CS747: Assignment 1 Report

Fundamentals of Intelligent & Learning Agents

Name: **Siddharth Saha**
Roll no: **170100025**
Due Date: **25 Sept, '20**

1 Introduction:

The objective of this report is to present a comparison across different algorithms that sample the arms of a stochastic multi-armed bandit in order to minimise regret. We consider as given:

- i.i.d rewards from each arm
- Rewards are drawn from a Bernoulli distribution
- Reward means lie in open interval $(0, 1)$

2 Task 1: Sampling Algorithms

2.1 Assumptions in Algorithm Implementation

1. In the very first few pulls, empirical probability means are undefined in *Epsilon-greedy*, *UCB* and *KL-UCB*. The implementation handles this by performing one cycle of **Round Robin** exploration.
2. Thompson-sampling algorithm did not require Round Robin exploration.
3. In Epsilon-greedy, the value of ϵ was set as `0.02`.
4. Ties are broken based on the arm returned by `numpy.argmax()` function. The arm corresponds to the first occurrence of that value in the numpy array.
5. The implementation chooses $c = 3$ in KL-UCB as per the algorithm description that $c \geq 3$. It is to be noted, the paper suggests that $c = 0$ can be chosen for further optimal performance in simulations.
6. The implementation of KL-UCB numerically solves for `q` using binary search with precision of `1e-3`.
7. The implementation is forced to invoke `numpy.random.seed(n)` specifically for beta sampling. The output of beta sampling in the docker image (`python 3.8`) and virtual environment (`python 3.6`) were discovered to be different for the same `random.seed(n)`.

2.2 Interpretations and Observations

1. Instances 1 and 2 clearly demonstrate the order of regrets:
   Epsilon-greedy > *UCB* > *KL-UCB* > *Thompson-sampling*
2. Instance 3 indicates increasing slope of Epsilon-greedy and near-constant slope of UCB. Thus, even here Epsilon-greedy would surpass UCB over a larger horizon. Asymptotically, the same order as above would be achieved.
3. UCB and KL-UCB start off worse than Epsilon-greedy. Over larger horizons, Epsilon-greedy surpasses KL-UCB earlier than UCB.

2.3 Plots

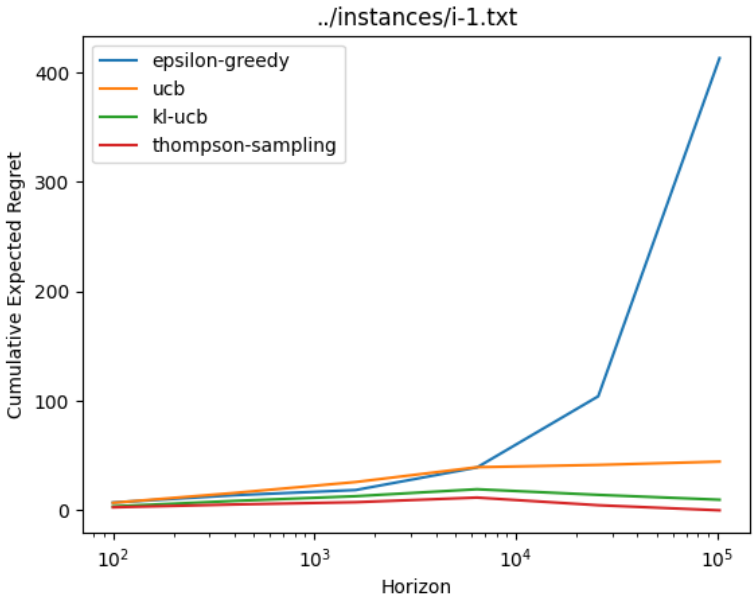

Figure 1: Comparison over 50 randomSeeds on Instance 1 (Number of arms = 2)

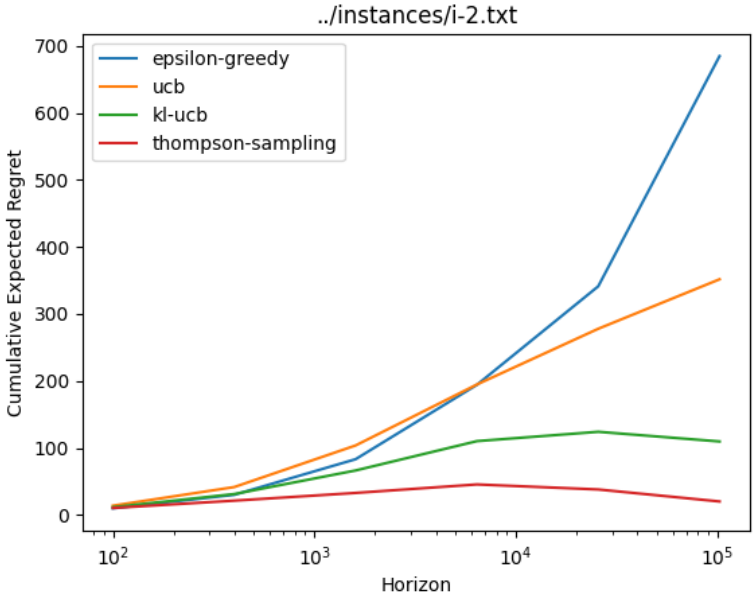

Figure 2: Comparison over 50 randomSeeds on Instance 2 (Number of arms = 5)

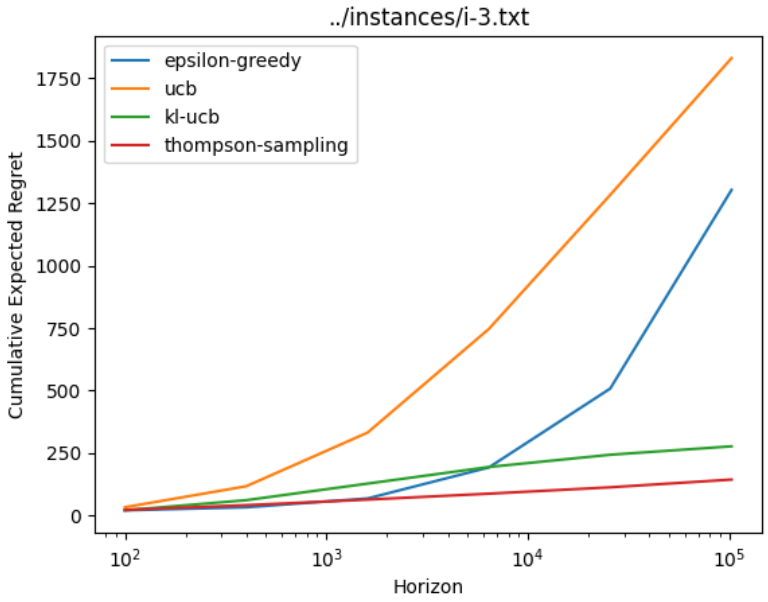

Figure 3: Comparison over 50 randomSeeds on Instance 3 (Number of arms = 25)

3 Task 2: Thompson Sampling and Hint

3.1 Algorithm Design

1. The implementation is passed the sorted list of reward means to be utilised as hint.
2. We utilise some ideas from Bayesian Inference. Here we are provided discrete values that each reward mean could hold, rather than the continuum that we had to consider in earlier algorithms.
3. Here we consider the prior that each arm is equally likely to bear any of these discrete values.
4. We tabulate these probabilities in the form of a confidence matrix. We pick the arm that we have highest confidence as being the optimal arm. (Tiebreaks using `argmax`)
5. Based on the outcome from that pull, we evaluate conditional probability corresponding to each discrete value for that arm.
6. We normalise across all the probabilities for the pulled arm and update the table.

3.2 Interpretations and Observations

1. Across all instances and horizons, hint algorithm performs significantly better.
2. An odd behaviour in the plot is that regret decreases with increase in horizon in Instance 1 and 2. To investigate this behaviour, the code where optimal arm is pulled every time was executed. Averaging across randomSeeds `0...49`, that yielded negative regret too.
3. Thus a possible reason for this behaviour can be attributed to not averaging across sufficient number of distinct randomSeed runs.
4. The key observation is that the size of the solution space significantly diminishes from a continuum to a set of discrete values with the given hint.
5. We have been successfully able to incorporate that hint in our pulls using Bayesian Inference.

3.3 Plots

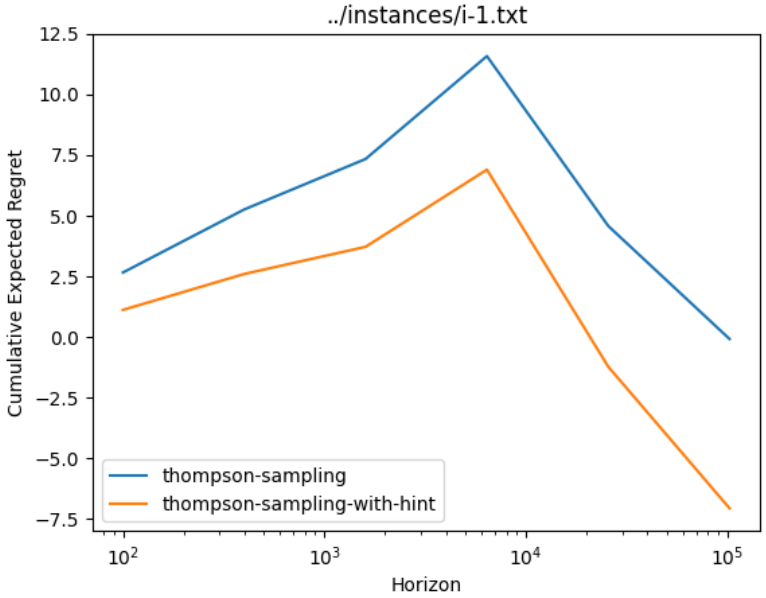

Figure 4: Comparison over 50 randomSeeds on Instance 1 (Number of arms = 2)

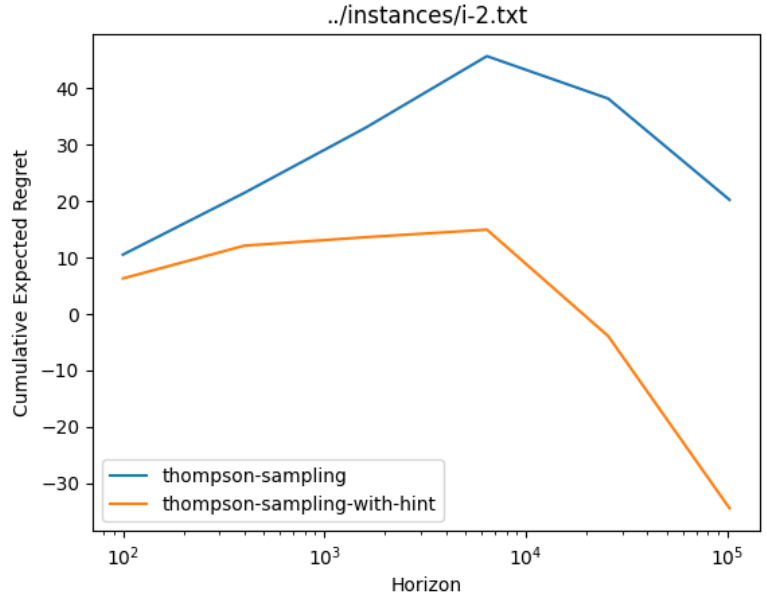

Figure 5: Comparison over 50 randomSeeds on Instance 2 (Number of arms = 5)

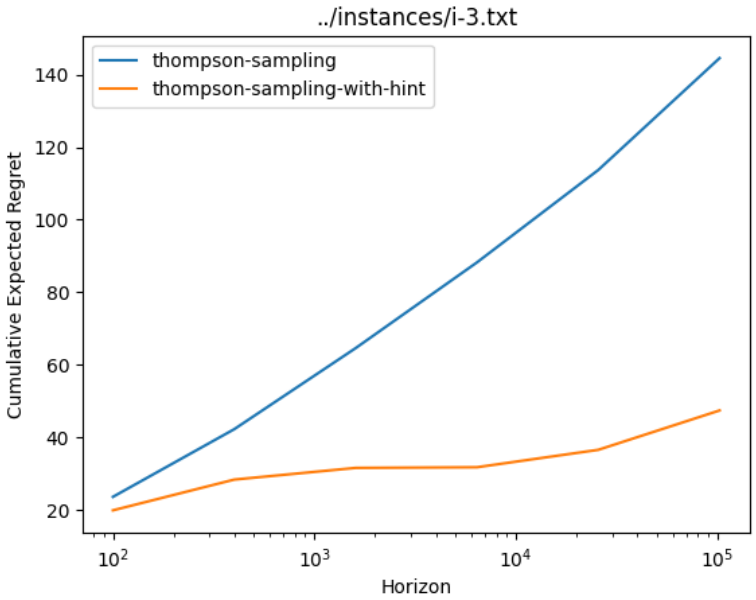

Figure 6: Comparison over 50 randomSeeds on Instance 3 (Number of arms = 25)

4 Task 3: Obtaining Epsilon Values

$$\epsilon_1 = 0.0002 < \epsilon_2 = 0.002 < \epsilon_3 = 0.1$$

Instance	$\epsilon_1 = 0.0002$	$\epsilon_2 = 0.002$	$\epsilon_3 = 0.1$
1	1868.98	237.5	2036
2	5617.68	1951.34	2071.16
3	2230.3	1556.92	4304.82

Cumulative expected regret averaged over 50 randomSeeds is least for ϵ_2 in above table.