# RL- HW 1: Multi Armed Bandits

## Q1

Generating the figure 2.2 of the book as it is, with $\varepsilon = 0{,}0.01{,}0.1$ . All the arms have a normal distribution with variance 1 and mean chosen randomly from a gaussian distribution of mean 0 and variance 1.

Average performance of ε-greedy action-value methods on the 10-armed testbed. The following plots show the data averaged over 2000 runs of 1000 timesteps each.

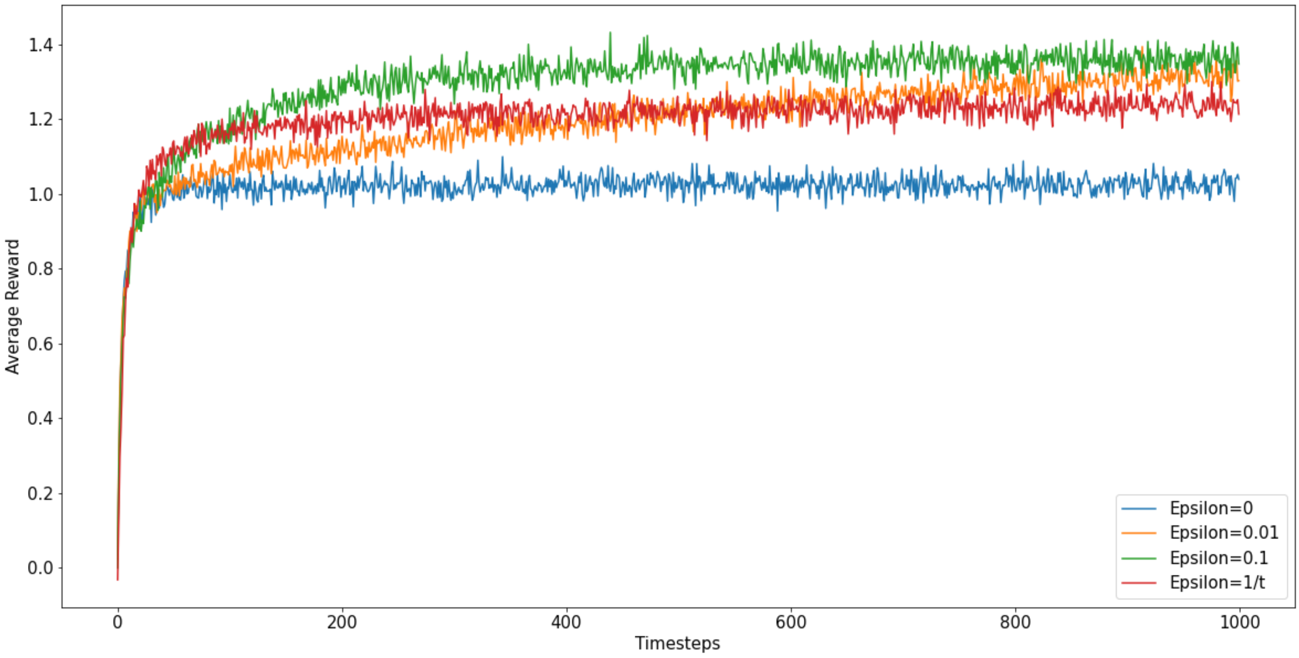
The arms are chosen greedily with a probability of 1-ε (exploitation) and randomly with a probability of ε (exploration).

Sample average method has been used for estimating the rewards using the following recursive formula:

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

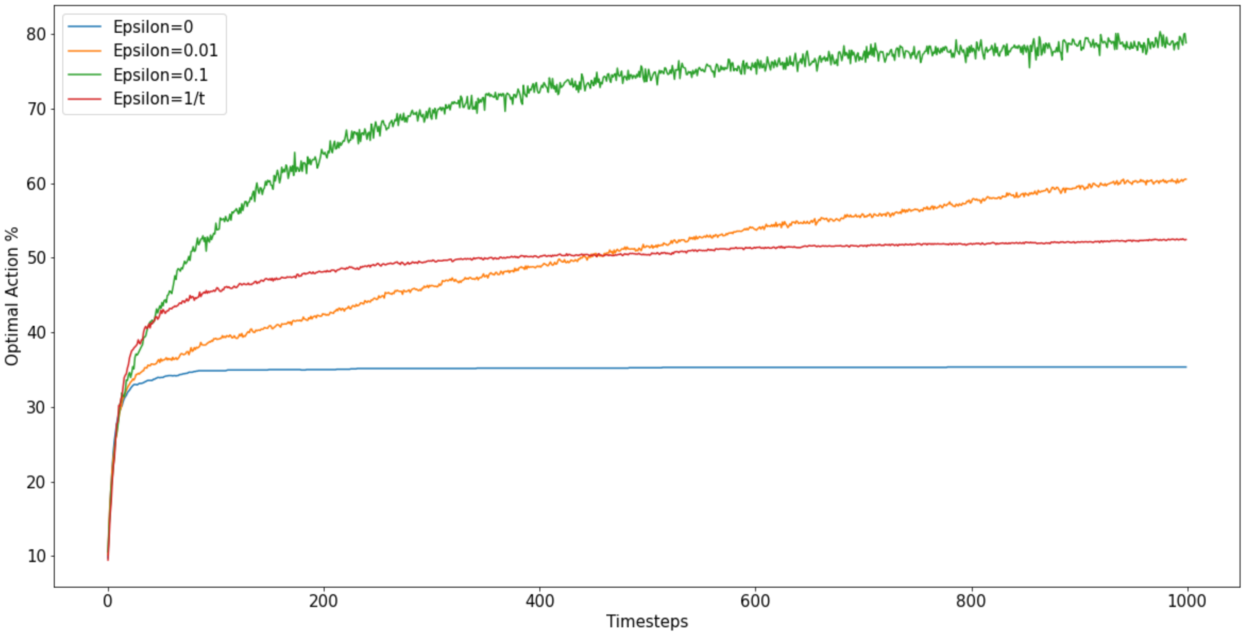
Average Rewards:

$$Average\ Reward(t) = \frac{Sum\ of\ rewards\ in\ all\ runs\ at\ time = t}{Total\ no.of\ runs}$$



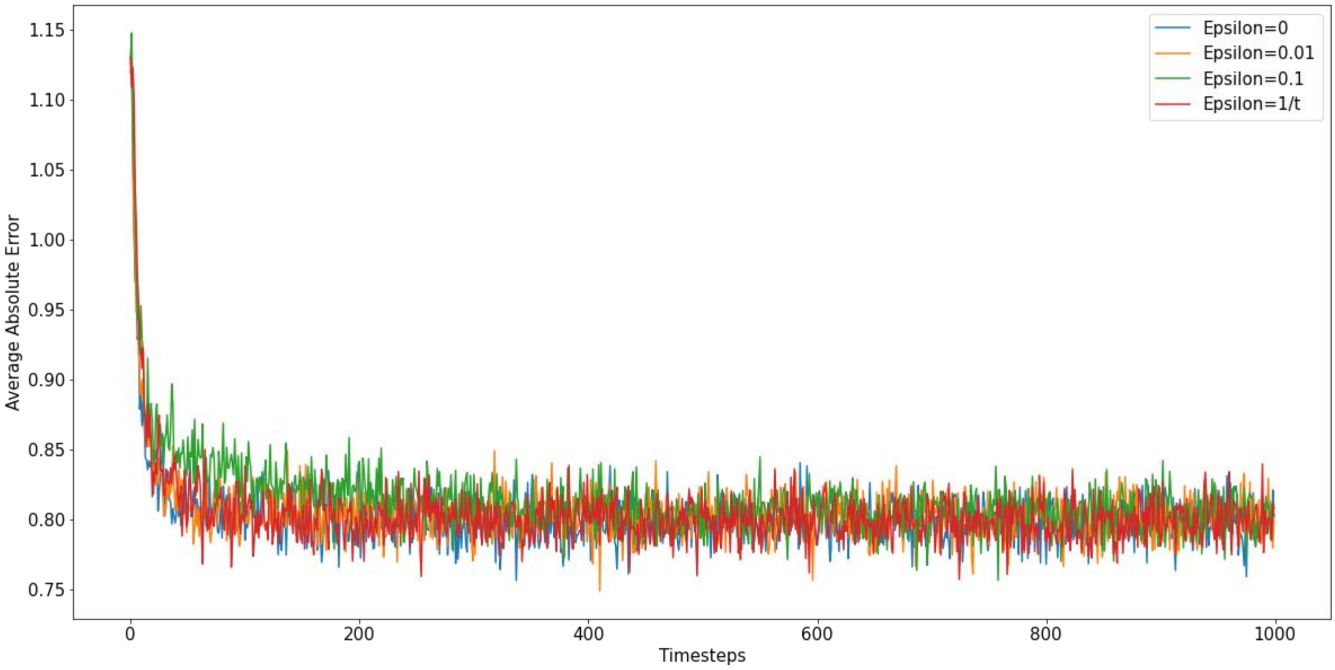
## Optimal Action %:

$$Optimal\ Arm\ \%\ (t) = \frac{No.of\ times\ optimal\ arm\ is\ chosen\ at\ time{=}t}{Total\ No.\ of\ runs} \times 100$$



## Average Absolute Error:

$$Average\ Absolute\ Error\ (t) = \frac{Actual\ Reward - Estimated\ Reward}{Total\ No.of\ Runs}$$

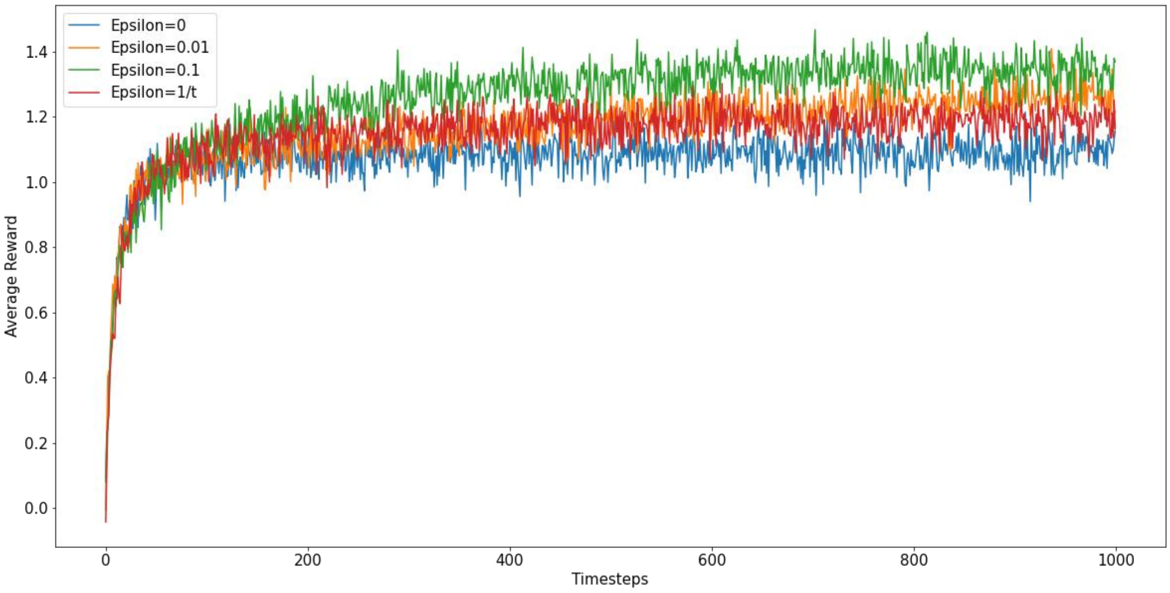


# Q2

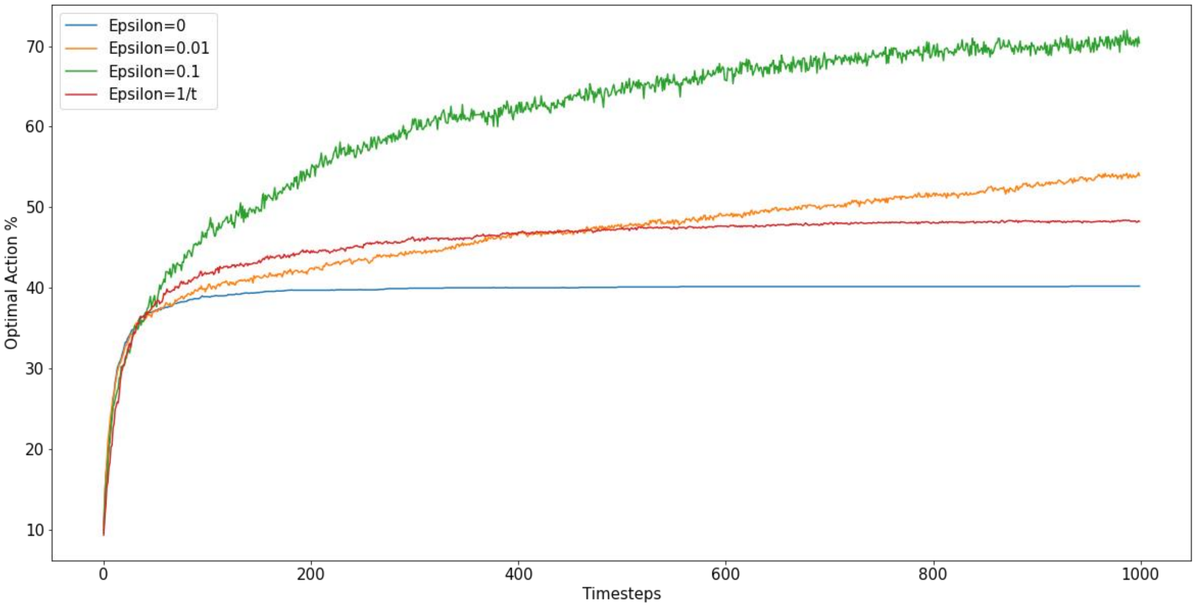
Generating plots similar to those in figure 2.2 of the book, with $\varepsilon = 0,0.01,0.1$ . All the arms have a normal distribution with variance 4 and mean chosen randomly from a gaussian distribution of mean 0 and variance 4.

(Repetition of Q1, only change is variance = 4).

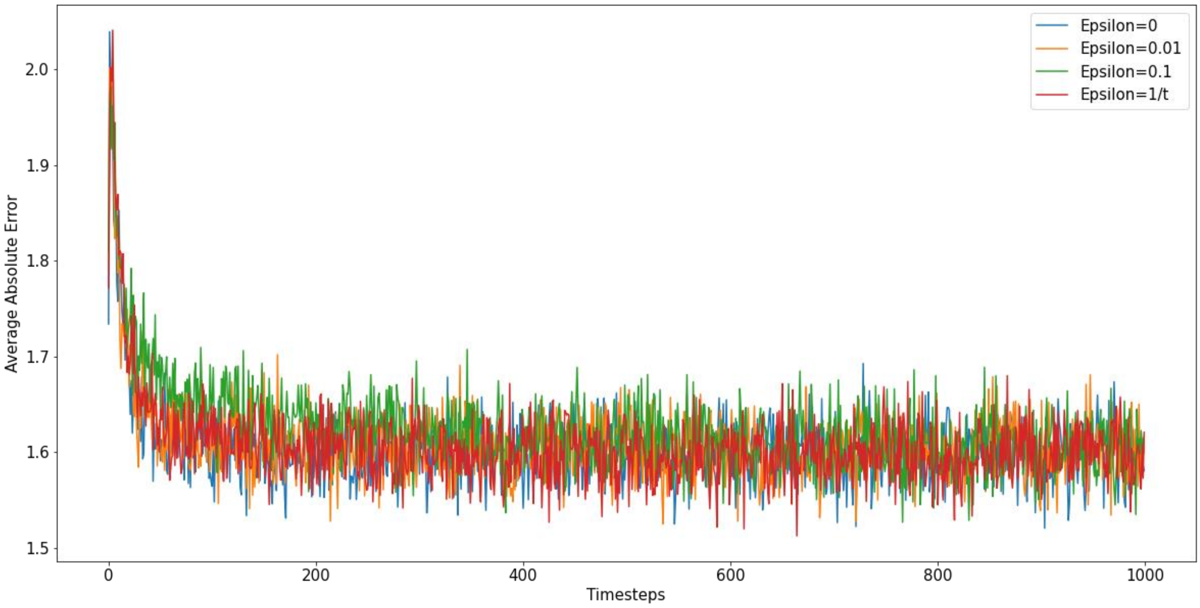
Average Reward:



Optimal Action %



Average Absolute Error:



# Q3

Assuming the long run consists of infinite no. of steps, we can say that the Ɛ-Greedy methods will achieve a nearly correct estimation of the value of the arms, and find the most optimal arm.

Let $p = P[selecting\ the\ optimal\ arm]$ =1- Ɛ + $\frac{\varepsilon}{|A|}$

Given $|A| = 10, p = 1 - \frac{9Ɛ}{10}$

For $Ɛ = 0.01, p = 0.991$ , i.e. The optimal arm is chosen 99.1% of times

For $Ɛ = 0.1, p = 0.91$ , i.e. The optimal arm is chosen 91% of timesType equation here.

When $Ɛ = \frac{1}{t}, p = 1 - \frac{9}{10t}$

This ensures that Ɛ reduces as time increases, which as a result reduces the amount of exploration done as $Q_t(a)$ approaches $q*(a)$, with an increasing percentage of choosing the optimal arm.

Since we are looking for the long run, $\lim_{t\to\infty} p = 1$, this means the probability of choosing the optimal arm is maximum in this case. This also results in the greatest cumulative reward in the long run.

When $Ɛ = 0$ , no exploration is done, so the algorithm focuses on the immediate best reward, only exploitation without maximizing the total reward over the long run.

Therefore, $Ɛ = \frac{1}{t}$ performs best in the long run in terms of both cumulative reward and the probability of selecting the optimal action.

# Q4

Sample Mean:

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

Since every $Q_{n+1}$ only depends on $Q_n$ and the current reward, and $Q_2$ does not depend on $Q_1$ as $Q_2 = Q_1 + \frac{1}{1}(R_1 - Q_1) = R_1$, therefore, sample mean is not influenced by the initial choice of $Q_1(a)$.

Constant Step Size:

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha(R_n - Q_n) \\
&= \alpha R_n + (1-\alpha)Q_n \\
&= \alpha R_n + (1-\alpha)(\alpha R_{n-1} + (1-\alpha)Q_{n-1}) \\
&\quad . \; . \; . \\
&= (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i
\end{aligned}
$$

Therefore, $Q_{n+1} = (1-\alpha)^n Q_1 + \sum_{i=1}^{n} \alpha(1-\alpha)^{n-i} R_i$ , we can clearly see that $Q_n$ is a function of $Q_1\,(a)$.

Since $\alpha < 1 \;\Rightarrow\; 1-\alpha < 1$, this means smaller the α, greater the value of $1-\alpha$, greater will be $(1-\alpha)^n$ i.e. the coefficient of $Q_1$ in the above equation, and greater the dependence of $Q_n$ on $Q_1$ .

Constant Step Size with No Dependence on $Q_1$:

$$
\begin{aligned}
Q_{n+1} &= Q_n + \alpha_n\,(R_n - Q_n) \\
&= \alpha_n R_n + (1-\alpha_n)Q_n \\
&= \alpha_n R_n + (1-\alpha_n)(\alpha_{n-1}R_{n-1} + (1-\alpha_{n-1})Q_{n-1}) \\
&= \alpha_n R_n + (1-\alpha_n)(1-\alpha_{n-1})Q_{n-1} + (1-\alpha_n)\alpha_{n-1}R_{n-1} \\
&\quad . \; . \; . \\
&= \sum_{j=1}^{n}\prod_{i=j+1}^{n}(1-\alpha_i)R_j\alpha_j \;+\; \prod_{i=1}^{n}(1-\alpha_i)Q_1
\end{aligned}
$$

Now, $Q_{n+1} = \sum_{j=1}^{n}\prod_{i=j+1}^{n}(1-\alpha_i)R_j\alpha_j \;+\; \prod_{i=1}^{n}(1-\alpha_i)Q_1$

Let $\alpha_i = \frac{\alpha}{\delta_i}$ where α is the constant step size parameter, and $\delta_i$ is defined by,

$\delta_{i+1} = \delta_i + \alpha(1-\delta_i)$ and $\delta_0 = 0$

We know that, $\alpha_1 = \frac{\alpha}{\delta_1}\; and\; \delta_1 = \delta_0 + \alpha(1-\delta_0) \Rightarrow \delta_1 = \alpha \Rightarrow \alpha_1 = 1$

Consider the coefficient of $Q_1$,

$\prod_{i=1}^{n}(1-\alpha_i) = (1-\alpha_1)(1-\alpha_2)\ldots(1-\alpha_n) = (1-1)(1-\alpha_2)\ldots(1-\alpha_n) = 0$

Therefore, using this trick we can modify the constant step size in a way that $Q_n$ becomes independent of $Q_1$.

## Q5

Aim: To demonstrate the difficulties that sample average methods have in nonstationary problems.

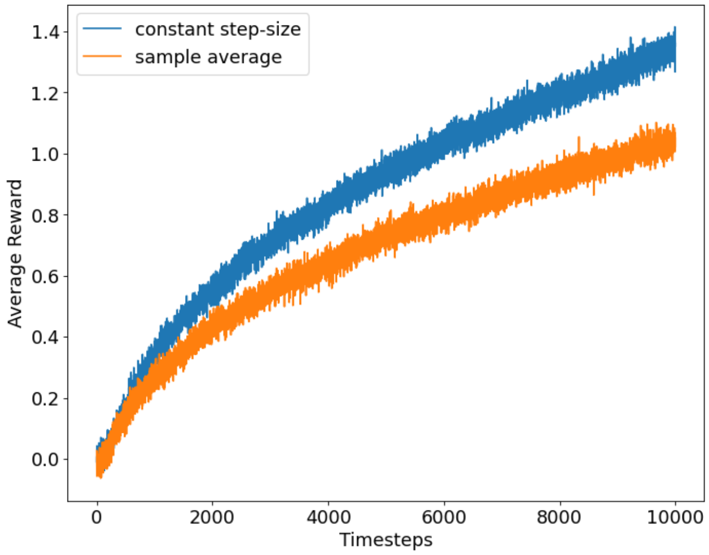
In both the methods, exploitation is done with a probability of 1- Ɛ and exploration with a probability of Ɛ.

For sample Mean: $Q_{n+1} = Q_n + \frac{1}{n}(R_n - Q_n)$ ,$\varepsilon = 0.1$

For constant step-size: $Q_{n+1} = Q_n + \alpha(R_n - Q_n)$, $\varepsilon = 0.1$, $\alpha = 0.1$

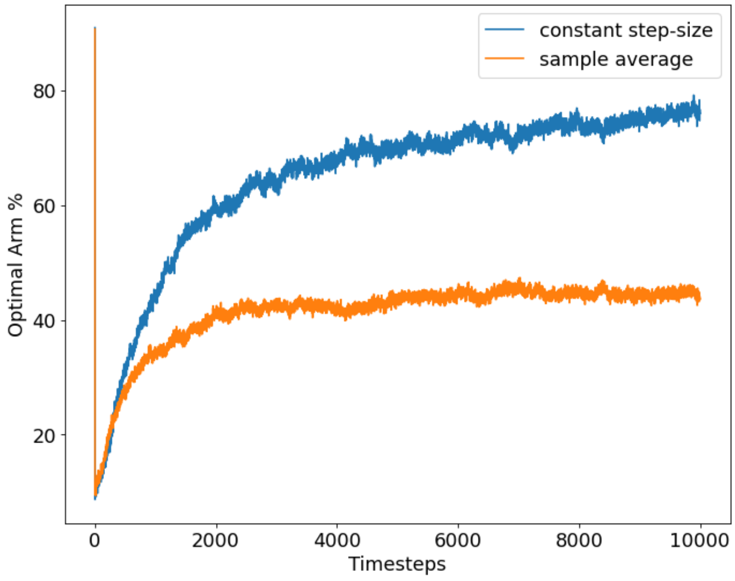
<u>Average Reward</u>:

$$Average\ Reward(t) = \frac{Sum\ of\ rewards\ in\ all\ runs\ at\ time{=}t}{Total\ no.of\ runs}$$



<u>Optimal Arm %:</u>

$$Optimal\ Arm\ \%\ (t) = \frac{No.of\ times\ optimal\ arm\ is\ chosen\ at\ time{=}t}{Total\ No.\ of\ runs} \times 100$$



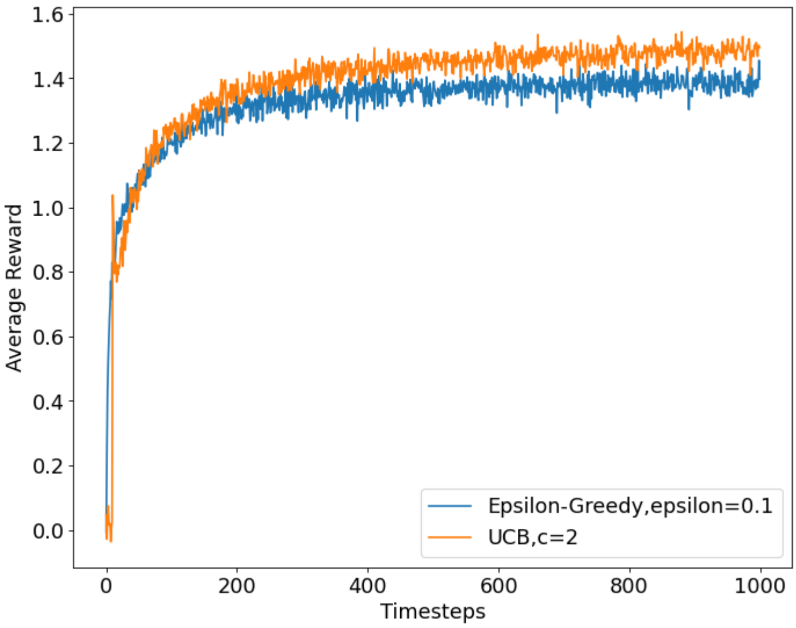
We can see that the constant step size method performs considerably better

## Q6

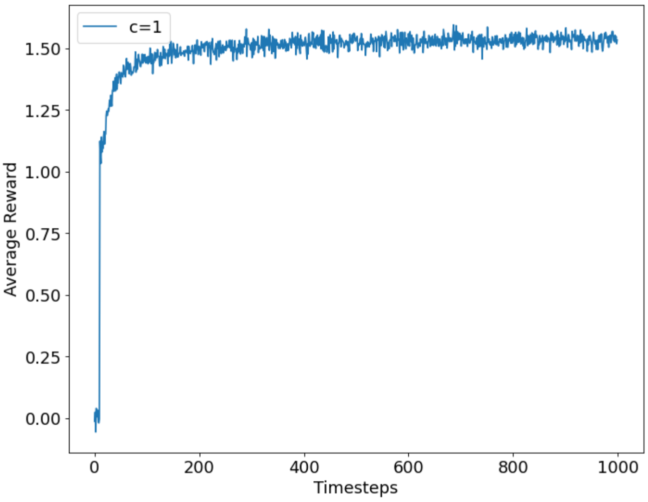
In the upper confidence bound action selection, we take into account the uncertainty in the our estimation of $Q_t(a)$, the term c $\sqrt{(lnt/N_t(a))}$ corresponds to the uncertainty level, decreasing with increase of $N_t(a)$ , the no. of times the arm $a$ is chosen, $c$ denotes the degree of exploration. The optimal arm is selected using:

$$A_t = \operatorname{argmax}\left[Q_t(a) + c\sqrt{(lnt/N_t(a))}\right]$$

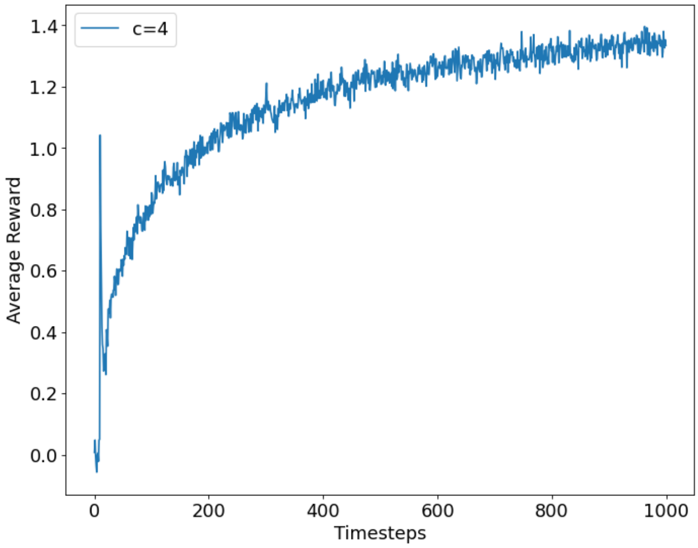
Following is a comparison of the Ɛ-greedy method and the upper confidence bound action selection:



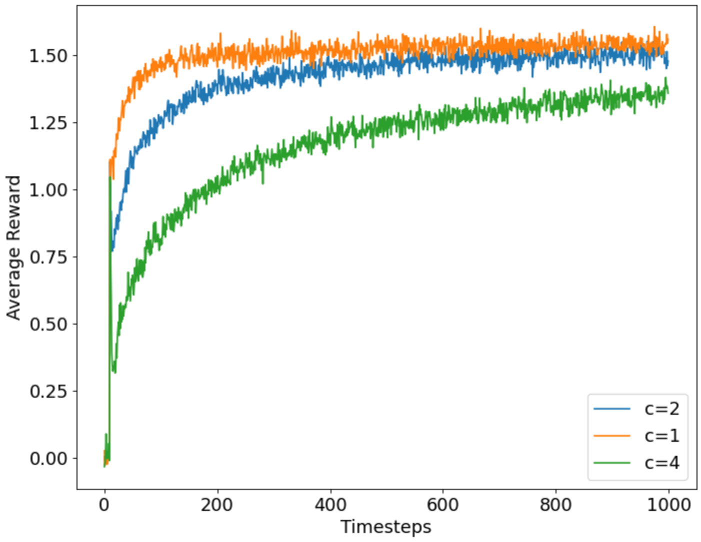
For c=1,



For c=4,



Following is a plot of expected average reward with varying degrees of exploration, $c = 1{,}2 \textit{ and } 4$.



In the equation, $\mathrm{A}_t = \operatorname{argmax}\left[Q_t(a) + \mathrm{c}\sqrt{(lnt/N_t(a))}\right]$

For the first k=10 timesteps, each of the arms is picked once, since when $N_t(a) = 0$ , the action is considered maximizing. At the 11th step, $Q_t(a)$ for the $i^{th}$ arm will now have received a value based on the move in which the $i^{th}$ arm was chosen, and $\mathrm{c}\sqrt{(lnt/N_t(a))}$ is same for all arms. Now the action picked is somewhat greedy, picking the one with maximum $Q_t(a)$ with the least uncertainty and hence the spike.

At the next step, the $\mathrm{c}\sqrt{(lnt/N_t(a))}$ term no longer remains same for all the arms and uncertainty increases, resulting in the drop seen in the graph.

We also see, that with increasing c, also increases the spike at the 11th action. This is because of the increasing degree of exploration.

## Q7

Generating figure 2.5 of the textbook, plotting the average performance of the gradient bandit algorithm with and without baseline on the 10-armed testbed when $q * (a)$ are chosen to be near +4 rather than 0 for both $\alpha = 0.1$ and $\alpha = 0.4$ .

In the gradient bandit algorithm, the arms are picked with a probability of $\pi_t(a)$, defined as

$$\pi_t(a) = \frac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}}$$

Where, $H_t\ (a)$ is the action preference function which updates at every step in the following manner,

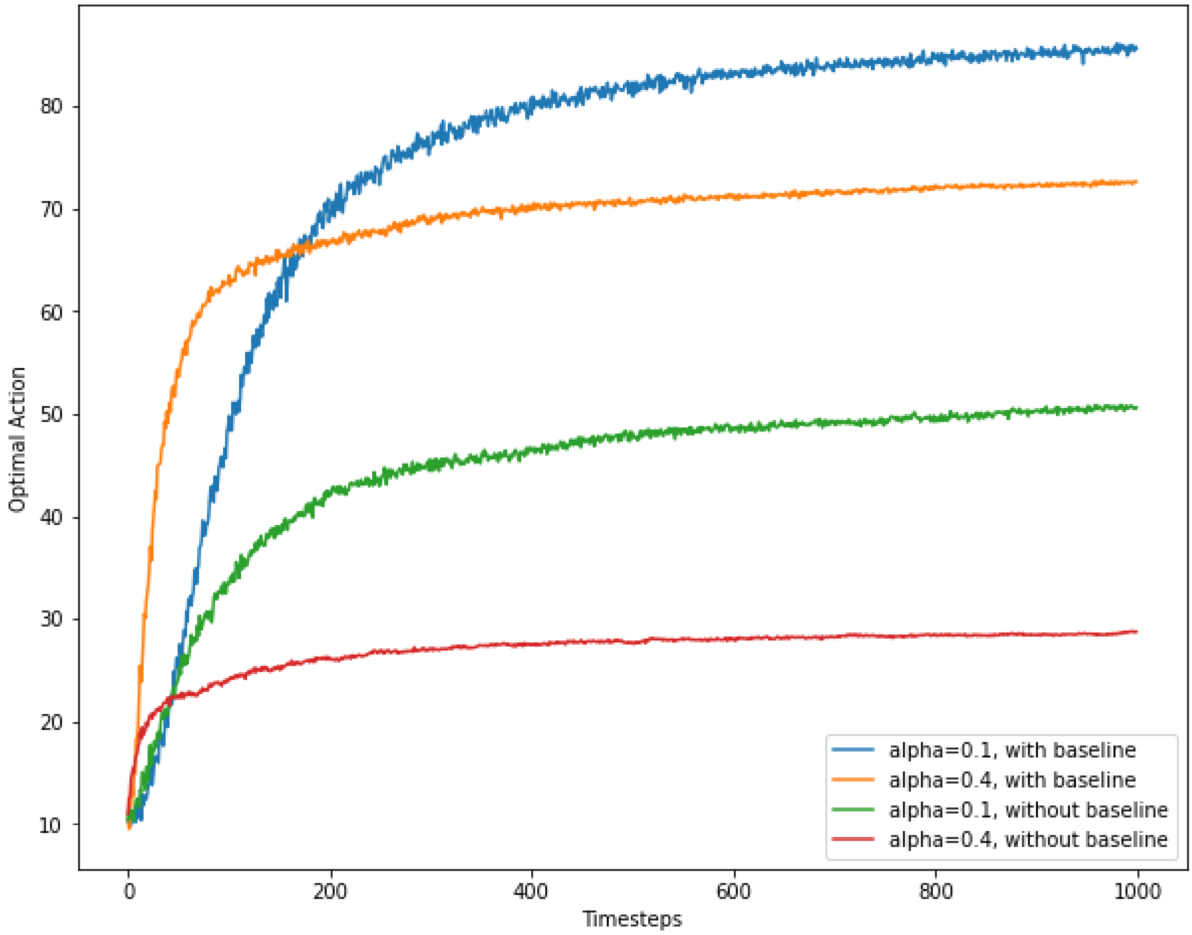
$H_{t+1}(A_t) = H_t(A_t) + \alpha(R_t - \bar{R}_t)(1 - \pi_t(a))$ if $a$ is the chosen arm

$H_{t+1}(A_t) = H_t(A_t) - \alpha(R_t - \bar{R}_t)\pi_t(a)$ for all the non-chosen arms.

$H_1(a) = 0$, for all arms

$\bar{R}_t$ is called the baseline.

Following is the plot of the optimal action % for various conditions:


Optimal Action
Timesteps
alpha=0.1, with baseline
alpha=0.4, with baseline
alpha=0.1, without baseline
alpha=0.4, without baseline