CS5180 Ex1

Name: Sudhendra Kambhamettu
NUID: 002786797

Q1.

This question essentially asks us to identify the occurrences which are definitely exploration steps and also identify which steps could be exploratory.

To do this, we can observe the following epsilon -greedy behavior which sates – the algorithm selects the best-known action most of the time (exploitation) but occasionally chooses an action at random (exploration). The steps where exploration "definitely" occurred would be those where the chosen action deviates from the one that would have been selected based on the highest estimated value Q or total average reward. Steps where exploration "could" have occurred would be less definitive and might include situations where the chosen action coincidentally aligns with the best-known action, even though it was chosen randomly.

However, knowing this we can still estimate the definite and possible exploration steps.

Definite occurrence of exploration:

1. Time step 5: $[A_5 = 3, R_5 = 0]$ is definitely an exploration step since up until this point, the action 2 had the highest average reward. Choosing the action 3 despite it not having the highest estimated reward suggests that this step is definitely an exploration step.

Possible occurrence of exploration:

1. Time step 1: $[A_1 = 1, R_1 = -1]$ could be an exploration step since it's the first action.
2. Time step 2: $[A_2 = 2, R_2 = 1]$ could be exploitation assuming the policy dictates the agent to take the next action influenced by the negative reward in step 1. But this step could also be an exploration step if the epsilon value was met with the set value.
3. Time step 3, 4: $[A_3 = 2, R_3 = -2]$ & $[A_4 = 2, R_4 = 2]$ could again be an exploitation step considering that the action 2 has the highest average reward after the time step 2 (A_2). But it could very well be an exploration step if the epsilon condition had met.

Q2.

To derive the estimate for the n-dependent alpha we know that –

$$Q_{n+1} = Q_n + \alpha_n[R_n - Q_n] - \boldsymbol{eq.1}$$

Expanding this equation, we get:

$$Q_{n+1} = Q_n + \alpha_n R_n - \alpha_n Q_n$$

$$= \alpha_n R_n + Q_n(1 - \alpha_n)$$

$$= \alpha_n R_n + (1 - \alpha_n)\big[Q_{n-1} + \alpha_{n-1}[R_{n-1} - Q_{n-1}]\big] \because eq.1$$

Similarly, by expanding this equation we get:

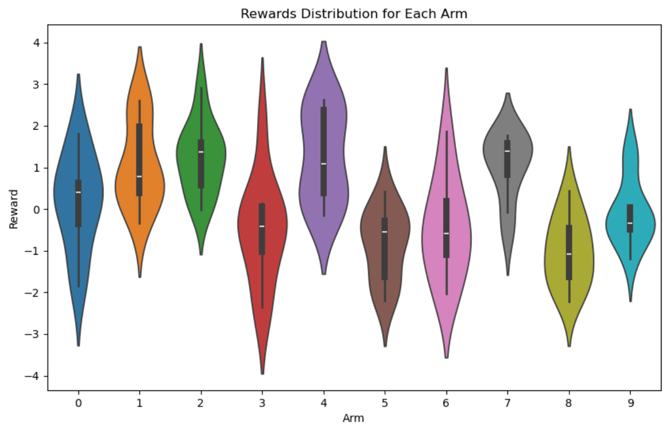
$$= \alpha_n R_n + \alpha_{n-1}R_{n-1} - \alpha_n\alpha_{n-1}R_{n-1} + (1 - \alpha_n)(1 - \alpha_{n-1})Q_{n-1}$$

$$= \vdots$$

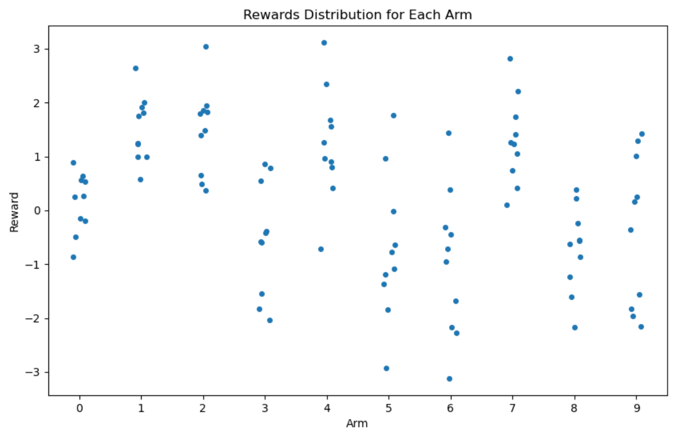
$$\therefore \sum_i^n \alpha_n R_n - R_{n-1}\prod_i^n \alpha_n + Q_1 \prod_i^n (1 - \alpha_n)$$

Q3. **Plot:**

Violin Plot:



Strip Plot:

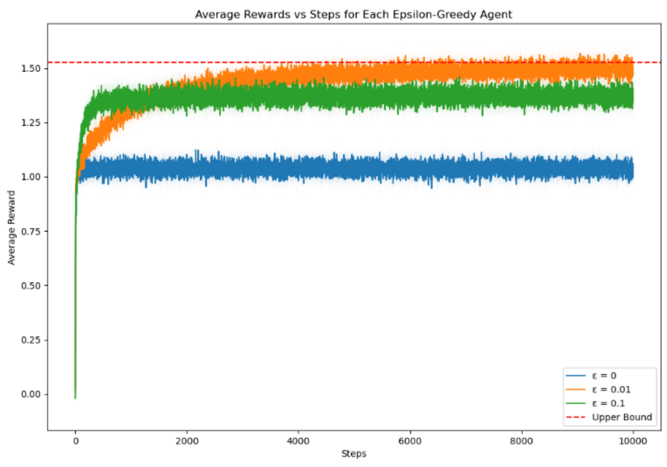


Q4.

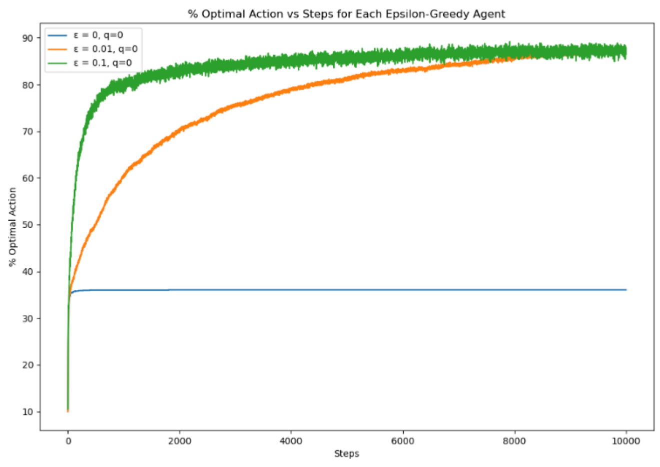
For the "long run", let $t \to \infty$ both non-zero ε-greedy policies will converge to learn the optimal action and the corresponding value function q_*. However, the key difference lies in the frequency of selecting the optimal action. The policy with ε = 0.01 will be more effective in this regard, as it will choose the optimal action significantly more often (99% of the time) compared to the policy with ε = 0.1 (90% of the time). Specifically, the policy with ε = 0.01 is expected to select the optimal action 10 times more frequently than the policy with ε = 0.1.

This increased frequency in selecting the optimal action directly translates to a higher total reward. As both policies eventually converge to $Q_t \to q_*$, the total reward and the probability of choosing the optimal action will be proportionally greater for the policy with ε = 0.01. In essence, the lower value of ε in this policy balances the trade-off between exploration and exploitation more efficiently over an extended period, leading to enhanced performance and reward accumulation.

Q5. **Plot:**
Average Rewards Plot:



% Optimal Actions Plot



Written: Yes, the averages reach the asymptotic levels predicted. As the plots above show, $\epsilon = 0.1$ although picks optimal actins faster, it reaches 90% of the upper-bound value and $\epsilon = 0.01$ although picks optimal actions slowly, reaches 99% of the upper-bound value which are accurate with the asymptotic predictions made in the previous question.

Q6.

A. The sample average estimate – $Q_n = \frac{R_1+R_2+R_3+\cdots+R_{n-1}}{n-1}$ which is given in Equation 2.1 is not biased since the $\mathbb{E}[Q_n] = q_*$. The sample average estimate simply calculates the average of all the past rewards & therefore as the number of samples increase, the estimate converges to the true expected value.
B. For the exponential recency weighted average estimate if $Q_1 = 0$ then the $Q_n\ for\ n > 1$ can be biased since, the initial estimate Q_1 may not be equal to the true estimate. This can cause the weighting to lean towards the recent rewards as the exponential recency weighted average gives more importance to them.
C. The following conditions should be satisfied for the exponential recency weighted average estimate to be unbiased:

a. The initial estimate Q_1 should be equal to the true estimate q_*.

b. The learning rate or the step size α should be adjusted over time (typically decreasing in a way that accounts for the number of samples seen) to give appropriate importance to all the rewards so that Q_n can converge to an unbiased estimate of q_*.

D. Consider the exponential recency-weighted average equation –

$$Q_n = Q_{n-1} + \alpha(R_n - Q_{n-1}) - eq.1$$

Expanding this equation recursively we can write Eq.1 as follows:

$$Q_n = (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i \ \because RLEq.2.6$$

For a $Q_1 = 0$ and a constant α $(0 < \alpha < 1)$ the above equation becomes:

$$Q_n = \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i - eq.2$$

So, for $n \rightarrow \infty$ the weight $(1-\alpha)^{n-i}$ diminishes for rewards with smaller i value (older rewards). Thereby making the effect of any single reward R_i on Q_n becomes smaller as n becomes larger.

Also, assuming the rewards are sampled from a stationary environment with a mean q_* the weighted sum of all the rewards will approach q_* as $n \rightarrow \infty$. Since, the weight parameter approaches a value of 1 as $n \rightarrow \infty$ i.e.,

$$\lim_{n\rightarrow\infty}\left(\sum_{i=1}^{n} \alpha(1-\alpha)^{n-i}\right) = 1$$

Therefore, for $n \rightarrow \infty, Q_n \rightarrow q_*$ i.e., Q_n is asymptotically unbiased for large n values.

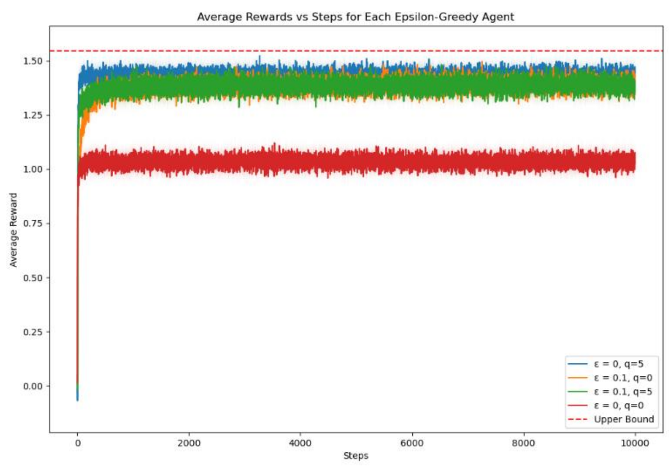
E. This estimate is generally biased because it favors recent rewards more than the older ones where the step size parameter controls the extent of this bias. Generally, the most recent rewards are not reflective of the future expectation and therefore leading to biased estimates.

Q7.

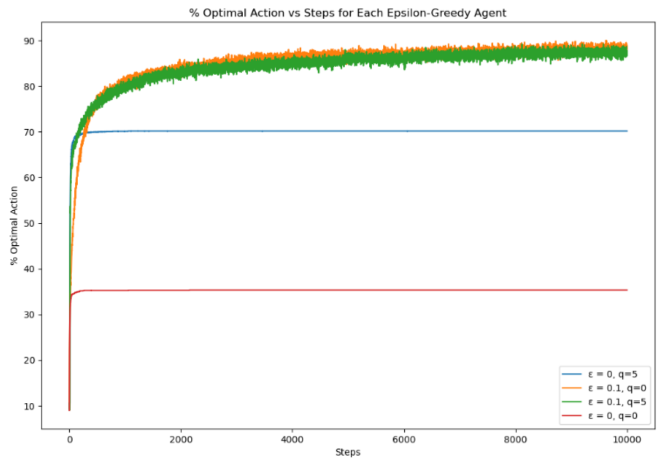
Plots:

Supplemental figures for 2.3 & 2.4 for the all the epsilon & Q_1 combinations as mentioned in the question and displayed in the plots below.

<u>Average Rewards Plot</u>

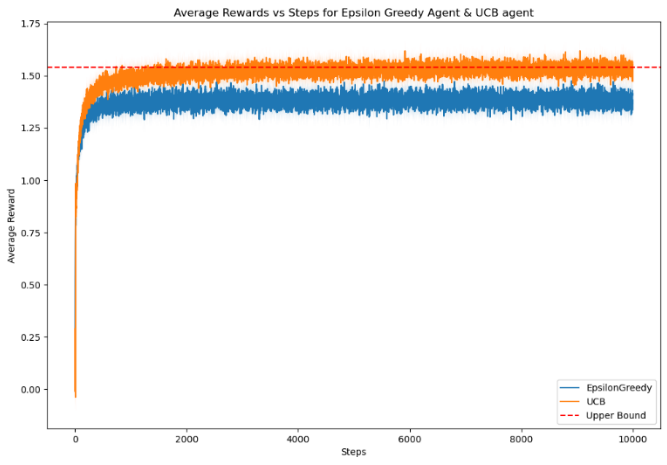


% Optimal Actions Plot

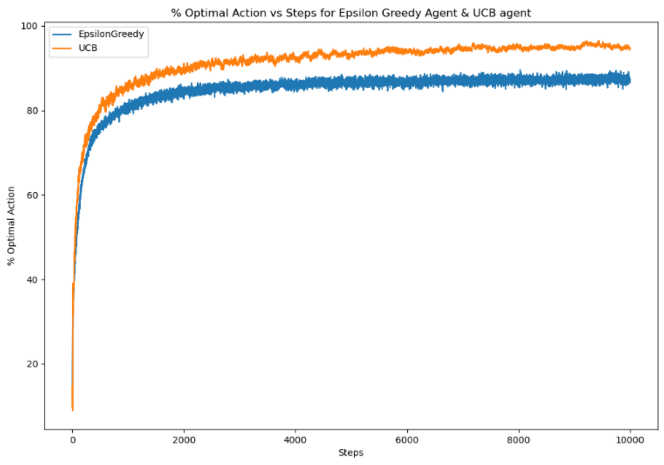


The following are the supplementing figures for 2.3 & 2.4 comparing epsilon-greedy method and the UCB algorithm.

Average Rewards Plot



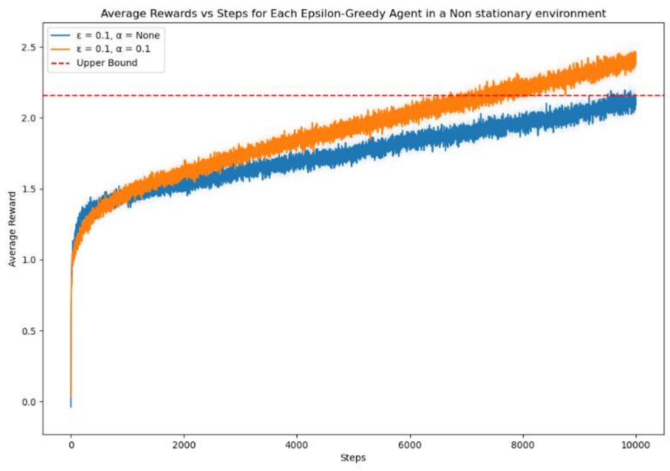
% Optimal Actions Plot



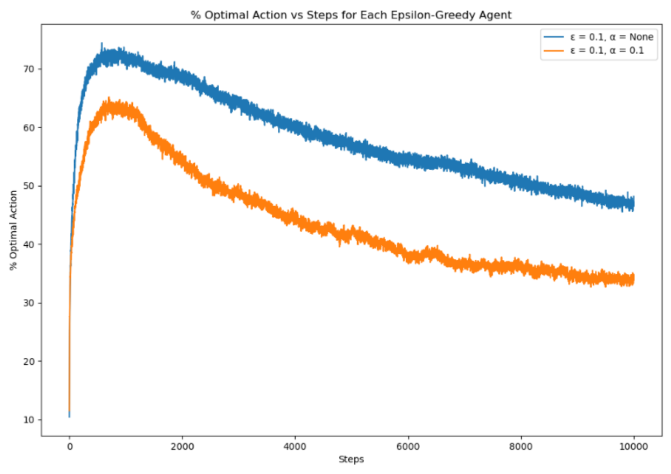
Written: The spikes in the plot for optimistic initialization and UCB reflect initial overestimation. Both methods start with high reward estimates, leading to exploration and a rapid increase in perceived rewards. The subsequent sharp decrease occurs as the algorithms exploit and correct their overestimations, stabilizing around the true action values. This trend is consistent with the theoretical understanding of these methods and is empirically evidenced by the early performance peaks in the plot.

Q8. **Plots**

Average Rewards Plot:



<u>% Optimal Actions Plot:</u>



Q9. **<u>Written:</u>**

<u>The Experiment:</u> Introducing costs for switching between arms (because the agent is a robot that physically moves)

<u>The Experimental Conditions:</u> A Switching Cost Bandit environment is created inheriting the original BanditEnv. In this environment for each step the agent takes, if the last action of the agent is not None value and if the last action is not the same as the current action, the agent incurs a cost of switching (switch_cost=0.1) which is reduced from the reward obtained for the agent taking that step.

In this setup of the environment the performance of the following agents is examined:

1. Epsilon-Greedy agent with $[\varepsilon = 0.01, Q_1 = 5,, \alpha = None]$
2. UCB agent $[Q_1 = 5, c = 2, \alpha = 1]$.

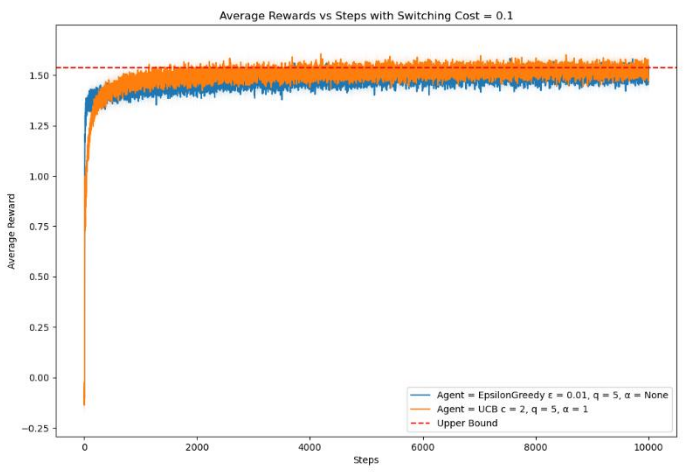
A 10-armed bandit is run for 2000 trials and 10000 steps in each trial for both the agents mentioned above.

<u>Plots:</u>

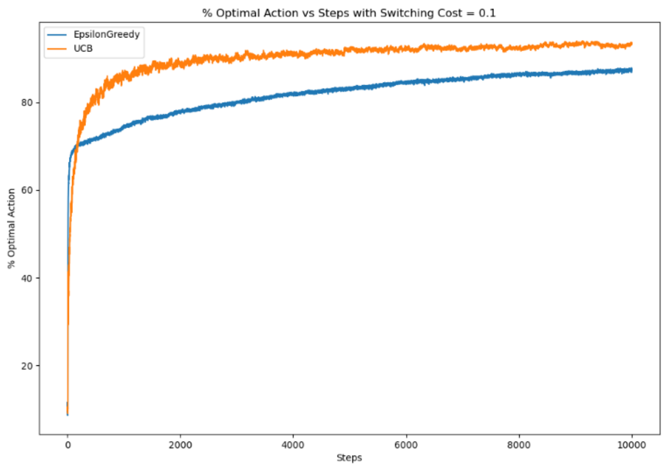
This experiment is analyzed based on the following criteria.

1. Average Rewards vs Steps
2. % Optimal Actions vs Steps
3. % of Switching arms vs Steps

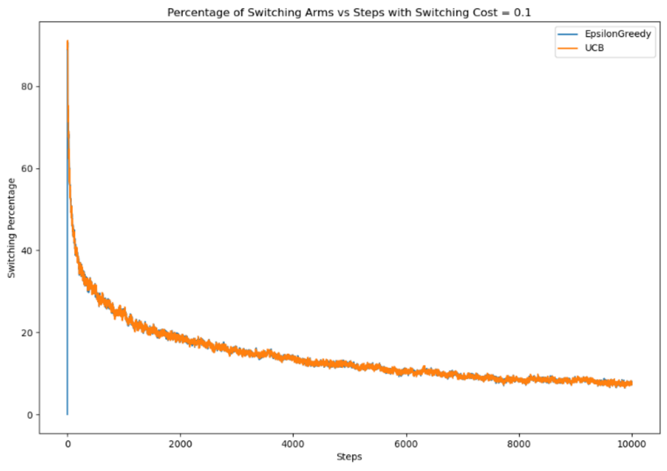
Average Rewards vs Steps



% Optimal Actions vs Steps



% of Switching arms vs Steps



Analysis:

Observing the empirical results obtained in the plots, it’s evident that the average rewards do indeed reach the asymptotic levels of the upper bound value. It’s surprising that despite of the switching cost, the agent finds out a way. We observe that UCB outperforms epsilon greedy by a small margin & UCB finds the optimal solution relatively quickly as compared to the epsilon-greedy policy.

Additionally, it is observed that the switching percentage rapidly falls with a sharp decline within the initial 1000 steps and continues to decline slowly thereafter before adjusting around 10-15% of the time.