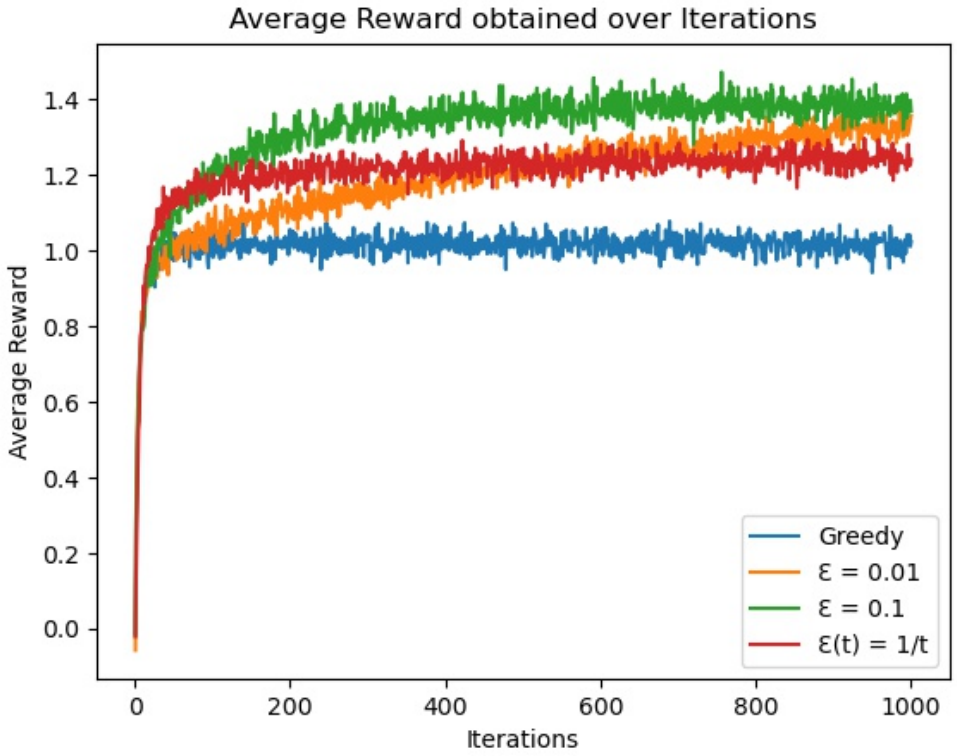
Reinforcement Leaning Homework 1

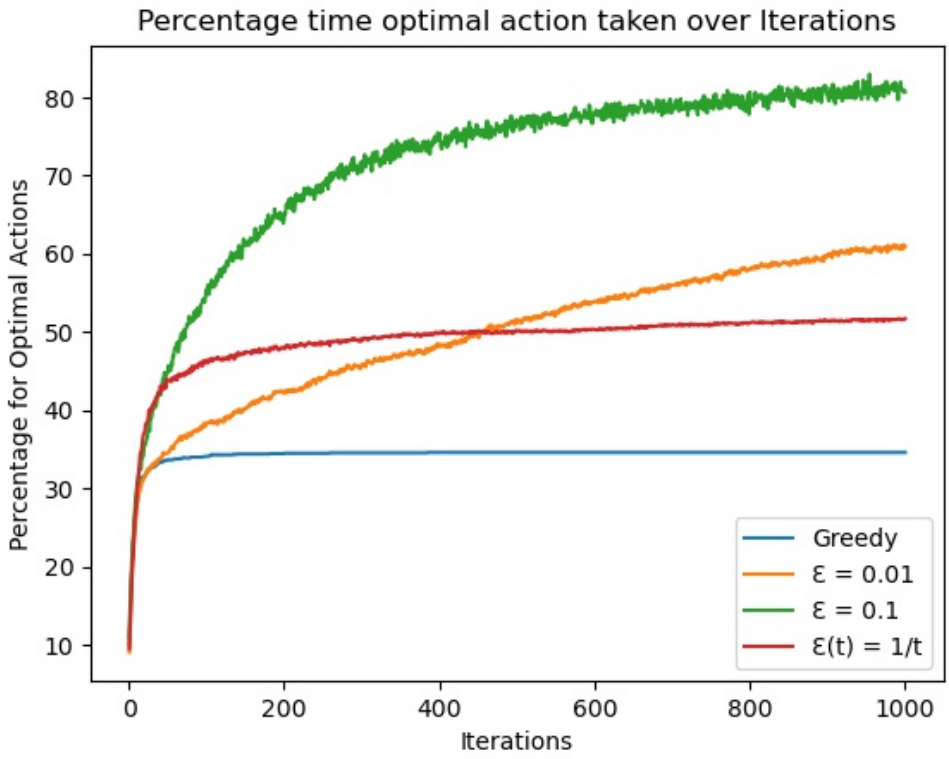
Devansh Gupta

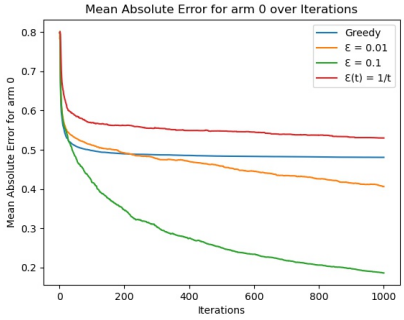
2019160

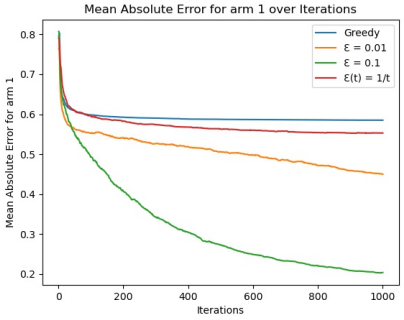
1 Answer 1

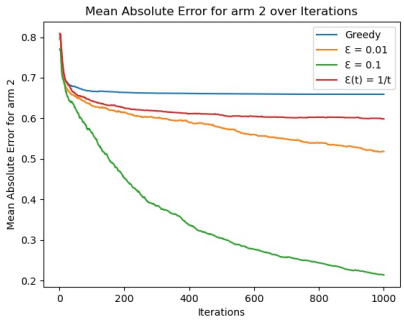
We have chosen $\epsilon(t) = \frac{1}{t}$ for the variable epsilon part of the plots as it satisfied both the equations in Eq. 2.7 mentioned in the book.

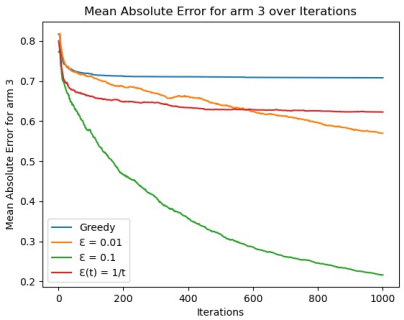


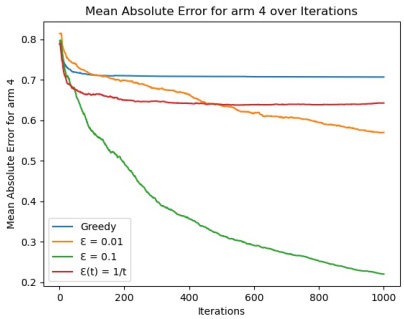

Percentage time optimal action taken over Iterations
Percentage for Optimal Actions
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

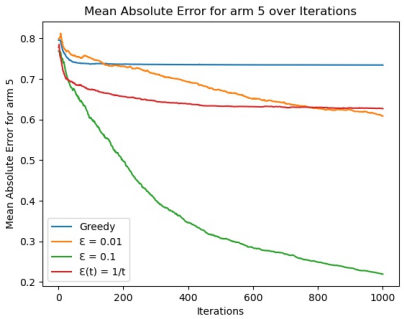

Mean Absolute Error for arm 0 over Iterations
Mean Absolute Error for arm 0
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

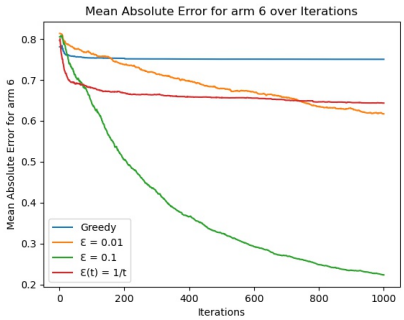

Mean Absolute Error for arm 1 over Iterations
Mean Absolute Error for arm 1
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

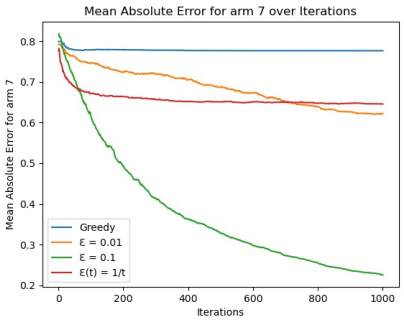

Mean Absolute Error for arm 2 over Iterations
Mean Absolute Error for arm 2
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

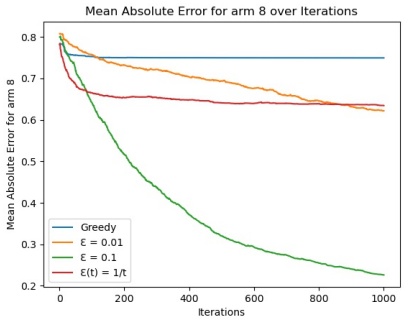

Mean Absolute Error for arm 3 over Iterations
Mean Absolute Error for arm 3
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

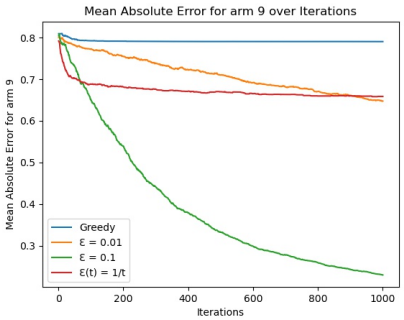

Mean Absolute Error for arm 4 over Iterations
Mean Absolute Error for arm 4
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t


Mean Absolute Error for arm 5 over Iterations
Mean Absolute Error for arm 5
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

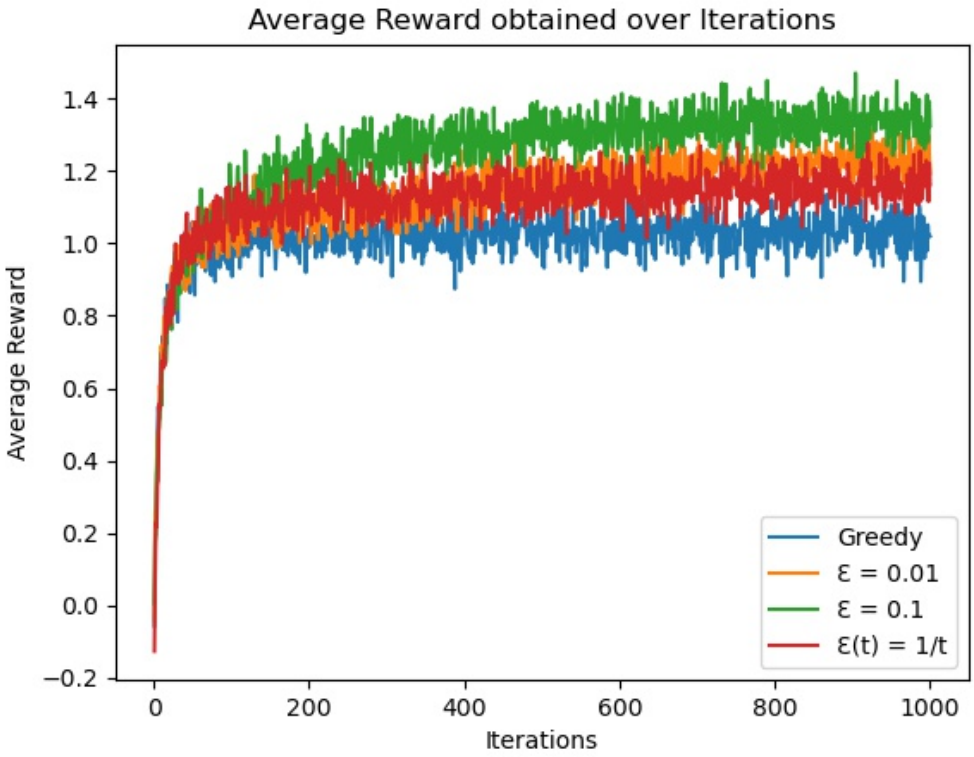

Mean Absolute Error for arm 6 over Iterations
Mean Absolute Error for arm 6
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

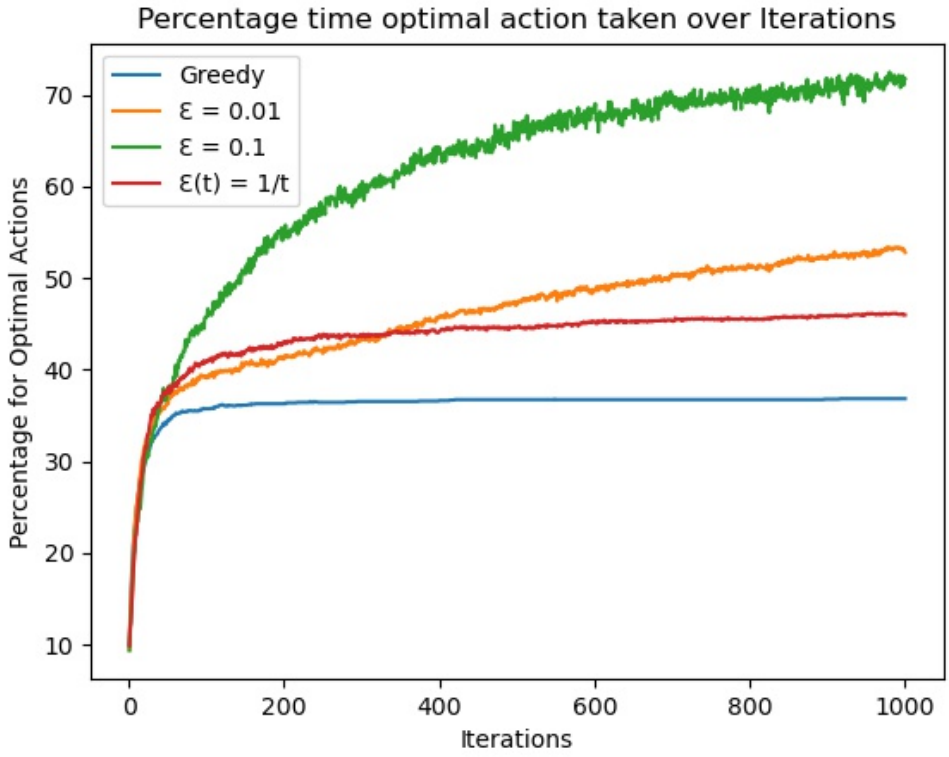

Mean Absolute Error for arm 7 over Iterations
Mean Absolute Error for arm 7
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

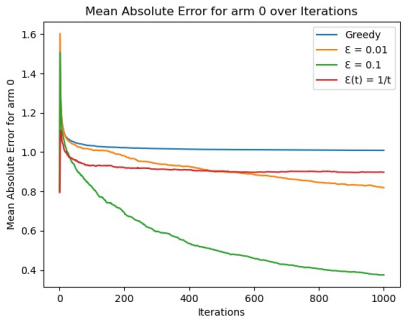

Mean Absolute Error for arm 8 over Iterations
Mean Absolute Error for arm 8
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

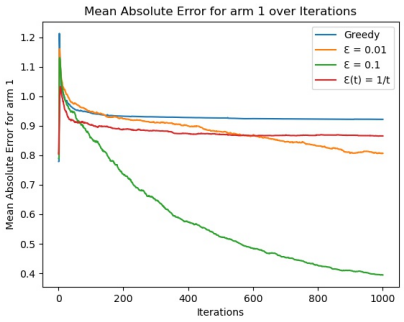

Mean Absolute Error for arm 9 over Iterations
Mean Absolute Error for arm 9
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

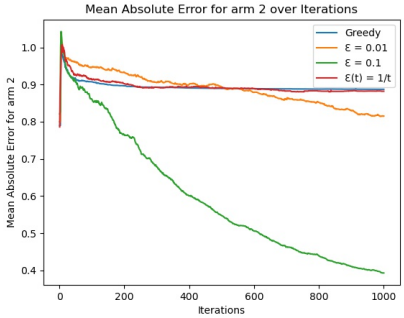
2 Answer 2

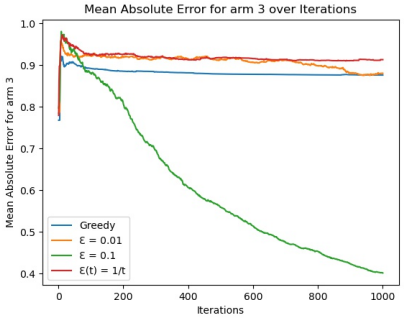

Average Reward obtained over Iterations
Average Reward
Iterations
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
0
200
400
600
800
1000
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

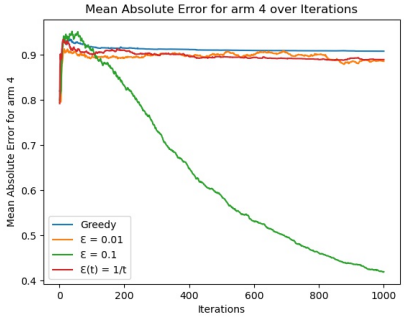

Percentage time optimal action taken over Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t
Percentage for Optimal Actions
Iterations
10
20
30
40
50
60
70
0
200
400
600
800
1000

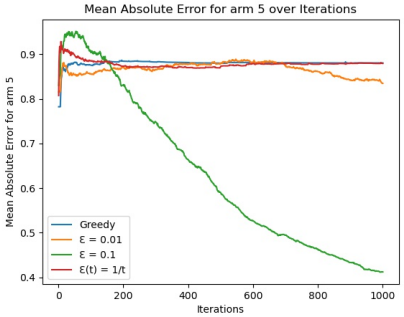

Mean Absolute Error for arm 0 over Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t
Mean Absolute Error for arm 0
Iterations
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0
200
400
600
800
1000

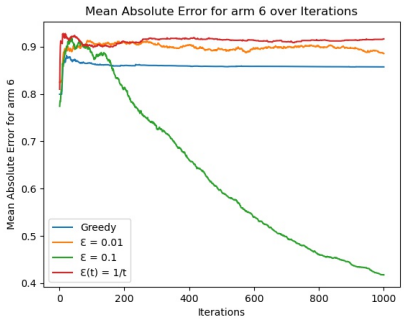

Mean Absolute Error for arm 1 over Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t
Mean Absolute Error for arm 1
Iterations
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
0
200
400
600
800
1000

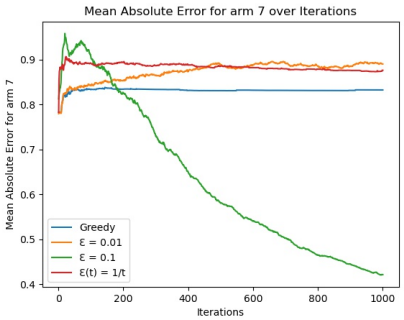

Mean Absolute Error for arm 2 over Iterations
Mean Absolute Error for arm 2
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

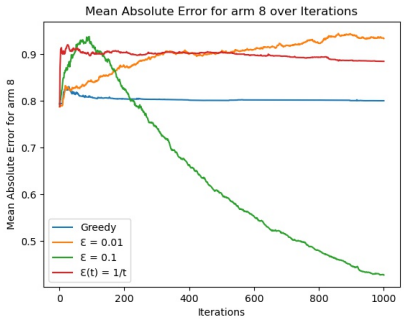

Mean Absolute Error for arm 3 over Iterations
Mean Absolute Error for arm 3
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

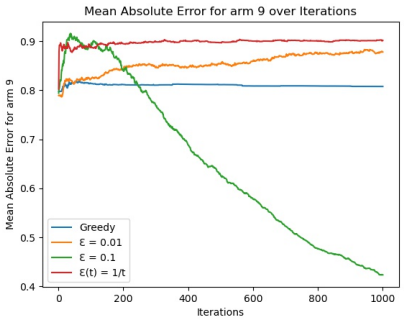

Mean Absolute Error for arm 4 over Iterations
Mean Absolute Error for arm 4
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t


Mean Absolute Error for arm 5 over Iterations
Mean Absolute Error for arm 5
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t


Mean Absolute Error for arm 6 over Iterations
Mean Absolute Error for arm 6
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t


Mean Absolute Error for arm 7 over Iterations
Mean Absolute Error for arm 7
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t


Mean Absolute Error for arm 8 over Iterations
Mean Absolute Error for arm 8
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t


Mean Absolute Error for arm 9 over Iterations
Mean Absolute Error for arm 9
Iterations
Greedy
ε = 0.01
ε = 0.1
ε(t) = 1/t

3 Answer 3

For the above given options for ϵ, we can claim that in the long run the greedy method would give the best results in terms of probability of selection of the best action assuming that the sample mean would be able to estimate $q^*(a)$ in the long run and hence the action which picks the optimal action with maximum probability would be preferred in this scenario. For any ϵ, the probability of picking the optimal action in the long run would be $1 - \epsilon + \frac{\epsilon}{|A|}$ where $|A|$ is the number of arms.

Therefore, for the greedy action, $\epsilon = 0$ the probability of selection of the optimal action is 1. While for $\epsilon = 0.1$, assuming 10 arms, we get the probability to be 0.91, for $\epsilon = 0.01$, we get the probability of 0.991, and for $\epsilon = \frac{1}{t}$, we get the probability of $1 - \frac{9}{10t}$ and in the long run, when $t \to \infty$, the probability approaches 1 but doesn't exactly become 1 but the greedy action gives this result.

Through the lens of the cumulative reward, we can also say that the above values of ϵ would do good. We calculate the expected reward of an iteration with the given Q-values as specified in equation (1), and assume that $Q \to q*$ when approaching infinity. We can see that after substituting various values of ϵ we can see that a greedy method would deem fit for the maximum cumulative reward in the long run mathematically. $\epsilon = \frac{1}{n}$, would be a similar case but like the argument above, this ϵ approaches 1 while greedy already becomes 1, hence increasing the upper bound of the cumulative reward in the long run.

$$E[R_n] = \left(1 - \epsilon + \frac{\epsilon}{|A|}\right) max_{a \in A}(Q_n(a)) + \sum_{a \neq argmax_{a \in A}(Q_n(a))} \frac{\epsilon}{|A|}(Q_n(a)) \quad (1)$$

But, as a side-note on practical settings, we can say that the greedy method does not fare well since an initial bias may fix an action for selection for the future rewards which may be a sub-optimal solution for our setting and hence we can say that it roughly has a probability of selecting the optimal arm in the long run with a probability of $\frac{1}{|A|}$ which becomes exact if there are no negative rewards and initial estimates are equal for all arms. Using an initialization method like optimistic initial values, can helping in exploring more in the greedy method. Hence, we can see in the above plots that the agents which explore a bit more perform better since there is not an inherent bias in the method to select the arm with a positive initial reward, thus helping it get to estimate more accurate Q-values thus helping it making a more informed decision.

4 Answer 4

The recursive expression for sample mean is defined in equation (2) where n is the number of times action A has been chosen and R_n is the reward obtained by the agent on choosing action A.

$$Q_{n+1}(A) = Q_n(A) + \frac{1}{n}(R_n - Q_n(A)) \tag{2}$$

Now if we substitute n = 1 in equation (2), we get

$$Q_2(A) = Q_1(A) + \frac{1}{1}(R_1 - Q_1(A))$$

$$Q_2(A) = Q_1(A) + (R_1 - Q_1(A))$$

$$Q_2(A) = R_1$$

Thus we can see that $Q_2(A)$ has no dependence on $Q_1(A)$, thus ensuring that there is no dependence of $Q_1(A)$ on $Q_n(A)$ because of a recurrence in equation (2).

In case of a constant step size say α, we use the equation (3), where R_n is the reward obtained by the agent on choosing action A.

$$Q_{n+1}(A) = Q_n(A) + \alpha(R_n - Q_n(A)) \tag{3}$$

Now, we can rewrite the equation (3) as given below

$$Q_{n+1}(A) = (1-\alpha)Q_n(A) + \alpha R_n$$

Then, we can simply use this recursive expression to get the equation in terms of previous rewards and possibly the initial estimated expectation.

$$Q_{n+1}(A) = (1-\alpha)((1-\alpha)Q_{n-1}(A) + \alpha R_{n-1}) + \alpha R_n$$

$$Q_{n+1}(A) = (1-\alpha)^2 Q_{n-1}(A) + (1-\alpha)\alpha R_{n-1} + \alpha R_n$$

$$Q_{n+1}(A) = (1-\alpha)^2((1-\alpha)Q_{n-2}(A) + \alpha R_{n-2}) + (1-\alpha)\alpha R_{n-1} + \alpha R_n$$

$$Q_{n+1}(A) = (1-\alpha)^3 Q_{n-2}(A) + (1-\alpha)^2\alpha R_{n-2} + (1-\alpha)\alpha R_{n-1} + \alpha R_n$$

Thus, we can see a pattern emerging which will eventually summarize to (4)

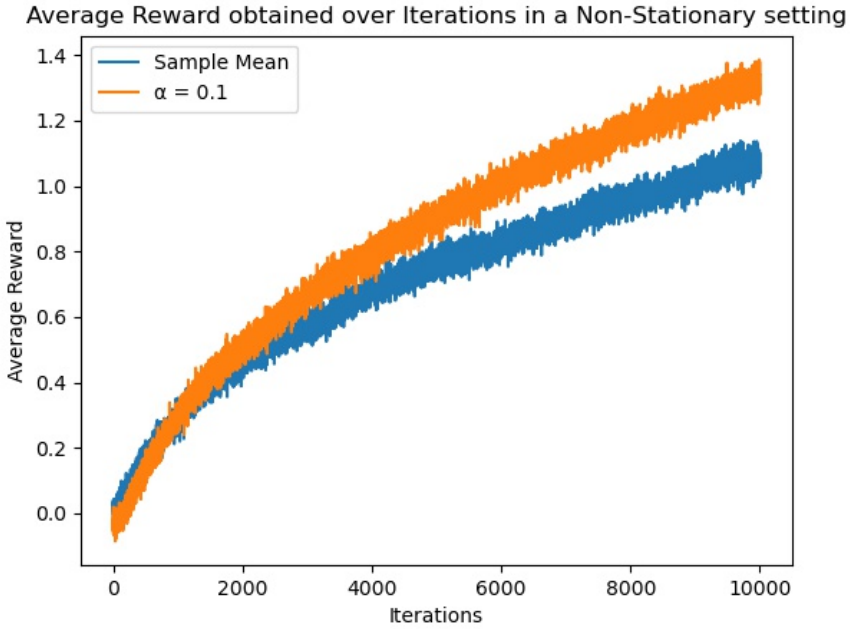
$$Q_{n+1}(A) = (1-\alpha)^n Q_1(A) + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i \tag{4}$$

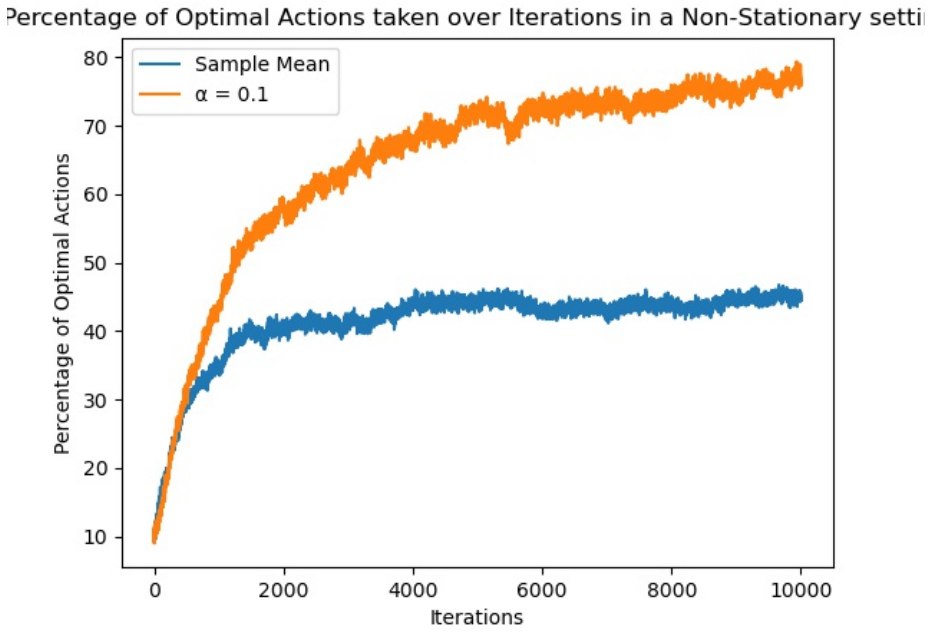
We can clearly see that there is a dependence of $Q_{n+1}(A)$ on the initial value $Q_1(A)$ in the final expression and hence can conclude that $Q_{n+1}(A)$ is a function of $Q_1(A)$.

The dependence of $Q_{n+1}(A)$ on $Q_1(A)$ is measured by the coefficient of $Q_1(A)$ in equation (4), i.e. $(1-\alpha)^n$. Therefore, if the value of $(1-\alpha)^n$ is larger, that would imply a larger dependence of $Q_{n+1}(A)$ on $Q_1(A)$, hence a smaller value of α would increase the dependence.

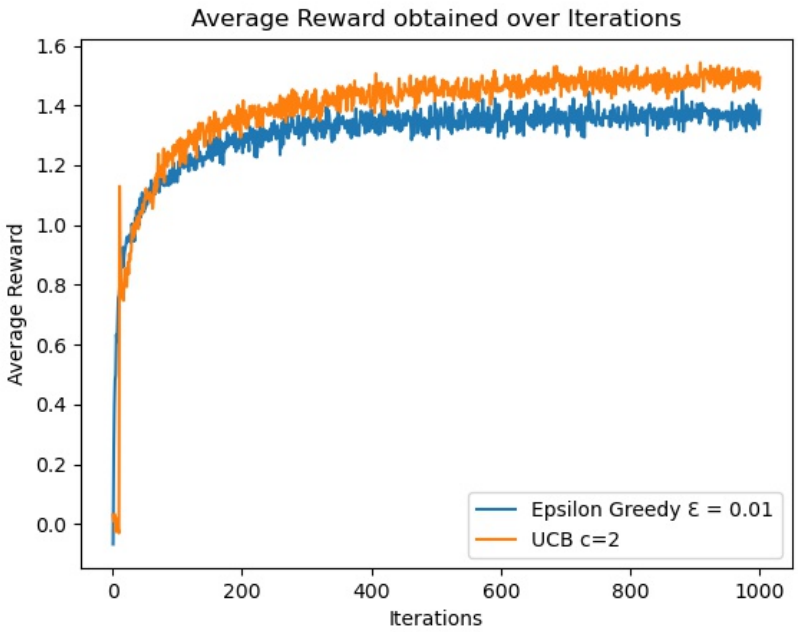
A method when the final estimate will not depend on the initial value is when the value of $(1-\alpha)^n$ is equal to zero. Hence, the value of the step size must be equal to 1, in order to ensure no dependence on the initial value for a constant step size value estimate.

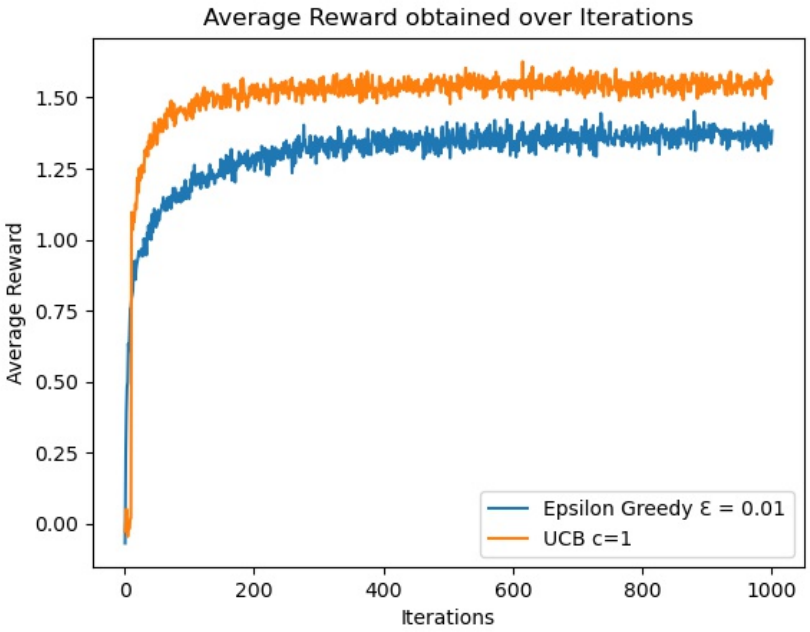
5 Answer 5


Average Reward obtained over Iterations in a Non-Stationary setting
Sample Mean
α = 0.1
Average Reward
Iterations
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
0
2000
4000
6000
8000
10000


Percentage of Optimal Actions taken over Iterations in a Non-Stationary setti
Sample Mean
α = 0.1
Percentage of Optimal Actions
Iterations
10
20
30
40
50
60
70
80
0
2000
4000
6000
8000
10000

6 Answer 6


Average Reward obtained over Iterations
Average Reward
Iterations
Epsilon Greedy ε = 0.01
UCB c=2
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
0
200
400
600
800
1000

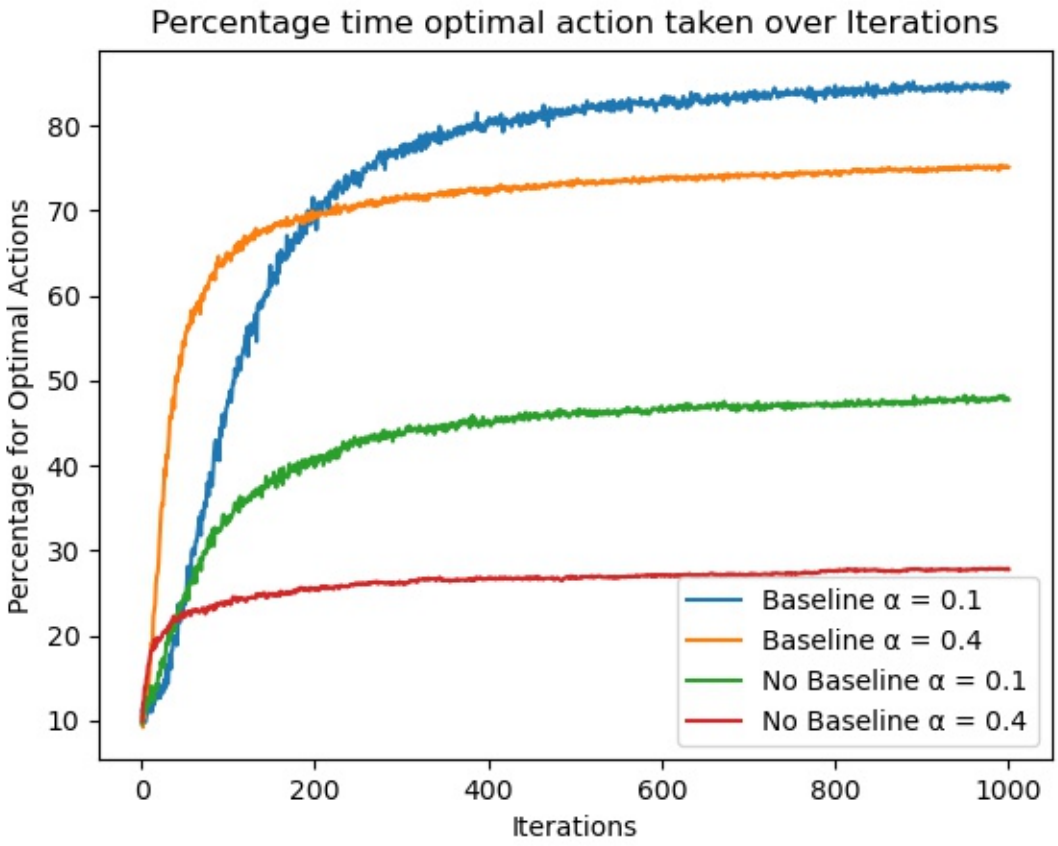

Average Reward obtained over Iterations
Average Reward
Iterations
Epsilon Greedy ε = 0.01
UCB c=1
0.00
0.25
0.50
0.75
1.00
1.25
1.50
0
200
400
600
800
1000

The spikes are present on the curve due to the initial exploratory nature of the UCB algorithm.

$$c\sqrt{\frac{ln(t)}{N_t(a)}} \tag{5}$$

In the initial stages $N_t(a) = 0$ for all the arms of the bandit, thus making each action as a maximizing action and selecting each arm in the first 10 steps to get the first estimate of the rewards. Then, it selects a greedy action based on the previous estimate at the 11th step since the value of the expression in (5) remains constant for all the actions on this step thus spiking the reward on the 11th step on an average, then the value decreases again because the value of $N_t(a)$ becomes 2 for the previously selected arm(which was greedy) and thus decreasing the entire sum of the UCB value for that arm compared to the other arms where $N_t(a)$ is still 1. The constant c in expression (5) controls the drop of the value because it is the constant which controls the contribution of the value of expression (5). Hence for a higher value of c, the drop becomes more prominent for the greedy action and forcing exploration, thus giving a greater dip of reward on an average over the runs right after the 11th step.

7 Answer 7



8 Conclusion

For the code of Answers 1, 2, 5, 6, and 7, please refer to the link `https://github.com/devanshgupta160/RL_Assignments_M2021_2019160`.