Parinaz Shiri (202381576), Himani Thakkar (202292672)
Reinforcement Learning (DSCI 6650)
June 14, 2024

Assignment 1

Bandit Algorithms

Part 1

Greedy with non optimistic initial values.
Solution. The incremental update for the action-value estimate $Q(a)$ in a bandit problem is given by:

$$Q_{n+1}(a) = Q_n(a) + \frac{1}{n}\left[R_n - Q_n(a)\right] \tag{1}$$

where:

- $Q_n(a)$ is the estimated value of action a after n iterations,
- R_n is the reward received after taking action a at iteration n,
- n is the number of times action a has been taken so far.

We used 1000 bandit problems with 10 actions, set the initial values to zero and ran it for 10,000 steps. Over time, the action-value estimates Q(a) become more accurate as they incorporate more information from received rewards. This method only exploits from the previous information that it has from the reward that it got from previous actions and it will never explore for finding other actions which may or may not have better rewards as it focused on. Hence, in the long run, it does not produce a higher reward. This is shown by the second chart which depicts the percentage of optimal actions taken, we see that the algorithm was able to take the optimal action only 35% of the time.

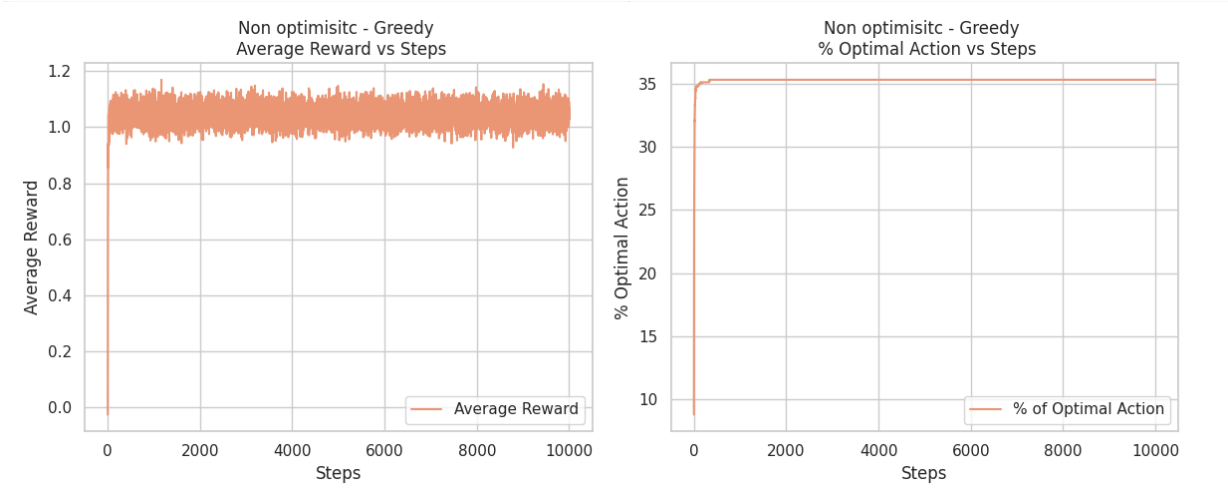

FIGURE 1. Greedy Approach

epsilon-greedy with different choices of epsilon.
Solution. In the epsilon-greedy approach, we balance exploration and exploitation by introducing a parameter ϵ. Using this method, we randomly generate a number and if it is less than epsilon we are going to explore and choose a random action from all possible actions. If the randomly generated number is higher than epsilon it is going to act as greedy algorithm and chooses to exploit which means choose the action that assumes would have the most reward. This algorithm explores with probability of ϵ and it exploits with the probability of 1 - ϵ. The choice of ϵ is crucial as it influences the performance of the bandit algorithm.

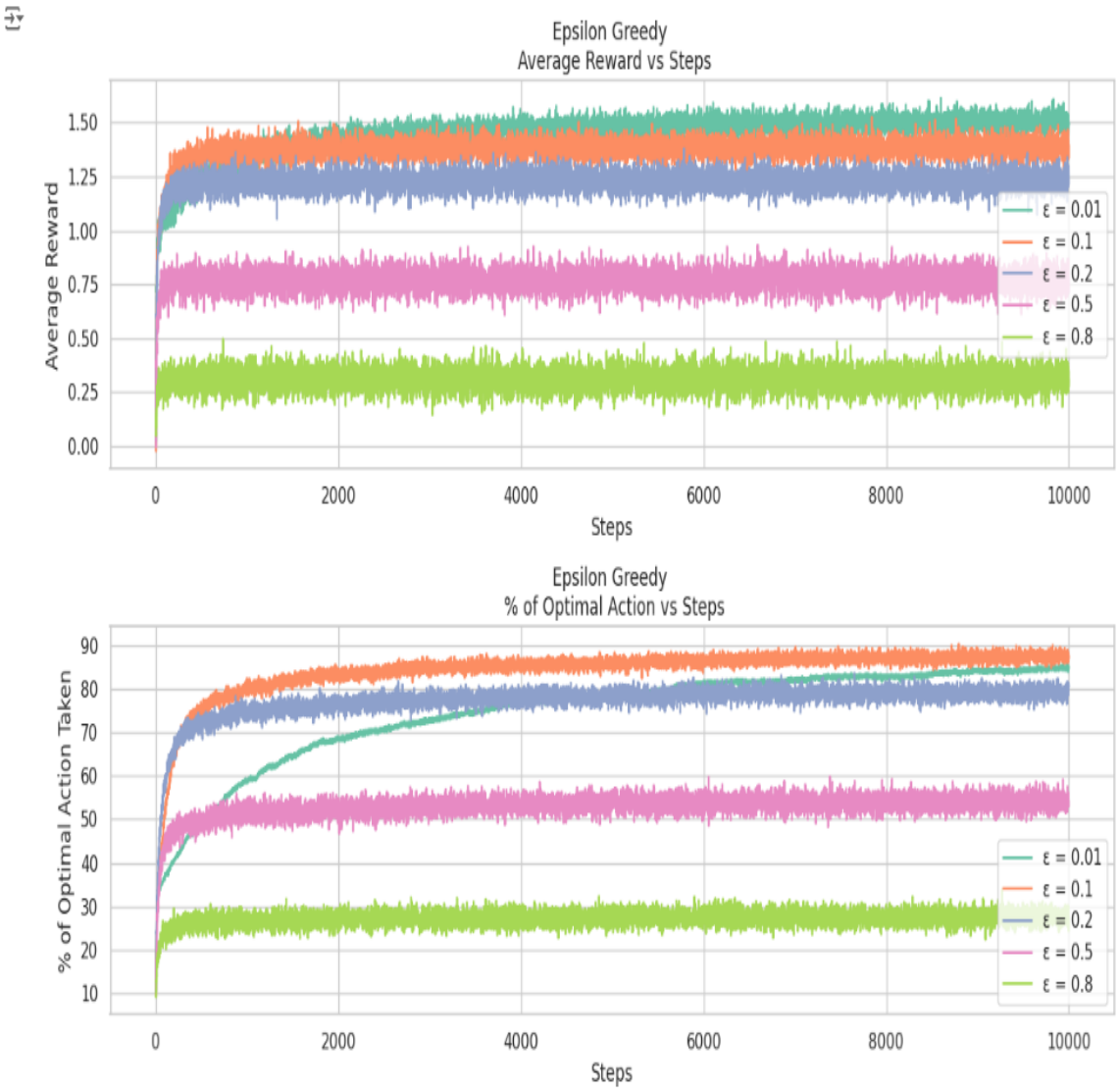

FIGURE 2. Different values of Epsilon

It can significantly impact the trade-off between exploration and exploitation. To determine an appropriate ϵ, we conducted pilot runs on a set of example bandit problems. We evaluated three different values of ϵ: 0.1, 0.01, and 0.005 for 10,000 steps and 1000 bandit problems. By tracking the evolution of the rewards curve, we aimed to identify the ϵ value that yields the best performance.
The plot illustrates the performance of the epsilon-greedy algorithm in a bandit problem context with three different values of epsilon (ϵ): 0.1, 0.01, and 0.5. The performance metrics tracked over 10,000 steps across 1,000 bandit problems include the average reward and the percentage of optimal actions selected.

Observing the average reward plot we can conclude that when the epsilon is lower it learns slower by comparing green and red plots in first steps. However, in future steps we can clearly see that green plot which has $\epsilon = 0.01$ reaches to higher average reward because when the algorithm learned we prefer to decrease the exploration and exploit more but in higher values of epsilon it keeps exploring even though it learned.

Furthermore, at the optimal action diagram we can found at as the epsilon value increases the chances of choosing the optimal action increases since the probability of exploring is more.

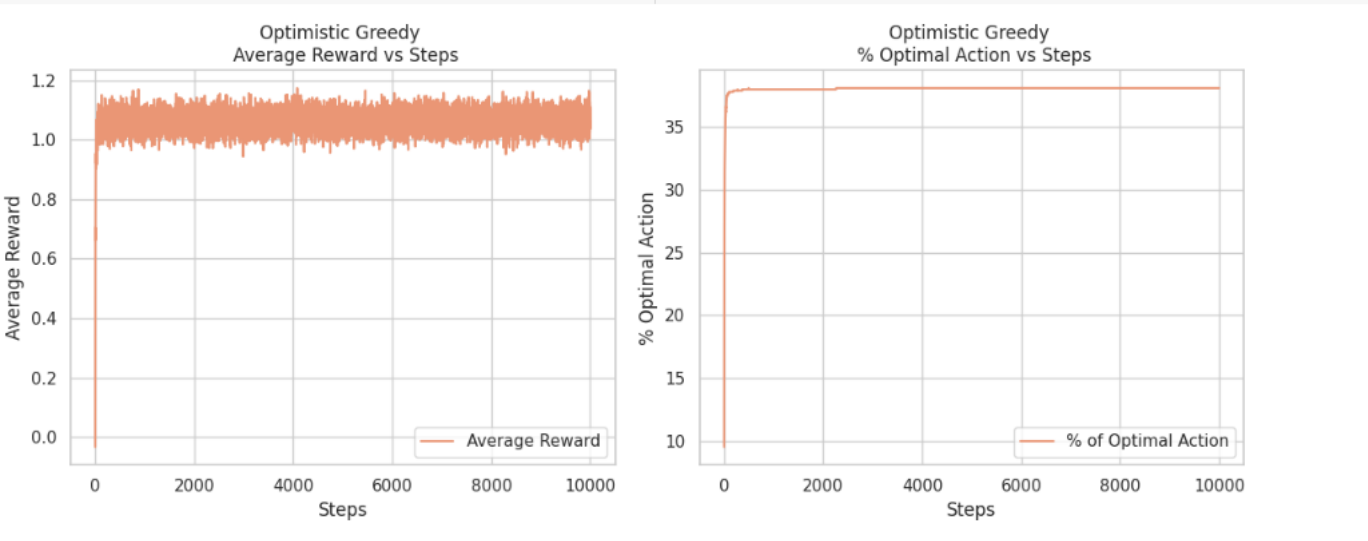

FIGURE 3. Initial Optimistic Values

optimistic starting values with a greedy approach.
Solution. The previous methods, were biased on the initial estimates. The initial action values were set to zero. However, initial action values can be used as a simple way to encourage exploration. We took our initial estimates to be +5, even though the greedy actions are always selected, it will always explore more because of the over estimation of the initial values.

By comparing Figure 1, which depicts the greedy approach, with Figure 3, illustrating the optimistic starting values approach, it is evident that both the average reward and the percentage of optimal actions selected increased. This improvement can be attributed to the higher exploration rate in the optimistic greedy method. The optimistic initial estimates encourage more exploration, which helps balance the critical trade-off between exploration and exploitation. As a result, the algorithm is more likely to discover and exploit high-reward actions over time, leading to enhanced performance metrics.

gradient bandit algorithm.
Solution. In the gradient bandit approach, we focus on learning a numerical preference for each action a, denoted as $H_t(a) \in \mathbb{R}$. These preferences guide the probability of selecting each action but do not directly correspond to the reward. Instead, the relative preference of one action over another is what drives the selection process. To formalize this, we introduce $\pi_t(a)$, which represents the probability of taking action a at time t. Initially, all action preferences are set equally to zero for all a, ensuring that each action has an equal chance of being selected. Over time, the preferences are updated based on the rewards received. This method balances exploration and exploitation by adjusting action probabilities dynamically, leading to a more adaptive and responsive decision-making process.

The plot illustrates the performance of the gradient bandit algorithm in a bandit problem context with three different values of learning rate (α): 0.1, 0.01, and 0.2. The performance metrics tracked over 10,000 steps across 1,000 bandit problems include the average reward and the percentage of optimal actions selected.

In terms of mean reward and optimal action the learning rate $\alpha = 0.2$ exhibits the highest variance with frequent and large fluctuations, occasionally achieving higher rewards and more choice of optimal actions but also experiencing significant drops. This suggests that a higher learning rate can lead to greater rewards but with less stability. On the other hand, the learning rate $\alpha = 0.01$ shows more stability with less fluctuation, but the average rewards and optimal action selection rate generally remain lower and more consistent, often dipping below zero. This indicates that a lower learning rate results in more stable

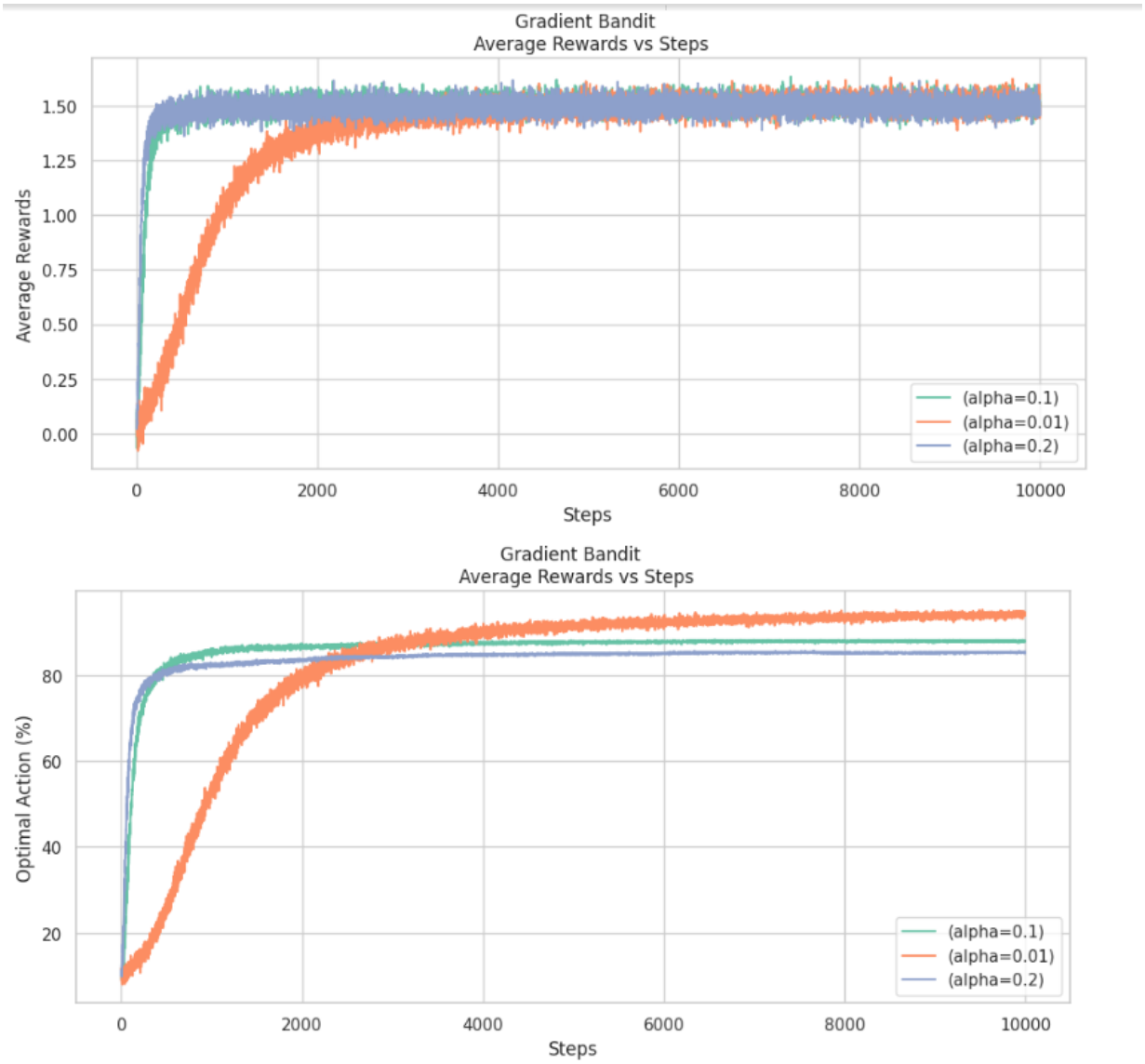

FIGURE 4. Gradient Bandit Algorithm

rewards. The learning rate $\alpha = 0.1$ strikes a balance between the two extremes, showing moderate stability with occasional peaks and troughs, suggesting it provides a reasonable compromise between reward magnitude and stability.

These comparisons highlight the critical impact of learning rate selection in the Gradient Bandit algorithm. A higher learning rate $\alpha = 0.2$ facilitates greater exploration, leading to potential higher rewards and a higher rate of optimal action selection, but with significant instability. Conversely, a lower learning rate $\alpha = 0.01$ ensures stability and consistency but at the cost of lower rewards and slower adaptation. The moderate learning rate $\alpha = 0.1$ offers a balanced compromise, achieving reasonable stability and performance.

Conclusion

Out of the four methods, in a stationary setting, gradient bandit and epsilon greedy had the best performance. The optimal action taken was highest for Epsilon greedy with $\epsilon = 0.1$. Similarly, when keeping the learning rate as 0.01 resulted in high average rewards and the highest percent of optimal action. The parameters should be chosen in a way that they keep the trade-off between exploration and exploitation and the one that keeps the balance better will permorm better as well.

Part 2

In Part 1, the reward probabilities remained constant over time. However, in real-time environments such as traffic control and robotic applications, these probabilities often change gradually or abruptly. In such situations, algorithms must adapt to non-stationary settings. The following experiments compare methods for handling non-stationary problems.

2.1 - Gradual Changes.

We compare the three methods based on their ability to adapt to gradual changes:

1) **Optimistic Greedy Method**: This method initializes action-value estimates optimistically high, encouraging exploration initially.
2) **ϵ-Greedy with Fixed Step Size**: This method selects the best-known action most of the time but occasionally explores other actions with a fixed step size for updating action-value estimates.
3) **ϵ-Greedy with Decreasing Step Size**: This method uses a decaying step size over time (1/n), averaging rewards for action-value estimation.

Drift Change

To simulate Gradual Changes, we introduced a drift change in the reward probabilities, modeled as:
The update equation for μ_t is given by:

$$\mu_t = \mu_{t-1} + \epsilon_t$$

where $\epsilon_t \ = \mathcal{N}(0, 0.001^2)$ is drawn from a normal distribution with mean 0 and variance 0.001^2.

Comparing the three methods, the epsilon-greedy strategy with a fixed step size indicated the largest number of outliers, but the epsilon-greedy approach with a decreasing step size had a more narrow interquartile range (IQR).

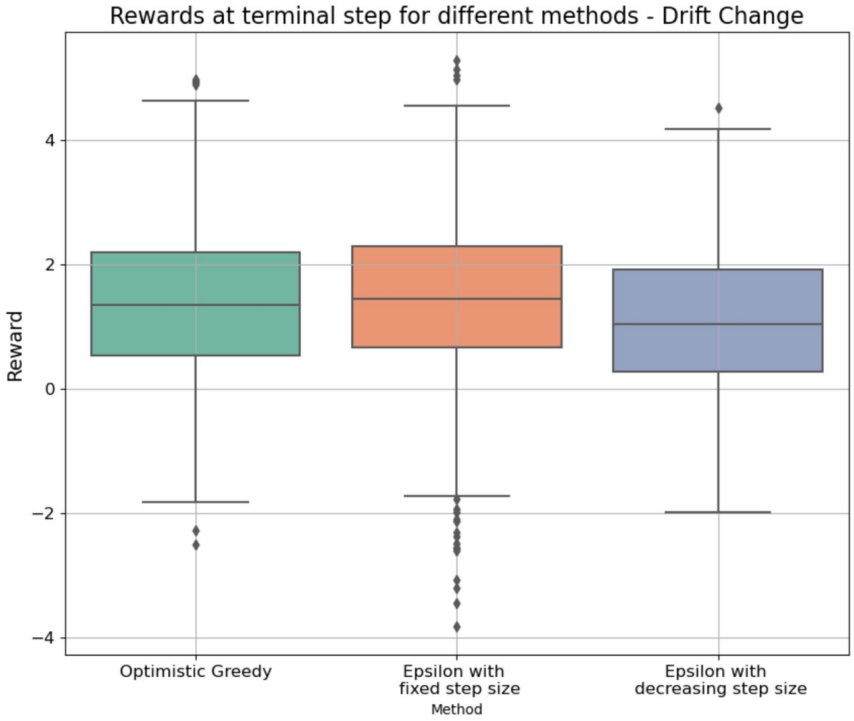

FIGURE 5. Drift Change

Additionally for all three methods, the average reward accumulation for the terminal step ranged from 1 to 1.5. The epsilon-greedy strategy with a fixed step size $\alpha = 0.1$ specifically scored best in terms of mean reward, with a mean reward of 1.4 (table-1). Moreover, it also had the highest value of median, similar to that of the optimistic greedy method.

Mean Reverting Change

For a different non stationary environment, we introduced mean reverting change in the reward probabilities given by:

$$\mu_t = \kappa \mu_{t-1} + \epsilon_t$$

where $\epsilon_t = \mathcal{N}(0, 0.001^2)$ is drawn from a normal distribution with mean 0 and variance 0.001^2.

The average rewards at the terminal step were close to zero for all three methods. This is true because, when the value deviates significantly from its long-term average, it tends to revert back towards the mean which in our case was set to 0. For example, if a stock price is significantly above its historical average, it is expected to fall, and if it is below, it is expected to rise.

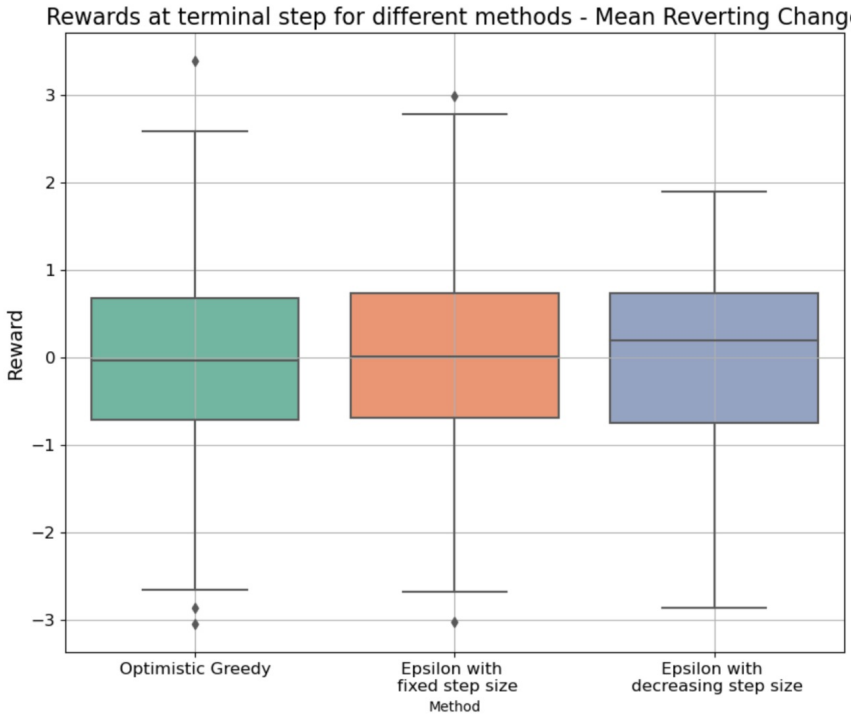

FIGURE 6. Mean Reverting Change

Between the three used methods, Epsilon with a decreasing step size gave slightly higher value of mean rewards at terminal step for mean reverting change and less outliers. Compared to drift change, this method had significantly lesser number of outliers too.

2.2 - Abrupt Changes

Abrupt change mechanism is, at each time step, with a probability of 0.005, the means of the reward distributions are permuted. The way to achieve that is by generating a number between 0 and 1, if it is less than 0.005, the means of the distribution will be permuted otherwise, the means constant. This permutation disrupts the previously learned value estimates, forcing the algorithms to continuously adapt to new reward structures.

The provided box plot compares the performance of three reinforcement learning (RL) algorithms—optimistic greedy, epsilon-greedy with fixed step size, and epsilon-greedy with decreasing step size—under conditions of abrupt change.

The optimistic greedy algorithm shows a median reward around 1 with (IQR) spanning from approximately less than 0 to 2. It has a relatively wide spread, with several outliers indicating significant variability.

The epsilon-greedy algorithm with fixed step size has a similar median reward of about 1, but its IQR ranges from below -2 to higher than 4, suggesting slightly better consistency. However, it also exhibits some outliers. The epsilon-greedy algorithm with decreasing step size has a median reward slightly above 0, with an IQR from below -2 to slightly below 4. The distribution is narrower and more stable compared to the other two methods, but it still shows some outliers.

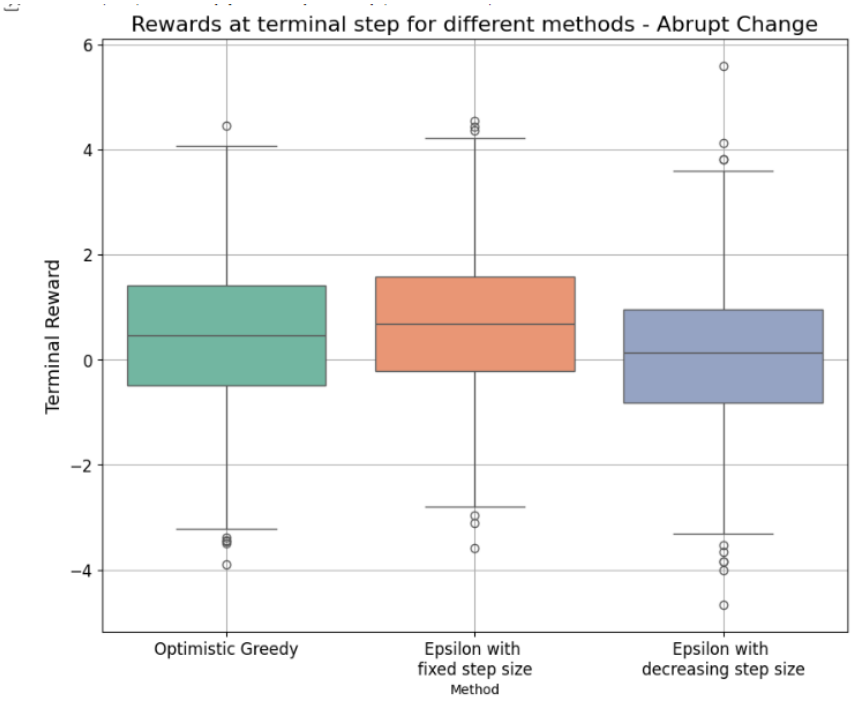

FIGURE 7. Abrupt Change

In summary, the epsilon-greedy with decreasing step size algorithm demonstrates the most stable performance under abrupt change conditions, as indicated by its narrower IQR and fewer extreme outliers. The optimistic greedy and epsilon-greedy with fixed step size algorithms both show similar median rewards, but the fixed step size method exhibits slightly better consistency. Despite this, both methods have notable outliers, reflecting their susceptibility to abrupt changes.

Conclusion

Comparing only the mean rewards, epsilon with fixed step size algorithm performs better than the other 2 algorithms. However, if we are considering, other factors such as the interquartile range (consistency with all bandit problems) and outlier distribution, epsilon with decreasing step size has a better performance.

Method	Drift Change	Mean Reverting Change	Abrupt Change
Optimistic Greedy	1.35	-0.02	0.41
ϵ with fixed step size	1.41	0.014	0.66
ϵ with decreasing step size	1.06	0.017	0.1

TABLE 1. Average rewards of different methods for non-stationary problems

REFERENCES

Reinforcement Learning: An Introduction, Second Edition.
Understanding Reinforcement Learning Hands-on: Non-Stationary.
Solving the multi-armed bandit problem.