# Assignment 1

EE932: Introduction to Reinforcement Learning

April 19, 2024

## Instructions

- Kindly name your submission files as '`RollNo_Name_A1.ipynb`'.
- You are required to work out your answers and submit only the iPython Notebook. The code should be well commented and easy to understand as there are marks for this.
- You may use the notebook given along with the assignment as a template. You are free to use parts of the given base code but may also choose to write the whole thing on your own.
- Submissions are to be made through iPearl portal. Submissions made through mail will not be graded.
- Answers to the theory questions, if any, should be included in the notebook itself. While using special symbols use the LaTeX mode
- Make sure your plots are clear and have title, legends and clear lines, etc.
- Plagiarism of any form will not be tolerated. If your solutions are found to match with other students or from other uncited sources, there will be heavy penalties and the incident will be reported to the disciplinary authorities.
- In case you have any doubts, feel free to reach out to TAs for help.

## Multi arm bandits

Consider a two-armed Bernoulli bandit scenario with true means given by $\mu(0) = \frac{1}{2}, \mu(1) = \frac{1}{2}+\Delta$, for some $\Delta < \frac{1}{2}$. In the Bernoulli bandit scenario, at each time instant, the environment generates a binary reward (either 1 or 0) by flipping a coin with the true mean of the chosen arm as the bias. Let the time horizon be $T = 10000$.

### Example

**Explore-then-commit (ETC) algorithm** is a popular algorithm used in multi-armed bandit problems. Its key idea is to first explore the arms to get and estimate of their means and then commit to (exploit) the best arm based on its empirical mean. The exploration is done for a fixed number of steps $m = T^{2/3}(\log T)^{1/3}$ times for each arm to get an estimate of arm means.

Regret is a measure to calculate the performance of an algorithm. The more the regret the less effective the algorithm is in choosing the arm with the highest reward. For our simple two arm setting, the sample regret is given by

$$\mu(1) \cdot T - \sum_{t=1}^{T} R_t,$$

where $R_t$ is the reward obtained in time step $t$.

The jupyter-notebook accompanied with the assignment demonstrates the ETC algorithm and quantifies its empirical performance using the notion of sample regret first for $\Delta = 1/4$ in part **E1** and then for various values of $\Delta \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.45\}$ in part **E2**.

Go thorough sample code for the above example here: https://tinyurl.com/ee932-assignments and answer the following questions.

## Questions (Deadline - 28th Apr 2024)

**Q1 Upper Confidence Bound (UCB) algorithm** is a popular algorithm used in multi-armed bandit problems. Its key idea is to balance exploration and exploitation by choosing arms based on both their empirical means and an exploration term. The exploration term is designed to account for uncertainty in the estimates of arm means. Here's how UCB works:

---

**Algorithm 1** UCB Algorithm

---

**Require:** Time horizon: $T$, Number of arms: $K$

$\bar{\mu}(a) \leftarrow 0$, $n(a) \leftarrow 10^{-6}$, $\forall$ arms $a \in [K]$

**for all** $t \in \{1, \cdots, T\}$ **do**

Calculate the Upper Confidence Bound (UCB) for all arms

$$UCB(a) = \bar{\mu}(a) + \sqrt{\frac{2 \log T}{n(a)}} \; \forall \; a \in [K]$$

Choose arm with the highest UCB:

$$\text{chosen\_arm} = \arg\max_{a} UCB(a)$$

Observe the reward $R_t$ for the chosen arm,
Update the mean and count for the chosen arm

$$n(\text{chosen\_arm}) = n(\text{chosen\_arm}) + 1$$

$$\bar{\mu}(\text{chosen\_arm}) = \frac{\bar{\mu}(\text{chosen\_arm}) \cdot (n(\text{chosen\_arm}) - 1) + R_t}{n(\text{chosen\_arm})}$$

**end for**

---

For the two-armed Bernoulli bandit scenario with true means given by $\mu(0) = \frac{1}{2}, \mu(1) = \frac{1}{2} + \Delta$, for some $\Delta < \frac{1}{2}$.

1. Run the Monte Carlo simulations to estimate the expected regret of the UCB algorithm.

Specifically, you run the UCB algorithm to compute the sample regret

$$\mu(1) \cdot T - \sum_{t=1}^{T} R_t,$$

where $R_t$ is the reward obtained in time step $t$.

Repeat this experiments 500 times (similar to given in the example) and estimate the expected regret by taking the average of the sample regrets you obtained in all those 500 experiments for various values of $\Delta \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.45\}$. [10 Marks]

2. Plot the estimated regret as a function of $\Delta$ for UCB and ETC algorithm and compare. [5 Marks]

**Q2 $\varepsilon$-greedy algorithm (Bonus)** Repeat the above experiment for the $\varepsilon$-greedy algorithm ($\varepsilon = 0.1$) and plot estimated regret as a function of $\Delta$ and compare with ETC and UCB algorithms. What do you observe? [5 Marks]