CS747

Assignment #1

Report

Sibasis Nayak (190050115)

Department of Computer Science and Engineering
Indian Institute of Technology Bombay

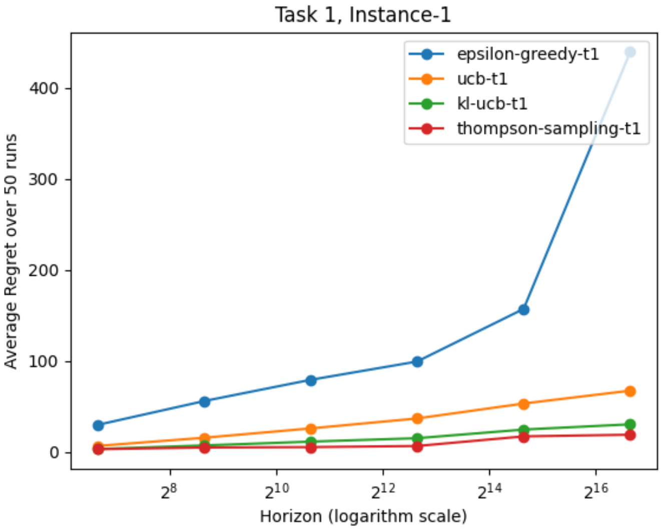
2021-2022

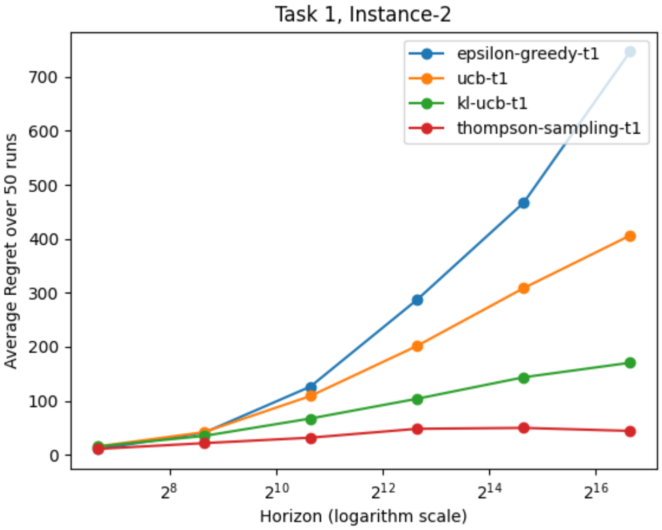
Task 1

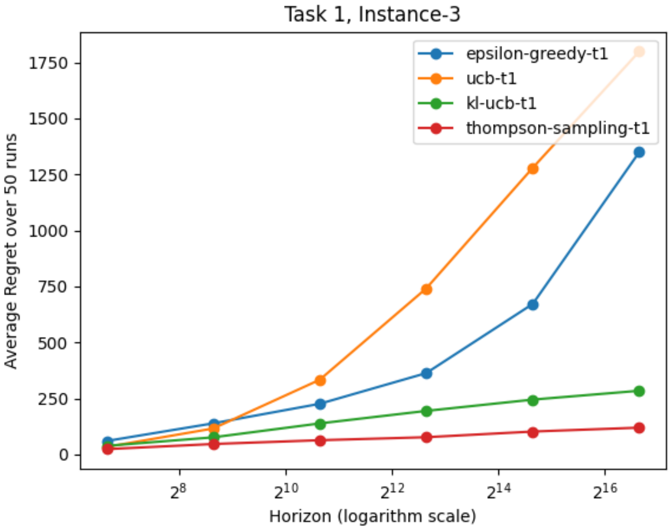
1. For `epsilon greedy` $\epsilon = 0.02$ has been taken.

2. For `UCB` initial exploration has been done by pulling each arm once, and then for the rest of the horizon the UCB-algorithm is implemented.

3. For `KL-UCB`, initial exploration is same as done in `UCB`, and the KL-UCB param c has been taken 3 (as per slides). One more assumption while calculating, UCB-KL if empirical mean so far for an arm is 0 or 1, then we pull those arms first. For 0 it means that it is heavily under-explored hence pulled, and for 1 it means it has never failed so far, hence it is pulled. Also for finding the optimal q binary search is used, which will make the algorithm more efficient.

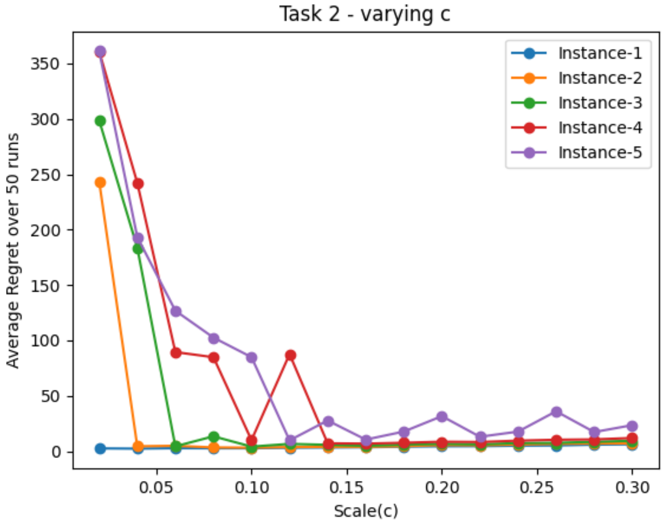






From the plots we can draw comparision between algorithms. We clearly see Thompson-sampling performs best, followed closely by KL-UCB, followed by UCB, followed by epsilon-greedy.
As we keep on increasing horizon this difference gets more and more evident.
One anomaly we see that is in Instance-3 where epsilon greedy performs better than UCB. Looking at the bandit instances we see that the arms are more in number here, and are very competitive(means are closer). UCB seems to have not finished exploration yet even over the largest horizon we have taken. So if we keep on increasing horizon we can expect it to cross epsilon-greedy at after some point.

Task 2

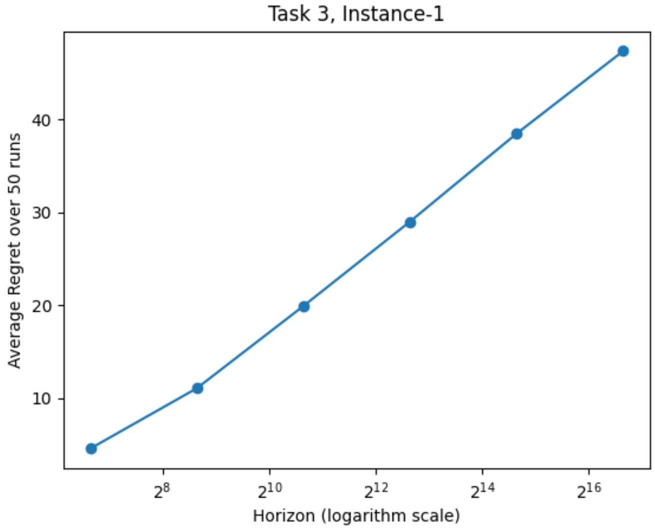


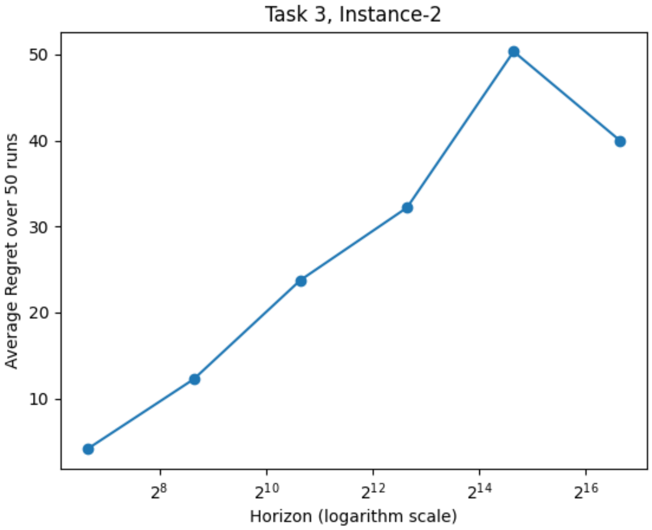
Instance	Minimizing scale
Instance-1	0.04
Instance-2	0.10
Instance-3	0.10
Instance-4	0.16
Instance-5	0.12

We see that as we go on increasing the exploration rate regret drops off substantially and after a point it almost remains constant and should increase after a certain point(not shown in plot). The reason being for this drop-off being, for very low exploration rates, all the arms could not be properly explored before we converge to an optimal arm, hence we will have high regret. As we increase the weighatge for exploration after a certain point, it does not matter anymore, we quickly get enough samples and proceed to converge to optimal arm. Although very high values would mean agent has extremely highly exploration, and very low exploitation.

We also see that the instances, in which the arms are more competitive(means are close enough like in i1 to i5 in increasing order of closeness), they have higher regret, which is expected since it will take a lot of time to explore. They will also be be benefited more by a higher exploration constant, since more exploration is required for competitive arms.

Task 3





I have used modified version of Thompson algorithm here. After choosing an arm as per beta prior distributions, I update the beta distributions as
`success = success + reward`
`failure = failure + 1 - reward`.
All other choices remain same as Thompson algorithm. The reasoning behind such choice being that after long enough exploration the expectation value of mean of of beta distribution of each arm will tend to it's empirical mean, which is what we want more or less (we were doing the same intuition in Thompson's algorithm).
For large enough t we will have

$$\begin{aligned} E[\beta(s_a^t+1, f_a^t+1)] &= \frac{s_a^t+1}{s_a^t+f_a^t+2} \\ &= \frac{\Sigma_t r_a^t}{u_a^t+2} \\ &\approx Empirical\ reward_a \end{aligned}$$

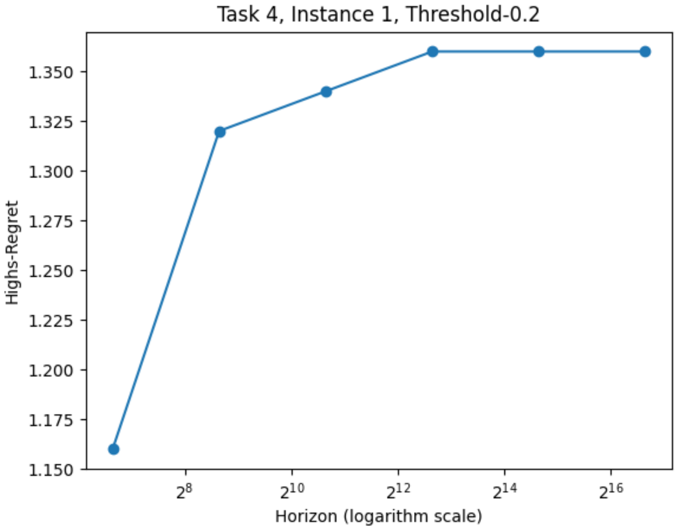
From the plots we see that we have very good regret values for our algorithm. The reason for the regret values increasing with horizon should just be exploration is not properly complete for the instances, and once we are done with complete exploration we can expect it to not deteriorate more further.

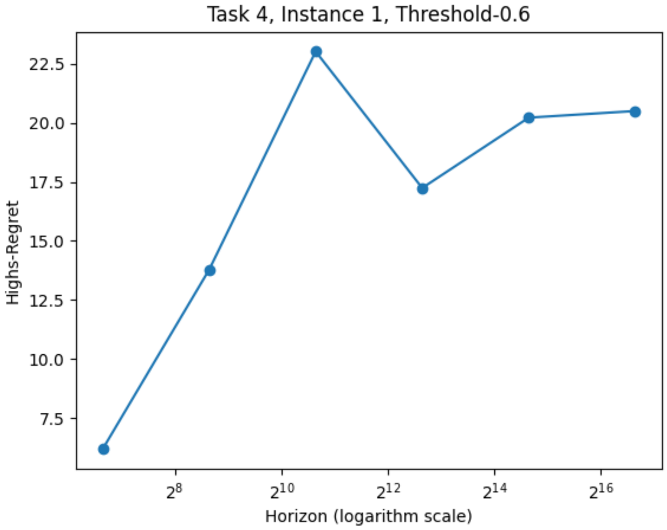
Task 4

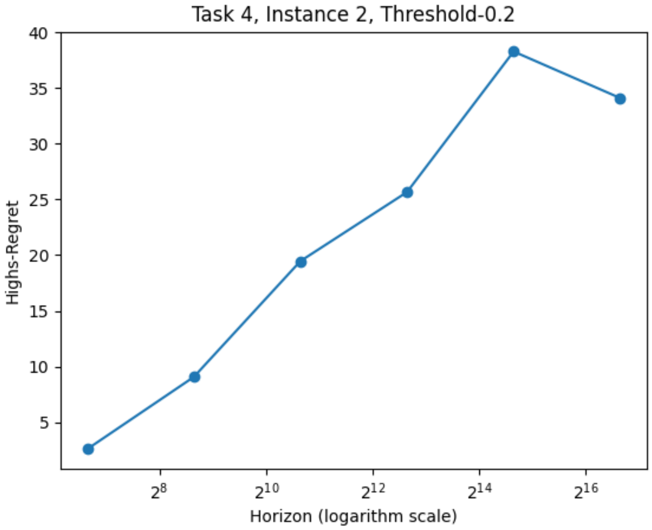
I have used modified version of Thompson algorithm here. After choosing an arm as per beta prior distributions, I update the beta distributions as
`success = success + I(reward>threshold)`
`failure = failure + I(reward<=threshold)`
All other choices remain same as Thompson algorithm.

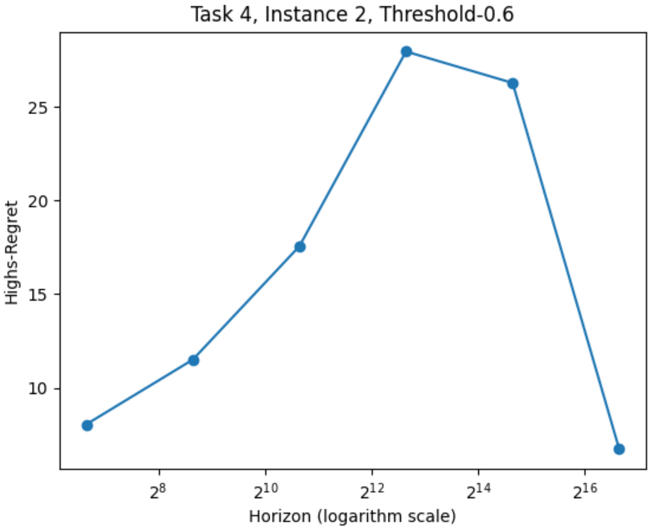
The intuition being that HIGH here can be modelled as our reward. We see that under this model this is same as normal multi-armed-bandit.

The probability for getting reward=1 being $p_a' = \Sigma_r I(r > th) \times p_a^r \times r$, and reward=0 being $1 - p_a'$. Thus with this model we update see it is just a multi-armed-bandit instance, and the updates are modified to the new reward model.


Task 4, Instance 1, Threshold-0.2
Highs-Regret
1.150
1.175
1.200
1.225
1.250
1.275
1.300
1.325
1.350
2^{8}
2^{10}
2^{12}
2^{14}
2^{16}
Horizon (logarithm scale)


Task 4, Instance 1, Threshold-0.6
Highs-Regret
7.5
10.0
12.5
15.0
17.5
20.0
22.5
2^{8}
2^{10}
2^{12}
2^{14}
2^{16}
Horizon (logarithm scale)


Task 4, Instance 2, Threshold-0.2
Highs-Regret
5
10
15
20
25
30
35
40
2^{8}
2^{10}
2^{12}
2^{14}
2^{16}
Horizon (logarithm scale)



From the plots we see that we have good regret values with our modified algorithm. We notice that for higher threshold in instance-1 we have higher regret, which is expected. For instance-2 we see some anomalies, since the arms are competitive here, hence more exploration.