armed Bandit problems

Systematic Exploration Mechanism. The ϵ-greedy method explores by randomly selecting actions that are not the best, without any prioritization of which action is likely to be better. In contrast, UCB explores by considering the potential of being the best option, calculated from how close it is to the maximum value along with the uncertainty term.

Handling Uncertainty. UCB uses a calculation formula that combines $Q_t(a)$ with the uncertainty term.

- If an action is selected fewer times, the uncertainty term increases, leading UCB to explore that action further for confidence.
- This mechanism helps UCB reduce the frequency of selecting suboptimal actions over time, while ϵ-greedy with a fixed ϵ continues to waste of the time on randomly selecting suboptimal actions indefinitely.

From Reinforcement Learning: An Introduction by Sutton and Barto [1], UCB tends to achieve higher average rewards than ϵ-greedy. In parameter studies, it was found that when tuned optimally, UCB outperforms other methods in terms of average rewards, as compared to other methods discussed in Chapter 2.

In conclusion, the UCB in terms of **more systmatic exploration**, leading to higher cumulative rewards in typical multi-armed bandit problems, but may lack the flexibility and simplicity that ϵ-greedy offers when dealing with large-scale learning problems.

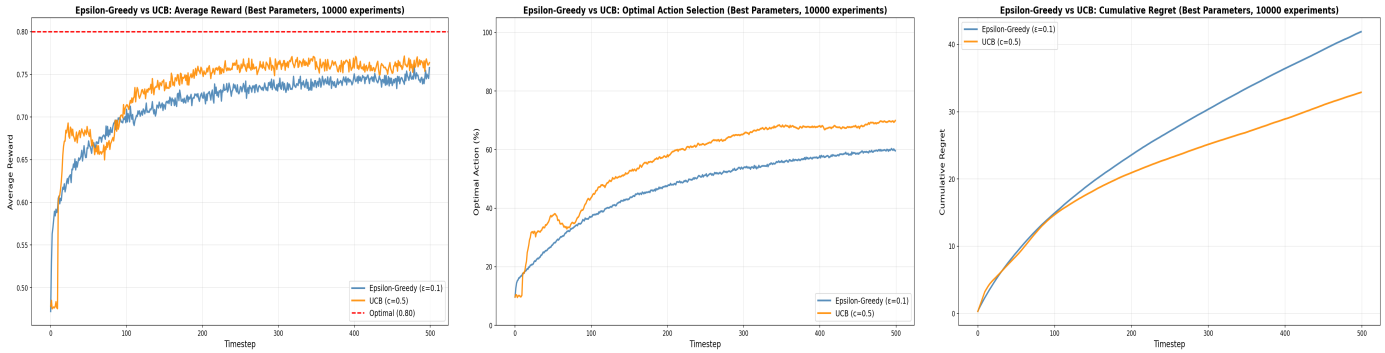

Figure 4: The best reward, optimal action, and regret for each algorithm.

5.2 Exploration–Exploitation Tradeoff

The exploration–exploitation tradeoff is the fundamental challenge in RL, specifically in multi-armed bandit problems.

From this experiment, we gain the knowledge from observing, analyzing, reviewing the recommended RL textbook, and exploring the internet world. That we can conclude the trade-off.

There is no single perfect number. The best parameter depends on the reward distribution and whether the environment changes over time. Based on an RL reference textbook [1], the approaches used for finding the *sweet spot* for the parameters like ϵ or c are:

- **The Inverted-U Shape:** Performance typically follows an inverted-U curve. If the exploration parameter is too low, learning is slow and prone to error. If it is too high, the agent remains too distracted by random choices to ever perform well.
- **Stationary vs. Non-stationary Environments:** In the stationary, rewards do not change over time. Exploration can be gradually reduced (decayed) as the agent becomes more confident. On the other hands, the non-stationary rewards change over time. The agent must maintain a constant level of exploration to detect if a previously bad arm has become the new best arm.

6 Alternative

The textbook has experimented with the **Optimistic Initial Values** Q_0, which are included in both algorithms and also the backbone of the exploration problem.

By setting initial reward estimates higher than any possible actual reward, the agent is naturally disappointed by its first few pulls. This forces it to explore every available option at least once before settling on a favorite.

References

[1] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2 edition, 2018.