CS 747 Programming Assignment 1

Name : Tamoghno Kandar
Roll no : 190100126

September 9, 2021

Task 1 - Implementing the Algorithms

Epsilon Greedy

This algorithm explores with a small probability ϵ uniformly at random. A list of means of each arm is maintained and updated at every time step. It is initialised with all zeros. At each time step, it generates a random number between 0 and 1, checks if this is greater than ϵ and accordingly exploits the arm with the maximum mean or explores respectively. In case of multiple arms having equal maximum mean, an arm is chosen at random from all the maximal arms. Once this is done, the arm is sampled and according to the reward obtained, the means are updated.

UCB

This algorithm takes into account the means as well as the uncertainity in the mean by keeping track of how many times the arm has been sampled from the start. For the implementation, an array of means and an array of UCBs (means summed up with the corresponding uncertainities terms) are stored and updated. The algorithm starts with round robin sampling to have some initial estimate of the UCBs for each arm. At each time step, the arm with the maximum value of UCB is chosen and sampled and the mean of that arm and the UCB values of all the arms are updated. Again, in case of ties, they are broken randomly. Upper confidence bound(UCB) is the sum of the empirical mean & exploration bonus for each arm.

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

KL-UCB

This algorithm is similar intuitively to UCB for the fact that the means and the confidence bounds are used to decide which arm is to be sampled next. The only implementational change is the way this is defined. In order to avoid

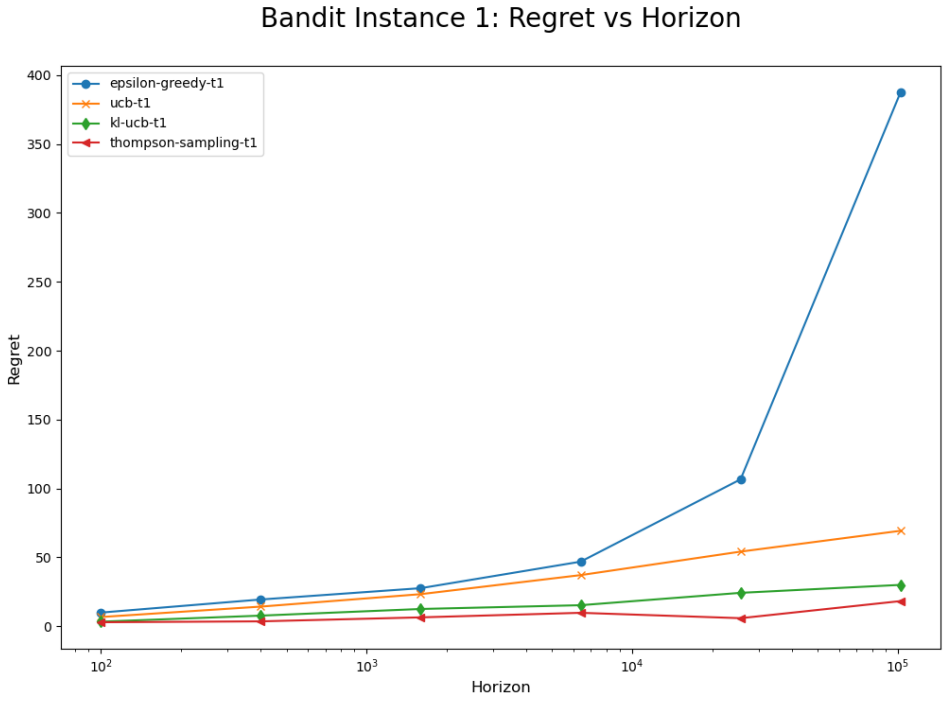
unboundedness during the implementation, two round robins are carried out and then the estimates are calculated by solving an equality which gives us a reward estimate for each of the arm at each time step. Based on that, we sample the arm with the maximal value, breaking ties randomly. Here the upper confidence bound is defined in a slightly different manner such that $ucb\text{-}kl_a^t$ is given by the maximum value of q $\in [\hat{p}_a^t, 1]$ such that:

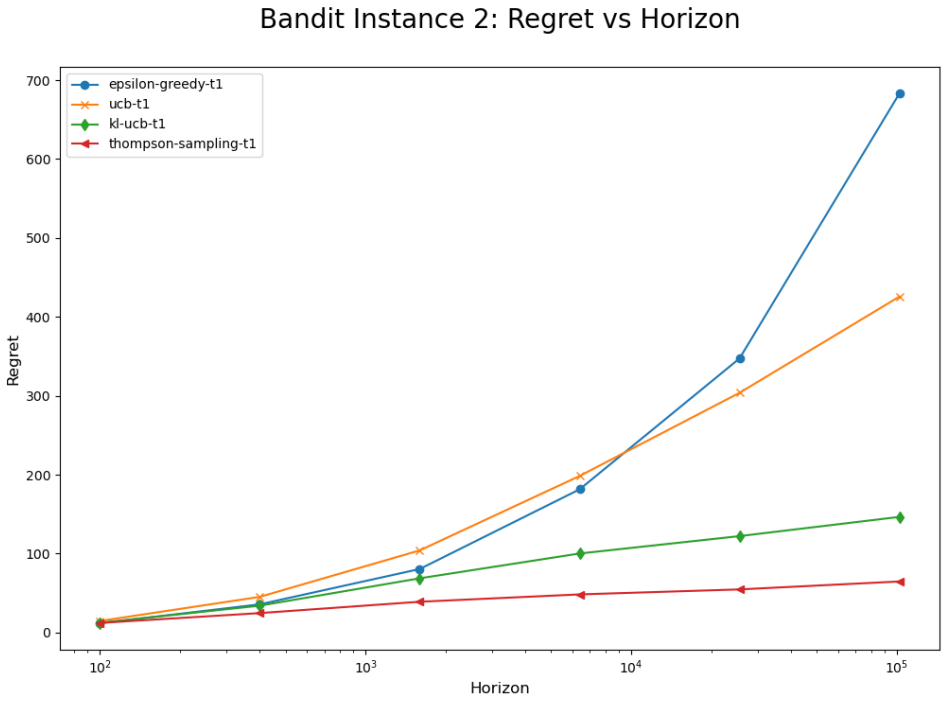
$$u_a^t KL(\hat{p}_a^t) \leq \ln t + c. \ln(\ln t)$$

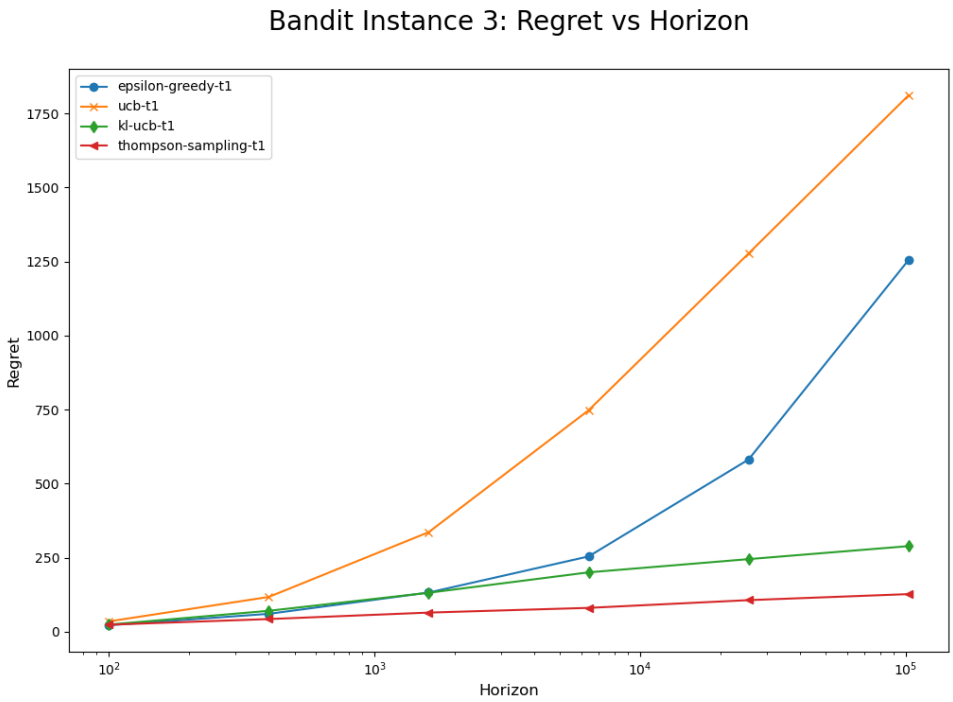
Thomson Sampling

This algorithm relies on a belief of the actual means of the arms, estimated using a Beta distribution over the number of successes and failures for each arm. At each time step, the arm with the maximum belief is sampled and the beliefs are updated according to the reward that it achieves. We start with a round robin to have some notion of belief initially.

Task 1 - Plots




Bandit Instance 2: Regret vs Horizon
epsilon-greedy-t1
ucb-t1
kl-ucb-t1
thompson-sampling-t1
Regret
Horizon
0
100
200
300
400
500
600
700
10^2
10^3
10^4
10^5

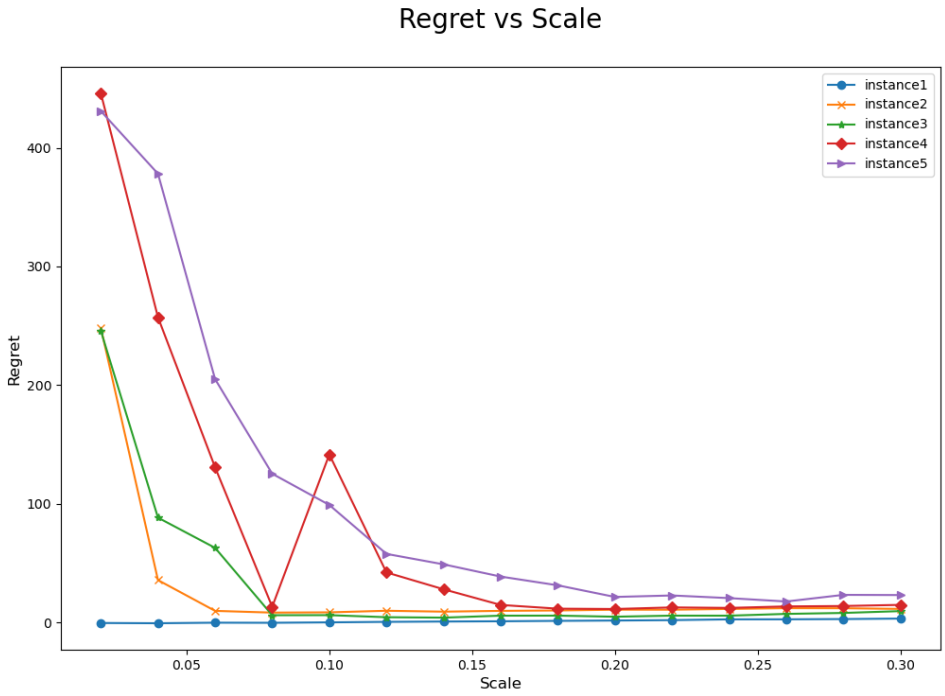


Observations

- Overall, it can be seen that the ϵ-greedy algorithm performs worst and Thompson Sampling performs best for different instances with consistently better performance than other algorithms.
- The graphs for UCB, KL-UCB and Thompson Sampling appear linear for higher horizons for a logarithmic scale of the x-axis thus implying that these algorithms accumulate logarithmic regret in the long run.
- ϵ-greedy performs best because even though it might pull non-optimal arms during exploration with high probability and the regret gets accumulated.
- KL-UCB performs better than regular UCB as it provides a tighter upper confidence bound than UCB.

Task 2

Aim: To optimise the "scale" c separately on the five instances provided and to interpret the results obtained.



Observations

- At c = 0.3 lowest regret was achieved for all the instances.
- As the value of scale (c) increases from 0.2, there was a sharp decrease in the regret for instances 4 and 5. The fall in regret was less steep for instances 2 and 3. For instance 1, the decrease in regret with increase in scale was most gradual.