CS 747: Course Project Proposal

Regret Minimization Algorithms

Guide : Prof. Shivaram Kalyanakrishnan

Sahil Konkyana - 140050030
Naveen Bhookya - 140050034
Chanukya Vardhan - 140050043

Abstract

An important aspect of most decision making problems concerns the appropriate balance between exploitation and exploration of the environment. Algorithms based on Upper Confidence Bounds for balancing exploration and exploitation are efficient, effective and easy to implement. Many regret minimization algorithms have been proposed like UCB, KL-UCB, Thompson Sampling for the cases where rewards are assumed to be bounded. But most of these algorithms only take the empirical mean information of different arms into account to choose the arm having the highest upper confidence bound. In this project we analyze the performance of highest upper confidence bound algorithms which also take the *empirical variance* of arms into account and compare them with algorithms that only consider mean.

Introduction

Bandit problems have been extensively used to model trade-offs between gaining knowledge by exploring the environment and exploiting the currently known knowledge. A k-armed bandit problem consists of a set of k probability distributions $\langle D_1, D_2, ..., D_k \rangle$ with associated expected means $\langle \mu_1, \mu_2, ..., \mu_k \rangle$ and variances $\langle \sigma_1^2, \sigma_2, ..., \sigma_k \rangle$. The distributions are initially unknown to the player. The goal of the player is to maximize the reward obtained by pulling these arms one by one.

At a turn t, the reward received by the player if he plays the arm $j(t)$ is given by $D_{j(t)}$. Bandit algorithms give strategies to choose the arm $j(t)$ at every turn. The two performance measures we've used to analyze the algorithms are *regret* and *cumulative regret* which are defined respectively as follows:

$$regret_t = reward_t - \mu^*$$
$$R_T = T\mu^* - \sum_{t=1}^{T} \mu_{j(t)}$$

where $regret_t$, $reward_t$ are regret and reward obtained at time t, $\mu^* = \max_{i=1..k} \mu_i$ is the expected reward from the best arm and R_T is the cumulative regret from $t = 0$ to $t = T - 1$.

In UCB algorithms, we have an upper confidence bound associated with each arm and the arm with maximum upper confidence bound is sampled next. UCB algorithms vary in the way this bound is calculated. UCB algorithm only takes the mean information to calculate the bound whereas β-UCB, adaptive β-UCB and UCB-tuned take into account the associated variance as well. In the next section, we describe in detail the algorithms and the way these bounds are calculated.

Algorithms

In this section we present the four algorithms that are implemented. All the four algorithms are based on selecting the optimal arm based on an upper confidence bound. The algorithms are as follows:

UCB:

Let the upper confidence bound of arm k be defined at a given instance be

$$UCB_k^t \triangleq \hat{p}_k^t + \sqrt{\tfrac{2}{u_k^t} log(t)}$$

where $\hat{p}_k^t$ is the empirical mean and u_k^t is the number of times arm k is pulled. The UCB policy is to select the arm that has the maximum UCB_k^t value. At time t, pull arm a where

$$a = \text{argmax}_{k \in A}\, UCB_k^t$$

β-UCB:

Let $0 < \beta < 1$ be some fixed confidence level. Consider the sub-confidence levels β_s defined as

$$\beta_s \triangleq \tfrac{\beta}{4Ks(s+1)}$$

Let

$$B_{k,s} \triangleq (\bar{X}_{k,s} + \sqrt{\tfrac{2V_{k,s} log(\beta_s^{-1})}{s}} + \tfrac{16 log(\beta_s^{-1})}{3s}) \wedge 1$$

where $\bar{X}_{k,s}$ and $V_{k,s}$ are the empirical means and variances respectively.
At time t, pull arm a where

$$a = \text{argmax}_{k \in A}\, B_{k,T_k(t-1)}$$

where $T_k(t)$ is the number of times arm k is chosen during the first t plays.

Adaptive β-UCB:

This algorithm is similar to β-UCB algorithm except that the β confidence level is not a fixed constanct but a decaying with t. The value of β at a time t is given by $\beta = 1/t$. At time t, the arm that needs to be pulled is the one that has maximum β UCB bound, which is defined in the β-UCB section above.

UCB-tuned:

Define the confidence sequences of arm k

$$c_{t,s}^{(k)} \triangleq \sqrt{\frac{2V_{k,s}log(4t^p)}{s}} + \frac{16log(4t^p)}{3s}$$

The confidence bound for each arm is defined as

$$B_{k,s} \triangleq (\bar{X}_{k,s} + c_{t,s}^{(k)}) \wedge 1.$$

where$\bar{X}_{k,s}$ and $V_{k,s}$ are the empirical means and variances respectively and p is a real number greater than 2.
At time t, pull arm a where

$$a = \text{argmax}_{k \in A} \, B_{k,T_k(t-1)}$$

Experimental Setup

We have used the base code and structure of the bandit environment of [3] and made necessary changes as per requirements. The server has information about the bandits and samples the arms from beta distribution with the parameters α and β taken from the input files. The MDP instances are generated by passing the parameters of the beta distribution of each arm to the server. The random generator used is the same as that of [3] which is GNU Scientific Library(GSL). Each of the four UCB algorithms are implemented on the client side.

The goals of the experimental setup are to observe the change in regret with respect to time and also the cumulative regret with respect to horizon. We have chosen 4 different values for number of arms i.e $\{2, 5, 10, 25\}$. For each such bandit instance the mean of arm i is taken to be $i/(N + 1)$ where N is the number of arms.
To study the effect of variance over these algorithms, we have chosen a set of ranges for the σ value to be sampled from for each arm. The ranges are $\{[0.05 - 0.08], [0.1 - 0.13], [0.18 - 0.22]\}$. Since the sampling from beta distribution needs the values of α and β rather than the mean and variance of the distribution, we wrote a script to generate these values based on the mean and variance information. By taking β to be equal to $c * \alpha$, where c is a constant, we get these values from the below mentioned formulas

$$c = \frac{1}{\mu} - 1$$

$$\alpha = \frac{c - \sigma^2 (c+1)^2}{\sigma^2 (c+1)^3}$$

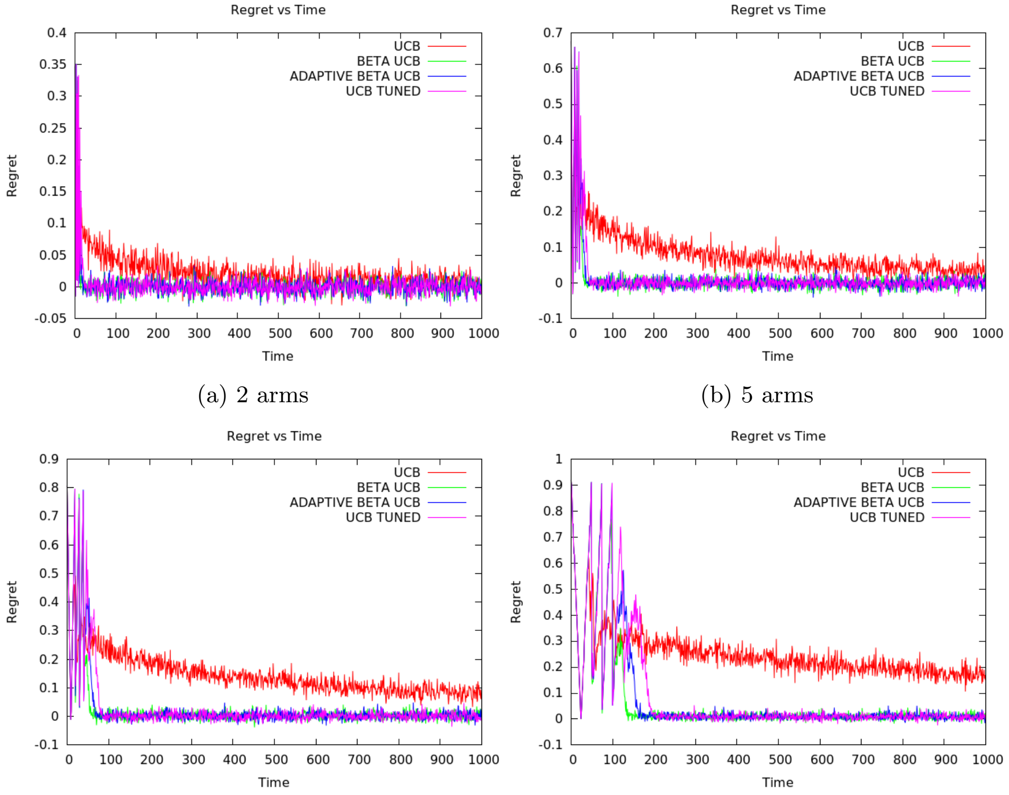
$$\beta = c * \alpha$$

So a bandit instance consists of the number of arms and the parameters for the beta distribution. Each such instance is run over 5 different values of horizons $\{10, 100, 1000, 10000, 100000\}$. For each horizon the experiment is run for 100 times with different random seeds and the average of these is computed to be the final value. Plots of regret vs time and also cumulative regret vs horizon are plotted. Different varieties of plots comparing each algorithm with the other and comparing an algorithm with variance and such are plotted.

Empirical Results

1) Regret vs Time

$\sigma \in [0.1 - 0.13]$
Horizon $= 1000$



(a) 2 arms

(b) 5 arms

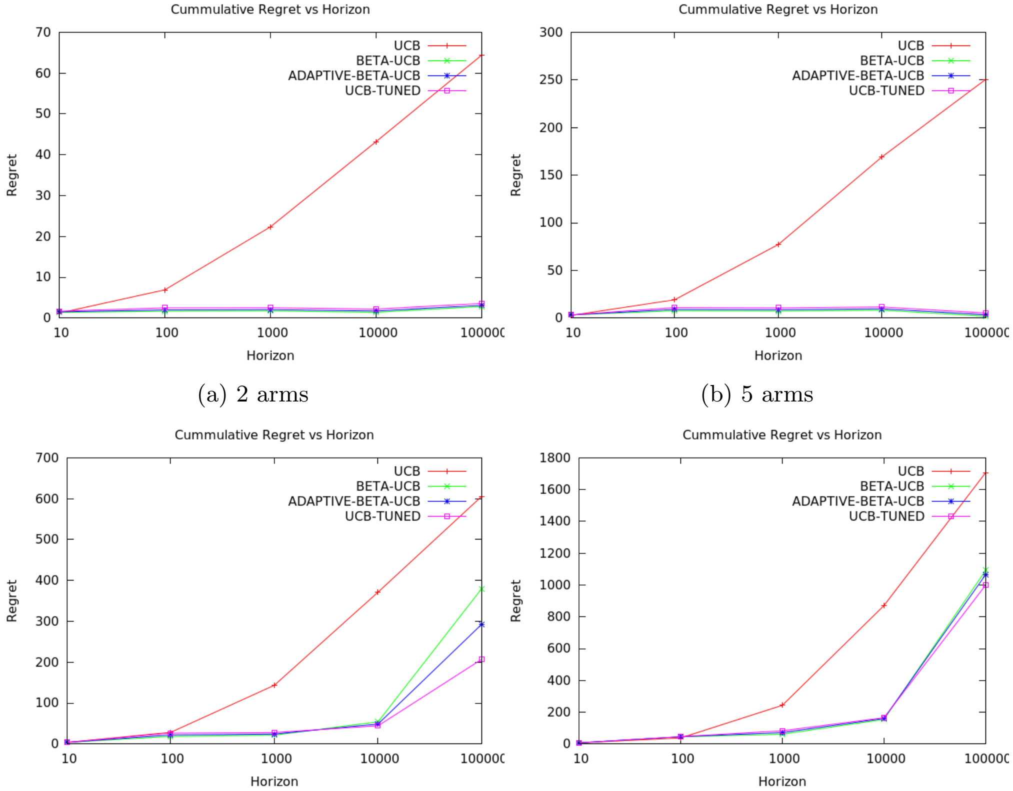
(c) 10 arms

(d) 25 arms

From the graphs above, we can infer that β-UCB, adaptive β-UCB and UCB-tuned algorithms perform better than UCB algorithm for varying number of arms. This is because variance information is captured in these three algorithms. As number of arms increase, number of exploration-exploitation steps increase leading to increase in convergence time.

2) Cumulative Regret vs Horizon

$\sigma \in [0.1 - 0.13]$



(a) 2 arms

(b) 5 arms

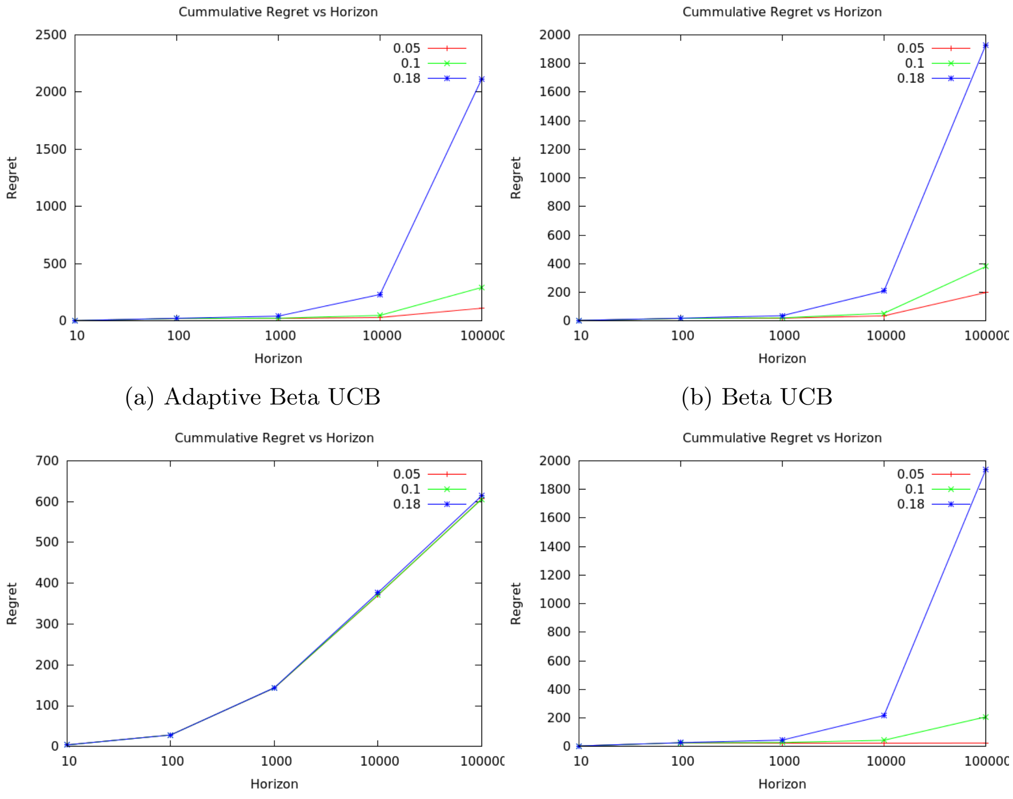
(c) 10 arms

(d) 25 arms

The UCB algorithm performs worse than the other three algorithms since UCB doesn't capture variance information. Among the three algorithms that make use of variance information, there is no big difference observed for 2 arms, 5 arms, 25 arms case, but a significant difference is observed for 10 arms case where UCB-Tuned performed better than Adaptive β-UCB which in turn performed better than β-UCB.

3) Cumulative Regret vs Horizon for different algorithms

$\sigma \in [0.05 - 0.08], [0.1 - 0.13], [0.18 - 0.22]$



(a) Adaptive Beta UCB

(b) Beta UCB

(c) UCB

(d) UCB Tuned

As variance increases the general tendency of these algorithms is reduction in performance. So the performance of β-UCB, Adaptive β-UCB and UCB-Tuned algorithms is worse for higher values of variance. But the performance of UCB algorithm is not affected effected by variance since it doesn't capture variance information. This is evident from the nearly same cumulative regret values with varying values of variance which is the general expected tendency.

The plots for the other parameter values are also generated and submitted along with this report as there are more graphs that can't be shown in the report itself.

Conclusions

In this project we have analyzed the performance of various upper confidence bound algorithms to solve the stochastic multi-armed bandit problem. Many of the existing algorithms do not take into account the variance of sampled rewards and hence perform poorly when the rewards from the arms are sampled from a high variance distribution. From this project we were able to work on three algorithms that make use of variance information and analyzed their performance. We can conclude that in general these algorithms perform better than normal upper confidence bound algorithms. In order to further minimize the regret, information about higher moments like kurtosis or skewness can be worked out into the confidence bounds. These algorithms might perform better than the algorithms studied in this project but require some theoretical and empirical results to support the assertion.

References

[1] J.-Y. Audibert, R. Munos, and C. Szepesvari. *Use of variance estimation in the multi-armed bandit problem.* In NIPS 2006 Workshop on On-line Trading of Exploration and Exploitation, 2006.

`http://videolectures.net/otee06_audibert_uvema/`
`http://imagine.enpc.fr/~audibert/ucbtuned0.5.pdf`

[2] Volodymyr Kuleshov and Doina Precup. *Algorithms for multi-armed bandit problems*, October 2010.
`http://www.cs.mcgill.ca/~vkules/bandits.pdf`

[3] Prof Shivaram Kalyanakrishnan. *CS747: Programming Assignment 1*, 2017.
`https://www.cse.iitb.ac.in/~shivaram/teaching/cs747-a2017/pa-1/programming-assignment-1.html`