Course Code: CS 5180

Course: Reinforcement Learning and Sequential Decision Making

Name: Pavan Rathnakar Shetty

Please find the entire submission in Canvas and https://github.com/Pavan-r-shetty/Reinforcement-Learning-2023.git as well

EX1 Assignment Submission

Course Code: CS 5180
Course: Reinforcement Learning and Sequential Decision Making
Name: Pawan Rathnakar Shetty

EX1 Assignment Solution

1. Given a K-armed bandit problem using an ε-greedy action selection, an action will be selected at random with probability ε and greedily (based on current action-value estimates) with probability $1-\varepsilon$

To determine on which step ε case definitely occured and on which it may have occured, we follow these steps:

1. Initialize the action value estimate $Q(a) = a$ for all actions.
2. Go through ~~action~~ given sequence of actions and rewards:

- If an action is taken that is not the current greedy action (highest action-value estimate), then the ε case either definitely occured (if the taken action's value estimate is less than current best estimate) or it could have possibly occured (if the taken action's value estimate is the same as the current best estimate).
- Update the action-value estimate for the taken action).

Let's break down the sequence:

1) $A_1 = 1$, $R_1 = -1$

$$Q(a) = \frac{\text{previous sum of reward for action a}}{\text{number of times a has been taken}}$$

$Q(1) = -1$
$Q(2) = 0$
$Q(3) = 0$
$Q(4) = 0$

Here action 1 was definitely taken due to ε case since it resulted in negetive reward and initial values for all action were 0.

2. $A_2 = 2$ $R_2 = 1$

$Q(1) = -1$
$Q(2) = 1$
$Q(3) = 0$
$Q(4) = 0$

Action 2 now has the highest value. It could be greedy action or due to Σ case.

3. $A_3 = 2$ $R_3 = -2$

After this, the average reward for action 2 becomes $\frac{1-2}{2} = -0.5$. It was not the best action at that time ((since $Q(2) = 1$) was best for this step), so this was definitely due to the Σ case.

4. $A_4 = 2$, $R_4 = 2$

After updating, the average reward for action 2 becomes $\frac{1-2+2}{3} = 1/3$. Given this reward, action 2 is now the greedy action since $1/3 > -1$, and it's greater than the other untouched actions which are still 0. Thus, this could have been Σ or greedy.

5. $A_5 = 3$ $R_3 = 0$

$Q(1) = -1$
$Q(2) = \frac{1}{3}$
$Q(3) = 0$
$Q(4) = 0$

Here, action 3 was not the best action at that time (since $Q(2) = \frac{1}{3}$ was the best), so this was definitely Σ case.

Summary: $A_1\ A_3\ A_5$ — Definitely ε case

$A_2\ A_5$ — Possible ε case

2. Given the incremental implementation of sample-average method:

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

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

update

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

substituting to our orignial equation

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

We can continue this recursive expansion for all previous rewards. For each reward R_K before time step n, the weight will be the product of all the "retention factor" after time step K.

The General weight of R_K in Q_{n+1} will be:

$$\text{weight of } R_K = \alpha_K \prod_{j=K+1}^{n} (1-\alpha_j)$$

This product term represents the compounded effect of retaining a fraction of the estimate at each time step from $K+1$ to n.

3.a) The sample average estimate is unbiased. This is because it is essentially the mean of all received rewards up to step n. The expected value of the mean of the sample is the true mean of the population.

So, $E[Q_n] = q^*$

b) Yes, if $Q_1 = 0$ and we are using the exponential recency - weighted average estimate), then Q_n is generally biased of $n > 1$. This is because the estimate is initialized to 0 and then weighs subsequent reward with a factor that makes old rewards have less influence. If the true q^* is not zero, this initial value can bias the estimate especially when n is small.

c) For Q_n to be unbiased, we need $E[Q_n] = q^*$ This condition is met if the initial estimate Q_1 is set to the true value q^* and if all rewards $R_1, R_2 \dots$ are identical and equal to q^*.

d) As n grows larger, the exponential recency-weighted average will give more importance to newer rewards and less to older rewards. Older rewards will have their impact decay exponentially. Given the weights of all rewards sum upto 1, and if rewards are drawn from a stationary distribution, as n becomes very large, the influence of the initial condition and older rewards become negligible making

Q_n approach q_i^*. Therefore, $E[Q_n]$ approaches q_i^* as n approaches infinity, making it assymptotically unbiased.

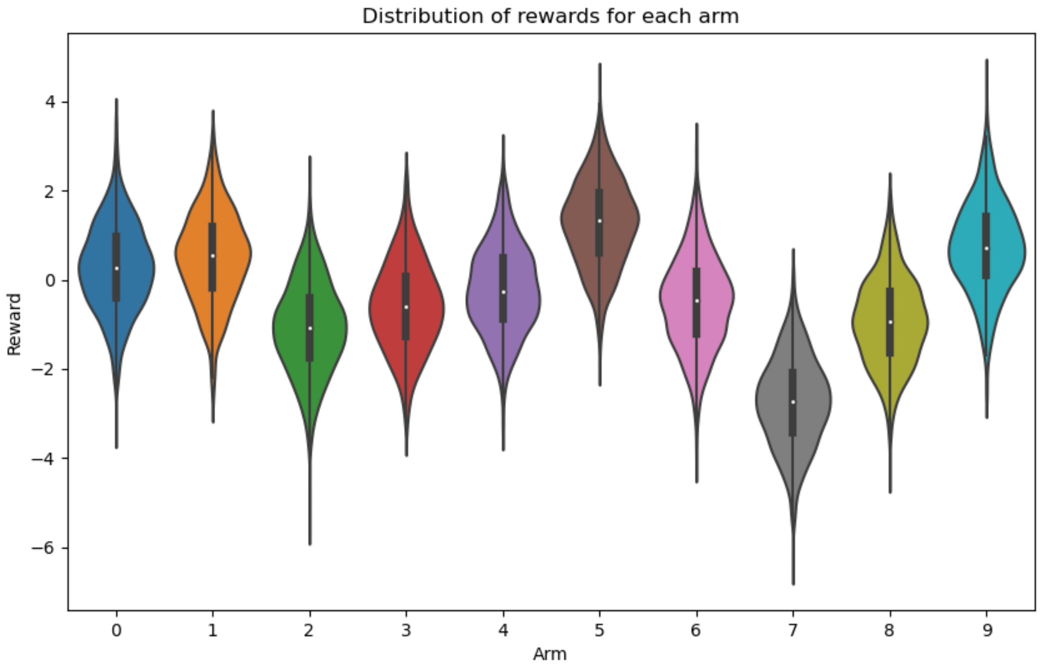
e) The exponential receny-weighted average is designed to give more weight to recent reward than older reward. If the environment of the reward distribution is non stationary, then older reward may not reflect the current state. However, since we are always weighting in the initial value and older rewards, there's an inherent bias introduced by past rewards, especially if they were diffent from current expectation. The exponential weighting ensures that this bias, ~~with~~ while decreasing over time, never completely goes away unless the reward distribution is stationary and initial estimate is accurate.

4Q)

uncomment line 33 in env.py, comment line 32 in env.py

Uncomment line 191, comment line 192, 193 in main.py

Plot:



5Q)

The methods typically compared in this context are:

Greedy method

ε-greedy methods (with various values of ε, such as 0.01 and 0.1)

Here's a breakdown of how these methods perform in the long run:

1. Greedy method:

This method always chooses the action that has the highest estimated value based on the past experiences. It doesn't explore any other actions, so if the initial estimates were off, it might stick to a sub-optimal action indefinitely.

2. ε-greedy methods:

This method will choose the action with the highest estimated value most of the time, but occasionally (with probability ε), it will choose an action randomly. This introduces a balance between exploration (trying out new actions) and exploitation (using the best known action).

Predicting the asymptotic behavior:

Cumulative Reward:

In the long run, ε-greedy methods tend to outperform the strict greedy method in cumulative reward because they continue to explore and can thus correct any initial incorrect assessments of action values. A greedy method might get stuck with a sub-optimal action if its initial estimates are misleading.

Among the ε-greedy methods, a smaller ε might perform better in the long run in terms of cumulative reward because it explores less frequently and exploits the best known action more often. But it's essential to strike a balance: too small an ε, and you're closer to a strict greedy strategy; too large an ε, and you're acting almost randomly.

Probability of Selecting the Best Action:

Again, ε-greedy methods have a better chance of consistently selecting the best action in the long run because of their exploration component. The greedy method may or may not find and stick to the best action, depending on its initial experiences.

Epsilon (ε) = 0 (Greedy method)

This method purely exploits the current best known action, without exploring other actions.

It seems that the average reward plateaus quickly at 1 and remains consistent thereafter.

Asymptotic Performance:

It will be 1 as time (or steps) goes to infinity.

Epsilon (ε) = 0.01

This method occasionally (1% of the time) explores random actions.

The reward is increasing slowly as the number of steps increases, indicating that this method benefits slightly from exploration.

Asymptotic Performance:

Given the trend, it's increasing but at a decreasing rate. It's unclear what the exact asymptotic performance is but it's going to be greater than 1.35 as time (or steps) goes to infinity.

Epsilon (ε) = 0.1

This method explores random actions 10% of the time.

After 250 steps, the reward increased significantly, and after 500 steps, it appears to plateau at 1.5.

Asymptotic Performance:

It will be 1.5 as time (or steps) goes to infinity.

Summary:

Greedy (ε=0): The asymptotic performance is 1.

ε=0.01: The asymptotic performance will be greater than 1.35, but the exact value isn't clear from the data.

ε=0.1: The asymptotic performance is 1.5.

Method with ε=0.1 will perform the best in the long run with an asymptotic performance of 1.5. This is followed by the ε=0.01 method (greater than 1.35 but exact value is not clear) and the pure greedy method with ε=0 (performance of 1).

6Q)

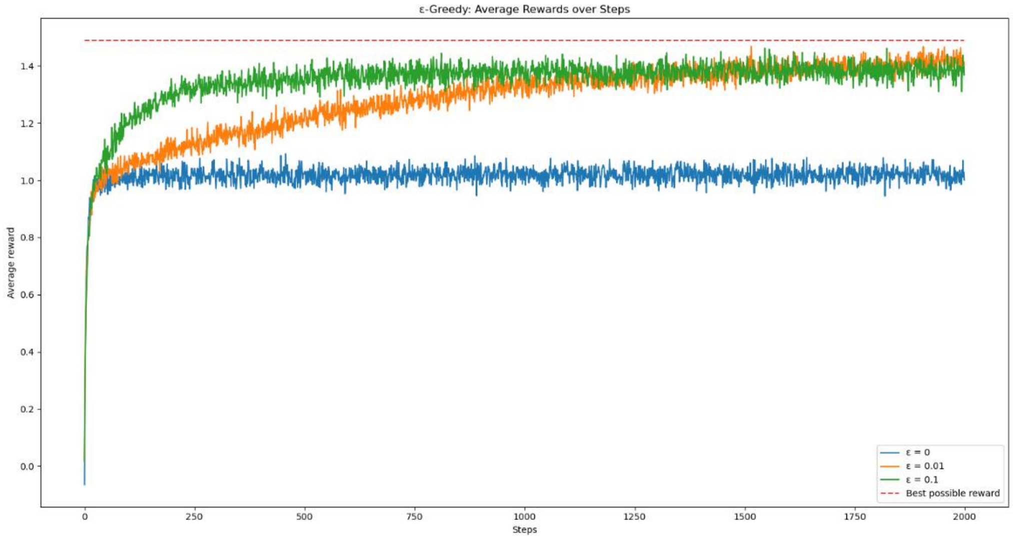
Uncomment line 192 and comment line 191, 193 in main.py

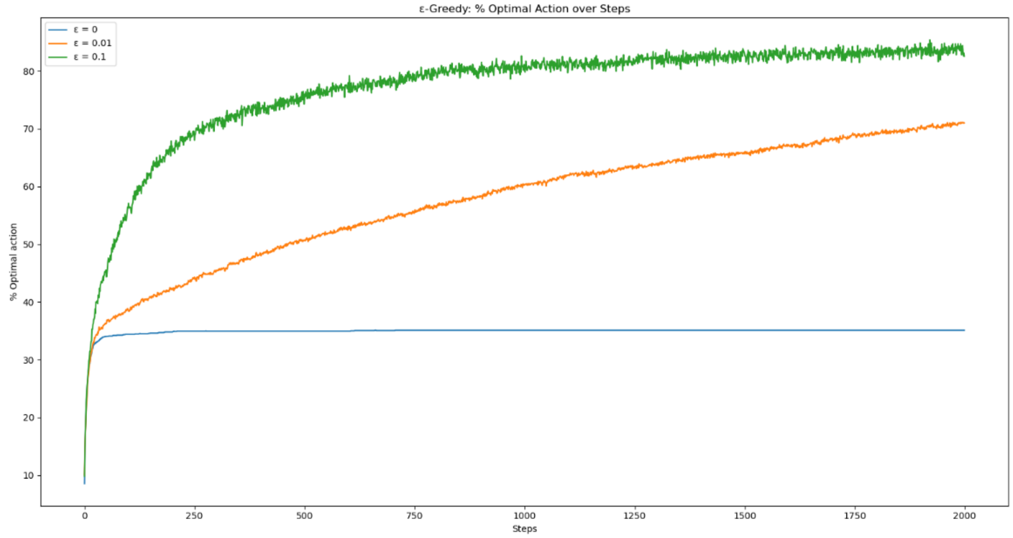
Comment line 33. Uncomment line 32 in env.py

Why is this the appropriate upper bound?

The reason for using **maxaq*(a)** as the upper bound is that this value represents the best average performance one could achieve if they always pulled the best arm (i.e., the arm with the highest expected reward) every single time. In a multi-armed bandit problem, the goal is often to find and exploit the best arm as quickly as possible to maximize the cumulative reward. By having this line on the plot, you're effectively saying: "This is the best average reward we could hope to get if we always made the optimal choice."

Plot:

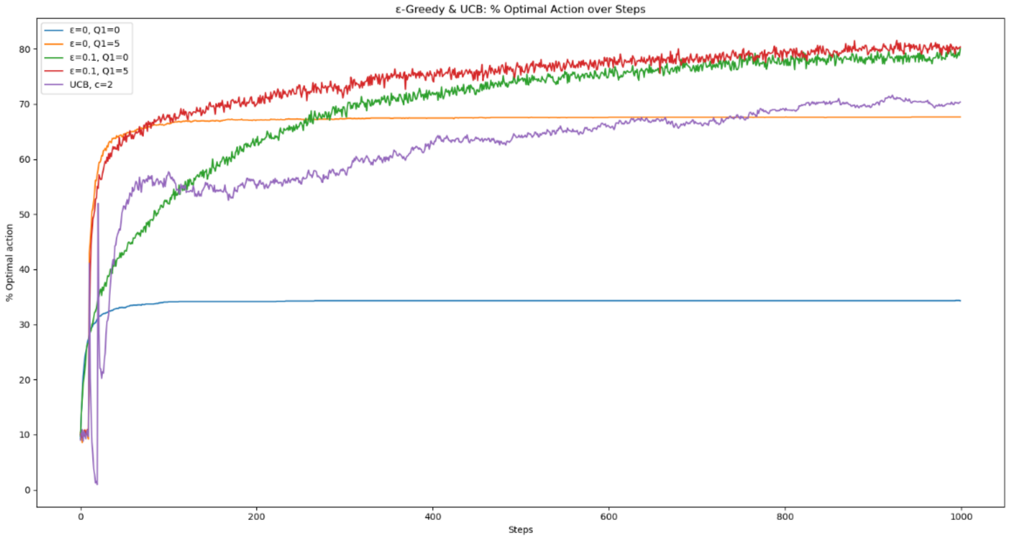


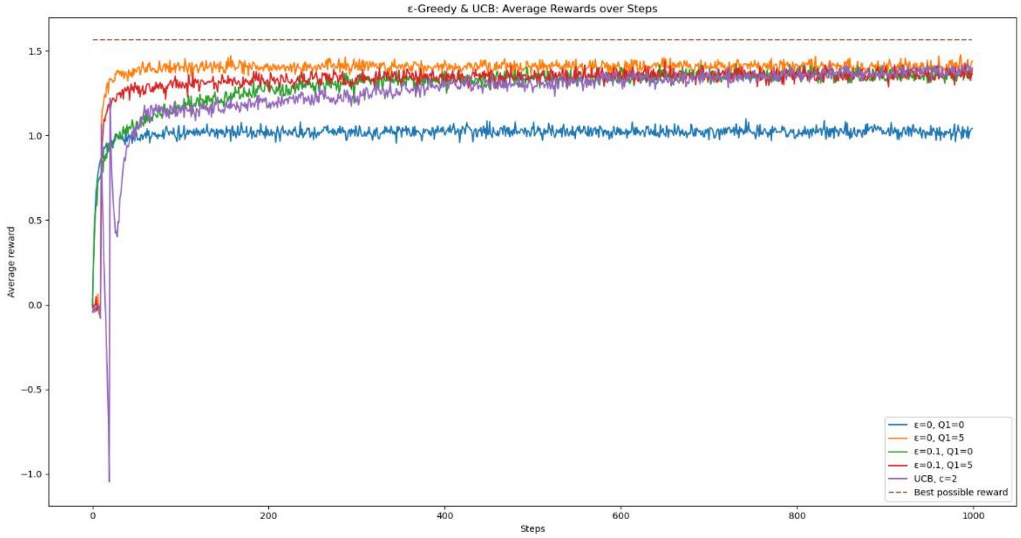


Written: Yes

7Q)

Plot:





Written:

The spikes observed in the beginning when using optimistic initialization and UCB can be understood by examining the underlying principles of both methods:

Optimistic Initialization:

Principle: The idea behind optimistic initialization is to set the initial Q-values of all arms to a value that is optimistically high. This high initial value encourages exploration of all the arms in the initial phases because every arm appears to be better than it might be in reality.

Sharp Increase: In the initial stages, because of optimistic values, the agent is trying out different arms. Since the rewards are not always the true average, sometimes the agent might get high rewards, leading to a spike.

Sharp Decrease: As the agent learns more about the true value of each arm, the expected reward starts to converge to the true average reward. If the optimistic value was set higher than the actual best arm's average reward, there will be a sharp decrease as the agent's estimates become more realistic and less optimistic.

UCB (Upper Confidence Bound):

Principle: UCB takes into account both the average reward of an arm and the uncertainty in that estimate. It ensures that arms that have not been tried as often are given a chance because the uncertainty term will be high for those arms.

Sharp Increase: In the initial stages, all arms have high uncertainty. As a result, the agent tends to try different arms, leading to a varied range of rewards. Sometimes, it might hit a streak of high-rewarding arms, leading to a spike.

Sharp Decrease: As the agent pulls arms, the uncertainty for those arms decreases, and the Q-value starts to converge to the true value. This means that spikes due to uncertain high-reward arms will decrease, leading to a more consistent average reward.

Analyzing Experimental Data:

To provide empirical evidence from your experimental data:

Frequency of Arm Pulls: One can analyze how often each arm was pulled in the initial phases. For both optimistic initialization and UCB, you should observe that there's a more uniform distribution across all arms in the initial stages as the agent is exploring more. This is in contrast to ε-greedy where one arm might dominate if it gave a high reward initially.

Average Reward Over Time: Plot the average reward for each arm over time. For methods with optimistic initialization, you should see that the average reward starts high but then converges down to the true average reward. For UCB, there should be more variance in the initial stages as the agent tries different arms, leading to occasional spikes in rewards.

Variance/Standard Error: As mentioned in the task, plotting confidence bands can also provide insights. A wider band in the initial stages indicates more exploration and uncertainty in rewards. As the agent learns more, the confidence band should become narrower, indicating more confidence in the estimated Q-values.

In conclusion, the spikes in the initial stages are a manifestation of the agent's exploration strategies. They showcase the agent's attempt to learn about each arm's true reward, driven either by optimistic initialization or the uncertainty principle in UCB. Over time, as the agent learns more, these strategies result in a more stable reward estimate, leading to the observed sharp decreases.