ID7123 : CENTER FOR COMPUTATIONAL BRAIN RESEARCH

MULTI-ARMED BANDITS

- Prof. Prashanth L. A. -

Student ID:
Ganga Meghanath
EE15B025

20 January 2019

Contents

1 INTRODUCTION

In a “Multi-armed Bandit problem”, a person must choose between multiple actions (originally comes from the idea of slot machines, the "one-armed bandits"), each with an unknown reward. The goal is to determine the best or most profitable outcome through a series of choices. At the beginning of the experiment, when odds and payouts are unknown, the gambler must determine which machine to pull, in which order and how many times.

Here, we have been provided with the following scenario :

Arms	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
P1	B(0.6)	B(0.4)	B(0.4)	B(0.4)	B(0.4)	B(0.3)	B(0.3)	B(0.3)	B(0.3)	B(0.3)
P2	B(0.62)	B(0.6)	B(0.6)	B(0.6)	B(0.6)	B(0.58)	B(0.58)	B(0.58)	B(0.5)	B(0.5)
P3	B(0.8)	B(0.4)	B(0.3)	B(0.2)	B(0.1)					
P4	U(20)	U(16)	U(12)							

where B stands for *Bernoulli distribution* with p-value inside the bracket and U stands for *Uniform distribution* with lower limit 0 and upper limit in brackets.

Here 2 algorithms have been portrayed :

1.1 Upper Confidence Bound (UCB)

Initialization: Play each arm once,

For t = K + 1, . . . , n, **repeat**

(1) Play arm $I_t = argmax_{k=1,\ldots,K} UCB_t(k)$, where

$$UCB_t(k) = \hat{\mu}_k(t-1) + \sqrt{\frac{8 \log t}{T_k(t-1)}}$$

(2) Observe sample X_t from the distribution P_{I_t} corresponding to the arm I_t.

1.2 Explore-then-Commit (ETC)

(1) **Exploration phase:** During rounds 1, . . . , mK, play each arm m times.
(2) **Exploitation phase:** During the remaining (n - mK) rounds, play the arm with the highest empirical mean reward, i.e., $argmax_{k=(1,\ldots,K)}\hat{\mu}_k(mK)$.

For **Gap-dependent bound for ETC**, the regret bound for horizon n is given by,

$$R_n \leq m\sum_{i=1}^{K}\Delta_i + (n-mK)\sum_{i=1}^{K}\Delta_i \exp\left(\frac{-m\Delta^2}{4}\right) \tag{1}$$

where m is the number of times each arm is played during the *exploration phase* and the Δ values are given by :

Arms	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
P1	0	0.2	0.2	0.2	0.2	0.3	0.3	0.3	0.3	0.3
P2	0	0.02	0.02	0.02	0.02	0.04	0.04	0.04	0.12	0.12
P3	0	0.4	0.5	0.6	0.7					
P4	0	2	4							

which were calculated as $\Delta_i = \mu_\star - \mu_i$ and in the above scenario, 1^{st} arm is optimal (max μ).

2 EXPERIMENTS

For the purpose of experiments, use the folder which is structured as :

- Multi-Armed-Bandit
  - Report
    - .tex file and compilatoin files
    - Figures : Plotted figures are saved here
  - main.py : Main function
  - alg.py : Contains ETC and UCB algorithms
  - plot.py : Contains the plotting functions

- constants.py : Contains values such as horizon, distribution, etc
- sampler.py : Contains the sampling distributions

The **command line arguments** are given as follows,

1. –distribution :
   - default="P1"
   - type = str
   - help='Choose from P1, P2, P3, P4'
2. –algorithm :
   - default="ETC"
   - type = str
   - help='Choose from ETC, UCB'
3. –m :
   - default=300
   - type = int
   - help='Int value : no. of times to explore each arm for ETC'
4. –err :
   - default=1
   - type = int
   - help='Int value : interval for error bar'
5. –compare :
   - default=False
   - type = bool
   - help='Whether to compare different algorithms'

For running the code, example commands are :

1. python main.py –distribution P1 –algorithm UCB

2. python main.py –distribution P3 –algorithm ETC –m 109

3. python main.py –distribution P3 –compare True

NOTE : For finding the **optimal *m*** values for **ETC**- algorithm, an online plotting resource called *Desmos-Graphing Calculator* has been used to plot **Eq(1)** where m is replaced by x.

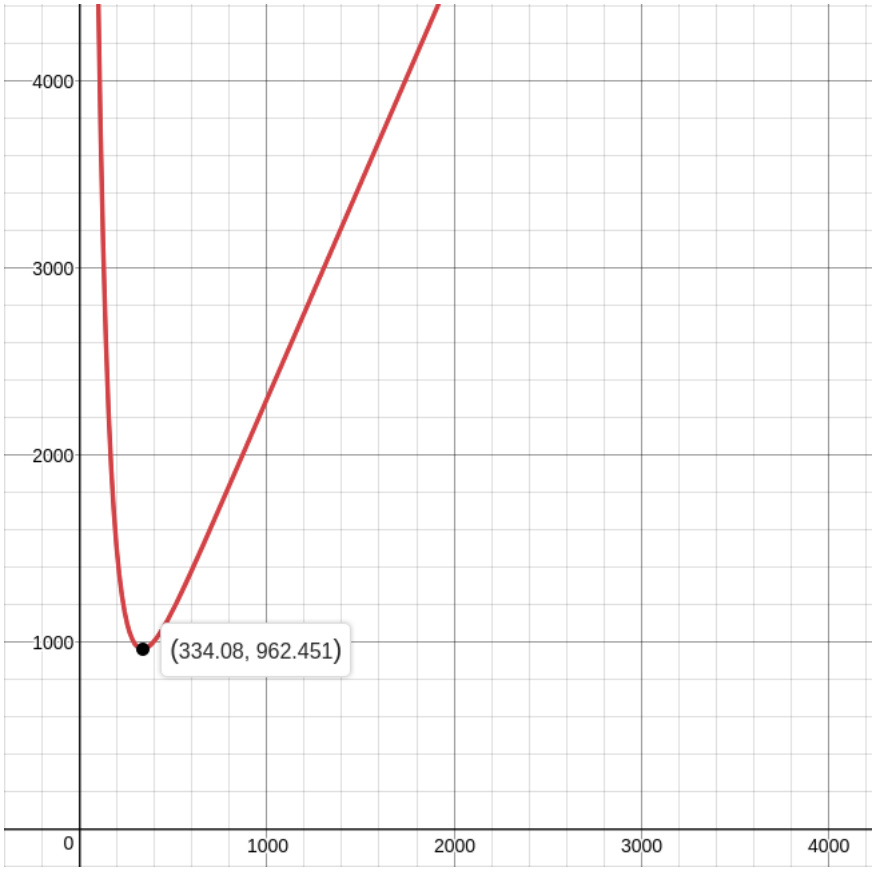
Also note that the error bars aren't visible in the figures for percentage of optimal arm pulled for ETC because the errors are extremely small across the experiments ($\sim 2.27e-13$). This is because, while using the optimal m, beyong the point, in almost all the experiments, we exploit the optimal arm and hence there is no variation before and after identification of an optimal arm (since the experimentally found best arm is the actual optimal arm).

2.1 P1 set of Arms

2.1.1 Explore-then-Commit (ETC)

For this set of arms, the optimal value of m was found to be 334 and the minimum value of the maximum regret was observed to be 962.45 at the minima of the graph as can be seen from the figure below. It was plotted using the equation:

$$x \cdot (0.8 + 1.5) + (10000 - x \cdot 10)\left(0.8 \cdot e^{-x \cdot 0.01} + 1.5 \cdot e^{-x \cdot \frac{0.09}{4}}\right)$$



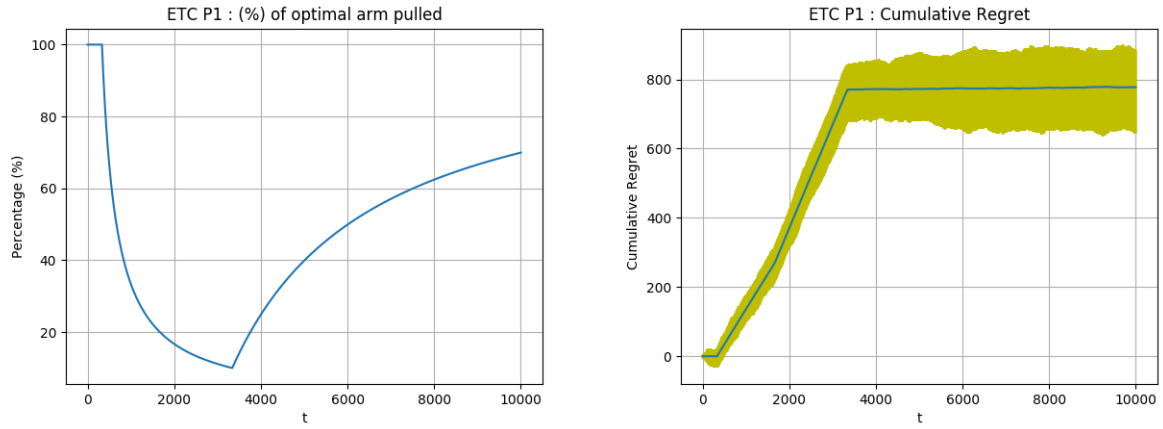


Figure 1: Evaluation plots for ETC for P1 set of arms

2.1.2 Upper Confidence Bound (UCB)

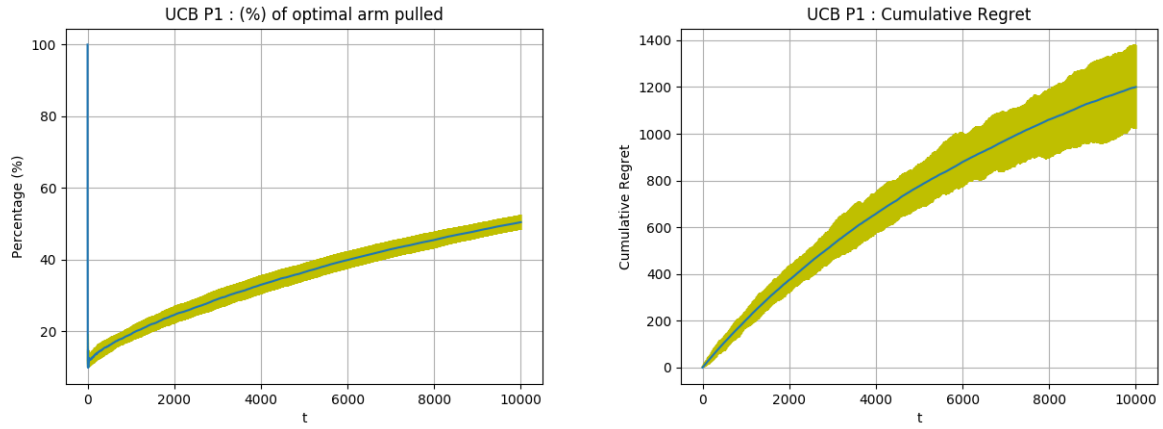


Figure 2: Evaluation plots for UCB for P1 set of arms

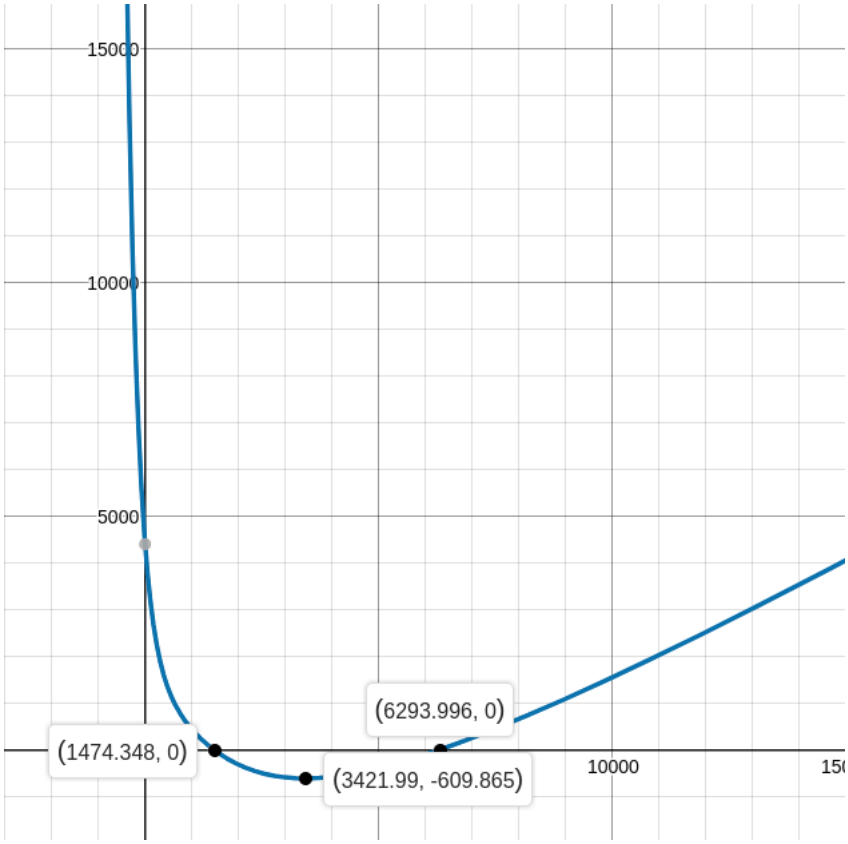
On comparison, we see that at t=n, the percentage of optimal arms pulled by using ETC surpasses UCB by around 20% and the final cumulative regret of ETC exceeds UCB by around a value of 450. In terms of the percentage of optimal arms pulled and the cumulative return at the end of the horizon using optimal m value, ETC seems to be performing better.

2.2 P2 set of Arms

2.2.1 Explore-then-Commit (ETC)

For this set of arms, the optimal value of m was found to be 1474 and the minimum value of the maximum regret was observed to be 0 before the minima of the graph as can be seen from the figure below. It was plotted using the equation:

$$x\cdot((4\cdot 0.02+3\cdot 0.04+2\cdot 0.12)+(10000-x\cdot 10)\cdot\left(4\cdot 0.02\cdot e^{-x\cdot\frac{0.02^2}{4}}+3\cdot 0.04\cdot e^{-x\cdot\frac{0.04^2}{4}}+2\cdot 0.12\cdot e^{-x\cdot\frac{0.12^2}{4}}\right)$$



Here, there seems to be something oddly wrong. On plotting, the regret takes negative values. This means that the horizon is not sufficient enough for us to find an optimal m. Hence, there seems to be a need to increase the horizon and recalculate the optimal m value and conduct the experiment. Since the horizon here is fixed at n=10000, we won't be proceeding with ETC algorithm by finding optimal m value.

2.2.2 Upper Confidence Bound (UCB)

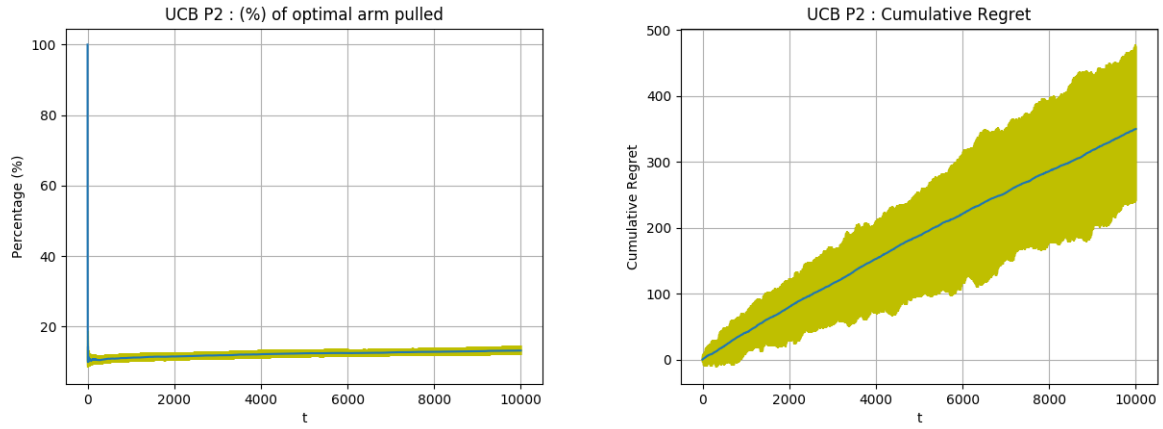


Figure 3: Evaluation plots for UCB for P2 set of arms

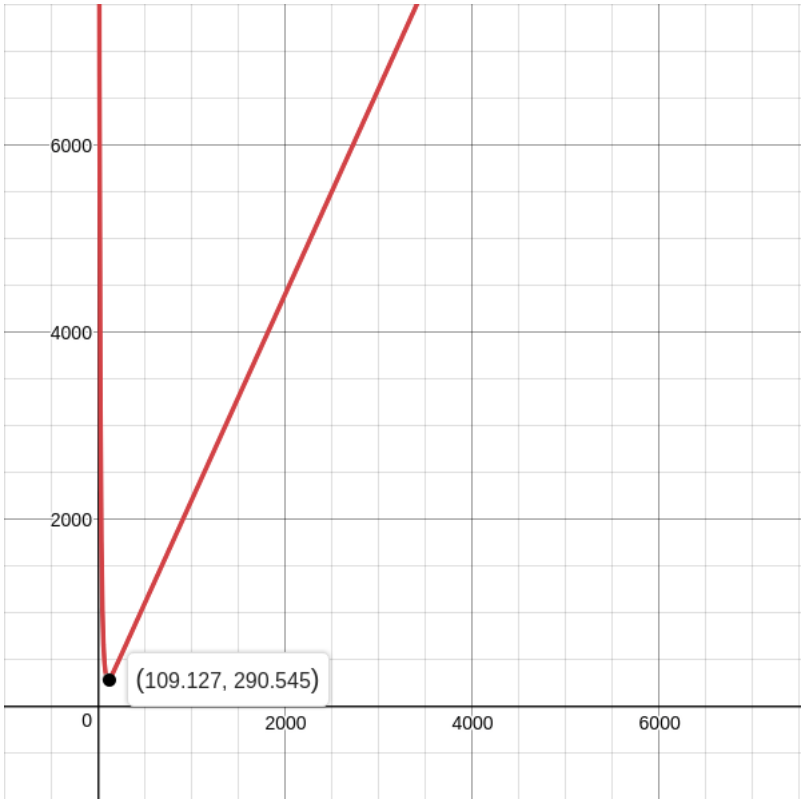
As can be seen from the above figures, UCB seems to be performing quite poorly on this set of ditributions for the arms where the actual mean values are very close to each other. UCB seems to be stuck in local minimas and hence the percentage of optimal arm pulled doesn't cross 15% at the end of the horizon. As for the regret, it's almost linear because the difference keeps increasing, ie., the cumulative regret keeps increasing. This is because the algorithm is stuck using sub-optimal solutions. If the difference between the mean values were larger, then we could hope for better performance of the algorithm. (Note that we're using bernoulli distribution for each arm here. This makes the problem of small differences between the mean values worse.)

2.3 P3 set of Arms

2.3.1 Explore-then-Commit (ETC)

For this set of arms, the optimal value of m was found to be 109 and the minimum value of the maximum regret was observed to be 290.54 at the minima of the graph as can be seen from the figure below. It was plotted using the equation:

$$x\cdot(0.4+0.5+0.6+0.7)+(10000-x\cdot 10)\cdot\left(0.4\cdot e^{-x\cdot\frac{0.4^2}{4}}+0.5\cdot e^{-x\cdot\frac{0.5^2}{4}}+0.6\cdot e^{-x\cdot\frac{0.6^2}{4}}+0.7\cdot e^{-x\cdot\frac{0.7^2}{4}}\right)$$



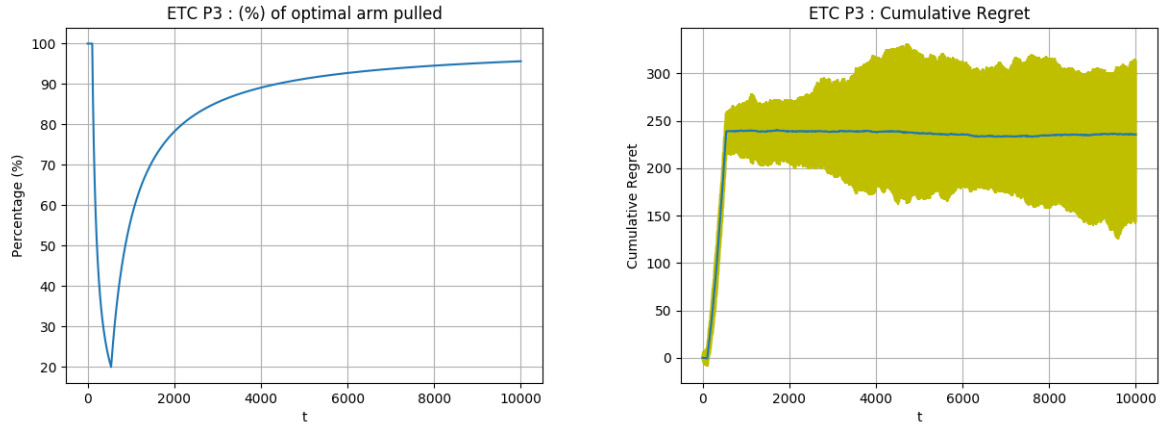


Figure 4: Evaluation plots for ETC for P3 set of arms

2.3.2 Upper Confidence Bound (UCB)

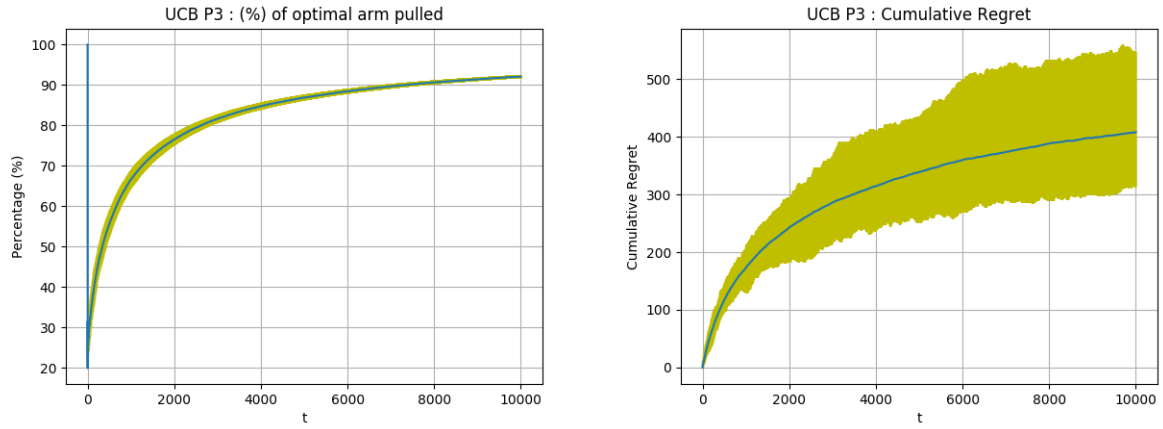


Figure 5: Evaluation plots for UCB for P3 set of arms

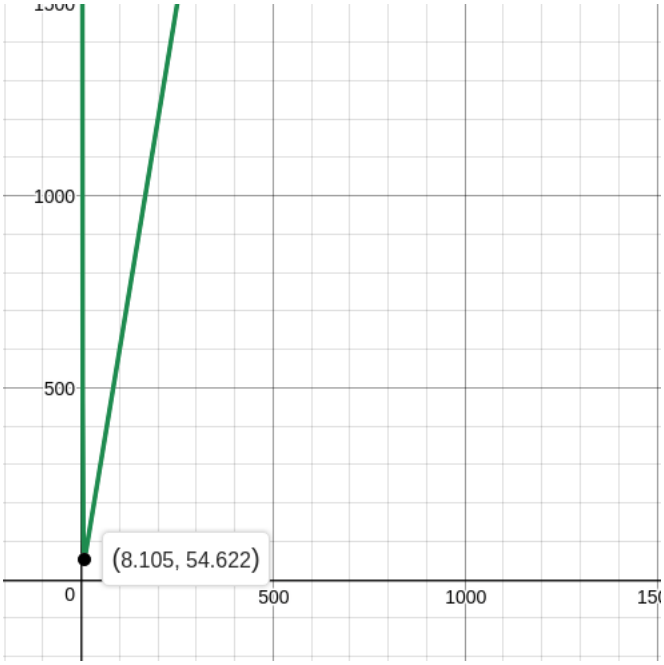
On comparison, we see that at t=n, the percentage of optimal arms pulled by using ETC exceeds UCB only by around 3% but the final cumulative regret of UCB exceeds ETC by around a value of 170. So both algorithms seems to be performing sort of equally at the end of the horizon in this case, using optimal m for ETC. But final cumulative return of UCB is higher.

2.4 P4 set of Arms

2.4.1 Explore-then-Commit (ETC)

For this set of arms, the optimal value of m was found to be 8 and the minimum value of the maximum regret was observed to be 54.6 at the minima of the graph as can be seen from the figure below. It was plotted using the equation:

$$x \cdot (2+4) + (10000 - x \cdot 10) \cdot \left(2 \cdot e^{-x \cdot \frac{2^2}{4}} + 4 \cdot e^{-x \cdot \frac{4^2}{4}} \right)$$



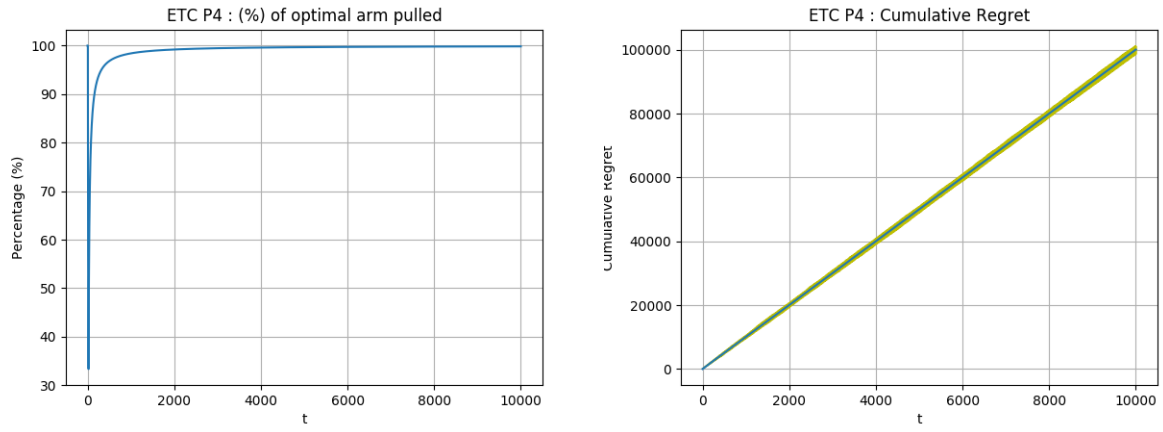


Figure 6: Evaluation plots for ETC for P4 set of arms

2.4.2 Upper Confidence Bound (UCB)

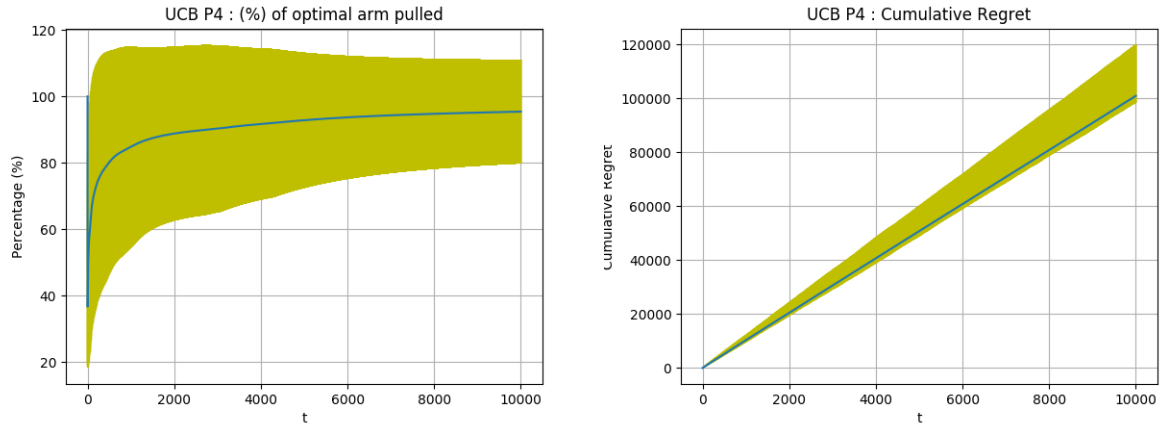
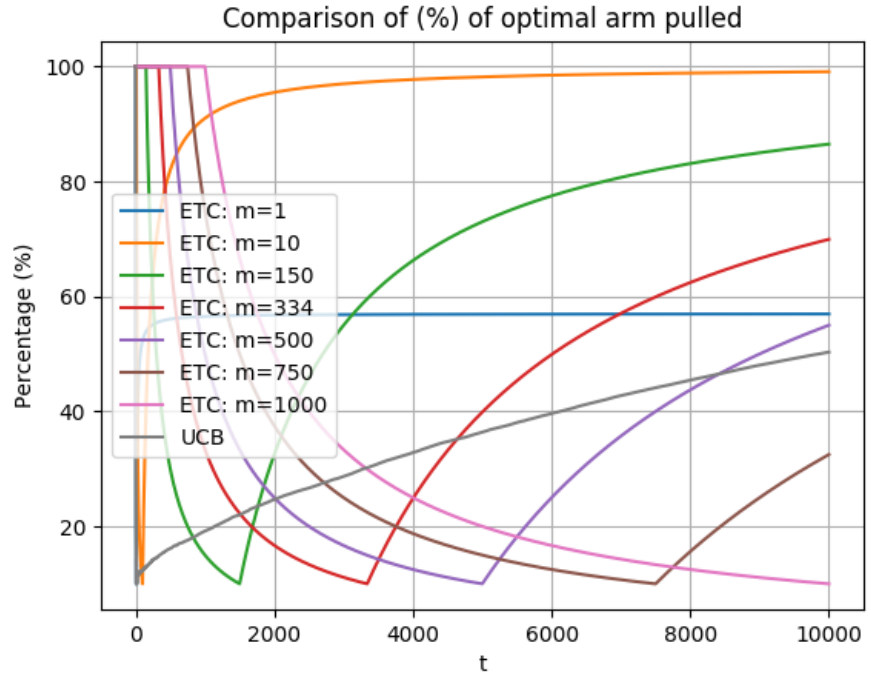


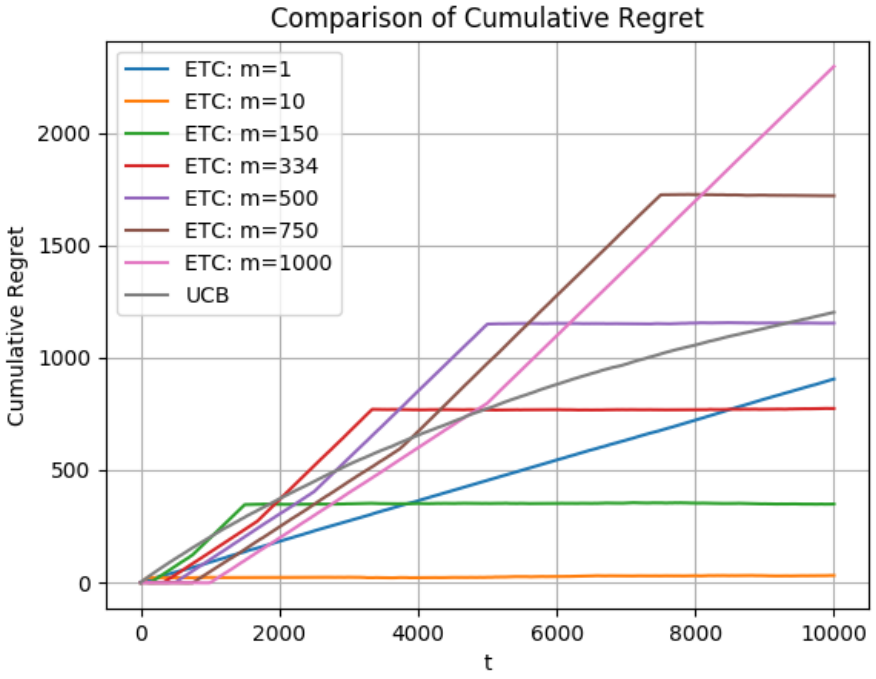
Figure 7: Evaluation plots for UCB for P4 set of arms

On comparison, we see that at t=n, the percentage of optimal arms pulled by using ETC exceeds UCB by around 5% but the final cumulative regret of ETC and UCB is almost same. So inorder to find which algorithm performs better in this case, let's analyse the figures showing percentage of optimal arm pulled. We seem that ETC identifies the optimal arm and almost completely exploits it. We see that the optimal m value is just 8 in this case. Hence exploitation happens in a very efficient manner and we get better rewards. And hence, in this case ETC outperforms UCB.

3 COMPARISON OF ALGORITHMS

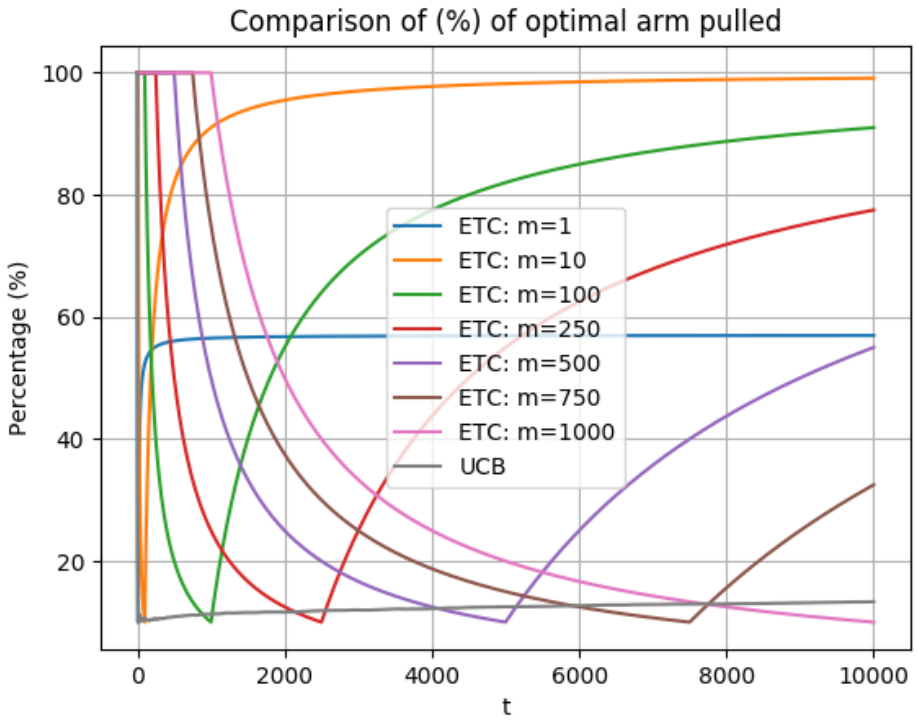
3.1 P1 set of Arms

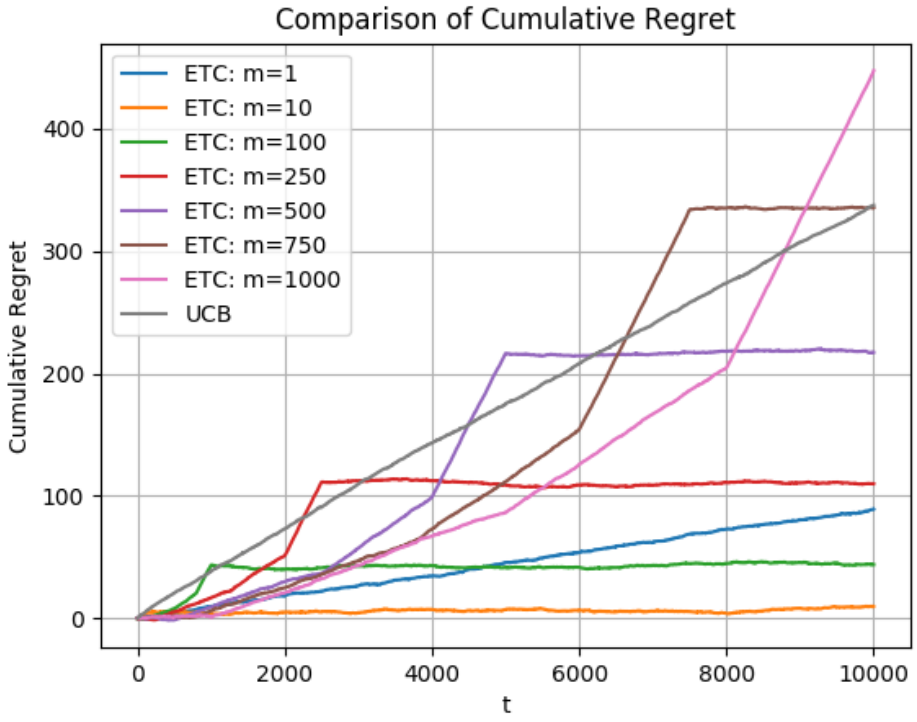




Here, we see that ETC with $m = 10$ seems to be outperforming the others in the figure for percentage of optimal arm pulled. As m is increased (except for m=1), there is a general decrease in the performance of ETC algorithm in this figure. This is because we're exploring more and exploiting less. With smaller values of m from the optimal m, ETC performs better than UCB as can be seem from the first figure. But when it comes to final cumulative regret, the opposite is observed. ETC with larger values of m have better final regret values than UCB. But we also have to remember that there is randomness involved while sampling for the experiment.

3.2 P2 set of Arms

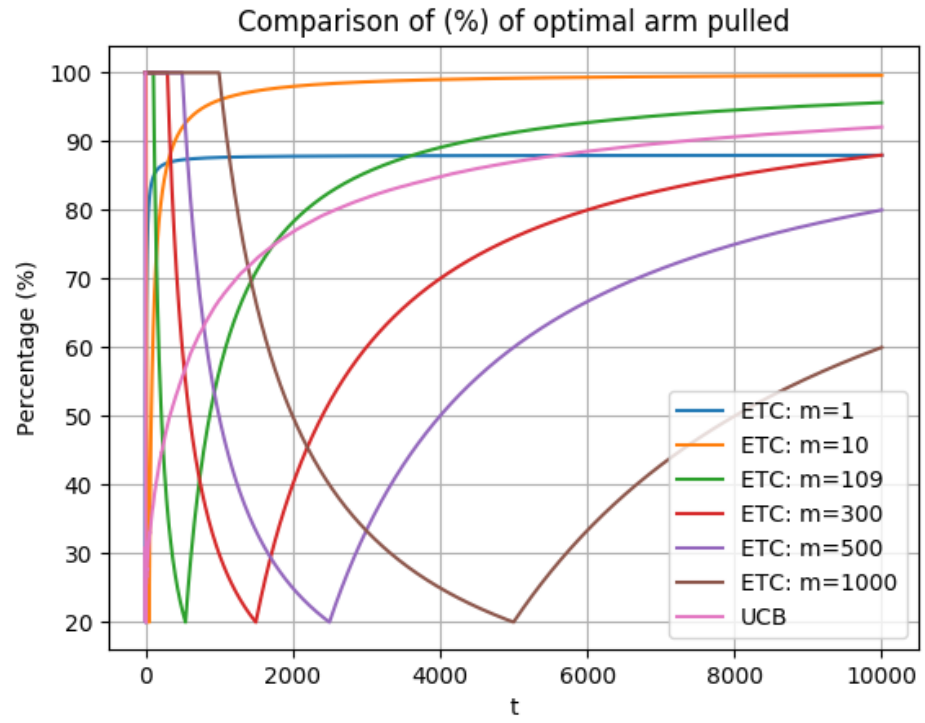


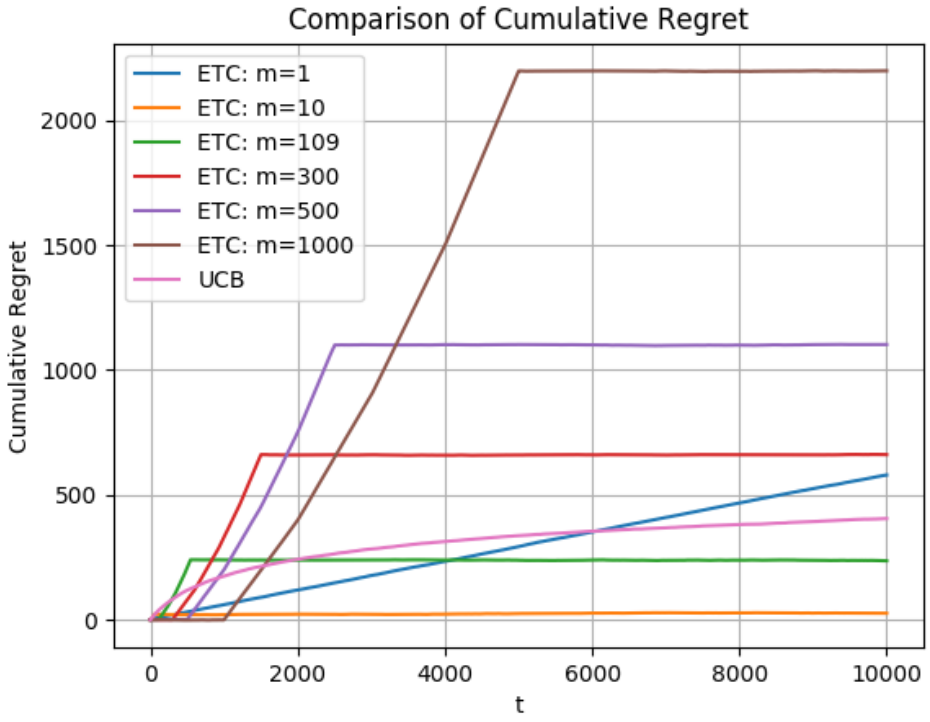


Here, we see that ETC with $m = 10$ seems to be outperforming the others in the figure for percentage of optimal arm pulled. As m is increased (except for m=1), there is a general decrease in the performance of ETC algorithm in this figure. This is because we're exploring more and exploiting less. Here we see that, ETC outperforms UCB for all values of m (except m=1000) as can be seem from the first figure. But when it comes to final cumulative regret, ETC with larger values of m have better final regret values than UCB. But we also have to remember that there is randomness involved while sampling for the experiment.

The reson for the ambiguity is same as that mentioned before in the case of P2. The mean values are extremely close and all the arms follow the same distribution (Bernoulli) and hence UCB performs very poorly in this case. On the other hand, even though the horizon is not sufficient to find an optimal m, ETC seems to perform fairly well. But then again in this case, because of close values of mean, the sub-optimal solutions found by UCB are not that bad.

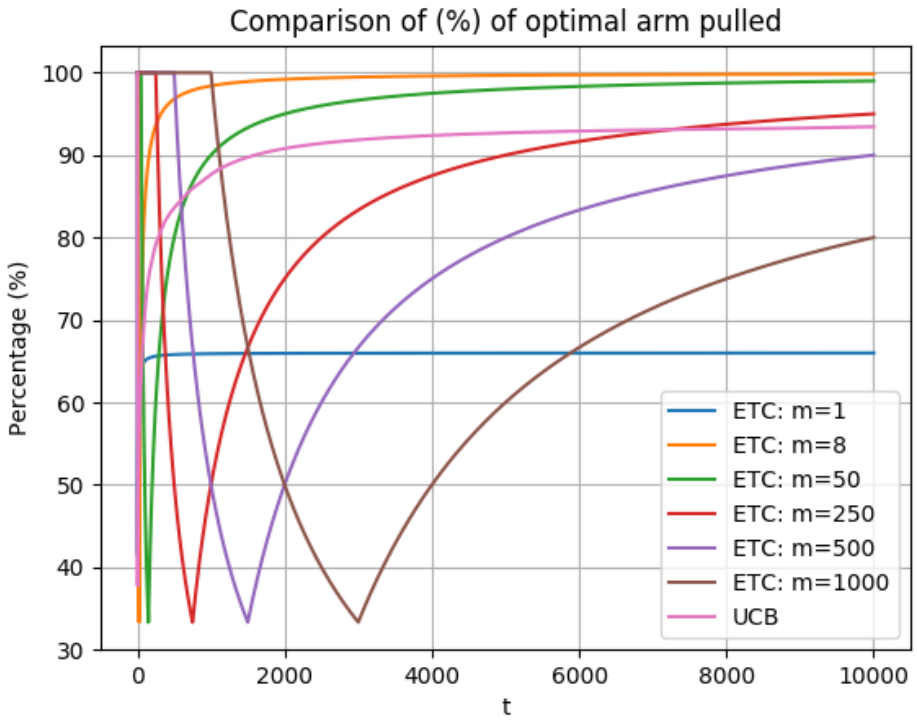
3.3 P3 set of Arms

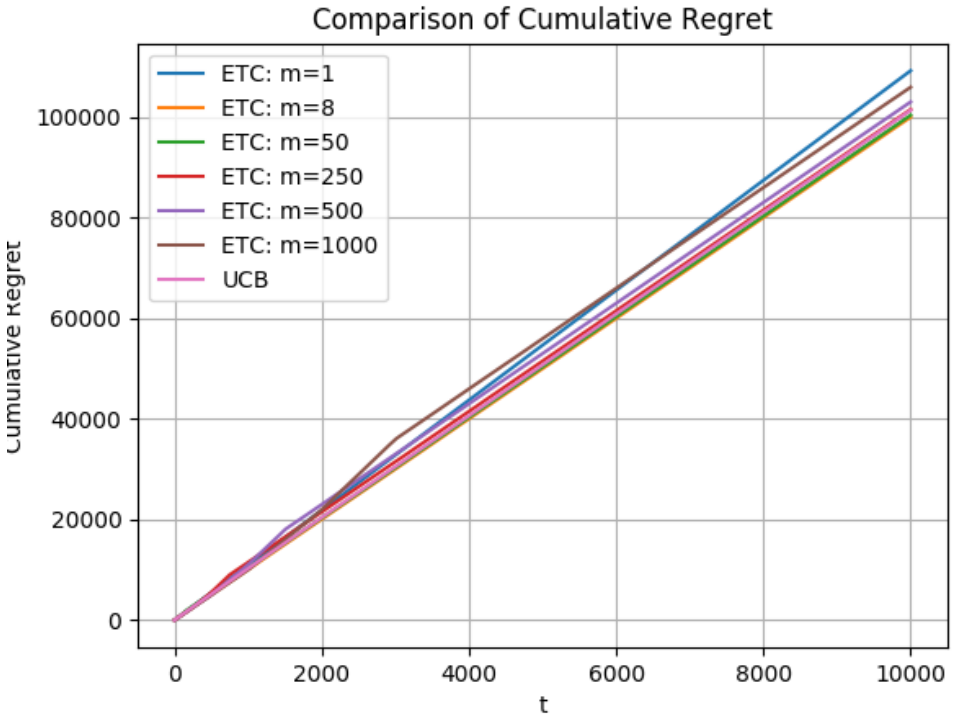




Here, we see that ETC with $m = 10$ seems to be outperforming the others in the figure for percentage of optimal arm pulled. As m is increased (except for m=1), there is a general decrease in the performance of ETC algorithm in this figure. This is because we're exploring more and exploiting less. With smaller values of m from the optimal m, ETC performs better than UCB as can be seem from the first figure. But when it comes to final cumulative regret, the opposite is observed. ETC with larger values of m have better final cumulative regret values than UCB. So here we see a big trade-off between the percentage of optimal arms puutlled and the cumulative return at the end of the horizon while analysing the values of m. But in general in this case, ETC performs better than UCB.

3.4 P4 set of Arms





g to be outperforming the others in the figure for percentage of optimal arm pulled. As m is increased (except for m=1), there is a general decrease in the performance of ETC algorithm in this figure. This is because we're exploring more and exploiting less. With small values of m, ETC performs better than UCB as can be seem from the first figure. But when it comes to final cumulative regret, ETC with larger values of m (except for m=1) have slightly better final cumulative regret values than UCB. But in general in this case, ETC performs better than UCB.

Bibliography

[1] Prof. Prashanth L. A. *CS6046: Multi-armed bandits: Course notes*, Indian Institute of Technology Madras, March 19, 2018

[2] https://www.desmos.com/calculator