CS747:Programming Assignment 1 Report

Debabrata Biswal
Roll no : 203050024

September 25, 2020

1 Introduction

The report contains the implementation details of the algorithms that I have coded for this assignment including the assumptions,some observations. P.S : All notations are according to the class conventions.So I am not defining them separately.

2 Epsilon Greedy

The epsilon was taken to be 0.02 as mentioned in the assignment.I am directly starting the algorithm without pulling each arm for some fixed times,as I think that's not really necessary for this algorithm.The regret is linear as it should be.

3 UCB

This was also a straightforward algorithm to implement.Although here I am pulling each arm once just so that for every arm u_t^a is non-zero as that will otherwise cause problem in calculation of upper confidence.For this algo I am setting upper confidence bound for arm a as

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

The regret for this algorithm was increasing logarithmically as should be the case.

4 KL-UCB

I really enjoyed implementing this algorithm,especially the part of optimizing this algorithm to decrease the time from 4 minute to just 45/55 seconds for instance 3 and horizon 102400.(All thanks to the vectorization).Firstly I am pulling each arm once before starting again just for convenience of calculations.Now since I pulled each arm once then empirical mean of each arm will now be either 0 or 1.But to avoid some obvious pitfalls (like $log(0)$ is $-\infty$ and $KL(p_a^t, 1)$ is ∞) I have added very small amount (1e-12) to the arm having empirical mean 0 and subtracted 1e-12 from arms having empirical mean 1.This was performing better than UCB algorithm as it is known to have smaller constant coeffcient and tighter upper confidence bound.I used C = 3 for this algo although having C less than 3 is performing better practically.

5 Thompson Sampling

Again This was also a fairly Simple and Straight Forward algorithm.We set a prior for mean of the each arm which is a beta distribution.Initially parameters of the beta are set to be alpha = 1 and beta = 1. Then we update our posterior distribution according to the reward. This was found to be performing better than KL-UCB for every instance.

6 Thompson Sampling With Hint

I learnt so many things solving this problem,especially I got clarity of how Thompson sampling works.Now the problem was to come up with a algorithm which performs better than Thompson sampling if you know the means of the arms beforehand but just the permutated means.
I first tried some algorithms based on the ideas of Median Elimination and all,But It seemed like I am not using much of the information that is being provided to me.I was just using the knowledge of Maximum mean but not the whole permutated means that is being given to me.
So after some thought I decided do just like Thompson sampling , But instead of using a beta prior I will use a Discrete Prior.For each arm a there will be a discrete distribution that will have $P(mean = \theta | arm = a)$ for each theta from the array of permutated means.Now according to the reward I get from a arm I will update the distribution for that arm.As i said It is exactly equal to the Thompson sampling but with a discrete prior instead of a beta prior.The update rule is straight-forward.

$$p(mean = \theta | reward) = \frac{p(reward | mean = \theta) p(mean = \theta)}{\sum_j p(reward | mean = \theta_j) p(mean = \theta_j)}$$

Now this algorithm worked as expected, But due to discrete values the sampling dis not work as expected,So this algorithm still lacked a little behind Thompson sampling for some instances.Now being out of ideas , I thought to again use the information we are provided.So when after some iteration the distribution of the optimal arm a saturates to the point that $p(mean = highest\theta|arm = a)$ is above 0.99 I choose that arm and starting from that point I only pull the chosen arm.Now as expected This modification faired way better.

7 Just a observation

For instance 3 When I compared UCB with Epsilon-Greedy I noticed that even till Horizon 102400 Epsilon Greedy was performing better than UCB.Although if you look at the slopes you can clearly see that regret for epsilon-greedy was increasing much more rapidly.So just to confirm it I checked for horizon 500000 over 50 random seeds and averaged the regret and found the following.
The mean Regret for epsilon greedy was - 4814.32
The mean regret for epsilon greedy was - 2393.04
As expected UCB surpassed epsilon greedy.

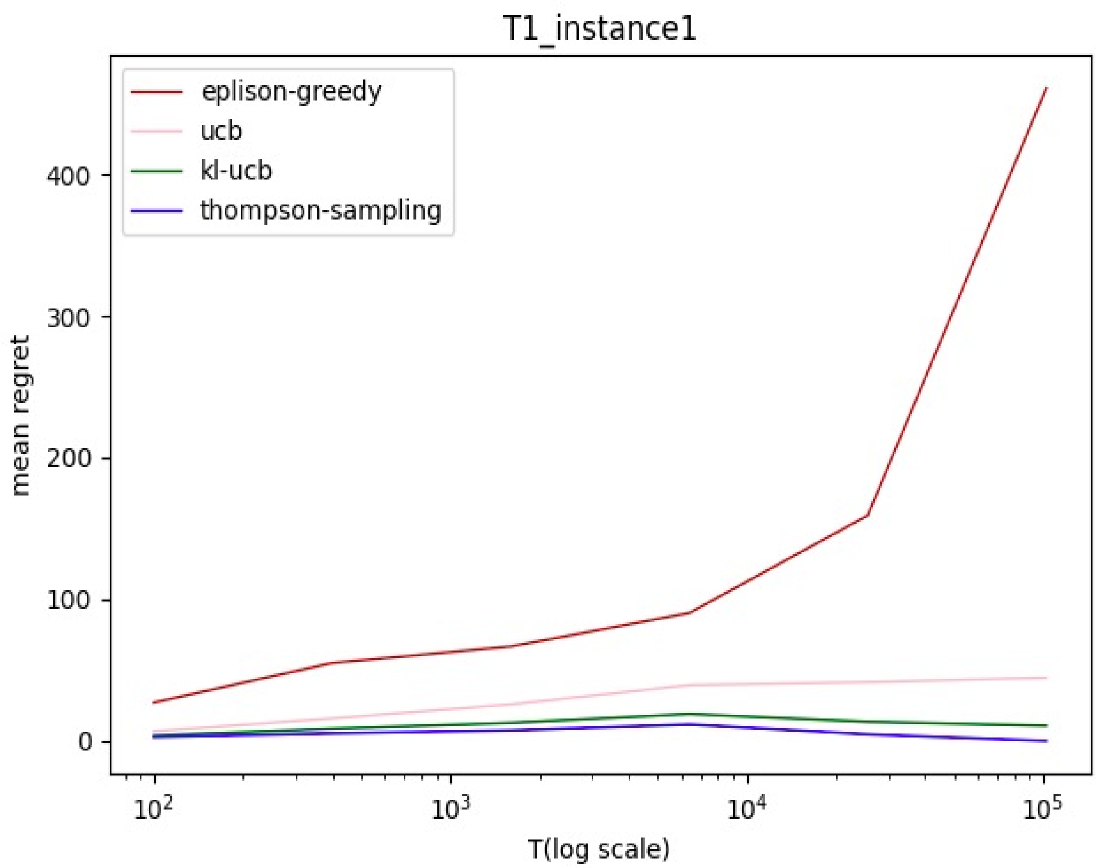

Figure 1: Comparison of different algorithms for instance 1

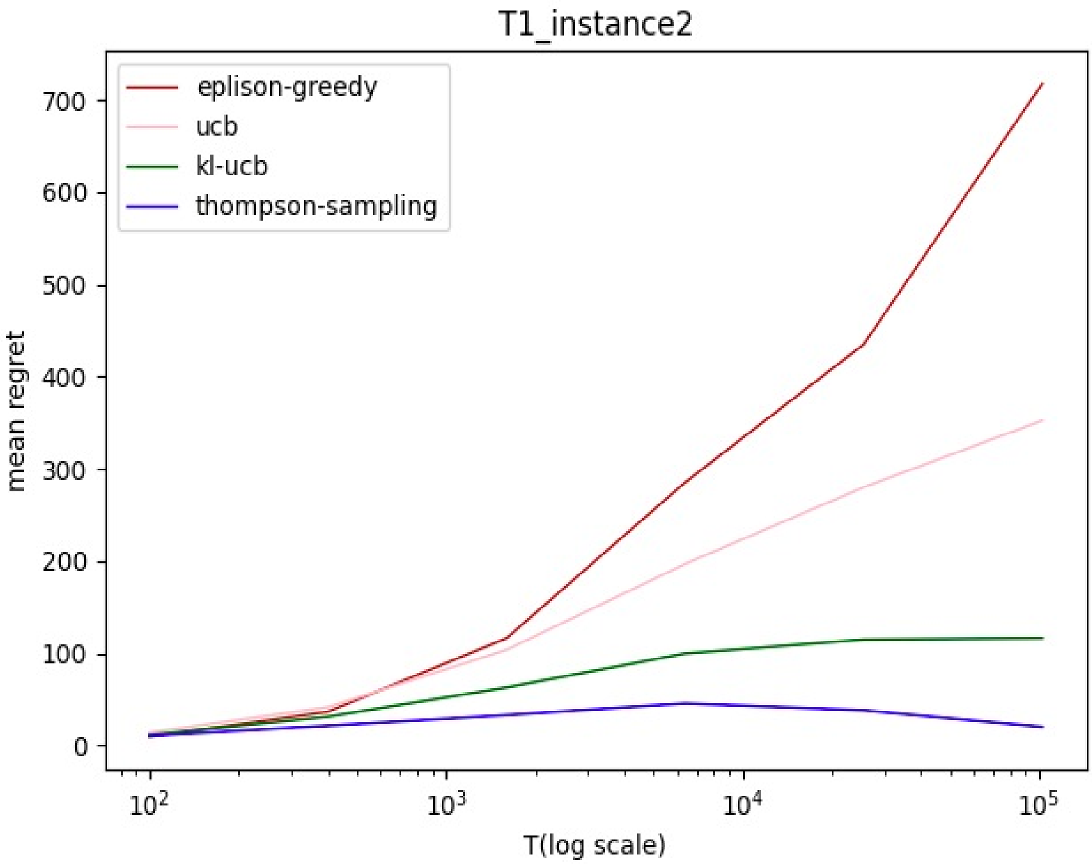

Figure 2: Comparison of different algorithms for instance 2

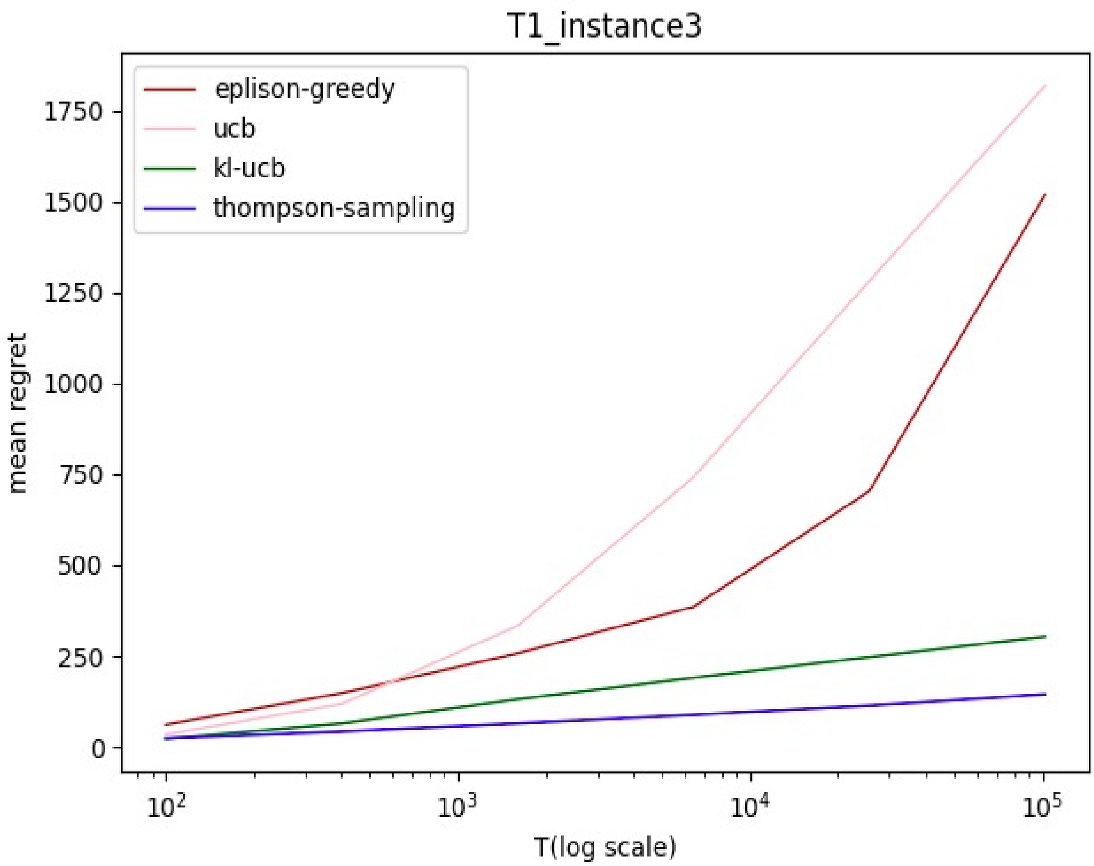

Figure 3: Comparison of different algorithms for instance 3

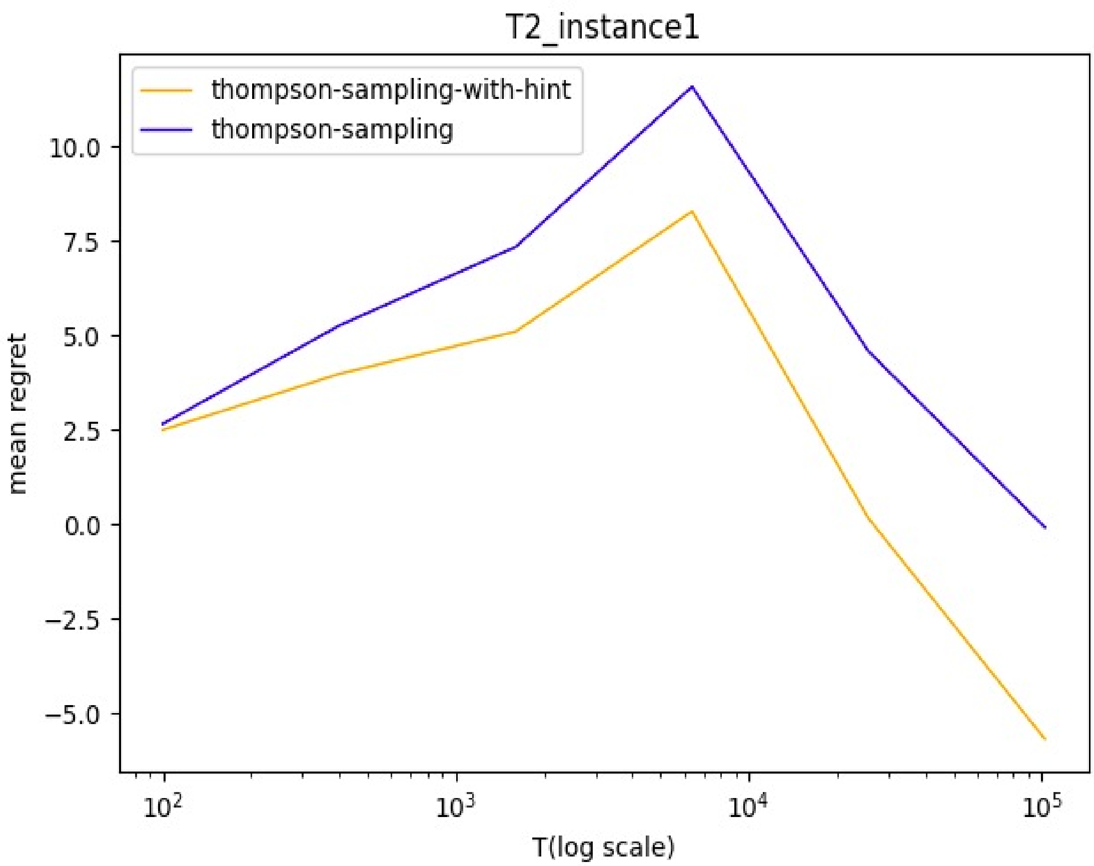

Figure 4: Comparison of Thompson Sampling and Thompson Sampling with hint for instance 1

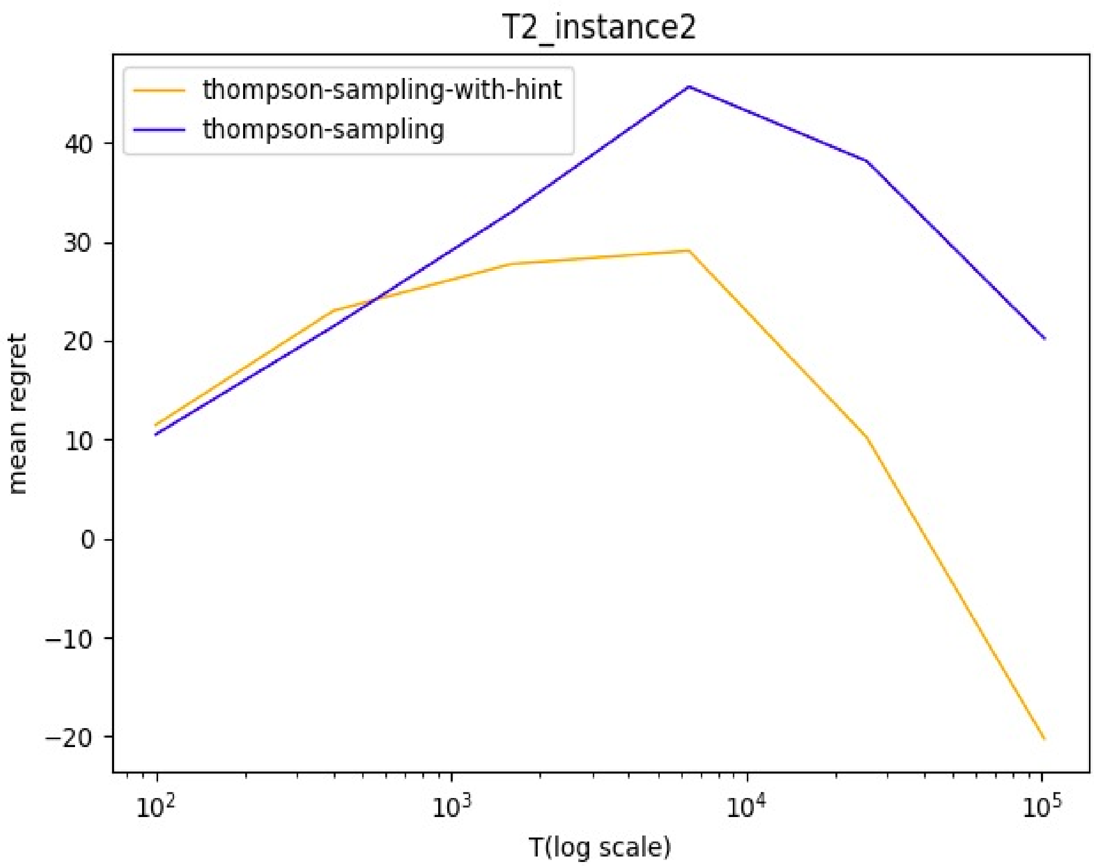

Figure 5: Comparison of Thompson Sampling and Thompson Sampling with hint for instance 2

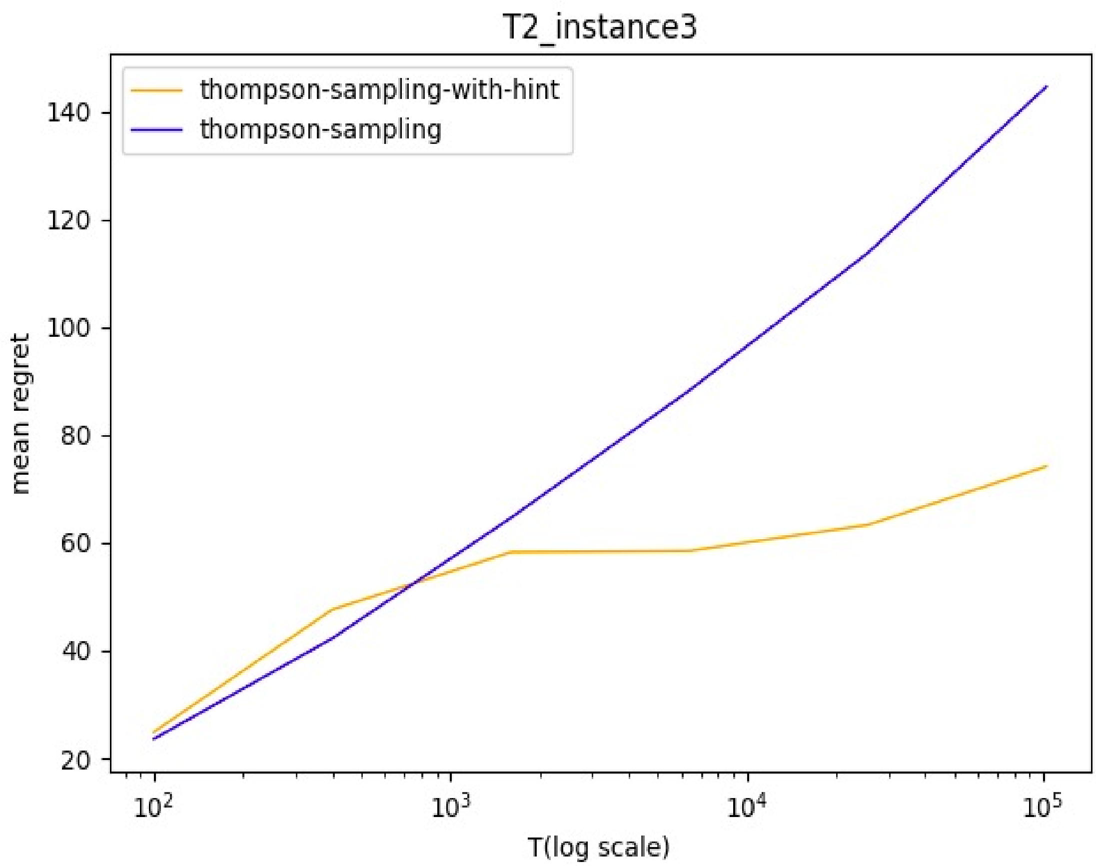

Figure 6: Comparison of Thompson Sampling and Thompson Sampling with hint for instance 3

8 T3

Here are ϵ values I found such that $\epsilon_1 < \epsilon_2 < \epsilon_3$ but regret corresponding to ϵ_2 is lowest.
for instance 1 : $\epsilon_1 = 0.001, \epsilon_2 = 0.005, \epsilon_3 = 0.1$
for instance 2: $\epsilon_1 = 0.0001, \epsilon_2 = 0.02, \epsilon_2 = 0.1$
for instance 3: $\epsilon_1 = 0.0001, \epsilon_2 = 0.02, \epsilon_3 = 0.1$