CS 747: Assignment 1

Mithilesh Vaidya
17D070011

1 T1

Methodology

- For Epsilon-Greedy, UCB and KL-UCB, the algorithms have a round-robin stage in the beginning i.e. each arm is pulled exactly once so that number of pulls $u_a = 1$ for all arms. This ensures that the empirical means of each arm are defined before the algorithm can use these values.
- For Thompson Sampling, the prior is set to the uniform distribution over [0,1] i.e. $\alpha = \beta = 1$
- For KL-UCB: $c = 0$ as mentioned in [1] since it is found to be empirically better. Binary search is used to calculate the value of q. We halt the search if the $|LHS - RHS|$ (in the inequality for q) dips below a tolerance parameter δ, which is set to: $\delta = 10^{-4}$

2 T2

For Thompson Sampling with Hint: the main idea is to maintain a discrete probability distribution of each arm's belief over the given sorted means. For each arm, we sample a value from the sorted means array. The arm which chose the highest true mean should be ideally pulled. If arm i has a mean which is the j^{th} element in the sorted means array, the distribution should ideally converge so that $armBelief(i,j) = 1 \; \forall \; i$ and $armBelief(i,k) = 0 \; \forall \; i, \; k \neq j$.

1. Let number of arms be N. Maintain a NxN matrix called *armBelief* where *armBelief(i, j)* denotes the probability that true mean of arm i is *sM[j]* where *sM* is the sorted array of means which are used as a hint. Initially, $armBelief(i,j) = 1/N \quad \forall \; i, j$ since we have no prior knowledge.
2. For each arm i, choose true mean j with probability *armBelief(i, j)* i.e. $score_i = sM[j]$ with probability *armBelief(i, j)*.
3. Play the arm which has the highest score i.e. play $a_t = \underset{i}{\text{argmax}} \; score_i$

4. Get reward $y \sim Bernoulli(\mu[a_t])$ where μ is the original unsorted vector of arm means.

5. Once we get a reward, update the belief distribution using the Bayesian Inference equation:

$$armBelief(a_t) \leftarrow \frac{armBelief(a_t) * \mu^y * (1-\mu)^{1-y}}{C} \tag{1}$$

   where C is a normalisation constant to ensure that the probabilities for $armBelief(a_t)$ sum up to 1.

Randomly breaking ties in *argmax* of step 3 in the above algorithm gives poor performance for instance 1.
To tackle this, we introduce another belief distribution: Of the true means over the arms called *muBelief* where *muBelief(i, j)* denotes the probability that *sM[i]* is the true mean of arm j. The intuition is: from the previous step, we know which true mean is the most likely. But we don't know which arm corresponds to that true mean value.

- In case of a tie in step 3 i.e. multiple arms choose $sM[j]$ which turns out to be $\max_i score_i$, we use the *muBelief* distribution to choose arm i with probability *muBelief(j, i)*.

- The Bayesian update rule for this distribution is:

$$P(sM[j] : \text{arm i}|y) = \frac{P(y|sM[j] : \text{arm i}) * P(sM[j] : \text{arm i})}{P(y)}$$

  where *sM[j] : arm i* means that sM[j] is the mean of arm i. This gives us the update equation:

$$muBelief(j,i) \leftarrow \frac{armBelief(i,j) * muBelief(j,i)}{K}$$

  where K is the normalisation constant to make it a probability distribution.

3 T3

I varied the value of epsilon (ϵ) and noted the regret by averaging out the results over 50 seeds. As expected, very small values leads to insufficient exploration while very large values leads to insufficient exploitation. Both cases give very high value of regret.
For each instance, we can observe a sweet spot - an intermediate value which strikes the right balance between exploration and exploitation for that particular instance. These values are the desired ϵ_2.

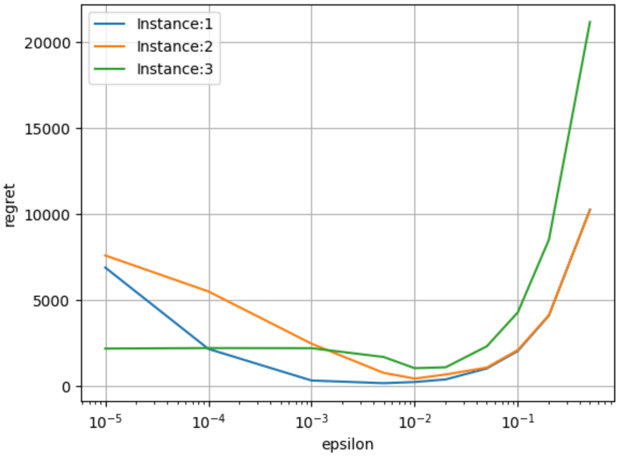

Figure 1: Regret as a function of ϵ

From the plot, we can conclude:

	i-1	i-2	i-3
ϵ_1	0.0001	0.001	0.001
ϵ_2	0.005	0.01	0.01
ϵ_3	0.1	0.2	0.5

4 T4

Given below are the plots for Regret vs Horizon for task 1.

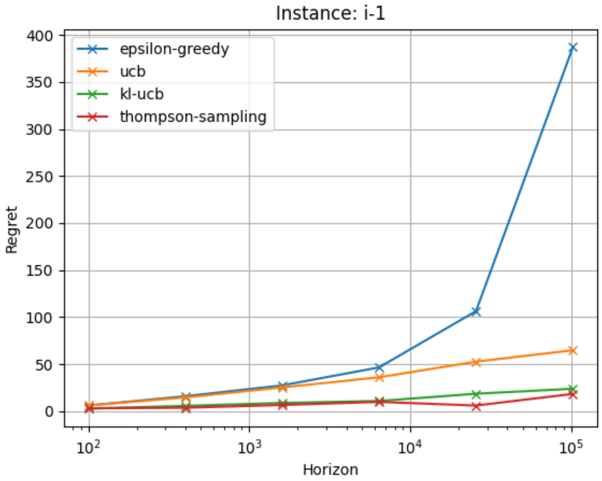

Figure 2: Regret for instance 1

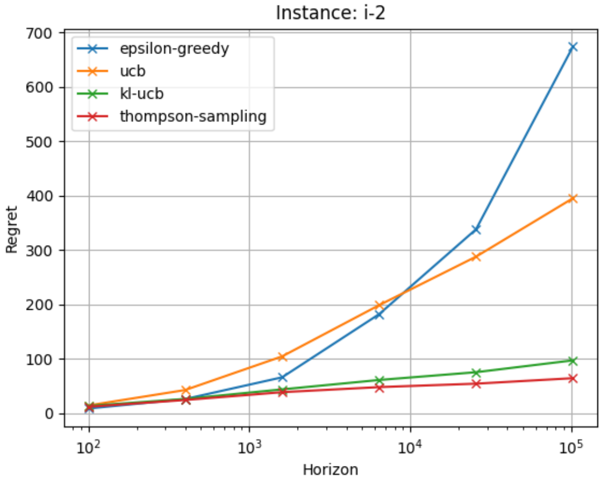

Figure 3: Regret for instance 2

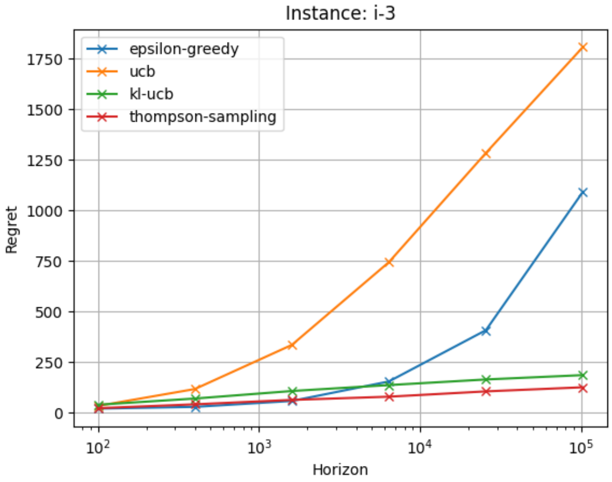

Figure 4: Regret for instance 3

Observations:

- As expected, regret for Epsilon-Greedy is linear (seen as an exponential curve since we are plotting $ln(T)$ vs Regret.
- On the other hand, UCB, KL-UCB and Thompson Sampling show logarithmic regret as predicted.
- Thompson Sampling gives the least regret for all instances for almost all horizons.
- We can observe the following general trend in regret:
  Thompson Sampling < KL-UCB < UCB < Epsilon-Greedy
- For instance-3, UCB performs worse than epsilon-greedy. However, it's plot is linear after T = 400 while the plot for Epsilon-Greedy is exponential. If we run the algorithm for even longer horizons, we can expect UCB to outperform Epsilon-Greedy. It is a sign that the constant for UCB in it's regret formula is significant.

Plots for task 2:

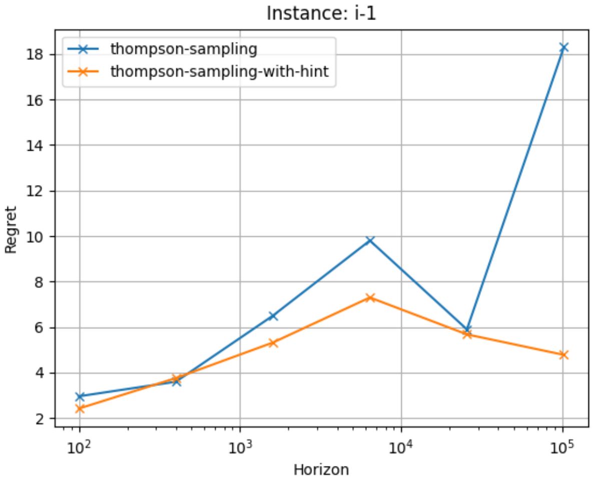

Figure 5: Regret for instance 1

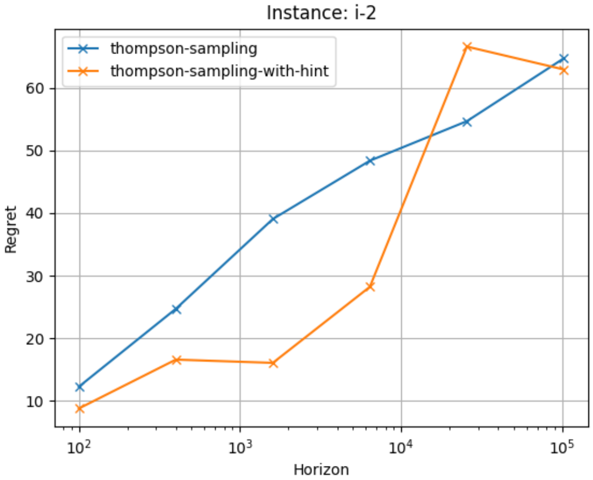

Figure 6: Regret for instance 2

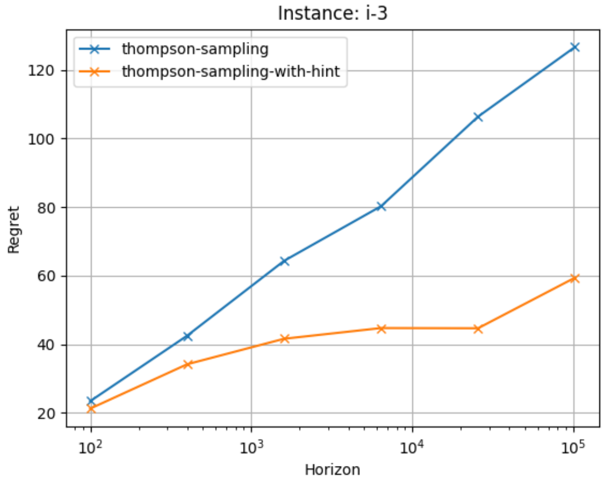

Figure 7: Regret for instance 3

Observations:

- Thompson Sampling With Hint (TSWH) outperforms the simple Thompson Sampling (TS) in (almost) all instances and all horizons.
- In instance 2, TSWH performs worse than TS at one horizon (T = 25600) but it is mostly due to random seeds. If we average out over different seeds or more runs, we can safely expect the aberration to vanish.
- The difference between the two widens as the number of arms increases. Hence, we can predict that as the number of arms goes up, knowing their means is much more valuable and it's exploitation gives even more benefit over TS.

References

[1] The KL-UCB Algorithm for Bounded Stochastic Bandits and Beyond Aurélien Garivier and Olivier Cappé, 2011