CS 747 (Spring 2025) Week 2 Test (Batch 1)

5.35 p.m. – 6.00 p.m., January 23, 2025, LA 001

Name: _____ Roll number: _____

Note. There is one question in this test. You can use the space on both pages for your answer. Draw a line (either vertical or horizontal) and do all your rough work on one side of it.

Question 1. Consider an `n`-armed bandit instance, $\texttt{n} \geq 2$, whose arms yield Bernoulli rewards. The arms are numbered 0, 1, ..., `n` - 1.

An algorithm's $\texttt{t} \geq 0$ interactions with the instance are recorded as an (arm pulled, reward obtained)-sequence `a[0]`, `r[0]`, `a[1]`, `r[1]`, ..., `a[t - 1]`, `r[t - 1]`, where `a[]` and `r[]` are arrays of size `t`. Each element of `a[]` is from $\{0, 1, \ldots, \texttt{n} - 1\}$, while each element of `r[]` is from $\{0, 1\}$. Write down pseudocode for a function call to Thompson Sampling that takes `n`, `t`, `a[]`, and `r[]` as input, and returns the next arm to pull. You can assume access to an inbuilt function `R()` for random number generation, but must precisely specify the inputs `R()` takes, as well as the relationship of its output to the inputs. [3 marks]

Answer 1.

Initialise `successes`[] and `failures`[] as arrays of size `n`, with all elements set to 0.

For $\texttt{i} = 0, 1, \ldots, \texttt{t} - 1$:
    If `r[i]` is 0:
        `failures[a[i]]` = `failures[a[i]]` + 1.
    Else:
        `successes[a[i]]` = `successes[a[i]]` + 1.

`maxSample` = -1; `maxIndex` = -1;
For $\texttt{a} = 0, 1, \ldots, \texttt{n} - 1$:
    `x` = `R(successes[a] + 1, failures[a] + 1)`.
    If `x` > `maxSample`:
        `maxSample = x`; `maxIndex` = a.

Return `maxIndex`.

We have assumed that a call to `R(a, b)` returns a sample from a Beta distribution whose parameters are `a` and `b`.

CS 747 (Spring 2025) Week 2 Test (Batch 2)

6.15 p.m. – 6.40 p.m., January 23, 2025, LA 001

Name: _____ Roll number: _____

Note. There is one question in this test. You can use the space on both pages for your answer. Draw a line (either vertical or horizontal) and do all your rough work on one side of it.

Question 1. Below is pseudocode for an algorithm called LCB that resembles the UCB algorithm presented in class, but has two important differences.

1. Instead of the upper confidence bound, we define a *lower* confidence bound (LCB) for each arm by *subtracting* the "exploration bonus" from the empirical mean.
2. The arm eventually pulled has the *smallest* LCB (rather than the largest UCB) in the set of arms A, with ties broken arbitrarily.

> //Assume that each arm has been pulled once initially.
>
> //t denotes the number of pulls already performed.
>
> - At time t, for every arm a, define $\text{lcb}_a^t = \hat{p}_a^t - \sqrt{\frac{2\ln(t)}{u_a^t}}$, where
>
> $\hat{p}_a^t$ is the empirical mean of rewards from arm a, and
>
> u_a^t the number of times a has been sampled at time t.
>
> - Pull an arm a for which lcb_a^t is minimum; that is, pull arm $\text{argmin}_{a \in A}\, \text{lcb}_a^t$.

Consider a 2-armed bandit instance I in which arm 1 yields Bernoulli rewards with mean p_1, and arm 2 yields Bernoulli rewards with mean p_2, satisfying $p_1 > p_2$. We already know that UCB guarantees sub-linear (in the horizon) regret on I. Does LCB also guarantee sub-linear regret on I? Answer yes or no and provide sufficient justification for your answer. [3 marks]

Answer 1.

No: LCB will not achieve sub-linear regret. Notice that LCB appears to be symmetric to UCB, and for similar reasons, will sample each arm infinitely often in the limit. However, it appears by visualising the progress of the algorithm that as the empirical means converge towards the means, it is arm 2 that will be pulled more often than arm 1. Hence, LCB is not greedy in the limit.

The intuitive argument above can be formalised as follows with a more precise proof. The arm pulled by LCB is

$$\operatorname*{argmin}_{a\in A} \mathrm{lcb}_a^t = \operatorname*{argmin}_{a\in A} \left(\hat{p}_a^t - \sqrt{\frac{2\ln(t)}{u_a^t}} \right) = \operatorname*{argmax}_{a\in A} \left(1 - \hat{p}_a^t + \sqrt{\frac{2\ln(t)}{u_a^t}} \right).$$

The RHS is exactly the arm that would be pulled by UCB if the 0-rewards and 1-rewards in each history were interchanged (notice that $\hat{p}_a^t$ is the ratio of the number of 1's to u_a^t, whereas $1 - \hat{p}_a^t$ is the ratio of the number of 0's to u_a^t.) It follows that the probability of generating any history h^t by running LCB on our bandit instance $I = (p_1, p_2)$ will be the same as generating the "complementary history" (with 1's and 0's switched) by running UCB under instance $I^c = (1 - p_1, I - p_2)$. Since UCB is known to achieve sub-linear regret, its fraction of pulls to arm 2 will approach 1 as the horizon becomes infinite. This identically means LCB will pull arm 2 all but a vanishing fraction of the time on I—implying it will achieve linear regret on I.