CS5180 Reinforcement Learning

Exercise 1: Multi-Armed Bandits

Anway Shirgaonkar

Q1. Exploration vs Exploitation

k-armed bandit problem $k = 4$

Action	Reward	$Q_n(a)$			
		1	**2**	**3**	**4**
-	-	0	0	0	0
1	-1	-1	0	0	0
2	1	-1	1	0	0
2	-2	-1	-0.5	0	0
2	2	-1	0.3	0	0
3	0	-1	0.3	0	0

A1: Since for the first action, the estimates $Q(a)$ are all set to 0, the greedy action would select the next action at random (breaking ties randomly). But the ϵ case may have occurred in this step

A2: It is highly likely that the algorithm chose a greedy action here. But the ϵ case may have occurred as well

A3: It is highly likely that the algorithm chose a greedy action here. But the ϵ case may have occurred as well

A4: It can be said with certainty that the ϵ case occurred at this step. Since the greedy algorithm would have chosen arm 3 or 4

A5: It can be said with certainty that the ϵ case occurred at this step. Since the greedy algorithm would have chosen arm 2

Q2. Varying step-size weights

Let the non-constant step size parameter be α_n, the reward R_n and the estimate Q_n
Now, the current estimate Q_n is given by:

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha_n(R_n - Q_n) \\
&= Q_n + \alpha_n R_n - \alpha_n Q_n \\
&= \alpha_n R_n + (1 - \alpha_n) Q_n
\end{aligned}
$$

expanding Q_n further, we get:

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

If we group these terms together, we get:

$$\boxed{Q_{n+1} = \sum_i^n \alpha_n R_n - R_{n-1}\prod_i^n \alpha_n + Q_1\prod_i^n (1-\alpha_n)}$$

Q4. Predicting Asymptotic Behavior in Figure 2.2

The ϵ-greedy method with $\epsilon = 0.01$ would perform better both in terms of cumulative reward and probability of selecting the best action. This is because the probability of selecting the greedy action in $\epsilon = 0.01$ is:

$$P(greedy|\epsilon = 0.01) = 0.99 + 0.01 * \frac{1}{10}$$

$$\boxed{P(greedy|\epsilon = 0.01) = 0.991}$$

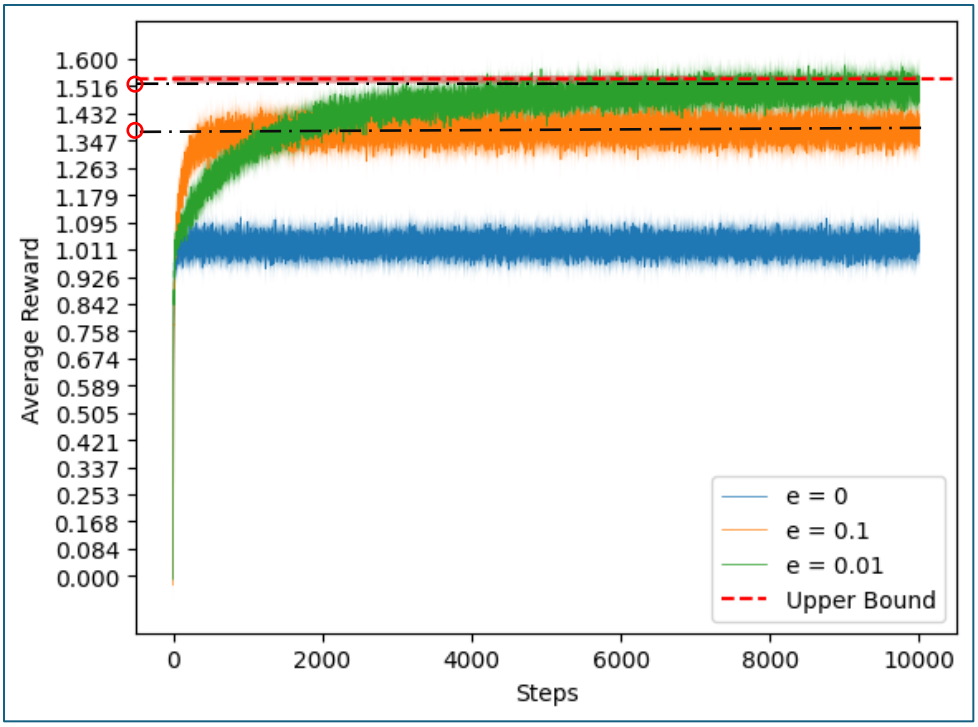
Now, the probability of selecting the greedy action in $\epsilon = 0.1$ is:

$$P(greedy|\epsilon = 0.1) = 0.90 + 0.1 * \frac{1}{10}$$

$$\boxed{P(greedy|\epsilon = 0.1) = 0.91}$$

Hence, the ϵ-greedy method with $\epsilon = 0.01$ is 10% more likely to choose the greedy action.

Hence it is clear that the ϵ-greedy method with $\epsilon = 0.01$ would perform better both in terms of cumulative reward and probability of selecting the best action. The ϵ-greedy method with $\epsilon = 0.1$ initially explores more, leading to a faster rise in curve of average rewards, but $\epsilon = 0.01$ would surpass this curve as $t \to \infty$. This is verified in the plot below with 2000 trials and 10000 steps:



Q5. Reproducing figure 2.2 (Written Part)

Do the averages reach the asymptotic levels predicted in the previous question?

Yes, the averages predicted in the previous question reach the asymptotic levels as denoted in the figure above. Now the mean of the maximum $q_*(a)$ for 2000 trials is 1.536 (determined empirically)

as $time \rightarrow \infty$,

the ϵ-greedy method with $\epsilon = 0.01$ will accumulate $1.536 * 0.991 = 1.5221$ average reward

And the ϵ-greedy method with $\epsilon = 0.1$ will accumulate $1.536 * 0.91 = 1.397$ average reward

Q6. Bias in Q-value estimates

a. The equation 2.1 is:

$$Q_t(a) \doteq \frac{\text{sum of rewards when } \mathbf{a} \text{ taken prior to t}}{\text{number of times } \mathbf{a} \text{ taken prior to t}}$$

In the event of $t \rightarrow \infty$, it is guaranteed that the estimated value Q_t will converge to the true value $q_*(a)$ by the law of large numbers, and since the $\mathbb{E}[Q_t(a)] = q_*(a)$, $Q_t(a)$ is an unbiased estimate

b. Considering the exponential recency-weighted average estimate, according to:

$$Q_{n+1} = (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

If $Q_1 = 0$,

$$Q_{n+1} = \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

It is evident from the equation that if $Q_1 = 0$, the weight, $\alpha(1-\alpha)^{n-i}$, given to the reward R_i depends on how many rewards ago, $n-i$, it was observed. The quantity $1-\alpha$ is less than 1, and thus the weight given to R_i decreases as the number of rewards increases.

$$E[Q_{n+1}] = \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} E[R_i]$$

For Q_n to be unbiased $\sum_{i=1}^{n} \alpha(1-\alpha)^{n-i}$ should be equal to 1, and that won't be possible in this case considering n does not tend to infinity. Hence Q_n is biased for $Q_1 = 0$

c. From the equation:

$$Q_{n+1} = (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

$E[Q_{n+1}] = q_*$ if Q_n is unbiased

Taking the expectation of the above equation:

$$E[Q_{n+1}] = (1-\alpha)^n E[Q_1] + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} E[R_i]$$

$$q_* = (1-\alpha)^n E[Q_1] + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i}\, q_*$$

Now, Q_n will be unbiased if and only if:
$Q_1 = 0$, and $\sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} = 1$

d. To show that Q_n is asymptotically unbiased,

$$Q_{n+1} = (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$$

Taking the expectation of the above equation:

$$E[Q_{n+1}] = (1-\alpha)^n E[Q_1] + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} E[R_i]$$

Now,

$$\sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} = \alpha \sum_{i=1}^{n} (1-\alpha)^{n-i} = 1 - (1-\alpha)^n$$

Therefore as $n \to \infty$,

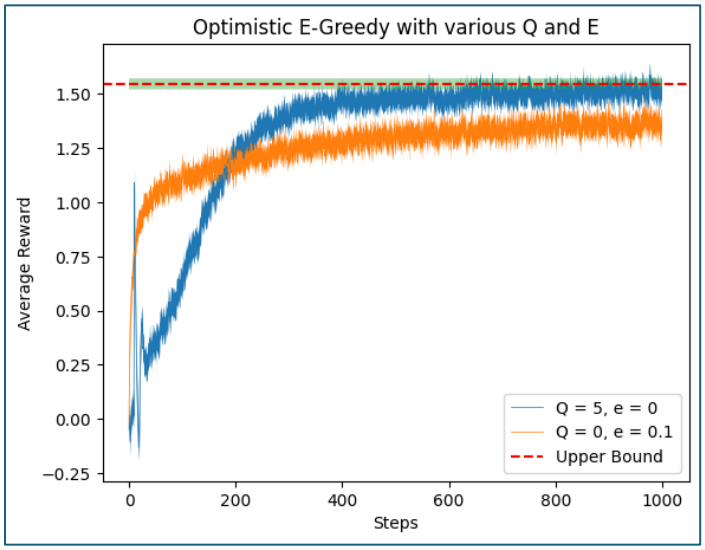
$$1 - (1-\alpha)^n \to 1 \quad and \quad (1-\alpha)^n \to 0$$

Which implies:
$E[Q_{n+1}] \to q_* \quad as \quad n \to \infty$ hence Q_n is asymptotically unbiased.

e. The exponential recency weighted average will be biased in general. As we proved in part d of the question that exponential recency weighted average is asymptotically unbiased. In most general cases, n does not approach infinity, and hence we can say that it will generally be biased

Q7. Reproducing and supplementing Figures 2.3 and 2.4 (Written Part)

- In the case of optimistic greedy, the initial estimate $Q_n(a)$ is set highly optimistic, meaning it is way above the actual maximum $q_*(a)$.
- The algorithm will begin with choosing any action randomly, since all have equal estimate. Then it would be "disappointed" to receive a low reward, which lowers the estimated $Q_n(a)$ of that action.
- The algorithm would now cycle through all of the actions in the next steps, i.e. basically exploring all the actions
- The change in the estimate $Q_n(a)$ of the optimal action would be the least after this exploration. Now the algorithm would choose the optimal action, since it has the highest value in $Q_n(a)$. This would result in a sharp increase in the average rewards.
- However, pulling the optimal lever over and over would quickly decrease the value of $Q_n(a)$ for it, which encourages exploration among other levers, leading to a sharp decrease in average rewards.
- This phenomenon can be observed in the plot below.



- In the case of UCB, it is the same case. Initially, the "uncertainty" value $\sqrt{\frac{\ln(t)}{N_t(a)}}$ is close to infinity when $N_t(a) = 0$. That is, initially, UCB explores a lot, and finds the optimal action. Once the optimal action is found, it will pull it until the uncertainty term for other actions grows larger and encourages exploration again.
- This phenomenon can be observed in the UCB plot below: