Memorial University of  
Newfoundland



Assignment 1

Experimentation with simple bandit learning algorithms

Submitted by:

Oviemuno Peter Utomakili (201477924)

Dhiraj Rajkarnikar (202382307)

Table of Contents

[**INTRODUCTION** 3](#_Toc169165243)

[**OBJECTIVES** 3](#_Toc169165244)

[**EXPERIMENTS** 3](#_Toc169165245)

[**Scenario 1 (Question 1):** 3](#_Toc169165246)

[**Scenario 2 (Question 2):** 6](#_Toc169165247)

[**RESULTS** 10](#_Toc169165248)

[**CONCLUSION** 12](#_Toc169165249)

# **INTRODUCTION**

A simple bandit learning algorithm refers to any rudimentary strategy that can be utilized to solve the multi-armed bandit problem. The multi-armed bandit problem is an iterative problem where a decision maker selects one of multiple fixed choices iteratively with limited information about the properties of each choice. This is sometimes referred to as K-armed or N-armed bandit problem. The goal of the problem is to maximize the cumulative reward over a series of trails. This is achieved by balancing the exploration-exploitation concepts in reinforcement learning. In simpler terms, figuring out the compromise between trying out different arms or choices to discover their rewards and leveraging the choices or arms that has provided the highest reward so far. Some of the classic simple bandit learning algorithms are:

1. Greedy Algorithm with non-optimistic initial values
2. Epsilon-Greedy Algorithm
3. Greedy Algorithm with optimistic initial values
4. Gradient Bandit
5. Upper Confidence Bound

Basically, the purpose of simple bandit algorithms is to provide various strategies for balancing exploration and exploitation.

# **OBJECTIVES**

The project aims to provide comparisons between various bandit learning algorithms in terms of performance with i. average accumulated reward and ii. The proportion of time the optimal action is taken. The goal is to start by experimenting with simple bandit problem with stationary reward distributions with k-armed testbed and then move on to non-stationary reward distributions. We also experiment on the types of non-stationary reward – gradual or abrupt or combination of both. The algorithms we are experimenting with are i. Greedy non-optimistic, ii. Epsilon-greedy, iii. Greedy optimistic and iv. Gradient bandit for stationary problems and i. Optimistic greedy, ii. Epsilon-greedy with fixed step size and iii. Epsilon-greedy with decreasing step-size for non-stationary problems. We take this further by fine-tuning the parameters for these algorithms and also repeat the total experiment runs to 1000 sets for both stationary and non-stationary problems.

# **EXPERIMENTS**

## **Scenario 1 (Question 1):**

Consider the so-called k-armed testbed, with k = 10, with normally distributed rewards. Generate a set of ten means μ1, . . ., μ10 from a N(0, 1) distribution and suppose that the arms 1 through 10 have N(μ, 1) reward distributions where i = 1, . . ., 10. You goal is to learn the action values corresponding to each of the 10 arms, i.e., the expected rewards q∗(a) for a = 1, . . . 10 using the different methods:

* 1. Greedy with non-optimistic initial values.
  2. Epsilon-greedy with different choices of epsilon.
  3. Optimistic starting values with a greedy approach.
  4. Gradient bandit algorithm.

Answer:

For k-armed testbed with k = 10 and normally distributed rewards, we conducted experiments with four different methods – Greedy with non-optimistic initial values, Epsilon-greedy with different choices of epsilon, Greedy with optimistic starting values and Gradient bandit with various alpha values. The plots below show the results of the experiment with descriptions.

We experimented with three different parameters for Epsilon-greedy, Greedy with optimistic values and Gradient bandit. The choices were:

1. Epsilon = 0.5, Optimistic initial value = 10, alpha = 0.5
2. Epsilon = 0.1, Optimistic initial value = 10, alpha = 0.1
3. Epsilon = 0.01, Optimistic initial value = 10, alpha = 0.01

A close-up of a graph

Description automatically generated

Figure 1: Simple Bandit problem with stationary reward

From the plot above we can see:

**Greedy non-optimistic** algorithm starts with under sub-par rewards and does not improve at all indicated from the steady red line. And for the percentage of times the optimal action is chosen, this algorithm quickly increases the optimal action percentage, stabilizing at the lowest value among all.

**Epsilon-greedy with epsilon value of 0.5** algorithm starts with relatively less rewards and stabilizes as shown by the dark blue line. It does not perform very well. Choosing the optimal actions in percentage, it chooses the optimal action in higher number and steadies around those values.

**Epsilon-greedy with epsilon value of 0.1** algorithm is denoted by orange line. It peaks fast performs relatively well and. The algorithm has the second fastest increase in optimal action percentage indicating good exploration.

**Epsilon-greedy with epsilon value of 0.01** is denoted by green line. It has a relatively mid average reward over time but with increasing steps, the average reward also seems to increase suggesting that if number of steps are increased, it might be good. Similarly, the optimal action chosen overtime also peaks around the mid-increases steadily.

**Greedy optimistic with initial value 10** denoted by pink line peaks higher and faster compared to others and continues with the same average reward from about 100 steps till the end. This performs better than most of the other scenarios. And same is true with the choice of optimal action percentage over time which indicates the algorithm has chosen the optimal action at a certain step and then sticks to it until the very end.

**Greedy optimistic with initial value 5** denoted by burgundy line performs similar with initial value of 10 but slightly worse and same can be said for the percentage choice of optimal action over time.

**Greedy optimistic with initial value 1** denoted by purple line has the worst performance among all Greedy algorithms with optimistic initial value but just by a slight margin. And it indicates same patter in optimal action chosen.

**Gradient Bandit with alpha 0.5** denoted by grey line takes a bit of time to get to peak compared to others and steadily keeps rising. It shows good average rewards over time and does not plateau. Regarding the optimal action percentage over time, the algorithm has a moderate increase in optimal action percentage stabilizing at around the middle of the pack of algorithms.

**Gradient Bandit with alpha 0.1** denoted by yellow line has a steady start but achieves the highest reward (average) overall, indicating effective learning process and adaptation. For choosing optimal action percentage, it achieves the highest percentage overall and demonstrates effective action selection.

**Gradient Bandit with alpha 0.01** has the slowest start but improves steadily reflecting slow or conservative exploration strategy. It does seem to be increasing and with very high number of steps, it might get better. And the algorithm has the slowest increase in optimal action percentage, struggling to explore adequately in the beginning but does improve a bit over the number of steps.

From the results above, it seems like Gradient Bandit with alpha value of 0.1 performs the best among all and stands out along with Epsilon-greedy with epsilon value of 0.1 coming close behind. This shows, in this setting, that these algorithms achieve high rewards and frequently select the optimal action.

## **Scenario 2 (Question 2):**

In this scenario, we consider non-stationary modifications to the problem in scenario 1. We consider the changes in rewards to be gradual, abrupt or a combination of both. To understand this, we compare the following algorithms:

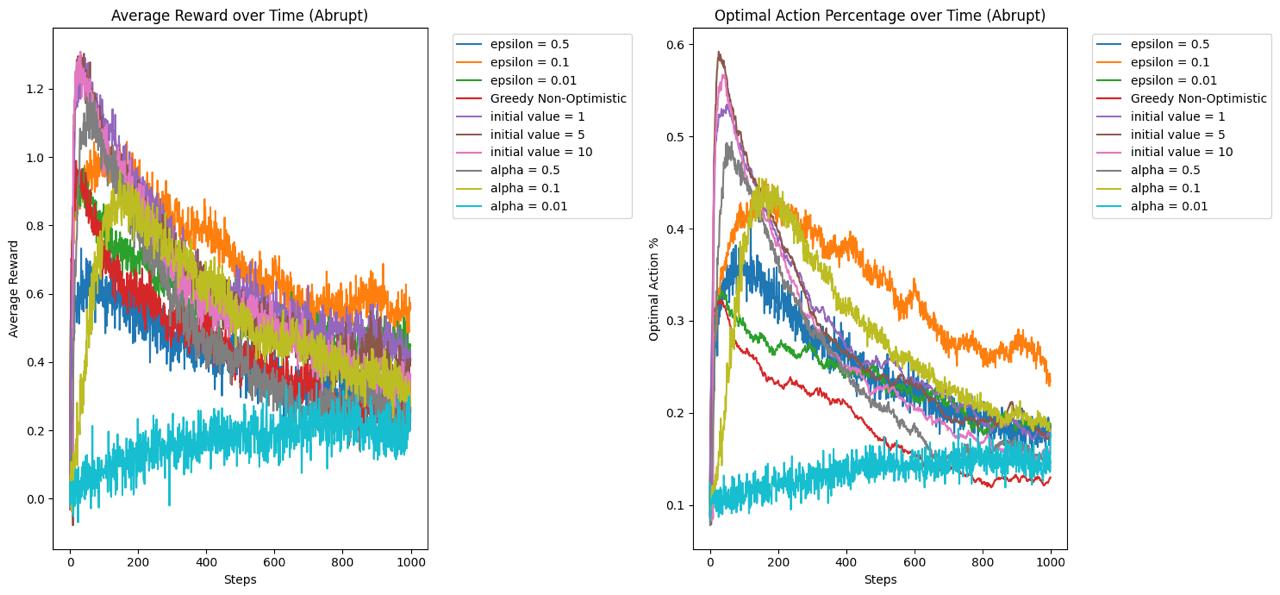
1. Optimistic greedy algorithm
2. Epsilon greedy algorithm with fixed step size
3. Epsilon greedy algorithm with a decreasing step size
4. 

Figure 2: Abrupt Change in Reward Distribution

The plot above shows the comparison the performance of different multi-armed bandit algorithms under abrupt changes in rewards distribution.

**Epsilon-greedy with epsilon value of 0.5** denoted by dark blue line exhibits a lower average reward stabilizing at around 0.2 to 0.4 and for optimal action percentage, it starts low and gets even lower and stays there around 0.15 to 0.25.

**Epsilon-greedy with epsilon value of 0.1** denoted by orange line shows moderate increase in average rewards and has the highest average rewards among all. For optimal action percentage, it peaks moderately but stays at highest optimal action percentage with fluctuations.

**Epsilon-greedy with epsilon value of 0.01** denoted by green line has moderate peak in average rewards then dips and stabilizes at around second highest among all. But the optimal action percentage does not stabilize like on the left plot.

**Greedy optimistic with initial value of 1** denoted by purple line has one of the highest peaks in average rewards but dips down over time and stabilizes at around 0.35 to 0.45. The optimal action percentage is right around the mid pack stabilizing at around 0.2.

**Greedy optimistic with initial value of 5** denoted by burgundy line starts high but eventually converges to a lower average reward and exhibits higher optimal action percentage but decreases rapidly.

**Greedy optimistic with initial value of 10** denoted by pink line peaks high as well but drops down around the middle of the pack as steps increase and exhibits similar behaviour for optimal action percentage overtime.

**Gradient Bandit with alpha 0.5** denoted by grey line show moderate increase in average rewards and falls down to almost the bottom of the pack and same is true for optimal action percentage.

**Gradient Bandit with alpha 0.1** denoted by yellow line peaks relatively moderate and slow and drops down to the mid of the pack but even though the optimal action percentage shows similar curve it stabilizes around the highest of the mid-pack.

**Gradient Bandit with alpha 0.01** denoted by blue line exhibits the lowest average rewards over time but has slight increase in slope and same can be said for optimal action percentage over time.

From the plot above, we can conclude that Epsilon greedy with epsilon value of 0.1 performs the best even though it does not peak very high, it stabilizes at the highest value over the duration of the experiment.

A close-up of a graph

Description automatically generated

Figure 3: Mean Reverting Change

The plot above shows the comparison of performance of various multi-armed bandit algorithms under mean-reverting changes in reward distribution. The left subplot shows average reward over time (mean-reverting) and right subplot shows optimal action percentage over time (mean-reverting).

From the plots we can see that most of the algorithms exhibits a fluctuating average reward, generally centred around zero. And for optimal action percentage, the algorithms show similar attributes. We can, from the plots, suggest that in mean-reverting environments, all the algorithms struggle to maintain consistent high rewards and optimal action percentages due to the inherently fluctuating nature of the rewards distribution. We can also conclude that the performance of these algorithms is almost inseparable compared to the performance during stationary rewards and abrupt changing rewards.

A close-up of a graph

Description automatically generated

Figure 4: Drift Change in Reward Distribution

From the plots above, we see:

**Epsilon-greedy with epsilon value of 0.5** denoted by dark blue line exhibits a lower average reward stabilizing at around 0.5 to 0.8 and for optimal action percentage, it starts low and stays there around 0.4 to 0.5.

**Epsilon-greedy with epsilon value of 0.1** denoted by orange line shows rapid increase in average rewards and has the peaks among the highs in average rewards. For optimal action percentage, it peaks moderately taking some steps but gets at second highest optimal action percentage among all the algorithms.

**Epsilon-greedy with epsilon value of 0.01** denoted by green line has rapid increase but small peak in average rewards then gradually increases to meet the mid pack. And the optimal action percentage shows the same trend.

**Greedy optimistic with initial value of 1, initial value of 5 and initial value of 10**  all show similar trend peaking and stabilizes around the mid pack among all the algorithms in both rewards and optimal actions.

**Gradient Bandit with alpha 0.5** denoted by grey line show moderate increase in average rewards and still shows some kind of slow increase and same is true for optimal action percentage.

**Gradient Bandit with alpha 0.1** denoted by yellow line starts relatively moderate and peaks the highest among all performing the best. And the optimal action percentage shows similar curve and still shows slight increase in the choice of optimal action.

**Gradient Bandit with alpha 0.01** denoted by blue line exhibits the lowest average rewards over time but has slight increase in slope and same can be said for optimal action percentage over time.

From the interpretation above, we can say that the Gradient bandit with alpha value of 0.1 performs the best among all with Epsilon-greedy with epsilon value of 0.1 close behind. Overall, the results suggest that in environments with drifting rewards distributions, lower learning rate for the gradient bandit and lower exploration rate of epsilon for Epsilon-greedy performs the best.

# **RESULTS**

From the experiments conducted, we have comparison of performance of the various algorithms we have experimented with. The results are shown by the box plots below along with interpretation.

A diagram of a diagram of a number of different numbers

Description automatically generated with medium confidence

Figure 5: Terminal Reward Distribution for Abrupt Change

From the boxplot above, we can see all three methods – Optimistic Greedy, Epsilon Greedy fixed and Epsilon Greedy decreasing have medians close to 0 with the variability also being similar among all the methods. Also, to notice is that the outliers are present above and below the main range of data. The three methods have similar performance in terms of terminal reward distribution for abrupt change.

A diagram of a diagram of a number of different numbers

Description automatically generated with medium confidence

Figure 6: Terminal Reward Distribution for Mean-Reverting

From the boxplot above, we can see all three have medians slightly above 0 with the variability also being similar among all the methods. Also, to notice is that there are no outliers in Optimistic greedy. The three methods have comparable performance in terms of terminal reward distribution for mean-reverting.

There is really not much difference between terminal reward distribution for abrupt change and mean-reverting from the box plots. Therefore, the choice of method between these two might be guided by other factors such as computation efficiency, ease of implementation or problem statement.

A diagram of a number of different numbers

Description automatically generated with medium confidence

Figure 7: Terminal Reward Distribution for Drift Change

From the boxplot above, we can see all three have medians between 1 and 2 with Epsilon-greedy fixed having higher than other two and variability among all methods seem to be similar but Epsilon-greedy fixed shows higher range of outliers below the data range. The three methods have comparable performance in terms of terminal reward distribution for drift change with Epsilon-greedy fixed having slightly higher median reward but also suggests that it might have higher risk of extreme negative outcomes suggesting there is a trade-off between reward and risk.

# **CONCLUSION**

The aim of the project was to compare the various simple bandit algorithms in various environments and scenarios. From our experiments with stationary reward in scenario 1 (question 1), we found out that the Gradient Bandit algorithm performs the best with learning rate of 0.1. It can be concluded that the learning rate of 0.5 was a bit worse than learning rate of 0.1 and learning rate of 0.01 was most of the time worst among all the algorithms which suggests that the learning rate chosen is adequate to provide convincing results. Following closely behind was Epsilon-Greedy algorithm with Epsilon value of 0.1.

Based on scenario 2 (question 2), the three methods – Optimistic Greedy, Epsilon-Greedy Fixed and Epsilon-Greedy Decreasing perform similar in terms of terminal rewards. There is not any indication of distinct and consistent advantage. There is a slight indication that Epsilon-greedy fixed may perform slightly better in drift change but also has a higher risk for risk-sensitive applications due to extreme negative outliers. Optimistic Greedy and ε-Greedy Decreasing methods offer more consistent performance with fewer extreme outcomes, making them suitable for more risk-averse applications.