Reinforcement Learning And Dynamic Optimization

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
TECHNICAL UNIVERSITY OF CRETE

Georgios Marios Tsikritzakis
2020030055
Assignment 1

April 17, 2024

1 Introduction

You own a web site that random users access for news. Your goal is to choose a news article to show to each user, that will maximize the chance that the user clicks on it (to read it further). This is also known as "the clickthrough rate". This is your problem setup more formally:

- **News Articles:**
  - When a user arrives at your site, there are a total of $K = 5$ news articles you can choose from.
  - If you recommend article i, then there is a probability that the user clicks article i, which is unknown and equal to p_i.
  - Assume you have a total of T rounds, during which you want to maximize the number of successful recommendations (i.e., clicks)

- **Users:**
  - For every user that visits your site, you know if they are: (i) male or female, and (ii) under or over 25 years old.
  - The "characteristics" of each new user visiting your site, are drawn in an IID manner (i.e., the next user has no dependence on who the previous user was).

- **User-News Preference:**
  - Unlike the standard bandits we've seen, it turns out that different types of users might prefer different articles!
  - Let p_0,p_1,p_2,p_3,p_4 denote the click probabilities for articles, 1,2,3,4,5, respectively. The taste differences are captured as follows:
    - female over 25 : $p_0 = 0.8$, $p_1 = 0.6$, $p_2 = 0.5$, $p_3 = 0.4$, $p_4 = 0.2$
    - male over 25 : $p_0 = 0.2$, $p_1 = 0.4$, $p_2 = 0.5$, $p_3 = 0.6$, $p_4 = 0.8$
    - male or female under 25 : $p_0 = 0.2$, $p_1 = 0.4$, $p_2 = 0.8$, $p_3 = 0.6$, $p_4 = 0.5$
  - **NOTE**: Your algorithm initially knows NEITHER the ranking of different articles (per category), NOR the exact click probabilities. It doesn't even know that males and females under 25 have similar preferences.

2 Simple Modification of the UCB algorithm

The idea is that because we have different user types (and not just a single user like in other bandits problems) we have to implement a UCB algorithm for each user type seperately.

2.1 implementation

The reward of each article is 0 or 1 if the article was clicked or not respectively and the $\hat{\mu}_{\text{ij}}(t)$ the estimation of the probability of article i to be recommended(success probability) in the group type j.It's basically a bernoulli bandit , where the reward is 0 or 1 with a probability $1-p$ and p respectively.

- **Initialization:**

  Try each article at least once for every user type to gather initial data about user preferences.So recommend each article once in every user type! In code , if the number of times article i recommended to user type j is 0 , recommend this article to gather initial info.

- **Article recommendation in every round:**

  Similar to the original UCB algorithm, calculate the Upper Confidence Bound for each article separately for each user type and recommend the article i that maximizes the

$$\text{ucb}_{\text{ij}}(t) = \hat{\mu}_{\text{ij}}(t) + \sqrt{\frac{2\text{logT'}}{N_{\text{ij}}(t)}}. \tag{1}$$

  That means that in the user type j , you recommend the arm i that maximizes the above variable. The $\hat{\mu}_{\text{ij}}(t)$ is the estimate of the success probability of article i in the user type j at time t.Similarly $N_{\text{ij}}(t)$ shows how many times article i has been recommended in user type j until time t. In the context of the modified algorithm with user preferences, we need to consider the effective horizon for each user type separately, because if we consider horizon T for every user type , then the extra upper bound term will be bigger and we may play more times a 'bad' arm.So if the incoming users are IID and have same probabillity we can consider $T' = \frac{T}{\#\text{user-types}}$. As we will see in the plots , the algorithm can have again sublinear regret even with T' = T, but with a little more regret.

- **Update Estimations:**

  Update each article's mean value estimation for every user type(i.e. probability of this article to be clicked) as

$$\hat{\mu}_{\text{ij}}(t) = \frac{\sum_{n=1}^{t} r_{\text{ij}}^{n} \cdot X_{\text{ij},n}}{\sum_{n=1}^{t} X_{\text{ij},n}} \tag{2}$$

  where r_{ij}^{n} is the reward (1 or 0 if the article was clicked or not respectively) of the article i at time n for the user type j , $X_{\text{ij},n}$ is 0 if article i was not recommended at time n for user type j else 1.

3 Regret Theoretical Upper Bound

In the theoritc upper bound , because we do not know the exact a-priori user type probabilities , we use as effective Horizon (for each user type) the term T.

$E[\text{Regret}] = \sum_{j=1}^{|U|} \sum_{i=1}^{k} \Delta_{ij} \cdot N_{ij}(T) = E[\text{Regret} \mid \text{good}] \cdot P(\text{good}) + E[\text{Regret} \mid \text{bad}] \cdot P(\text{bad})$

Where:

- $\Delta_{ij} = \mu^* - \mu_{ij}$ is the suboptimality of article i in the group type j .
- $N_{ij}(T)$ is how many times article i has been recommended in group type j until the time t=T.
- k is the number of articles (actions).
- $|U|$ is the number or different groups - user types.

Define the **good event** as in the normal UCB algorithm:

$$|\hat{\mu}_{ij}(t) - \mu_{ij}| \leq \sqrt{\frac{2\text{logT}}{N_{ij}(t)}}, \forall i, \forall j, \forall t \tag{3}$$

Then we can prove that $\Delta_{ij} \leq 2 \cdot \sqrt{\frac{2\text{logT}}{N_{ij}(t)}} \Rightarrow N_{ij}(t) \leq \frac{8\text{logT}}{\Delta_{ij}^2}$

Without loss of generality we can suppose that the max regret comes from group $j = u$ so:

$E[\text{Regret}] = \sum_{j=1}^{|U|} \sum_{i=1}^{k} \Delta_{ij} \cdot N_{ij}(T) \leq |U| \cdot \sum_{i=1}^{k} \Delta_{iu} \cdot N_{iu}(T)$

Because equation (3) is valid $\forall j$, is also valid for $j = u$ so :

$$N_{iu}(t) \leq \frac{8\text{logT}}{\Delta_{iu}^2} \tag{4}$$

- This proves that we can't play a bad arm (i.e., Δ_{iu} large) too many times (i.e., more than O(log T)) (for Good events)

So :

$E[\text{Regret} \mid \text{good}] \leq |U| \cdot \sum_{i=1}^{k} \frac{8\text{logT}}{\Delta_{iu}}$

For **instance dependent bound** where the minimum gap Δ_{iu} is not too small we can say that $E[\text{Regret} \mid \text{good}] = O(\text{logT})$

For the **bad event** :

$$P(\text{Bad}) = P\left(\exists i, j, t : |\hat{\mu}_{ij}(t) - \mu_{ij}| > \sqrt{\frac{2\text{logT}}{N_{ij}(t)}}\right) \leq |U| \cdot T \cdot k \cdot T^{-4} \tag{5}$$

Because the worst case is regret T (1 for every round):

$E[\text{Regret on bad events}] = E[\text{Regret} \mid \text{bad}] \cdot P(\text{bad}) \leq |U| \cdot k \cdot T^{-2} \to 0$ as T grows! So we can ignore the bad event!

Finally : $E[\text{Regret} \mid \text{good}] \cdot P(\text{good}) + E[\text{Regret} \mid \text{bad}] \cdot P(\text{bad}) \approx E[\text{Regret} \mid \text{good}] \cdot P(\text{good})$ $\leq E[\text{Regret} \mid \text{good}] = O(\text{logT})$ **Sublinear Regret**

4 Plots

4.1 Experiment for T = 1000

Running the simulation in the python notebook we can see the cummulative average Regret.

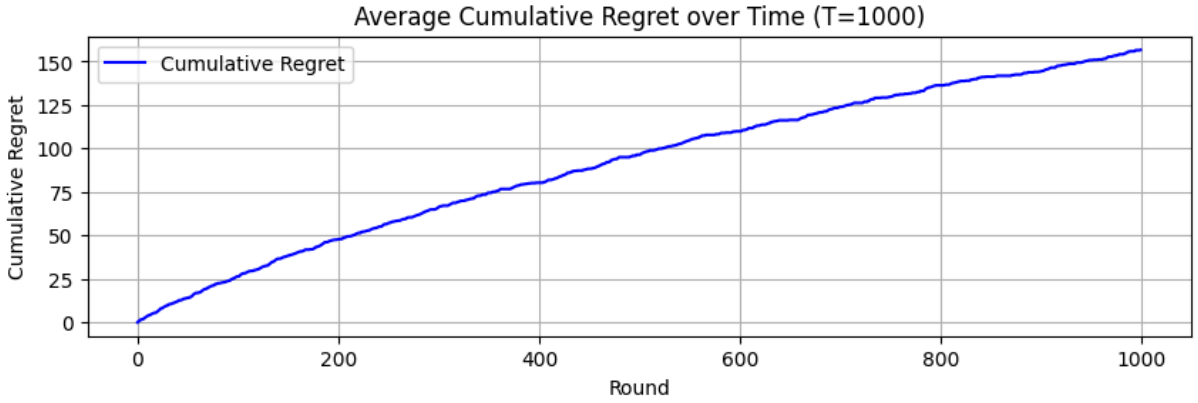

Figure 1: E[Regret] for modified UCB algorithm.

The Regret does not seem logarithmic , and that's because in the experiment we have a small horizon of 1000 rounds , but Theoretically $T \to \infty$. We will see that for more horizon rounds the plot gives a more 'logarithmic' curve.

We can also plot the cummulative regret per round , to see if the $\frac{\text{E[Regret]}}{T} \to 0$ as rounds increasing.That means that the algorithm is learning and is sublinear.

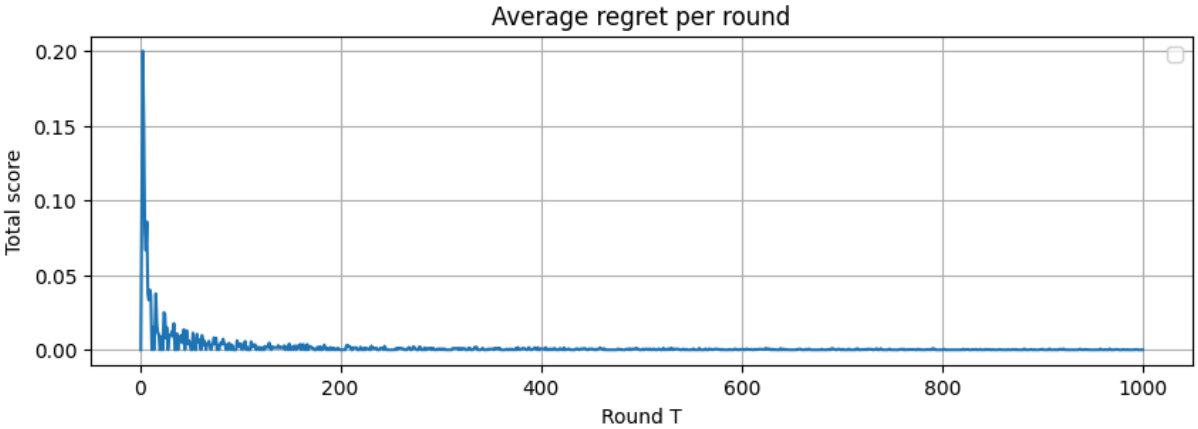

Figure 2: E[Regret] per round for modified UCB algorithm.

As we can see our algorithm is sublinear , because the regret as rounds increasing , is decreasing and actually approach zero.

4.1.1 Comparison of Theoretic and Experimental Upper bound

- From the Theoretical upper regret bound we saw that $E[\text{Regret}] \leq |U| \cdot \sum_{i=1}^{k} \frac{8\log T}{\Delta_{iu}}$. By substituting $\Delta_{iu} = 0.2$ which is the smaller suboptimality we can get for the 'worst' group, we can bound further the Regret. So $E[\text{Regret}] \leq |U| \cdot \sum_{i=1}^{k} \frac{8\log T}{0.2} = |U| \cdot K \cdot 40 \cdot \log T$. **So we found a less strict upper bound , but for $T \to \infty$ the constants around logT do not make such a difference.** Now it's easier to plot the theoretic and the experimental average regret in the same plot.

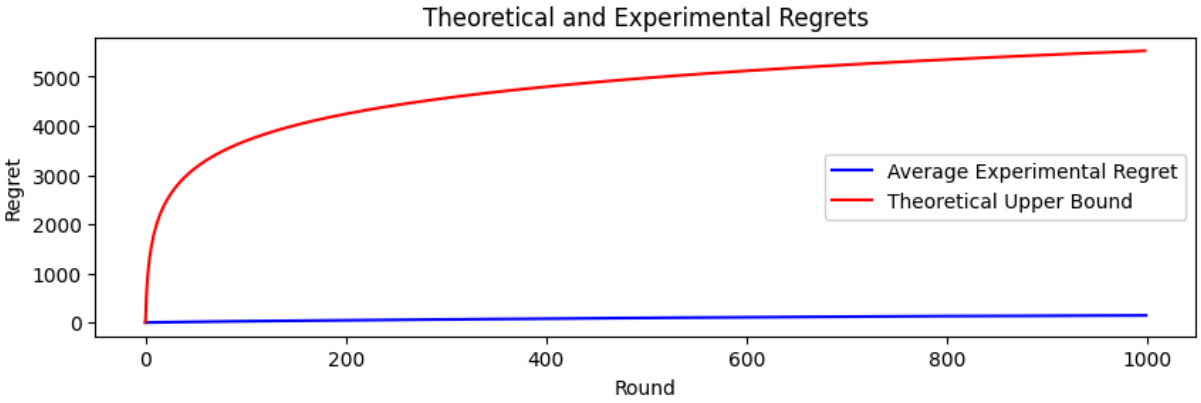

Figure 3: E[Regret] Experimentally and Theoretically.

We can see that the Theoretic Upper Bound is bigger , as we expected , because it's less strict.However they are both logarithmic.

4.2 Experiment for T = 10000

Running again the simulation for larger horizon T = 10000 we get:

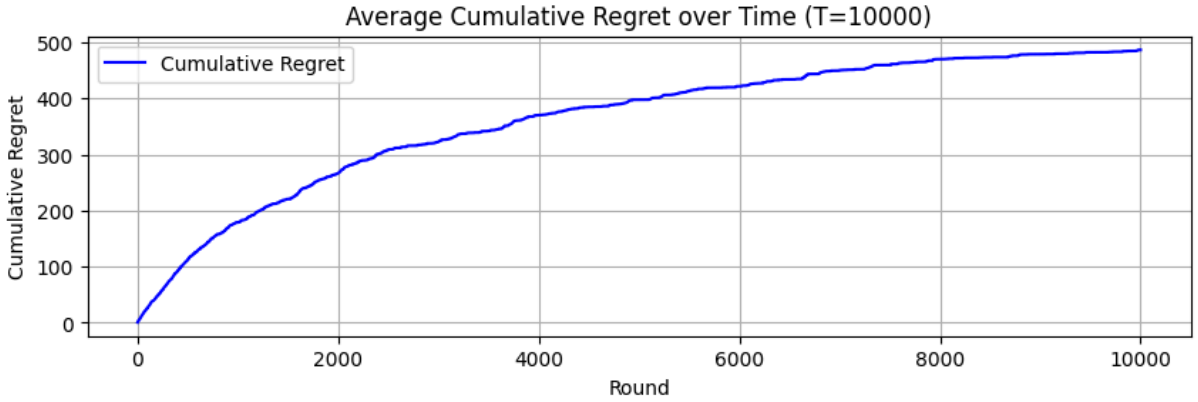

Figure 4: E[Regret] for modified UCB algorithm.

Now for a larger horizon , we can see a logarithmic curve that follows theory. That is expected because we increased the horizon!

4.3 Experiment for T = 10000 with effective horizon T' = T

If we run the simulation with the effective horizon of each user group as T , then we can see again the logarithmic regret of our algorithm as we saw in the mathematical proof at section 3.If we do not know the a-priori probabilities of users coming in our web site , we can't find the effective horizon. The algorithm works again with sublinear logarithmic regret but with a little bit more regret.That is expected because the extra term in the ucb variable will be larger and we may play more times the 'bad' arms.

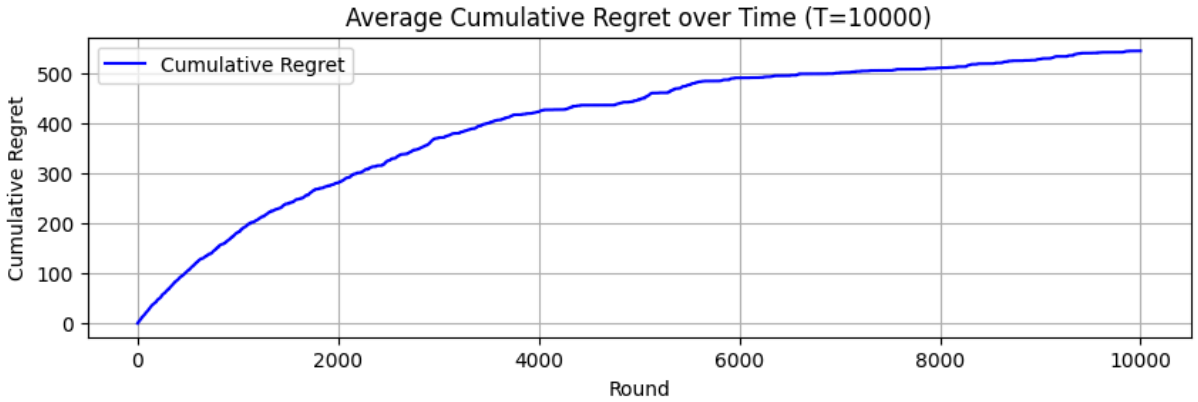

Figure 5: E[Regret] for modified UCB algorithm with T'=T.