Multiple-Play Stochastic Bandits with Shareable Finite-Capacity Arms

Malyala Preethi Sravani, 200070041
Sakshi Heda, 200070071

Introduction

In a conventional multi armed bandit (MAB) problem, the learner pulls one out of $K \in \mathbb{N}_+$ arms. Each arm has a distribution according to which the reward is selected; the mean of this distribution is not known to the learner. The objective being obtaining maximum reward, the learner has to either play the arm with high uncertainty rewards (exploration), or keep playing the arm with highest empirical mean (exploitation). An algorithm must be formulated that maximises the reward at the end of T time slots.

For many real life situations, we consider the ability to play multiple arms in a single time slot - Multi play multi armed bandit (MP - MAB). However, this setting allows the player to play an arm only once in each time slot.

We hence move to the shareable multi-play multi-armed bandits (*shareable* MP-MAB). In this setting, Each arm can be played any number of times. However, the reward obtained from each arm is limited by its capacity. Such models are used in cognitive radio networks, mobile edge computing, online advertisement placements etc.

Problem Statement

There are $K \in \mathbb{N}_+$ arms, indexed by $1, 2, ..., k$, and have rewards according to the distribution X_k. with means, μ_k (not known to the player). Each of them has a finite rewards capacity, m_k, which is not known to the player. The reward is load dependent, as long as the capacity is not reached. The

expression of reward can be given as: $\min\{a_k, m_k\} \times X_k$. Without loss of generality, we assume $\mu_1 \geq \mu_2 \geq ... \geq \mu_k$.

N number of plays are assigned in each time slot, and the player chooses how these N plays are distributed among the K arms.

In the ideal case, had the means of the distribution been known to the player, the optimal strategy would have been to assign m_1 number of plays to arm 1, m_2 of the remaining number of plays to arm 2 etc. As these values are not known, we estimate these values.

The objective is to estimate the means of the reward distribution and capacity of these arms, while maximising the reward.

Theoretical Background

We implemented the `OrchExplore` algorithm. Given below is the description of the algorithm:

Stage 1: At the initial iterations of the algorithm, it runs PIE in the odd time slots, and PUE in the even time slots. These functions will be defined shortly.

Stage 2: After each run of PIE or PUE, the upper and lower confidence bounds of the candidates are updated as according to the following equations:

$$m^l_{k,t} := max\{\lceil \hat{\nu}_{k,t}/\hat{\mu}_{k,t} + \phi(\tau_t, \delta)\phi(l_t, \delta)\rceil, 1\}$$
$$m^u_{k,t} := min\{\lceil \hat{\nu}_{k,t}/\hat{\mu}_{k,t} + \phi(\tau_t, \delta)\phi(l_t, \delta)\rceil, N\}$$

Once the PUE set $\mathcal{Y}_t = \phi$, `OrchExplore` runs only PIE.

Stage 3: Once both PIE and PUE are empty, ($\mathcal{Y}_t = \phi$ and $\mathcal{E}_t = \phi$), then PIE starts working as *exploitation* - it allocates plays to empirical optimal arms according to these arms' reward capacities.

```
while t <= T do:
    if t is odd or Yt = {} then
        run PIE
    else
        run PUE
    end if
end while
```

Parsimonious Individual Exploration (PIE)

The basic idea of PIE is to maximise the number of plays of empirically optimal arms and minimise that for empirically sub-optimal arms, while keeping the exploration of the empirically sub-optimal arms rare.

The `Oracle` function takes the reward capacity's lower bounds m_t^l and empirical means $\hat{\mu}_k$ of the arms as inputs and outputs a_t^{IE}, that assigns the highest mean to the arm with highest lower bound on capacity. Find $\mathcal{S}_t$, the set of all arms that will be played, and define L_t to be the least favoured among these.

Now pick all those arms whose **KL-UCB** index, $u_{k,t}$ is greater than or equal to the least favoured arm's mean $\hat{\mu}_{k,t}$. The KL-UCB is defines as follows:

$$u_{k,t} := \sup\{q \geq 0 : \hat{\tau}_{k,t}, \mathrm{kl}(\hat{\mu}_{k,t}, q) \leq \log t + 4 \log\log t\}$$

These arms are in a set, $\mathcal{E}_t$. With a probability of 1/2, assign one play from L_t to one arm randomly uniformly selected from $\mathcal{E}_t$. After updating a_t^{IE}, the algorithm pulls the arms the mentioned number of times and observes the rewards. The following values are updated:

- empirical mean, $\hat{\mu}_t$
- KL-UCB indexes, u_t
- effective times of IE, $\hat{\tau}_t$
- time slot index, t

Parsimonious United Exploration (PUE)

In PUE, we prioritise the exploration of high-empirical-mean arms, and assign the arm's upper capacity bound, $m_{k,t}^u$ number of plays to each of the these arms.

We define $\mathcal{Y}_t$ as all those arms in $\mathcal{S}_t - L_t$ for which $m_{k,t}^l \neq m_{k,t}^u$. We increase the empirical mean of these arms by a large positive value, M. These means are denoted by vector, $\hat{\mu'}_t$

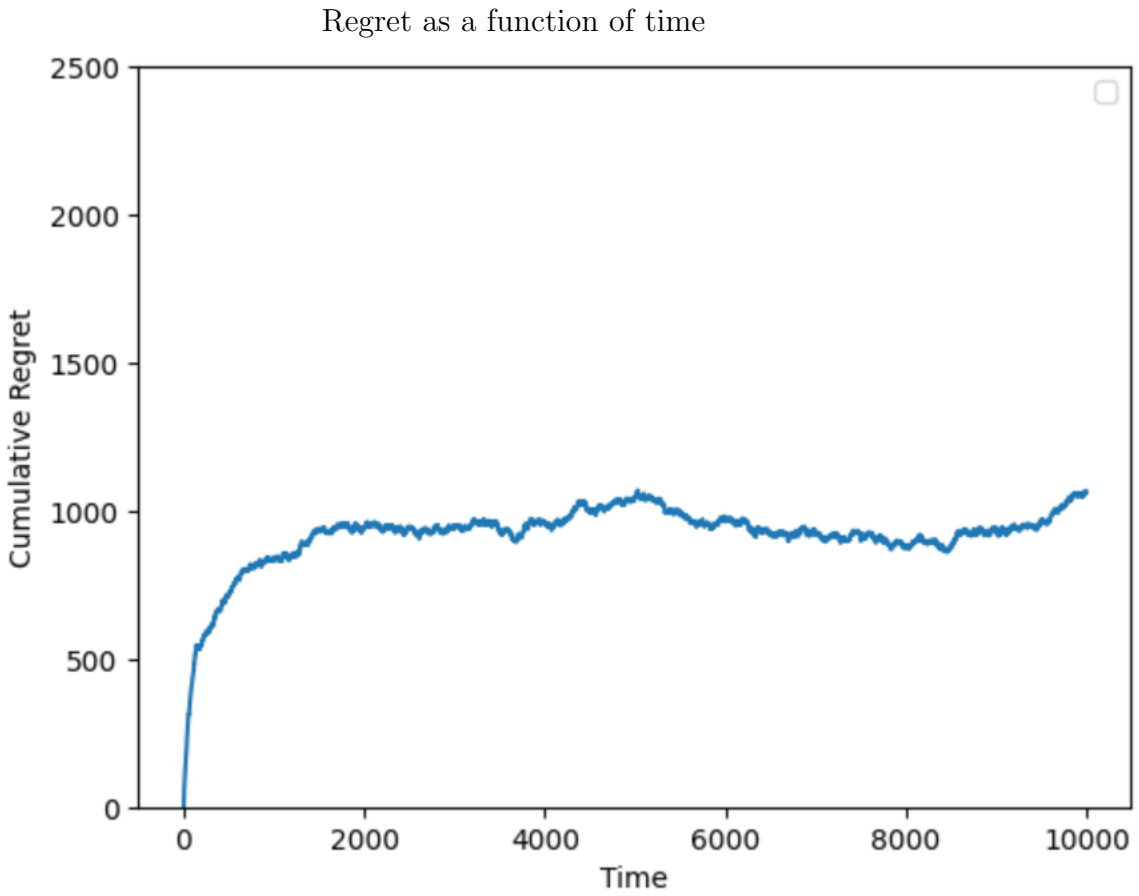
Now, $\hat{\mu'}_t$ and the upper capacity bound, $m_{k,t}^u$ are given as input to the `Oracle` function and the output, a_t^{UE} is used to play the round. The rewards are obtained and the following values are updated:

- "full load" reward mean, $\hat{\nu}_t$

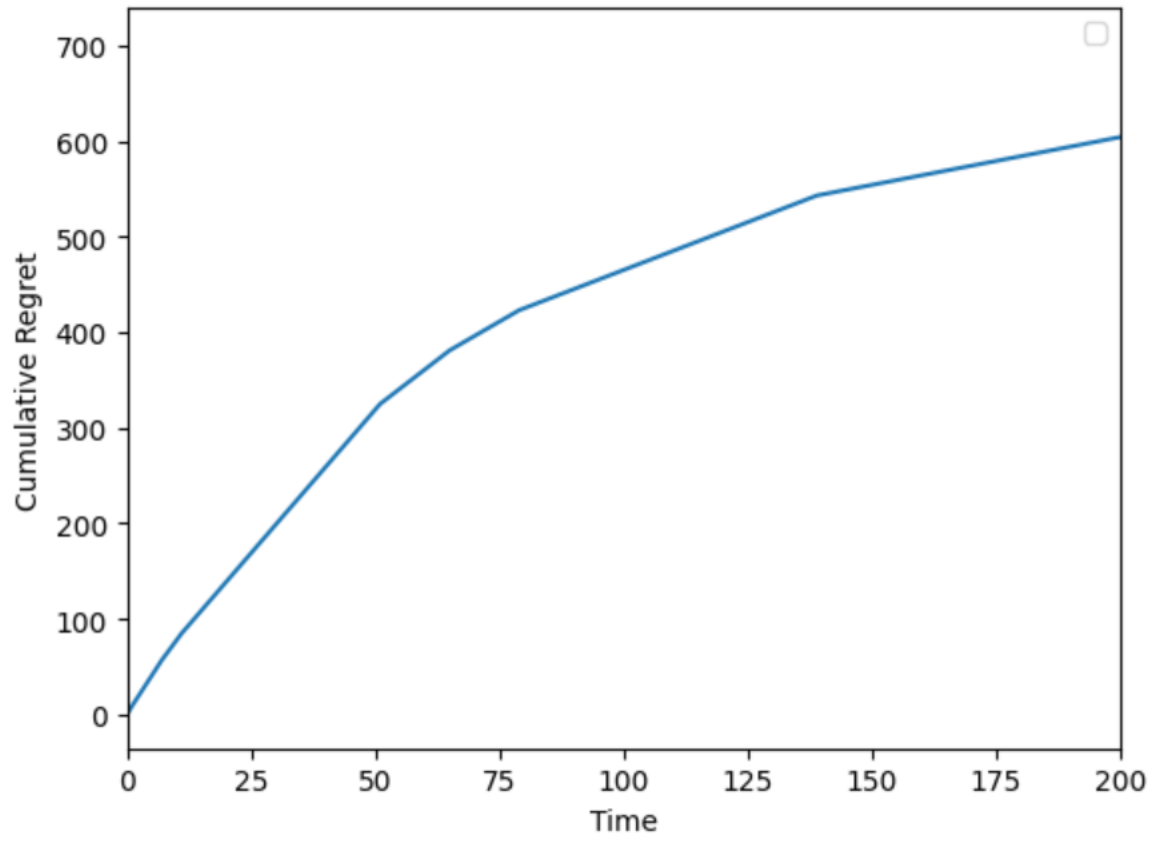
- effective times of UE, $\hat{\imath}_t$
- time slot index, t

Results and Observations

The following is the regret obtained on running the algorithm for 10,000 time slots:



Regret


Cumulative Regret
Time
0
100
200
300
400
500
600
700
0
25
50
75
100
125
150
175
200