CS 747 Report : Assignment 1

Siddhesh Pawar
17D170011

Task 1:

Implementation details and assumptions

Generating Bernoulli rewards:

For a given p, I used np.random to get a random number between 0 and 1 if the number is greater than p, then reward=0 else the reward =1.

This follows from the well-known property of extension of uniform distribution to generate any other probability distribution

1. Epsilon greedy: epsilon was set to 0.02. A decision variable is defined that chooses a number(a) with probability epsilon and another number with probability 1-epsilon. If the decision variable chooses number a, then randomly any arm is picked from the given arms(including the arm with max mean), and with probability 1-epsilon arm with the highest mean(empirical mean) is chosen
2. Kl-ucb: Each arm is sampled once initially, then after that Kl ucb is calculated for each arm followed by picking the arm with the highest Kl ucb. A list of Q is maintained. The Q is the maximum q that lies between p_a and 1 and satisfies:
   KL(p_a,q) is less than or equal to (lnT+3lnlnT)/u_a which is found through iterations starting with an initial guess as 0.1. A for loop is used for this purpose.
3. Ucb: Round robin sampling is done for first n iterations where n is the number of arms after each arm has been pulled once, then Upper confidence bound is calculated for each arm and the arm with the highest ucb is picked. Based on the reward, the estimated mean for that arm is updated(which is used for calculation of ucb in the next iteration). An array of UCB is maintained.
4. Thompson sampling: A vector of success and failures is maintained. At every iteration, we sample a point from t=beta(success+1,failure+1) for each arm and the arm with the highest value of t is played, and based on the rewards the success/failure of that arm is updated.

.

It is seen that Thompson sampling performs the best for all instances while epsilon greedy performs the worst for two instances.

For instance 3, kl-ucb performs as badly as epsilon greedy which implies that kl-ucb is not a good algorithm for that instance

KL-UCB performs better than UCB for instances 1 and 2.

Plots for all have been included at the end of the report(in Task 4 section)

The negative regret in some cases implies that the maximum estimated mean was more than the true mean of the optimal arm

Task 2:

Algorithm for Thompson sampling with a hint:
Maintain a vector of empirical means(assuming arms follow beta distribution) of the arms, and play the arm whose empirical mean is closest to the true mean of the optimal arm.
This is done as follows:

1. For each iteration in the horizon maintain a safe set C of arms whose empirical mean is greater than (true of optimal arm+true mean of second-best arm)/2, this ensures that set C contains arms whose empirical mean is close to or greater than true optimal mean.The mean of the beta distribution is given by alpha/(alpha + Beta) where alpha and beta are the parameters of beta distribution
2. Pick the arm with the highest empirical mean from the safe set C, this ensures exploitation
3. If set C is empty, carry out regular Thompson sampling with (succss+1,faiures+1) as parameters of beta distribution for the corresponding arm, this ensures exploration

Experiments tried:

1. For each iteration, firstly safe set C was defined as a set of arms whose empirical mean is greater than the second-best arm, which however didn't give as good results for the third instance
2. For each iteration, safe set C was defined as arms whose empirical means is more than the average of true means which too didn't give as good results (although the results were better than Thompson sampling in some cases)

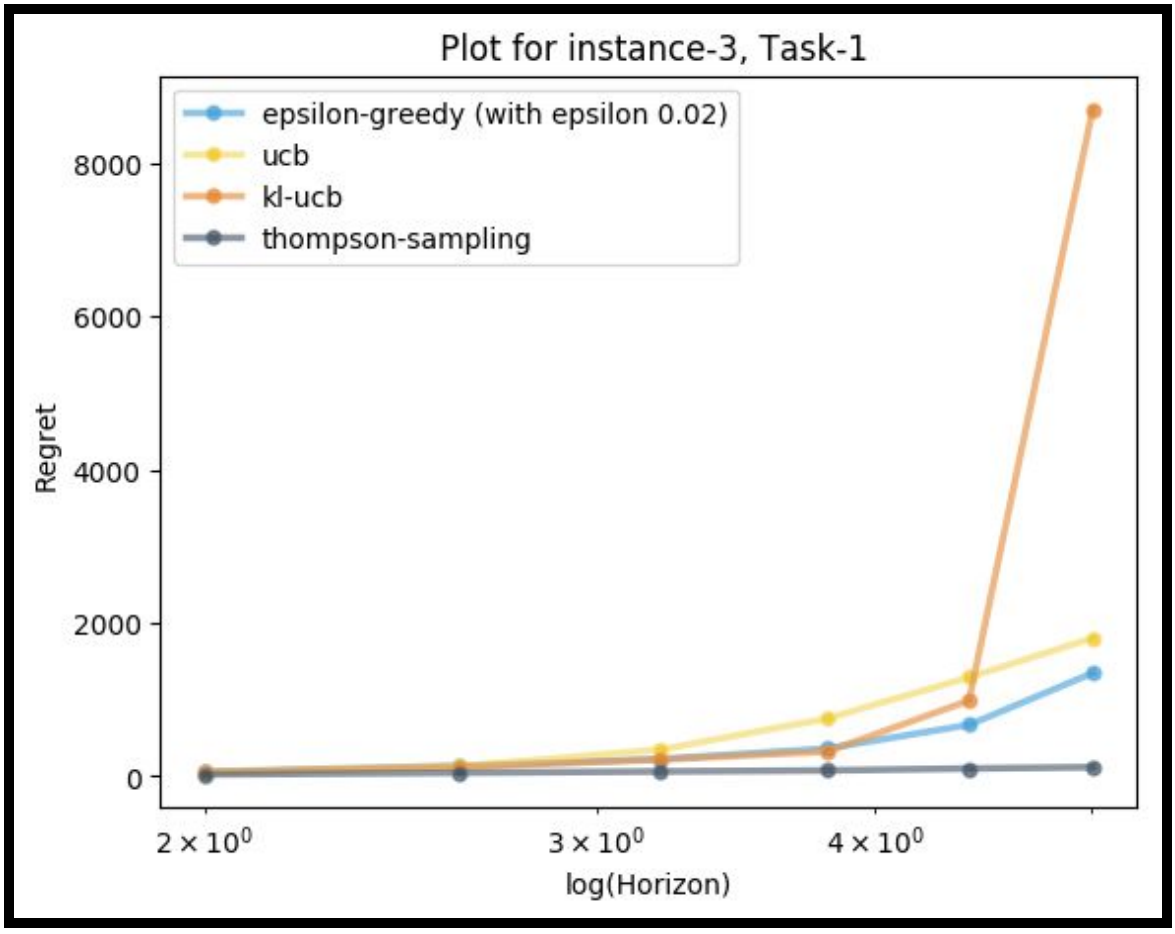
Plots for all have been included at the end of the report(in Task 4 section)

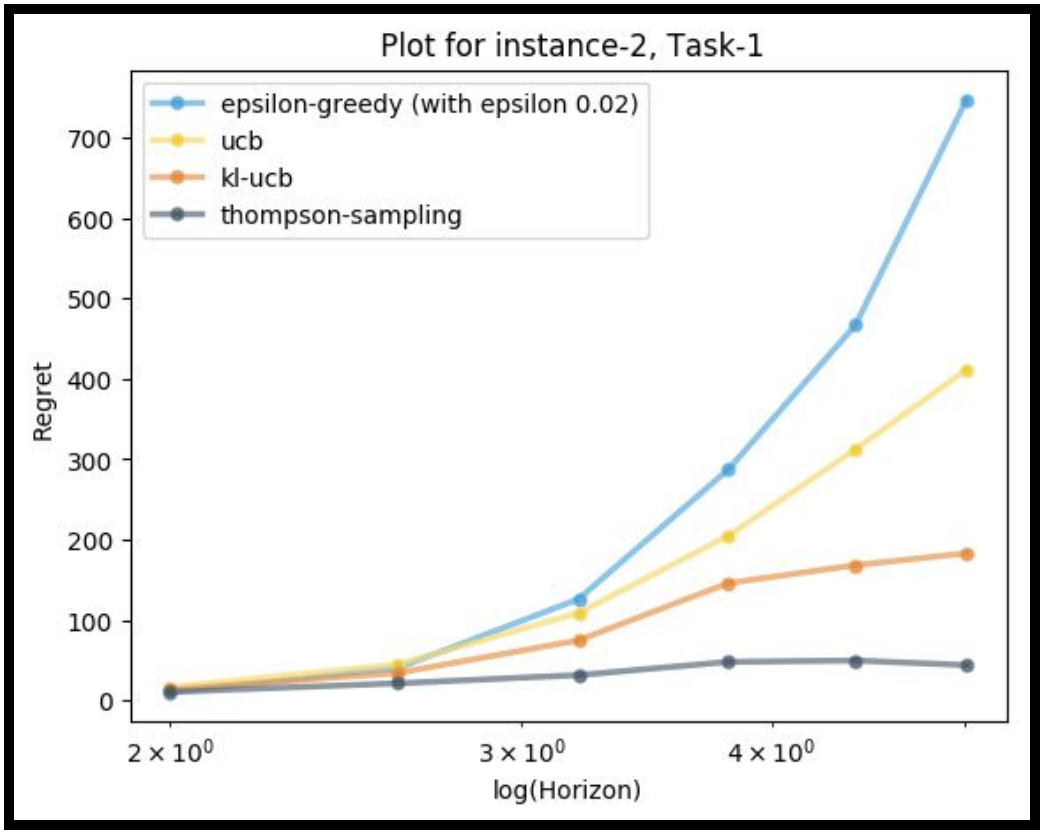
Task 3:

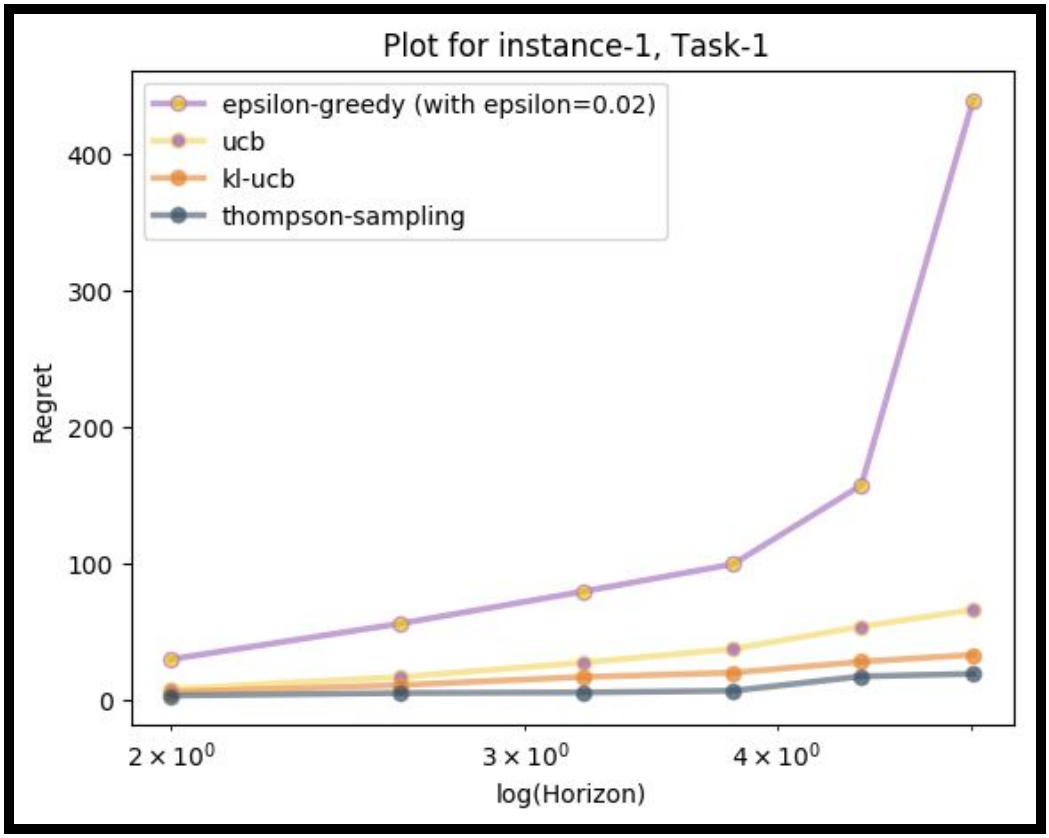
Epsilons tried were : 0.0002; 0.002;0.2;0.3;0.4;0.5;0.6;0.7;0.8;0.9
The average of regret was calculated over 50 seeds:

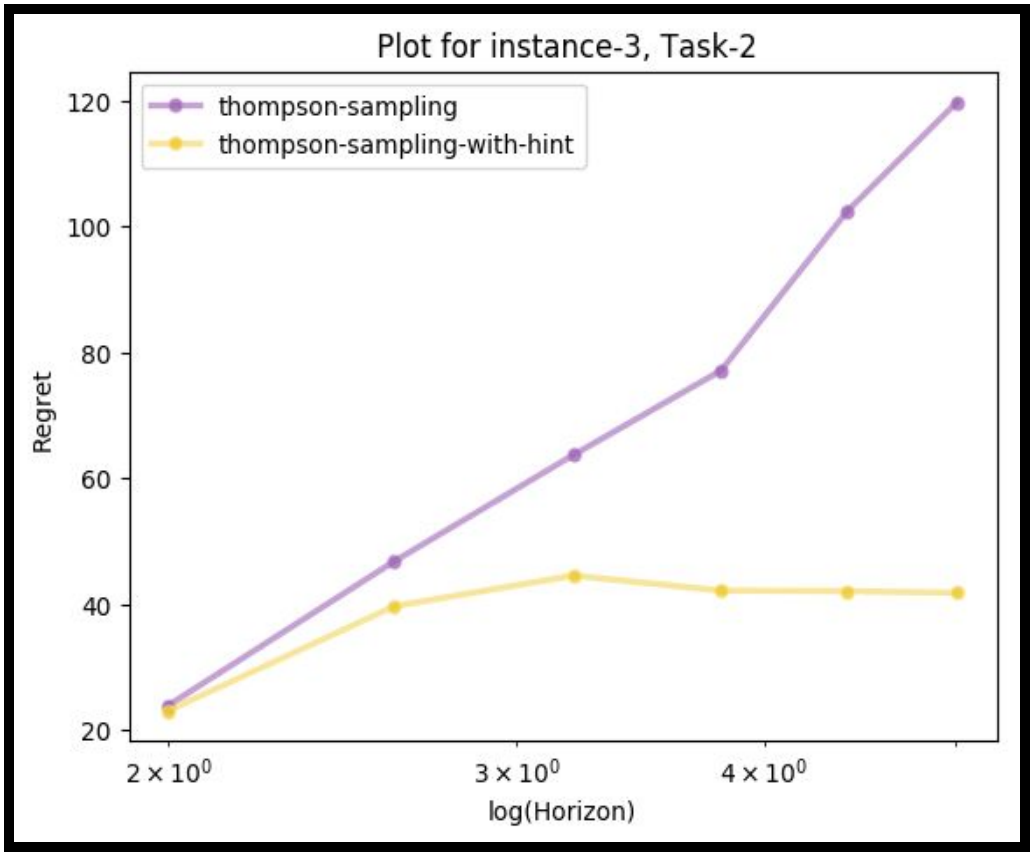
1. For epsilon=0.0002
   Regret for instance 1 was: 39.25
   Regret for instance 2 was: 50.25
   Regret for instance 3 was: 503.408
2. For epsilon=0.002
   Regret for instance 1 was: 7.355
   Regret for instance 2 was: 5.953
   Regret for instance 3 was: 21.51
3. For epsilon=0.2
   Regret for instance 1 was: 82.213
   Regret for instance 2 was: 79.74
   Regret for instance 2 was: 175.91
   Therefore those values of epsilon for all instances are:
   epsilon_1=0.0002
   epsilon_2=0.002(for that fact even epsilon_2 = 0.02 satisfies the conditions, given epsilon_1 and epsion_3 are the same)
   epsilon_3=0.2
   The regret of epsilon_2 is less than that of epsilon_1 and epsilon_3

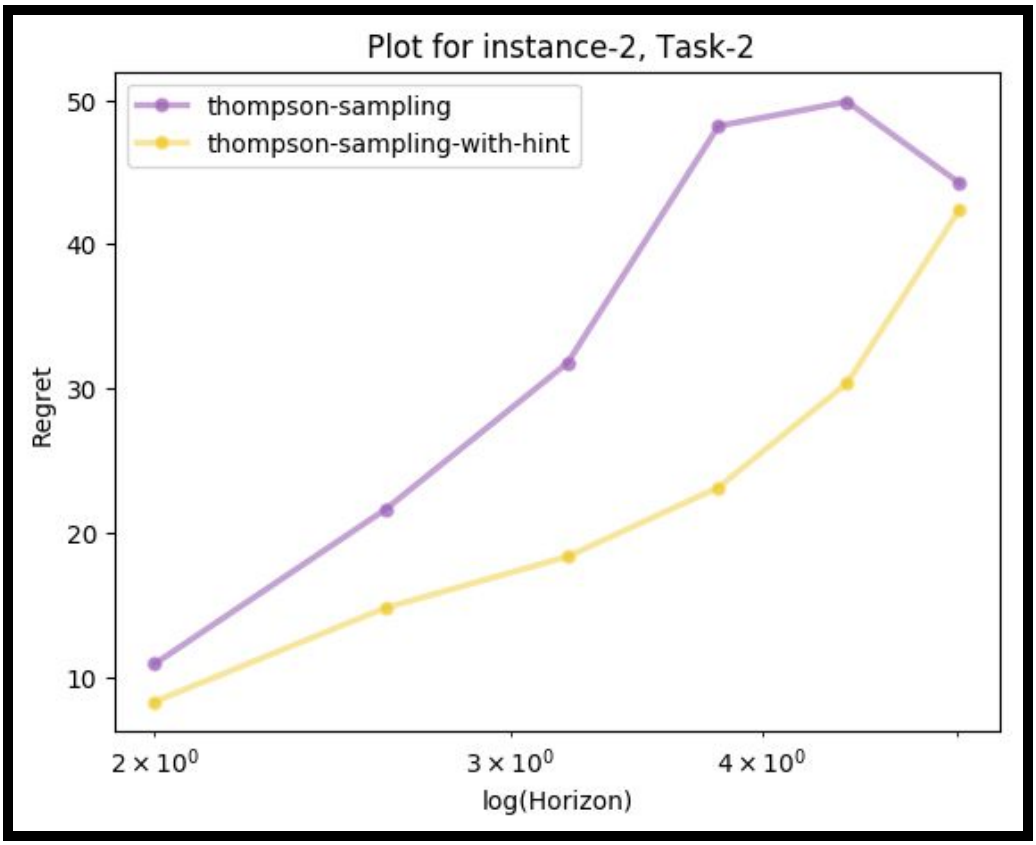
Task 4: Plots from data

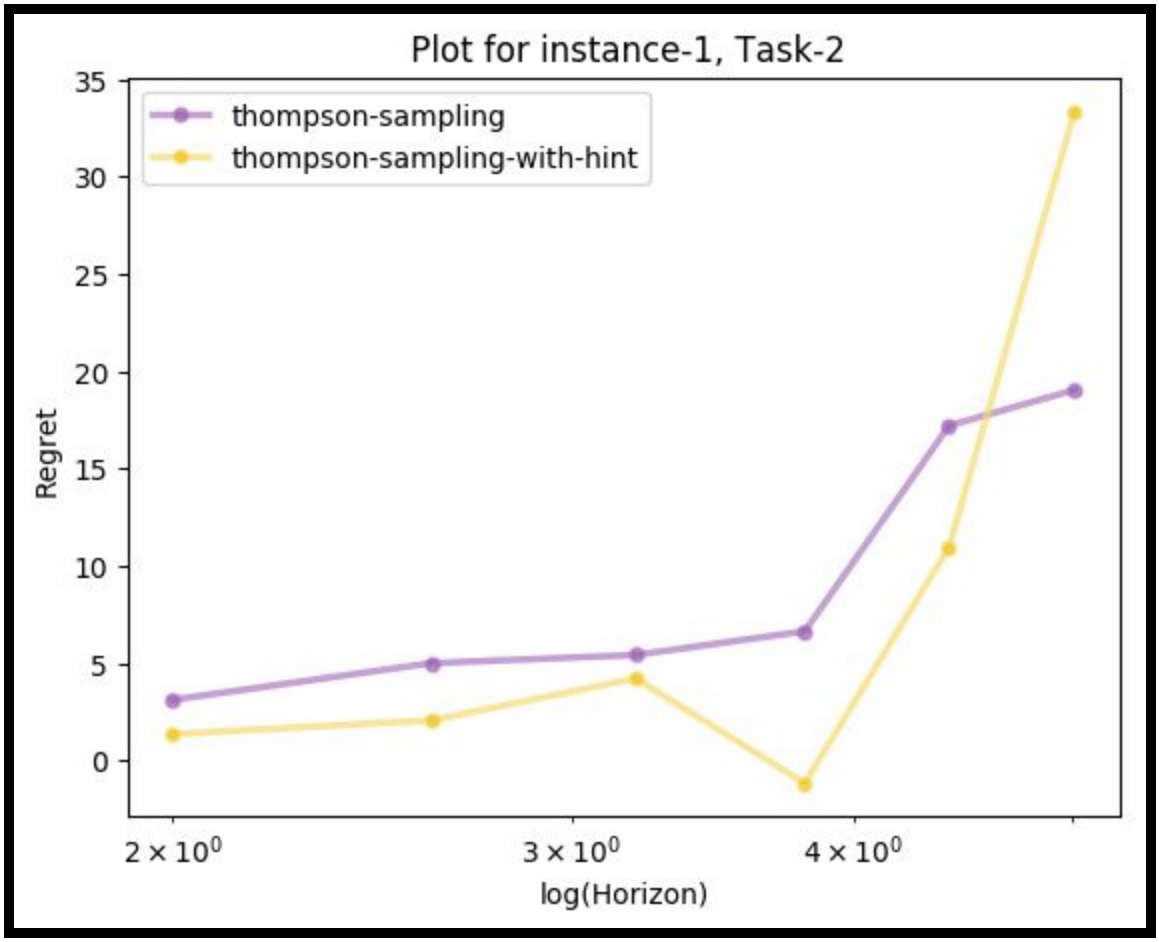



Plot for instance-2, Task-1
epsilon-greedy (with epsilon 0.02)
ucb
kl-ucb
thompson-sampling
Regret
0
100
200
300
400
500
600
700
2×10^0
3×10^0
4×10^0
log(Horizon)


Plot for instance-1, Task-1
epsilon-greedy (with epsilon=0.02)
ucb
kl-ucb
thompson-sampling
Regret
0
100
200
300
400
2×10^0
3×10^0
4×10^0
log(Horizon)


Plot for instance-3, Task-2
thompson-sampling
thompson-sampling-with-hint
120
100
80
60
40
20
Regret
2×10^0
3×10^0
4×10^0
log(Horizon)


Plot for instance-2, Task-2
thompson-sampling
thompson-sampling-with-hint
50
40
30
20
10
Regret
2×10^0
3×10^0
4×10^0
log(Horizon)


Plot for instance-1, Task-2
thompson-sampling
thompson-sampling-with-hint
Regret
35
30
25
20
15
10
5
0
2×10^0
3×10^0
4×10^0
log(Horizon)