CS 747 Programming Assignment 1

Adish Shah

3rd September, 2022

Contents

1 Task 1 **2**
1.1 UCB Algorithm 2
1.2 KL-UCB Algorithm 3
1.3 Thompson Sampling 4

2 Task 2 **5**

3 Task 3 **6**

4 References **7**

1 Task 1

1.1 UCB Algorithm

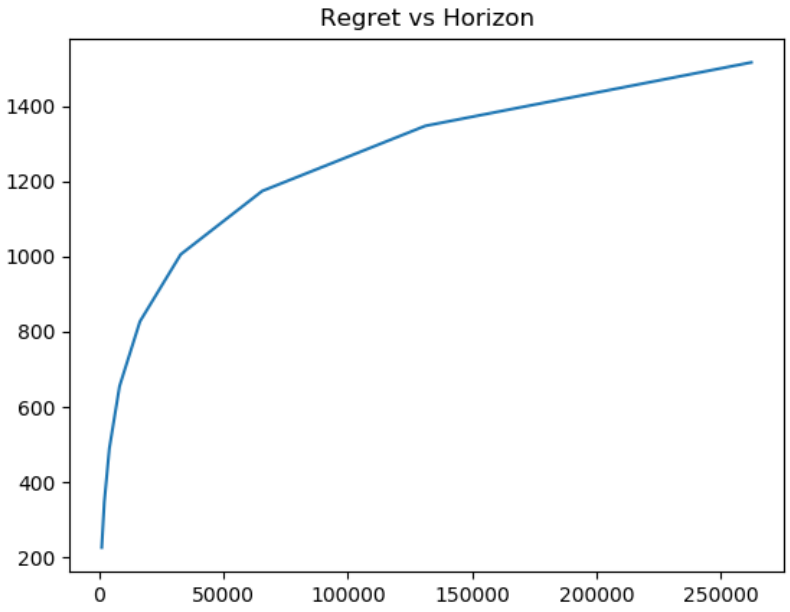

Figure 1: Regret v/s Horizon plot obtained using UCB for Task 1

The algorithm performs round-robin for the first num_arms time frames. After that at time t, for every arm a, we define

$$ucb_a^t = \hat{p}_a^t + \sqrt{\frac{2ln(t)}{u_a^t}}$$

where $\hat{p}_a^t$ is the empirical mean of rewards from arm a, and u_a^t is the number of times a has been sampled at time t. The arm a for which ucb_a^t is maximum is pulled. This is done using the `np.argmax()` function.

After receiving the reward, the corresponding empirical mean and u_a^t is updated.

We can observe that the graph is increasing and concave, which is expected as the regret is proportional to $log(T)$, where T is the horizon.

1.2 KL-UCB Algorithm

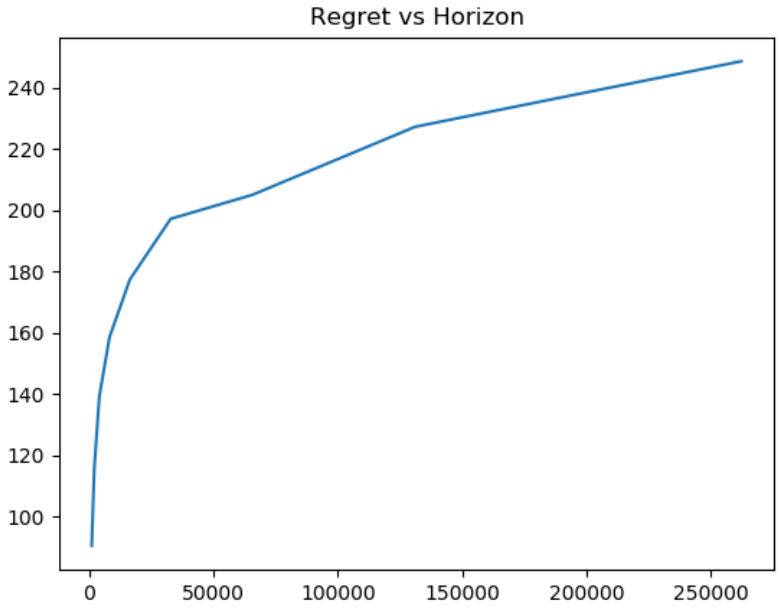

Figure 2: Regret v/s Horizon plot obtained using KL-UCB for Task 1

Similar to UCB, the algorithm performs round-robin for the first num_arms time frames. After that at time t, for every arm a, we define

$$ucb - kl_a^t = max\{q \in [\hat{p}_a^t, 1] s.t. KL(\hat{p}_a^t, q) \leq \frac{ln(t) + cln(ln(t))}{u_a^t}\}$$

where $\hat{p}_a^t$ is the empirical mean of rewards from arm a, and u_a^t is the number of times a has been sampled at time t.

I have chosen $c = 3$ and defined a function for $KL(p,q) = p * log(\frac{p}{q}) + (1-p) * log(\frac{1-p}{1-q})$. The value of q is found using binary search within 10^{-4} of the actual value.

The arm a for which $ucb - kl_a^t$ is maximum is pulled.This is done using the `np.argmax()` function.

After receiving the reward, the corresponding empirical mean and u_a^t is updated.

We can observe that the graph is increasing and concave, which is expected as the regret is proportional to $log(T)$, where T is the horizon. Also the regrets are lower compared to the UCB graph, which is also as expected, because KL-UCB achieves a tighter confidence bound than UCB.

1.3 Thompson Sampling

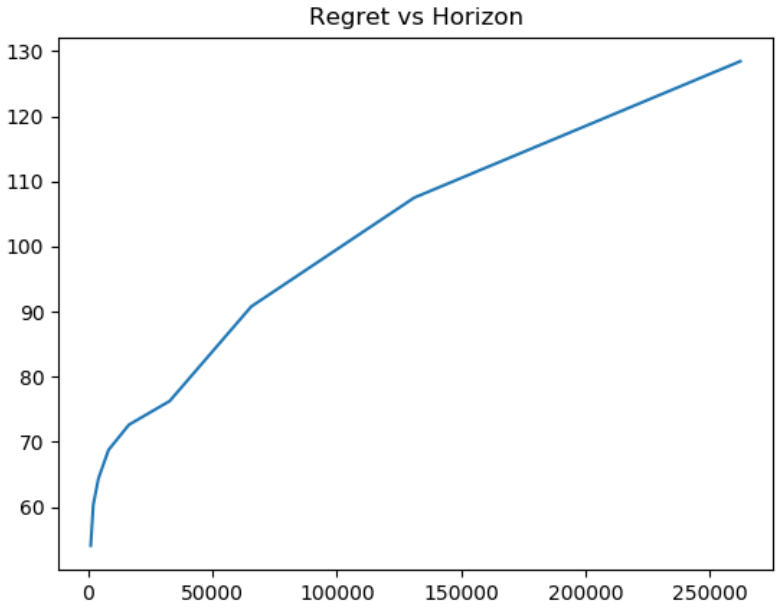

Figure 3: Regret v/s Horizon plot obtained using thompson sampling for Task 1

At time t, for every arm a, we have s_a^t successes and f_a^t failures. $\beta(s_a^t + 1, f_a^t + 1)$ represents a belief about the true mean of arm a. So, at time t, we draw a sample from this Beta distribution for every arm a. Finally, the one with the highest value of the sample is pulled. This is done using the `np.argmax()` function.

After receiving the reward, the corresponding s_a^t and f_a^t are updated.

We can observe that the graph is increasing and concave, which is expected as the regret is proportional to $log(T)$, where T is the horizon. Overall, the regret values obtained from Thompson sampling are lower than UCB and KL UCB, hence we can conclude it performs very well practically.

2 Task 2

Figure 4: Regret v/s Batch Size plot obtained using batched thompson sampling for Task 2

I have implemented a **batched thompson sampling** algorithm for this task.

The algorithm keeps track of the successes and failures of each arm. Till batch t, we have s_a^t successes and f_a^t failures for arm a. $\beta(s_a^t + 1, f_a^t + 1)$ represents a belief about the true mean of arm a. So, for every move in the new batch, we draw a sample from this Beta distribution for every arm a. The one with the highest value of the sample is recorded and added to the list of arms to be pulled for this batch. This gives us the set of arms to be pulled, and using the `np.unique` function gives us 2 lists of unique arms and the number of times each of them should be pulled.

After receiving the rewards, the corresponding s_a^t and f_a^t are updated.

We can observe that the regret linearly increases with batch size. This is because we are updating our belief of the true mean less frequently as bat as the batch size increases.

3 Task 3

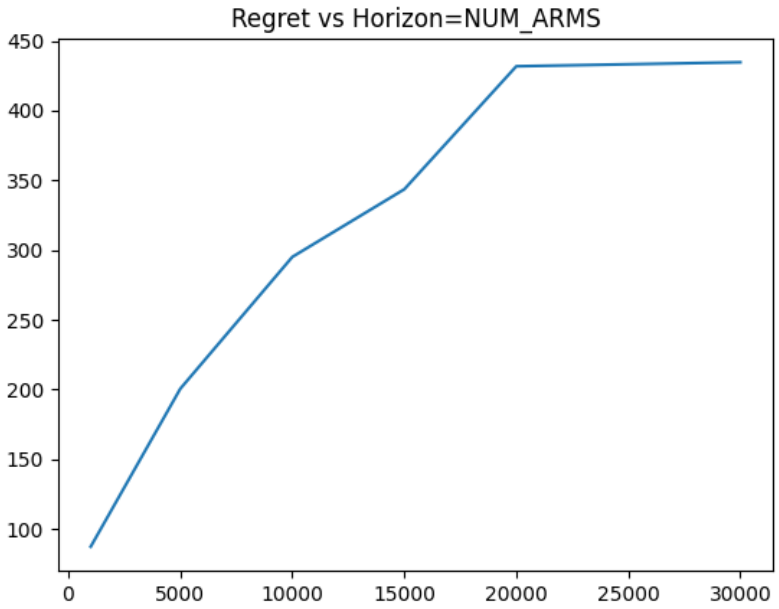

Figure 5: Regret v/s Horizon plot obtained for Task 3

In this task, since the number of arms is equal to the horizon, we cannot achieve sublinear regret, as the policies of Infinite Exploration and Greed in the limit cannot be satisfied.

Following is my approach to the problem. A subset of the arms is chosen randomly, which contains $\mathcal{X}$ arms. Since the given bandit instance is already randomised, I simply chose the first $\mathcal{X}$ arms to be my subset. With a very high probability we get an arm that has mean more than $1 - \frac{\mathcal{X}}{N}$, where N is the total number of arms.

After this, I have implemented the good old thompson sampling on this subset of arms. After experimenting a bit, I found that choosing $\mathcal{X} = \sqrt{N}$ gives low regret as compared to other values. The graph shows an increasing trend for the most part.

4 References

1. A Batched Multi-Armed Bandit Approach to News Headline Testing