EECE 5614: Reinforcement Learning

Name: Shalaka Padalkar

NU ID: 002831387

Project 1

Given:

An agent plays a 2-arm bandit, trying to maximize its total reward.

In each step, the agent selects one of the levers and is given some reward according to the reward distribution of that lever.   
  
Assumption:

The reward distribution for the first lever is a Gaussian with 𝜇1 = 6, 𝜎12 = 15, and

for the second lever is a binomial Gaussian with 𝜇21 = 11, 𝜎212 = 16, 𝜇22 = 3, 𝜎222 = 8,

which means that the resulting output will be uniformly probable from these two Gaussian distributions.

If these distributions were known (which in practice are not), we could compute the optimal/true action values as:

A mathematical equation with black letters

AI-generated content may be incorrect.

But, in this problem, we assume: reward distributions are unknown, and the agent only sees a realization of reward after selecting an action.

The agent acts according to the 𝜖-greedy action selection policy with parameter 𝜖:

A close up of black text

AI-generated content may be incorrect.

Consider: The agent selects 1000 actions (step/time).

For smoother results, repeat 1000 steps for 100 independent runs.

Solution:

Part A:

Set initial Q values at the beginning of each run as Q(a1) = Q(a2) = 0

If action a is selected at time step k and the reward rk is observed,

Q value for corresponding action will be updated as:



Consider:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 𝛼 | 1 | 0.9k |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 𝜖 | 0 | 0.1 | 0.2 | 0.5 |

Plot: Average accumulated reward w.r.t. time/step

Results: (details continued on the next page)

* Which of the greedy, random, in-between policies performed the best?

The greedy policy focused entirely on exploiting the best-known action, failing to discover better actions through exploration. This policy performed well for 𝛼 = 1.

The random policy ensured maximum exploration, resulting in most accurate Q estimates across all the learning rates. This policy delivered highest accumulated reward for slower decaying learning rates like the exponential (𝛼 = 0.9k) and the inverse (𝛼 = ).

The in-between policy balanced exploitation and exploration. It performed the best for the logarithmic (𝛼 = ) learning rate, achieving the highest accumulated rewards.

* Which learning rate was the best?

In my opinion, the logarithmic learning rate (𝛼 = ) achieved the best balance between stability and adaptability, making it effective for exploitation-exploration based policies.

* Which pair of 𝛼 and 𝜖 led to the maximum average accumulated reward?

The 𝛼 = and 𝜖 = 0.2 delivered the highest accumulated reward, letting the agent discover the exploit the optimal action.

Figure 1:

A graph showing different colored lines

AI-generated content may be incorrect.

Table 1: Average Q-values for 𝛼 = 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epsilon-greedy | Average of action value Q(a1) of 100 runs | True action value Q\*(a1) | Average of action value Q(a2) of 100 runs | True action value Q\*(a2) |
| 𝜖 = 0(greedy) | -1.89 | 6 | 2.61 | 7 |
| 𝜖 = 0.1 | 2.92 | 6 | 2.82 | 7 |
| 𝜖 = 0.2 | 3.66 | 6 | 3.81 | 7 |
| 𝜖 = 0.5 (random) | 5.06 | 6 | 5.31 | 7 |

Analysis:

Since the learning rate 𝛼 = 1, the agent replaces the existing Q(a) value with the new update every single time. It can be observed from the above plot that the blue curve (𝜖 = 0) has the highest accumulated reward whereas, the other curves stabilize at lower rewards. The agent does not explore and exploits it initial estimates of Q(a), hence, yielding higher accumulated rewards for 𝜖 = 0(greedy).

The greedy policy produces low Q estimates but achieves higher rewards because it does not explore. For higher 𝜖 values, produces better estimates but their corresponding accumulated rewards are lower due to over-exploration and instability.

Figure 2:

A graph of different colored lines

AI-generated content may be incorrect.

Table 2: Average Q-values for 𝛼 = 0.9k

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epsilon-greedy | Average of action value Q(a1) of 100 runs | True action value Q\*(a1) | Average of action value Q(a2) of 100 runs | True action value Q\*(a2) |
| 𝜖 = 0(greedy) | 5.28 | 6 | 0.74 | 7 |
| 𝜖 = 0.1 | 5.10 | 6 | 2.44 | 7 |
| 𝜖 = 0.2 | 4.92 | 6 | 4.28 | 7 |
| 𝜖 = 0.5 (random) | 5.37 | 6 | 6.17 | 7 |

Analysis:

The learning rate 𝛼 decays exponentially over time. While the blue curve (𝜖 = 0) stabilizes at the lowest reward, the red curve (𝜖 = 0.5) rises the fastest and stabilizes at the highest reward as the agent is able to explore and exploit the optimal action a2 effectively.

For action a1, the true action value is 6 and the Q estimates are close across all the 𝜖. For action a2, the true action value is 7 and the Q estimates improve significantly as the 𝜖 increases to 0.5, reflecting sufficient exploration.

Figure 3:

A graph of a graph

AI-generated content may be incorrect.

Table 3: Average Q-values for 𝛼 =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epsilon-greedy | Average of action value Q(a1) of 100 runs | True action value Q\*(a1) | Average of action value Q(a2) of 100 runs | True action value Q\*(a2) |
| 𝜖 = 0(greedy) | 5.59 | 6 | 0.48 | 7 |
| 𝜖 = 0.1 | 5.21 | 6 | 5.87 | 7 |
| 𝜖 = 0.2 | 5.34 | 6 | 6.48 | 7 |
| 𝜖 = 0.5 (random) | 5.83 | 6 | 6.74 | 7 |

Analysis:

It can be observed from the above plot that the green curve (𝜖 = 0.2) rises steadily and quickly and stabilizes at a higher reward as compared to other curves. The red curve (𝜖 = 0.5) rises very quickly due to exploration but stabilizes at lower rewards than 𝜖 = 0.2.

For action a1, the true action value is 6 and the Q estimates remain relatively close across all the 𝜖. For action a2, the true action value is 7 and the Q estimates increase drastically after slight exploration, performing the best at 𝜖 = 0.5 with highest average action value of 6.74.

Since the ultimate goal of our reinforcement learning task is to maximize the rewards, we can say that 𝜖 = 0.2 performs better in terms of accumulated rewards, even though 𝜖 = 0.5 has more accurate Q estimate values and slightly lower accumulated rewards.

Figure 4:

A graph with different colored lines

AI-generated content may be incorrect.

Table 4: Average Q-values for 𝛼 =

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Epsilon-greedy | Average of action value Q(a1) of 100 runs | True action value Q\*(a1) | Average of action value Q(a2) of 100 runs | True action value Q\*(a2) |
| 𝜖 = 0(greedy) | 5.18 | 6 | 0.84 | 7 |
| 𝜖 = 0.1 | 5.15 | 6 | 2.62 | 7 |
| 𝜖 = 0.2 | 5.44 | 6 | 3.79 | 7 |
| 𝜖 = 0.5 (random) | 5.79 | 6 | 6.16 | 7 |

Analysis:

The blue curve (𝜖 = 0) stabilizes at the lowest accumulated reward, followed by the orange (𝜖 = 0.1) and green (𝜖 = 0.2) curve with increased exploration. The red curve (𝜖 = 0.5) achieves highest accumulated rewards because the random policy explores extensively, combined with the decaying 𝛼 leading to slower updates or slow learning.

For action a1, the true action value is 6 and the Q estimates remain relatively close to it across all the 𝜖. For action a2, the true action value is 7 and the Q estimate improves as 𝜖 increases from 0 to 0.5, with highest being 6.16. It can be observed that 𝜖 = 0.5 performs the best, as the random exploration ensures sufficient coverage of the action space, giving enough chances to select and learn that action a2 is the optimal action.

Part B:

Consider: 𝛼 = 0.1 and 𝜖 = 0.1

For each pair of optimistic values 𝑄 = [0 0], 𝑄 = [6 7], 𝑄 = [15 15],

Where Q = [Q(a1) Q(a2)], plot the average accumulated reward w.r.t. step/time and compare the results

Figure 5:

A graph of a number of different colored lines

AI-generated content may be incorrect.

Table 5:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Initial Q values | Average of action value Q(a1) of 100 runs | True action value Q\*(a1) | Average of action value Q(a2) of 100 runs | True action value Q\*(a2) |
| 𝑄 = [0 0] | 0.88 | 6 | 0.99 | 7 |
| 𝑄 = [6 7] | 0.83 | 6 | 1.06 | 7 |
| 𝑄 = [15 15] | 0.94 | 6 | 0.90 | 7 |

Analysis:

Starting with low initial values (blue curve), the agent assumes that the actions are not rewarding so it explores to find the optimal ones and stabilizes at lower accumulated reward whereas, starting with higher initial values (green curve), the agent assumes that there are high rewards for all actions and hence, explores aggressively but stabilizes slowly. It can be observed from the plot that the orange curve, starting with initial values very close to the true values, achieves the highest accumulated rewards, showing efficient convergence.

The orange curve, on the other hand, starts closer to the true values, explores and exploits and stabilizes at comparatively higher accumulated reward.

Part C:

For a fixed 𝛼 = 0.1, use the Gradient-Bandit policy with H1(a1) = H1(a2) = 0

A math equations and formulas

AI-generated content may be incorrect.

Plot: Average accumulated reward w.r.t. step/time. Discuss how the results are different from 𝜖-greedy results with 𝑄(𝑎1) = 𝑄(𝑎2) = 0, 𝛼 = 0.1 and 𝜖 = 0.1

Result:

A graph with blue and orange lines

AI-generated content may be incorrect.

Analysis:

It can be observed from the above plot that the gradient-bandit policy achieves higher accumulated rewards than the epsilon-greedy policy. It maybe because the epsilon-greedy policy has a fixed exploration rate as 𝜖 = 0.1, while the gradient-bandit policy follows a adaptive exploration strategy.