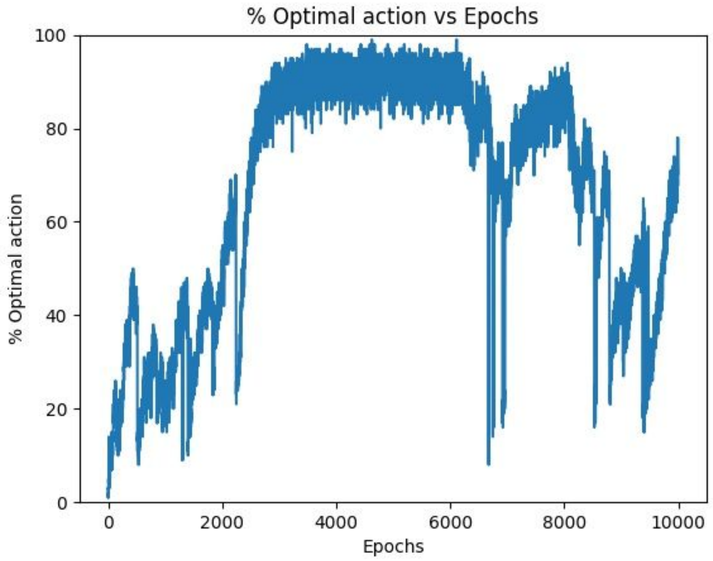
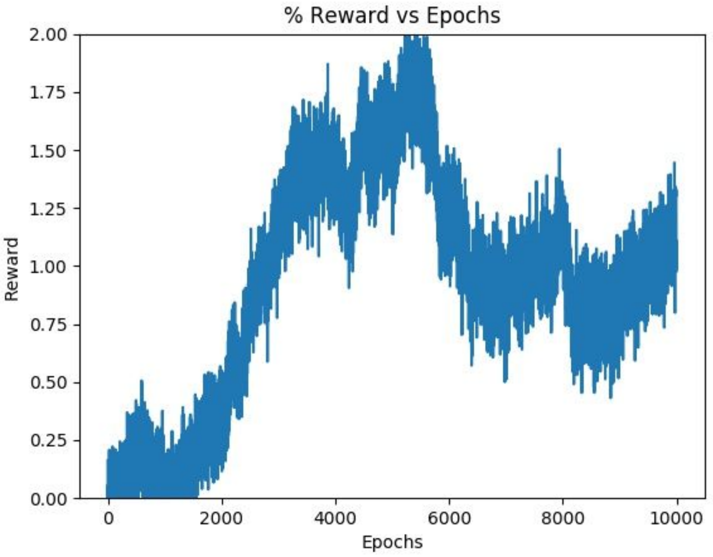
RL Assignment 1

Author
Nihesh Anderson (2016059)

Problem 1

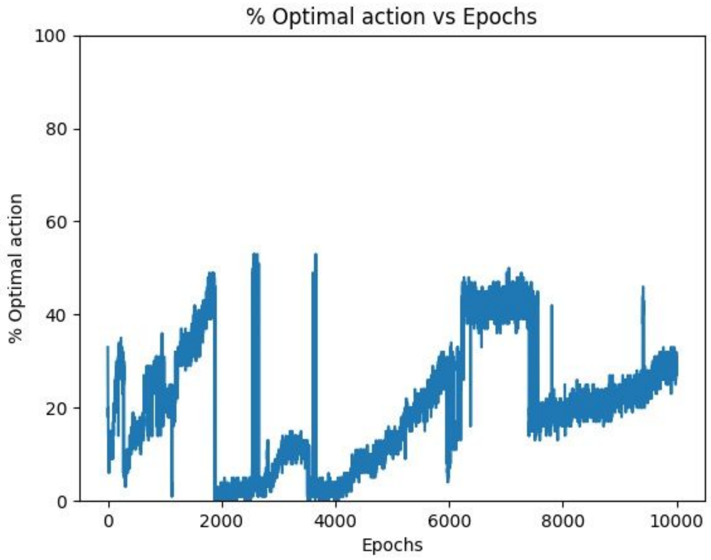
Alpha = 0.1 for estimating Q

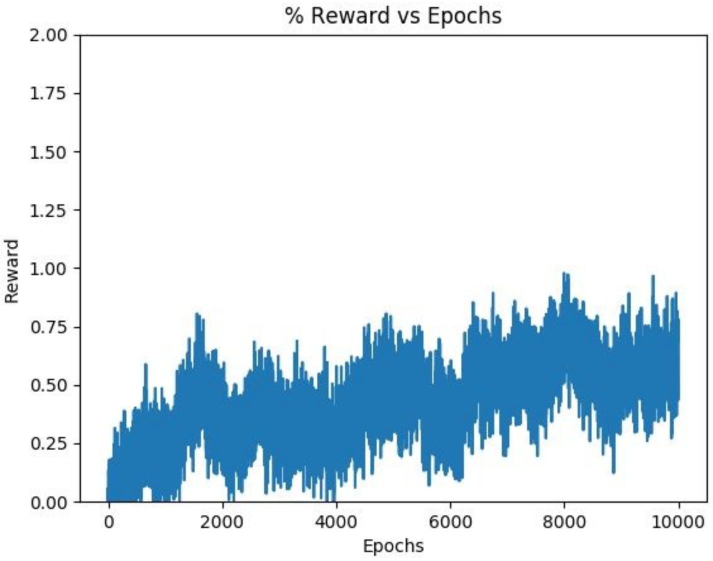




The environment itself is non stationary. Therefore, what the agent might think is an optimal action isn't optimal anymore, when a new bandit becomes more valuable. This is why there are pits in the first graph. However, the agent recovers from these pits far more quickly when compared to the mean based Q array estimator, as illustrated in the next page.

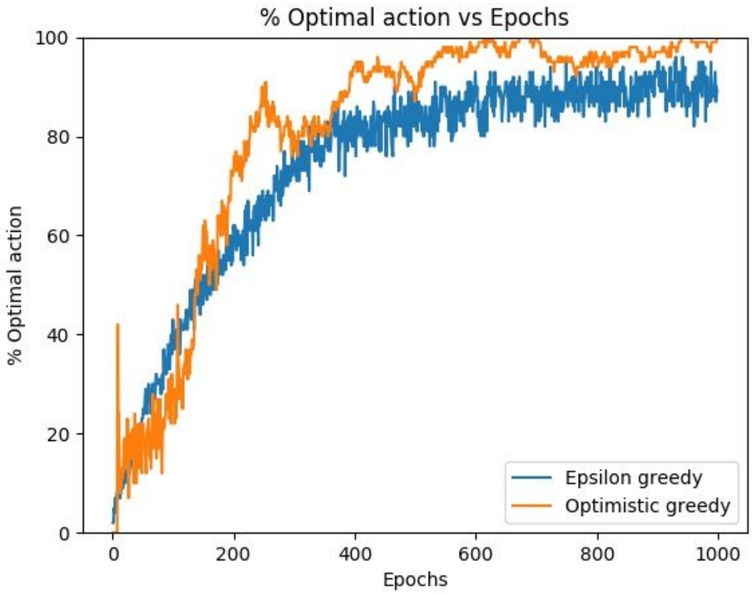
Mean based estimator





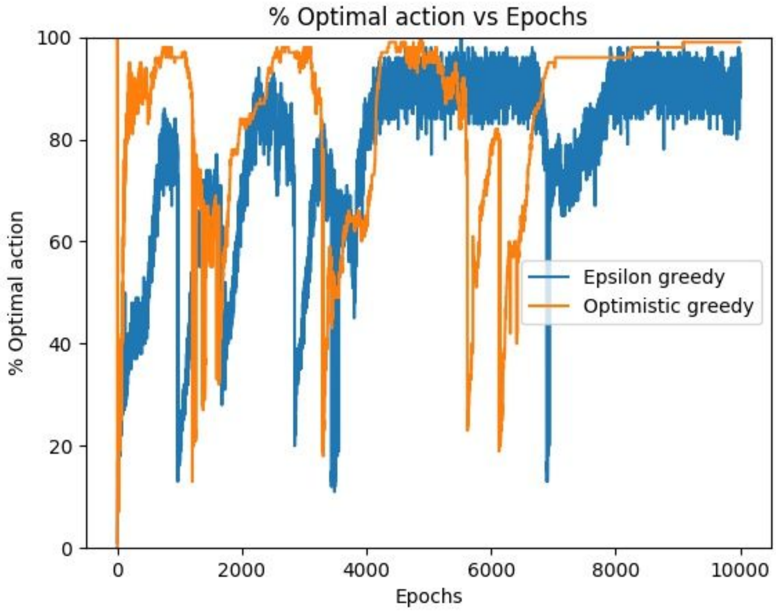
The first graph clearly shows that whenever the optimal action changes due to the dynamic nature of the environment, the agent takes a fairly long time to learn the changes because mean takes a lot of time to adapt.

Problem 2

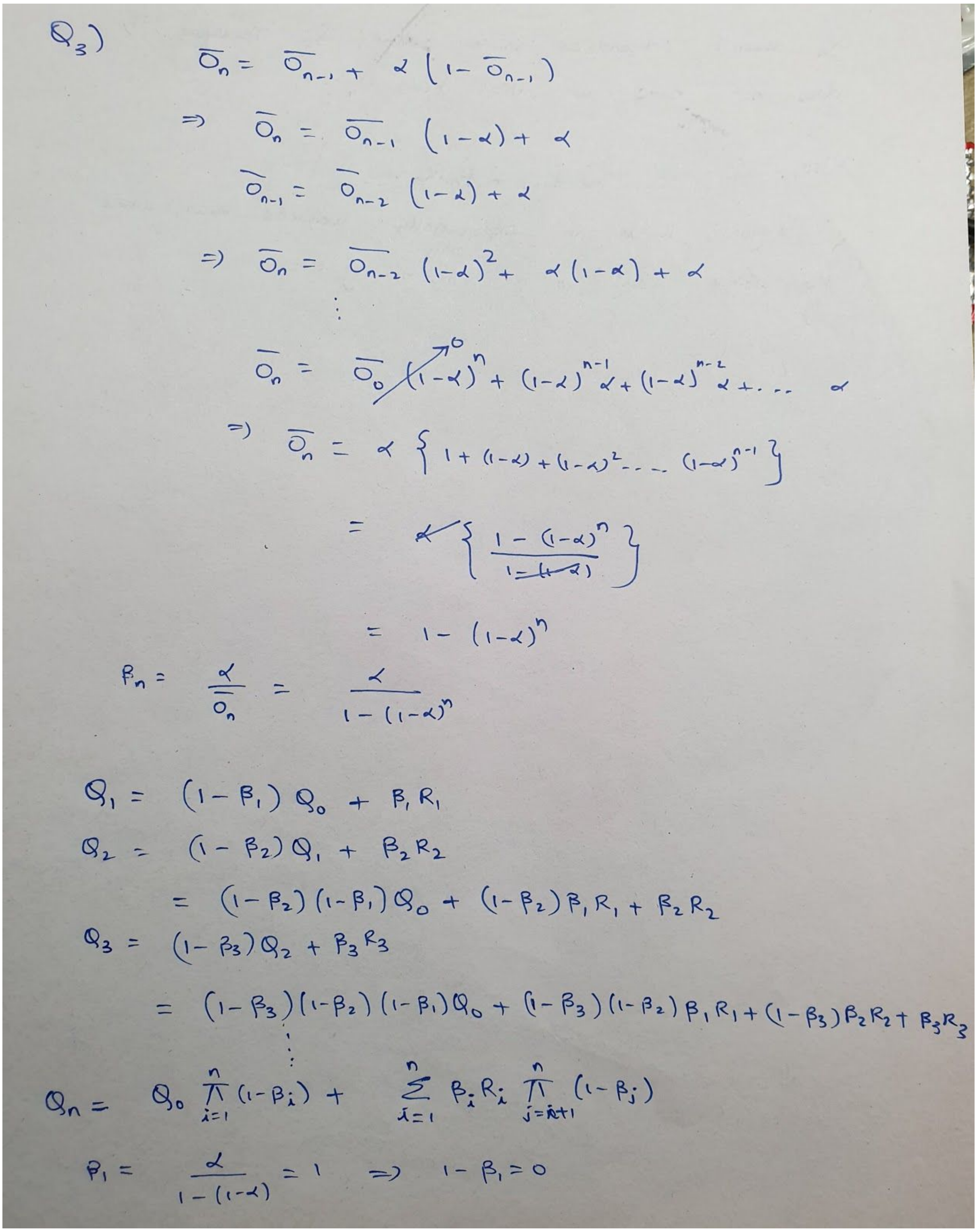


The above figure is equivalent to Fig 2.3 in course book

In case of non stationary environment, the performance of both the algorithms are more or less the same. Theoretically, we can say that epsilon greedy performs slightly better because it tends to explore and adapt to the continuous changes more quickly compared to optimistic greedy, which only gives an initial headstart.



Problem 3



Q_3)

$$\overline{O}_n = \overline{O}_{n-1} + \alpha\left(1 - \overline{O}_{n-1}\right)$$

$$\Rightarrow \quad \overline{O}_n = \overline{O}_{n-1}\left(1-\alpha\right) + \alpha$$

$$\overline{O}_{n-1} = \overline{O}_{n-2}\left(1-\alpha\right) + \alpha$$

$$\Rightarrow \quad \overline{O}_n = \overline{O}_{n-2}\left(1-\alpha\right)^2 + \alpha\left(1-\alpha\right) + \alpha$$

$$\vdots$$

$$\overline{O}_n = \cancelto{0}{\overline{O}_0\left(1-\alpha\right)^n} + \left(1-\alpha\right)^{n-1}\alpha + \left(1-\alpha\right)^{n-2}\alpha + \dots \alpha$$

$$\Rightarrow \quad \overline{O}_n = \alpha\left\{1 + (1-\alpha) + (1-\alpha)^2 \dots (1-\alpha)^{n-1}\right\}$$

$$= \cancel{\alpha}\left\{\frac{1-(1-\alpha)^n}{\cancel{1-(1-\alpha)}}\right\}$$

$$= 1 - (1-\alpha)^n$$

$$\beta_n = \frac{\alpha}{\overline{O}_n} = \frac{\alpha}{1-(1-\alpha)^n}$$

$$Q_1 = (1-\beta_1)Q_0 + \beta_1 R_1$$

$$Q_2 = (1-\beta_2)Q_1 + \beta_2 R_2$$

$$= (1-\beta_2)(1-\beta_1)Q_0 + (1-\beta_2)\beta_1 R_1 + \beta_2 R_2$$

$$Q_3 = (1-\beta_3)Q_2 + \beta_3 R_3$$

$$= (1-\beta_3)(1-\beta_2)(1-\beta_1)Q_0 + (1-\beta_3)(1-\beta_2)\beta_1 R_1 + (1-\beta_3)\beta_2 R_2 + \beta_3 R_3$$

$$\vdots$$

$$Q_n = Q_0 \prod_{i=1}^{n}(1-\beta_i) + \sum_{i=1}^{n} \beta_i R_i \prod_{j=i+1}^{n}(1-\beta_j)$$

$$\beta_1 = \frac{\alpha}{1-(1-\alpha)} = 1 \quad \Rightarrow \quad 1-\beta_1 = 0$$

$$\Rightarrow \quad Q_n = \sum_{i=1}^{n} \beta_i R_i \prod_{j=i+1}^{n} (1-\beta_j)$$

Q_n doesn't depend on initial value Q_0. Therefore, it does not have an initial bias.

Also, $Q_n = (1-\beta_n) Q_{n-1} + \beta_n R_n$.

Therefore, it is an exponentially weighted mean, where the weight itself is a function of time.

Left to prove: Sum of weights $= 1$, i.e, $\sum_{i=1}^{n} \beta_i \prod_{j=i+1}^{n} (1-\beta_j) = 1$

$$1-\beta_j = 1 - \frac{\alpha}{o_n} = \frac{\bar{o}_n - \alpha}{\bar{o}_n} = \frac{\bar{o}_{n-1}(1-\alpha) + \alpha - \alpha}{\bar{o}_n} = \frac{\bar{o}_{n-1}(1-\alpha)}{\bar{o}_n}$$

$$\Rightarrow \sum_{i=1}^{n} \beta_i \prod_{j=i+1}^{n} (1-\beta_j) = \sum_{i=1}^{n} \frac{\alpha}{\cancel{\bar{o}_i}} * \frac{\cancel{\bar{o}_i}}{\cancel{\bar{o}_{i+1}}} \frac{\cancel{\bar{o}_{i+1}}}{\cancel{\bar{o}_{i+2}}} \times \cdots \frac{\cancel{\bar{o}_{n-1}}}{\bar{o}_n} (1-\alpha)^{n-i}$$

$$= \sum_{i=1}^{n} \frac{\alpha}{\bar{o}_n} (1-\alpha)^{n-i}$$

$$= \frac{\alpha}{\bar{o}_n} \sum_{i=1}^{n} (1-\alpha)^{n-i}$$

$$= \frac{\alpha}{\bar{o}_n} \frac{1-(1-\alpha)^n}{1-(1-\alpha)} = \frac{\cancel{\alpha}}{\bar{o}_n} \frac{(1-(1-\alpha)^n)}{\cancel{\alpha}}$$

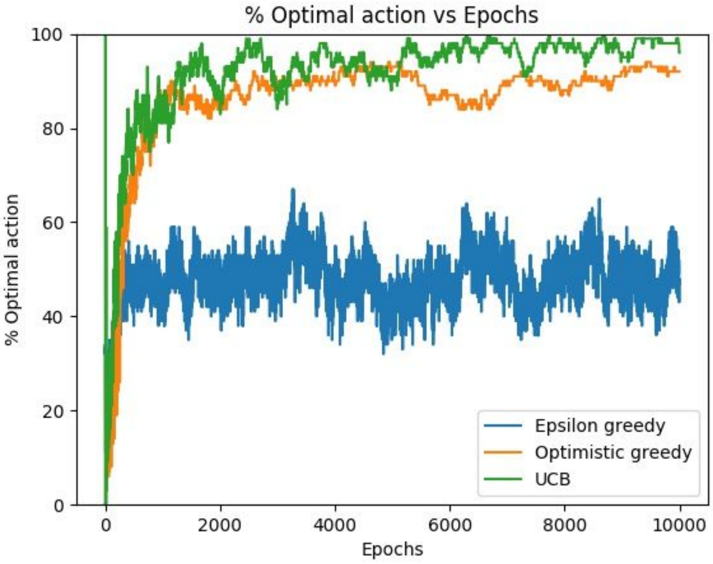
$\bar{o}_n = 1-(1-\alpha)^n \leftarrow$ from ①

$$\Rightarrow \sum_{i=1}^{n} \beta_i \prod_{j=i+1}^{n} (1-\beta_j) = \frac{(\cancel{1-(1-\alpha)^n})}{(\cancel{1-(1-\alpha)^n})} = 1$$

Hence, proved!

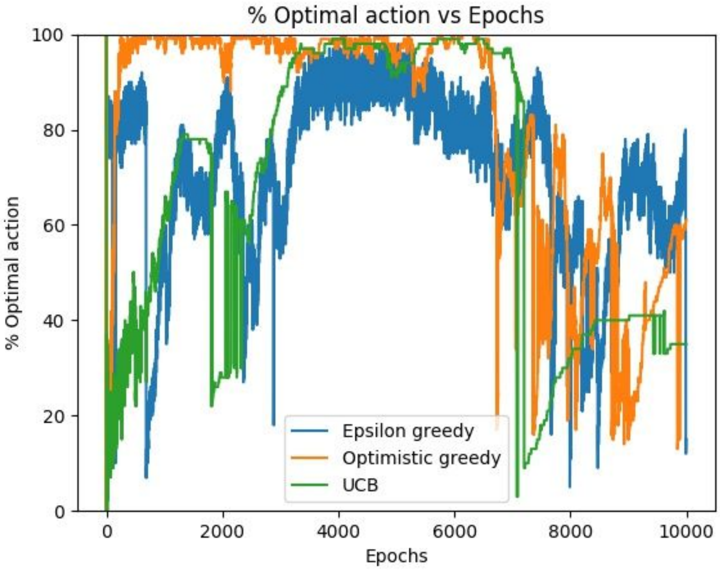
Problem 4

Performance of all the three algorithms in a stationary setting



It is clearly evident that UCB performs better than epsilon greedy and optimistic greedy in the stationary case. This is expected, as UCB gives a fair chance for all the actions, therefore obtaining better estimates of the underlying Q function.

Non stationary setting:



UCB does not seem to work as effectively as it did, in the non stationary setting. This is probably because UCB tries to pick values which are just less than the maximum Q value so that they get a fair chance, thereby decreasing the % optimal action.