EconML assignment Report

Team members:

i) Yash Virani
ii) Vijay Awatade
iii) Vipul Patil

Overview of code:

i) EXP3-IX algorithm:

Parameters are hard-coded in the code. To check regret for different time horizons values like 100, 110, 120 up to 1000 are considered and regret corresponding to each time horizon is calculated and then plotted against the Time horizon. For each time horizon EXP-IX3 algorithm is applied and regret is calculated.

Initially weights are initialized to 1 and regret as 0. In each iteration one arm is pulled using probability distribution calculated by normalizing the weights. After getting a reward for the selected arm, weights are modified. In each iteration regret value is calculated by subtracting reward of selected arm from maximum reward generated. And this value is added to the Regret variable which keeps track of total regret for a given time horizon.

ii) LinUCB algorithm:

In LinUCB, each arm (action) is associated with a feature vector that captures its characteristics. The algorithm maintains a linear model that estimates the expected reward for each arm based on its features. The model parameters are updated iteratively as the algorithm observes rewards. During action selection, LinUCB calculates an upper confidence bound (UCB) for each arm. The UCB incorporates both the estimated reward from the linear model and an exploration term that accounts for uncertainty.
After selecting an arm and receiving the reward, LinUCB updates its model parameters based on the observed data. The feature matrix and reward vector associated with the chosen arm are updated to refine the estimation of expected rewards.

Algorithm Implemented:

i) EXP3-IX algorithm:

Algorithm 1 EXP3-IX

Parameters: $\eta > 0, \gamma > 0$.
Initialization: $w_{1,i} = 1$.
for $t = 1, 2, \ldots, T$, **repeat**

1. $p_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{K} w_{t,j}}$.
2. Draw $I_t \sim \boldsymbol{p}_t = (p_{t,1}, \ldots, p_{t,K})$.
3. Observe loss ℓ_{t,I_t}.
4. $\widetilde{\ell}_{t,i} \leftarrow \frac{\ell_{t,i}}{p_{t,i}+\gamma}\mathbb{I}_{\{I_t=i\}}$ for all $i \in [K]$.
5. $w_{t+1,i} \leftarrow w_{t,i}e^{-\eta\widetilde{\ell}_{t,i}}$ for all $i \in [K]$.

ii) LinUCB algorithm:

Algorithm 1: Linear UCB

Input : λ, β_t
for $t = 0, 1, 2, \ldots$ **do**
  Execute

$$x_t = \text{argmax}_{x \in D} \max_{\mu \in \text{BALL}_t} \langle x, \mu \rangle$$

  and observe the reward r_t
  Update BALL_{t+1}.
end

Bandit Instance:

i) EXP3-IX algorithm:

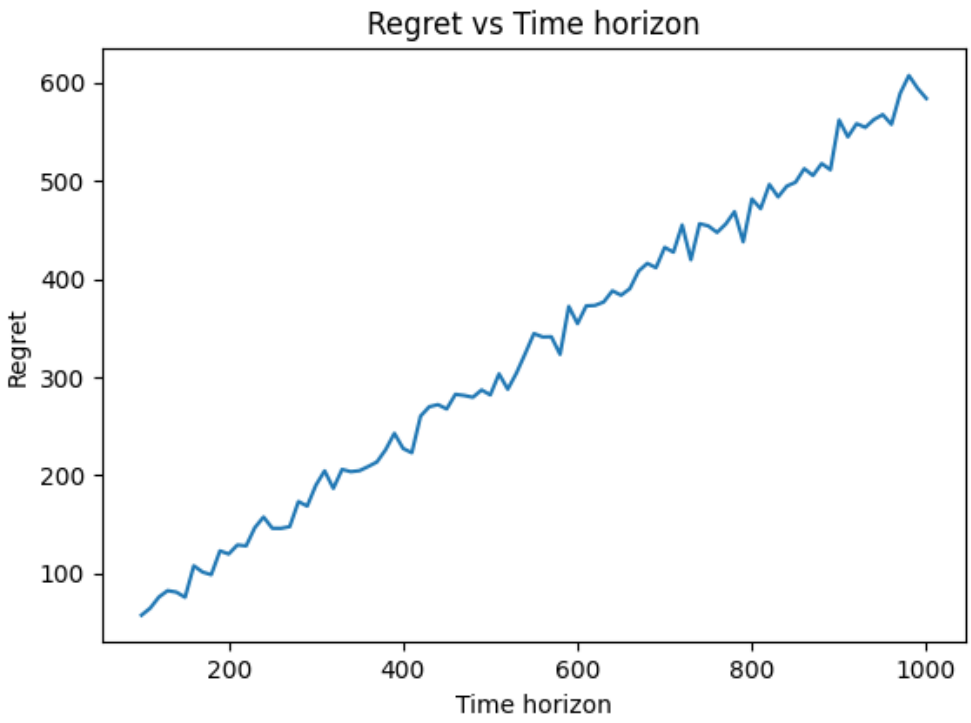
To create an adversarial bandit instance we generated random numbers between -1 to 1. When every time *get_reward(arm_index, Regret)* function is called it generates random rewards for all arms and returns reward attached with *arm_index* arm.

ii) LinUCB algorithm:

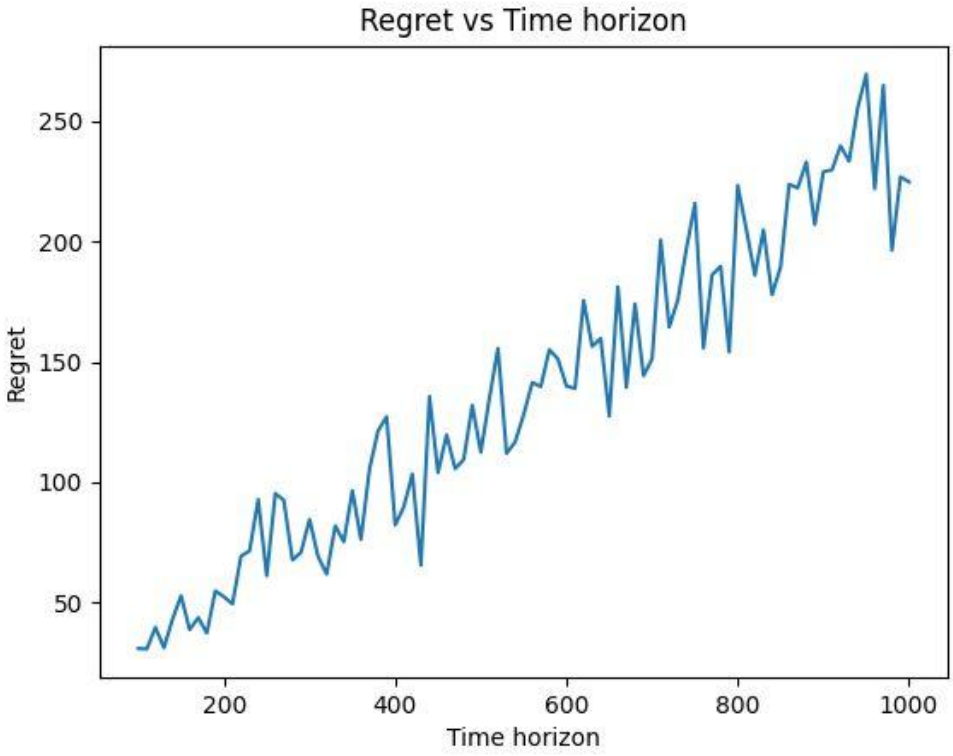
To create a linear bandit we defined mu = [`0.5, 0.2`] and 3 spaces as D = `[[0, 1], [0.25, 0.75], [0, 0.25]]`. After selecting an arm we are taking the dot product of mu and space D[selected arm] and adding some noise randomly (uniform(-1,1)).

Regret line plot:

i) EXP3-IX algorithm:



ii) LinUCB algorithm:



Resources used:

i) EXP3-IX algorithm

https://arxiv.org/pdf/1506.03271.pdf

ii) LinUCB algorithm

https://tor-lattimore.com/downloads/book/book.pdf (Section 19.2)

https://sites.cs.ucsb.edu/~yuxiangw/classes/RLCourse-2021Spring/Lectures/scribe_linear_bandit.pdf

Contextual Bandit: from Theory to Applications. - Vernade - Workshop 3 - C…

Contribution:

i) Vipul - EXP3-IX algorithm complete + LinUCB debugging

ii) Vijay - LinUCB algorithm complete