Homework1: Nonstationary Bandit Problem

Pang Liu

February 11, 2025

1 Introduction

In this homework, it is aimed to design and conduct an experiment that demonstrates the limitations of the sample-average method in nonstationary problems.

Previously, in the 10-arm bandit experiment, the true action values $q_*(a)$ were sampled once from a $\mathcal{N}(0, 1)$ distribution and kept fixed. I replicate the experiment to explore the performance of the ϵ-greedy strategy under three different ϵ values for 1000 steps. After replicating the experiment (running `1_experiment_2_2.py`), I obtained a result consistent with the textbook, as shown in Figure 1.

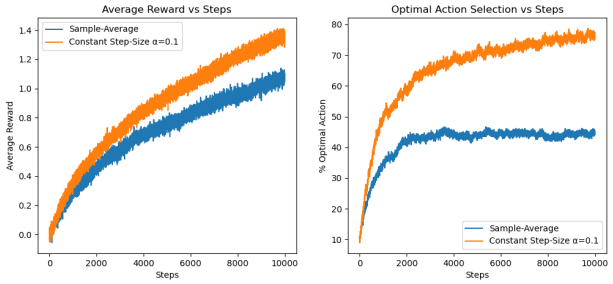

Figure 1: Replication of the 10-arm bandit experiment in the textbook

This experiment indicated that $\epsilon = 0.1$ is optimal among the three settings. Building on this, I am interested in investigating a new question: **if the environment is nonstationary, how do different Q-estimation methods perform?**

With the proven optimal $\epsilon = 0.1$, I will extend the number of steps and create a nonstationary testbed to examine:

- The sample-average method (incremental update)
- The constant step-size method

The expectation, based on theoretical knowledge, is that the constant step-size method will adapt more quickly to changes in $q_*(a)$ and thus perform better in a nonstationary environment.

2 Experiment Setup (Testbed)

Before comparing, let's call back the methods that we have learned. First of all is the **Sample-Average Method**,where we store all historical rewards to compute the mean:

$$Q(a) = \frac{1}{N(a)} \sum_{i=1}^{N(a)} R_i,$$

We also use the **incremental form**, which is equivalent in result to the sample-average but more efficient computationally:

$$Q(a) \leftarrow Q(a) + \frac{1}{N(a)}\big(R - Q(a)\big),$$

Another one is **Constant Step-Size Method**, where $\alpha = 0.1$. This method places more weight on recent data and is expected to adapt more quickly in a nonstationary setting:

$$Q(a) \leftarrow Q(a) + \alpha\big(R - Q(a)\big),$$

We use incremental form of Sample-Average Method and Constant Step-Size Method to continue testing.

2.1 Bandit Environment Setup

- **Initial values** of $q_*(a)$: All arms start with the same initial $q_*(a)$, which can be set to zero or another constant.
- **Random Walk**: At each step, all $q^*(a)$ undergo a random walk:

$$q^*(a) \leftarrow q^*(a) + \mathcal{N}(0,\, 0.01).$$

- **Reward**: When the agent selects an action a, the reward R is drawn from

$$R \sim \mathcal{N}\big(q^*(a),\, 1\big).$$

2.2 Agent Setup

- ϵ-greedy with $\epsilon = 0.1$.
- Compare:
  1. Sample-Average (incremental form)
  2. Constant Step-Size ($\alpha = 0.1$)

2.3 Experiment Procedure

- Each experiment runs for 10,000 steps.
- The experiment runs 2,000 independent trials and then average the results.
- The experiment records:
  1. The average reward at each step (to plot the reward curve).
  2. The proportion of choosing the optimal action at each step (to plot the optimal action percentage curve).

Based on theoretical reasoning, constant step-size is expected to adapt faster to changes in $q^*(a)$ and demonstrate better performance in both the reward curve and the optimal action selection curve.

3 Experimental Results

According to the above experimental setup, the result implemented by the codes is shown in Figure 2.

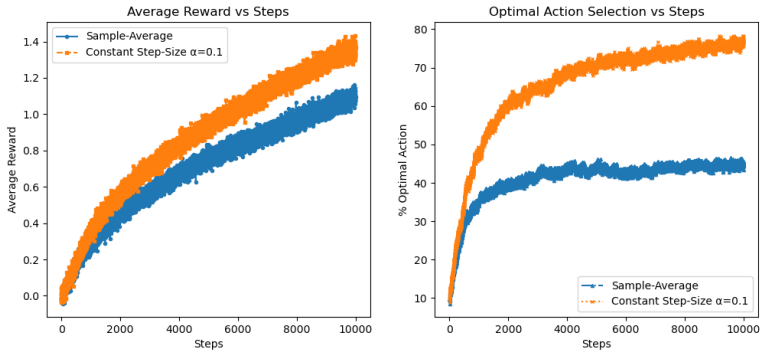

Figure 2: Raw results of the nonstationary bandit experiment

Because the raw lines are difficult to distinguish, I employed a smoothed (moving average) approach to highlight overall trends, shown in Figure 3.

From these results, we can observe:

- **Smoothed Average Reward vs. Steps (Left Subplot)**:
  The orange dashed line (Constant Step-Size $\alpha = 0.1$) achieves higher reward and continues to improve over time. The blue solid line (Sample-Average) grows more slowly and converges to a lower final value.
  Conclusion: Constant step-size ($\alpha = 0.1$) adapts more quickly in the nonstationary environment, leading to higher average reward.

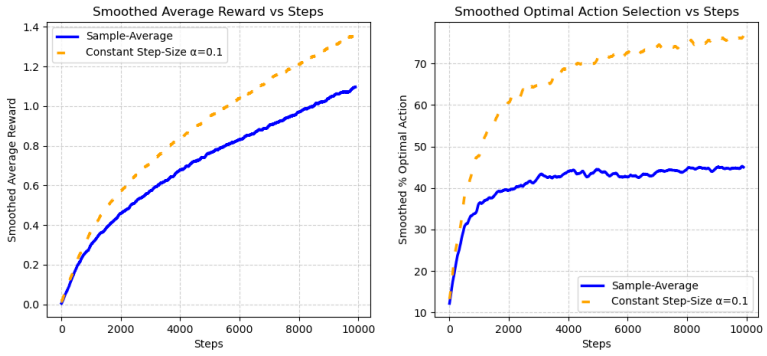

Figure 3: Smoothed results: (Left) Average Reward vs. Steps, (Right) Optimal Action Selection vs. Steps

- **Smoothed Optimal Action Selection vs. Steps (Right Subplot)**: The orange dashed line (Constant Step-Size) has a higher rate of optimal action selection, stabilizing around 70%, whereas the blue solid line (Sample-Average) stabilizes around 40%.
  Conclusion: Constant step-size ($\alpha = 0.1$) effectively tracks changes in $q^*(a)$, enabling the agent to select the optimal action more frequently.

Hence, these results confirm that the constant step-size method outperforms the sample-average method in nonstationary settings.

One important reason why the constant step-size method is more suitable for nonstationary environments can be seen by examining the sample-average (incremental) approach. Because the sample-average method continually averages all past rewards, if $q^*(a)$ is changing, older data may mislead current action-value estimates. In other words, when the underlying reward function shifts, the accumulated influence of outdated samples slows adaptation and can lead to suboptimal decisions.

In contrast, a constant step-size approach (with $\alpha = 0.1$) places greater weight on recent rewards, effectively "forgetting" older information. This design choice allows the agent to respond more rapidly whenever $q^*(a)$ changes. Hence, the constant step-size method is inherently better suited to nonstationary environments in which action values fluctuate over time.

4 Further Thinking

Although the experiment has shown that $\alpha = 0.1$ works effectively in this setup, there are additional investigations one can pursue.

4.1 Adaptive Step-Size

One idea is to let the step-size decrease over time, for instance:

$$Q_{t+1} = Q_t + \frac{1}{t}(R_t - Q_t),$$

where older data gradually lose influence. However, in nonstationary environments, if $\frac{1}{t}$ becomes too small, it can also slow down adaptation to new changes. Empirically, it can be found (Figure 4) that such a strategy might perform even worse than the pure sample-average method, likely because the step-size shrinks excessively as t grows.

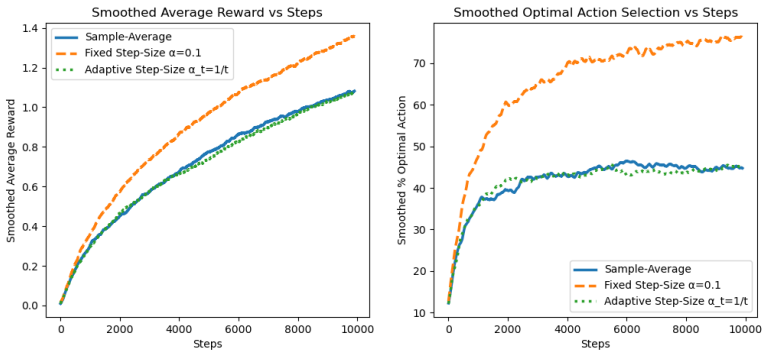

Figure 4: Results of an adaptive step-size approach

4.2 Sliding Window Average

Alternatively, we can apply a fixed-size sliding window of the most recent N rewards:

$$Q_t = \frac{1}{N} \sum_{i=t-N+1}^{t} R_i,$$

or in incremental form,

$$Q_t = Q_{t-1} + \frac{1}{N}\Big(R_t - R_{t-N}\Big).$$

Old data are discarded after they fall out of the window. In experiments (Figure 5), the sliding window average improves upon the pure sample-average but still tends to underperform compared to a properly chosen constant step-size.

4.3 Choosing the Best α

Exploring Different α values through varying α (e.g., 0.05, 0.2, 0.5) to see how different step-sizes affect performance. Extremely large α may lead to instability (too sensitive to noise), whereas extremely small α may adapt too slowly.

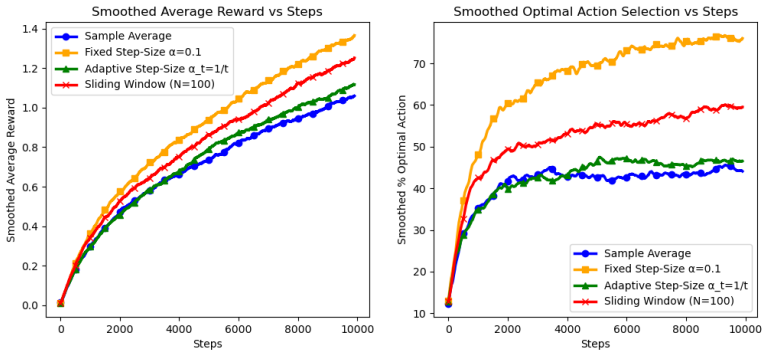

Figure 5: Results of the sliding window average

Figure 6 shows the performance of different α values: $[0.01, 0.05, 0.1, 0.2, 0.5]$. Indeed, $\alpha = 0.1$ strikes a good balance between adaptation and noise smoothing in our testbed, confirming it to be the best among the chosen set.

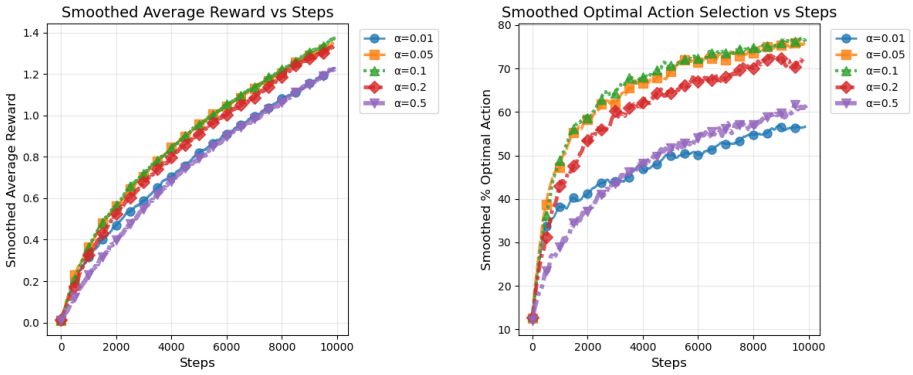

Figure 6: Comparison of different constant step-size values α

5 Conclusion

In this homework, we designed a nonstationary bandit experiment to compare the sample-average method and the constant step-size method ($\alpha = 0.1$). The results support the theoretical expectation that fixed step-size methods perform better in nonstationary environments, because they place more emphasis on recent rewards and thereby adapt more quickly to shifting action values.

Additionally, we explored variants such as an adaptive step-size approach and a sliding window average. While these methods can partially mitigate the

shortcomings of pure sample averaging, the fixed step-size method with a well-chosen α remains the most straightforward and effective strategy.

In summary:

- Sample-average methods are suitable for stationary environments but struggle when the true action values change over time.
- Constant step-size methods effectively "forget" outdated data and adapt to new conditions, leading to higher rewards and better optimal-action selection rates.
- The choice of α is crucial: too large can induce excessive variance, too small can slow learning. Empirically, $\alpha = 0.1$ works well for this setup.

Extra Credit

Exercise 1.1: Challenges in Self-Play

Potential issues in self-play:

1. The agent could get stuck in loops.
2. It may fail to learn the optimal strategy.
3. The policy could remain static since, on average, both agents might repeatedly draw in each iteration.

However, AlphaGo successfully utilized extensive self-play to improve its performance. Addressing these issues requires mechanisms beyond the scope of our current knowledge, introducing additional techniques to prevent these limitations.

Exercise 1.2: Symmetries in Reinforcement Learning

Advantages:

1. Reducing the state space improves search efficiency.
2. Theoretically, symmetrically equivalent positions should have the same value in a perfect-play scenario.

Challenges: The values assigned to symmetrically equivalent positions may vary depending on the opponent's behavior.

Thus, a balance must be maintained between theoretical optimality and practical adaptability.