# CS 747 Assignment 1 Report

Rishabh Dahale
17D070008

September 25, 2020

## 1 Assumption

Here is the list of assumptions for the algorithms where necessary.

1. KL-UCB: For this algorithm, there are 2 parameters to tune.

    (a) $c$ for the calculation of

    $$ucb\text{-}kl_a^t = max\{q\epsilon[\hat{p}_a{}^t, 1] \mid u_a^t KL(\hat{p}_a^t, q) \leq ln(t) + cln(ln(t))\}$$

    As the proof of this algorithm suggests that $c \geq 3$, I have selected $c = 5$. This is constant across all the instances and all the seeds

    (b) To calculate the value of q in the above equation, as the KL-Divergence is monotonically increasing in the range $[\hat{p}_a^t, 1]$, I have used binary search to get the required values. The terminating condition for the binary search is

    $$|ln(t) + cln(ln(t)) - u_a^t KL(\hat{p}_a^t, q)| \leq \delta$$

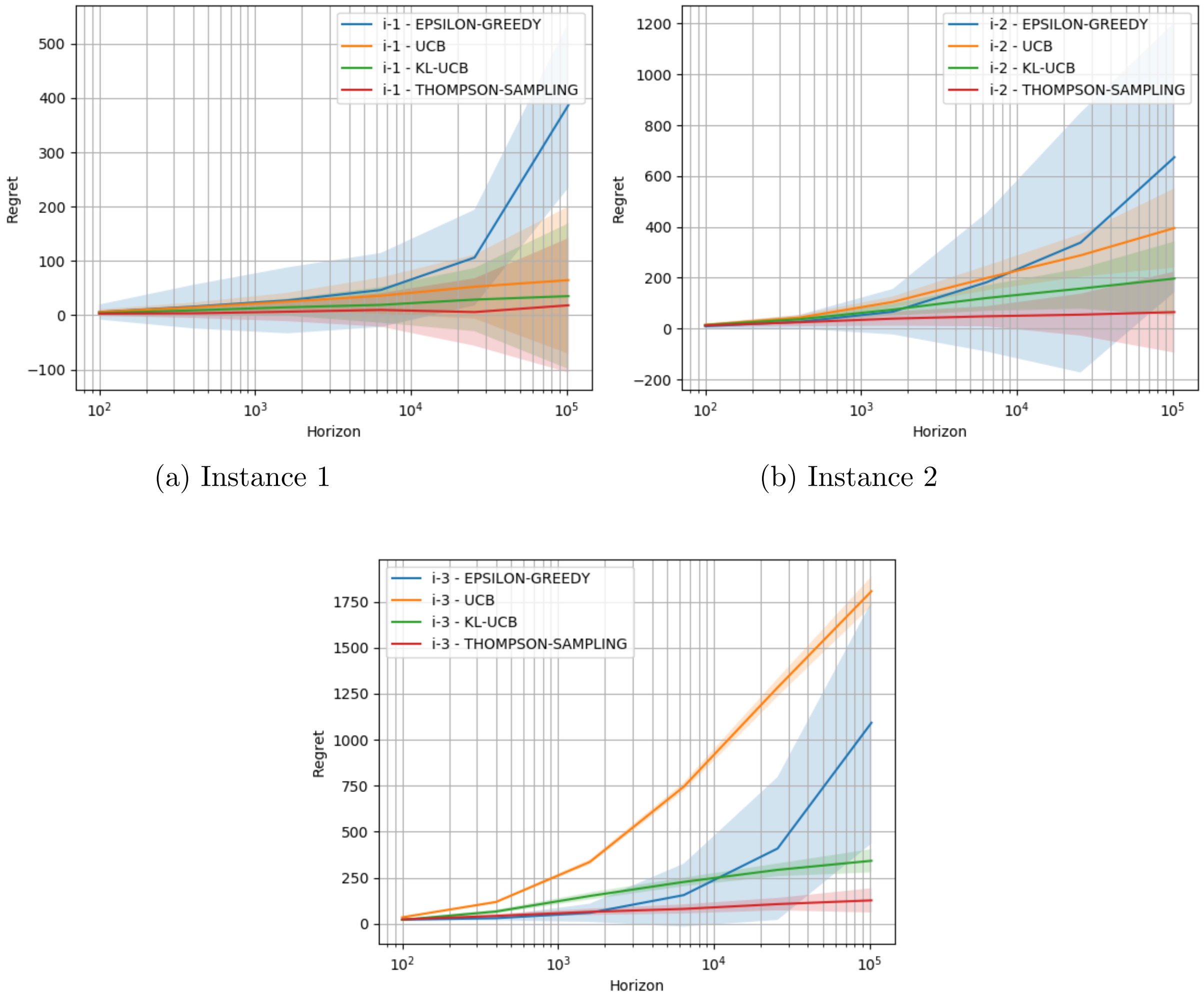
    For this I have used $\delta = 10^{-4}$

    Also it needs to be ensured that $ln(t) + cln(ln(t)) > 0$. For this, each arm is pulled for $\lceil \frac{3}{n} \rceil$ times before calculating the value of kl-ucb.

2. For the algorithms $\epsilon$-Greedy and UCB, each arm is pulled once before starting the actual algorithm. These initial pulls are accounted in the in the horizon.

## 2 Task 1

Instance 1 have 2 arms, instance 2 have 5 arms and instance 3 have 25 arms. From the plots we can observe that for instance 1 and 2 $\epsilon$Greedy algorithm have the worst regret for all the horizons, whereas the Thompson Sampling shows the best regret. For instance 3, although UCB shows worst regret, the slope of UCB tends to decrease indicating that as the horizon will tend to infinity, it will come below $\epsilon$Greedy. The shaded region around the plots is the standard deviation of the experiments. When the number of arms are less it can be seen that



(a) Instance 1

(b) Instance 2

(c) Instance 3

Figure 1: Regret vs horizon for the difference instances for algorithms $\epsilon$Greedy, UCB, KL-UCB, and Thompson sampling. Note that the x axis is logarithmic. The region shaded around the bold lines is the standard deviation of the experiments.

the standard deviation is high for almost all the strategies. This is because the difference if the probability of success for optimal arm and second optimal arm is huge, making a huge impact while exploration.
The results of instance 3 are specifically interesting from the point of view of variance. In instance 3, there is an increase in the number of arms, but there is also a decrease in the sub-optimal gap of the arms. The variance of $\epsilon$Greedy is still highest in this case because during exploration, it selects all the sub-optimal arms with equal probability, whereas for other algorithms incorporate the empirical mean of the arms during exploration. This results in non uniform exploration of the sub optimal arms, resulting in higher exploration weight to arms closer to optimal arm. Because of this, the mean reward for exploration phase for UCB, KL-UCB and Thompsom Sampling is much higher than that of $\epsilon$Greedy, making the

standard deviation of these algorithms very small.
It can be noted that the regret of thompson sampling is almost linear wrt horizon in the logarithmic scale, which suggest that the regret is $\mathcal{O}(log(T)$

# 3 Task 2

## 3.1 Thompson Sampling with Hint

The main principle behind Thompson Sampling is the updating of belief given a new observation. This is carried out through Bayes rule. Let the vector $\hat{\Theta}$ denote the permutation of the real means (hint given to the algorithm). Given this hint, we can maintain 2 belief matrices

1. For every arm we can maintain a belief over $\hat{\Theta}$. Let's call this `armBelief`. Each row of this matrix is the belief vector of an arm. $\texttt{armBelief}_{ij}$ is the probability with which `arm i` have the true mean $\theta_j$. The matrix should satisfy

$$\sum_{j=0}^{N-1} \texttt{armBelief}_{ij} = 1 \;\; \forall i$$

where N is the number of arms ($|\hat{\Theta}|$ to be precise).

2. For every $\theta \in \hat{\Theta}$, maintain a belief over arms. Lets call this `muBelief`. $\texttt{muBelief}_{ij}$ is the probability with which $\theta_i$ is associated with `arm j`. This matrix should satisfy

$$\sum_{j=0}^{N-1} \texttt{muBelief}_{ij} = 1 \;\; \forall i$$

Intuitively the two belief matrices may look like transpose of one another, but they are not. The update equation for both the belief are different.

1. Update of `armBelief`
After every time step `t`, we can write

$$\texttt{armBelief}_{ij}(t) = P(\theta = \theta_j | y_0, y_1, ...y_t)$$

Let the arm played at time t be `arm i`, reward obtained be $y_t$

$$P(\theta = \theta_j | y_0, y_1, ...y_t) = \frac{\texttt{armBelief}_{ij}(t-1) P(y_t | \theta = \theta_j)}{\sum_j \texttt{armBelief}_{ij}(t-1) P(y_t | \theta = \theta_j)}$$

As the problem statement specifies that the reward distribution is the Bernoulli, we can write $P(y_t|\theta = \theta_j) = \theta_j^y{}_t (1-\theta_j)^{1-y_t}$

2. Update of $\texttt{muBelief}_{ij}$
   After every time step t, we can write

$$\texttt{muBelief}_{ij}(t) = P(A = \texttt{arm j}|\theta = \theta_i)$$

. Let the $\theta$ predicted by $\texttt{arm i}$ at time t by $\theta_i$. By Bayes rule, we can write

$$P(A = \texttt{arm j}|\theta = \theta_i) = \frac{P(\theta = \theta_i|A = \texttt{arm j})P(A = \texttt{arm j})}{\sum_j P(\theta = \theta_i|A = \texttt{arm j})P(A = \texttt{arm j})}$$

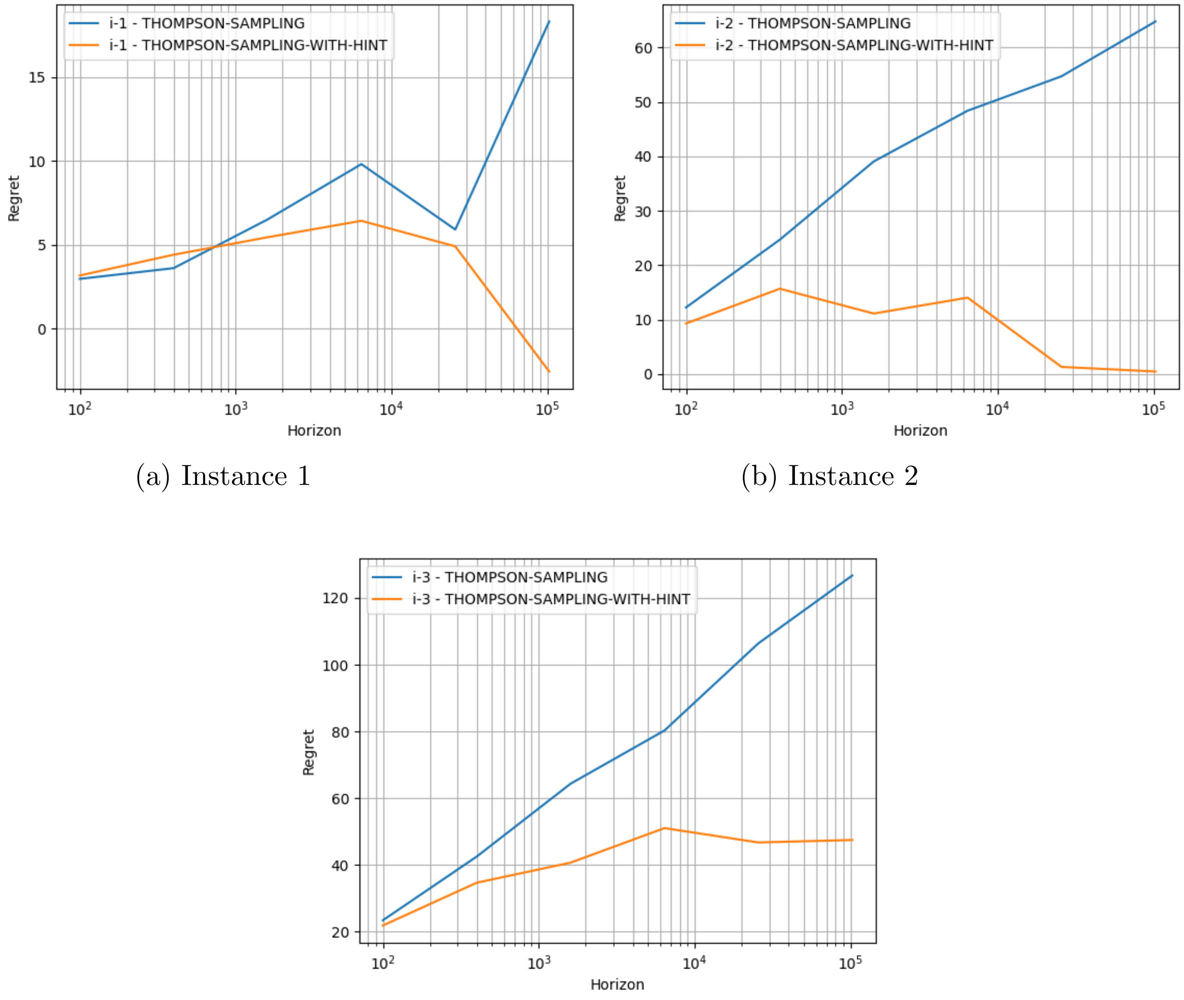
$$= \frac{\texttt{armBelief}_{ji}(t-1) \times \texttt{muBelief}_{ij}(t-1)}{\sum_j \texttt{armBelief}_{ji}(t-1) \times \texttt{muBelief}_{ij}(t-1)}$$

As it can be seen, the update equations for both the belief matrices are different. Using both the matrices optimally can give a very optimal regret compared to simple Thompson Sampling. Pseudo code for the algorithm is as follows

**Algorithm 1** Thompson Sampling with Hint

```
**Input**: HINT vector (permuted true means of the arms)
$\texttt{armBerilef, muBelief} = \begin{pmatrix} \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \\ \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \end{pmatrix}_{N \times N}, \begin{pmatrix} \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \\ \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \end{pmatrix}_{N \times N}$
for $t = 1, 2, \ldots, H$ do
    For each arm i sample $\theta_{a_i}$ from the distribution $\texttt{armBelief}_i$
    $\theta_{max} \leftarrow max_i \theta_{a_i}$
    $n \leftarrow$ number of arms who sampled $\theta_{max}$
    $i \leftarrow$ index of $\theta_{max}$ in the HINT vector
    if n=1 then
        playArm $\leftarrow argmax_i \theta_{a_i}$
    else
        playArm $\leftarrow$ sample an arm from the distribution $\texttt{muBelief}_i$
    end if
    reward $\leftarrow$ reward observed from playing arm playArm
    for j=1,2,...,N do
        $\texttt{muBelief}_{ij} = \frac{\texttt{armBelief}_{ji} \times \texttt{muBelief}_{ij}}{\sum_j \texttt{armBelief}_{ji} \times \texttt{muBelief}_{ij}}$
        $\texttt{armBelief}_{playArm,j} = \frac{\texttt{armBelief}_{playArm,j} \times \texttt{HINT}(j)^{reward}(1-\texttt{HINT}(j))^{1-reward}}{\sum_j \texttt{armBelief}_{playArm,j} \times \texttt{HINT}(j)^{reward}(1-\texttt{HINT}(j))^{1-reward}}$
    end for
end for
```

It can be seen that in all the instances, the above algorithm works better than Simple Thompson Sampling, which should be expected as we have extra information. For instance 1, although the graph shows negative regret for 102400 horizon, when the experiment was repeated over more random seeds (600-649), the final average regret was very close to 0.



(a) Instance 1

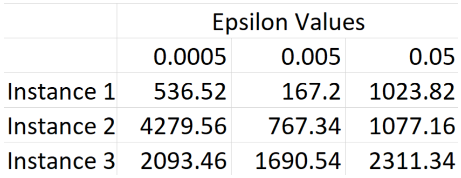
(b) Instance 2

(c) Instance 3

Figure 2: Regret of Thompson Sampling and Thompson Sampling with Hint using above algorithm

# 4 Task 3

For the following values of epsilon, we get the following regret for the $\epsilon$Greedy algorithm This variation is because $\epsilon$ controls the exploration factor of the algorithm. If the explo-



| | Epsilon Values | | |
|---|---|---|---|
| | 0.0005 | 0.005 | 0.05 |
| Instance 1 | 536.52 | 167.2 | 1023.82 |
| Instance 2 | 4279.56 | 767.34 | 1077.16 |
| Instance 3 | 2093.46 | 1690.54 | 2311.34 |

Figure 3: Results of varying $\epsilon$ for $\epsilon$Greedy algorithm

ration carried out by the algorithm is small, it can get stuck with sub-optimal arm during exploitation phase. Whereas if there is large exploration, the algorithm will tend to select sub-optimal arm more number of times resulting in larger regret. As a result, we can tune $\epsilon$ to control the exploration, and hence minimise the regret.