Problem Set 4 **(Graded)** — Due Friday, November 11, before class starts

For the Exercise Sessions on Oct 28 and Nov 4

Last name	First name	SCIPER Nr	Points

Problem 1: Lower Confidence Bound Algorithm

In the course we analyzed the Upper Confidence Bound algorithm. As was suggested in the course, we should get something similar if instead we use the Lower Confidence Bound algorithm. It is formally defined as follows.

$$A_t = \begin{cases} t, & t \le K, \\ \arg\max_k \hat{\mu}_k(t-1) - \sqrt{\frac{2\ln f(t)}{T_k(t-1)}}, & t > K. \end{cases}$$

Analyze the performance of this algorithm in the same way as we did this in the course for the UCB algorithm.

Hint: Is this algorithm well designed?

Problem 2: Bandits with Infinitely Many Arms

In the course we considered bandits with a finite number of K arms. In this problem we will see that the same ideas apply if we have infinitely many arms as long as there is some additional structure.

Assume that there is an unknown unit-norm vector $\theta \in \mathbb{R}^d$. For every unit-norm vector $u \in \mathbb{R}^d$, there is a bandit. It gives the reward $X_u = \langle u, \theta \rangle + Z_u$, where Z_u is a zero-mean unit-variance Gaussian that is independent over time and independent with respect to different bandits. The nature of the reward is known to the player.

Find a policy, i.e., a strategy of what bandit to probe at any given point in time given a specific history, that has a sublinear regret as time tends to infinity. You can assume that you know the horizon, i.e., we are looking for fixed-horizon policies.

Hint: Start with the simplest thing you can think of. If you do not have time to do the math, describe in words the basic idea of your strategy and why it should give us a sublinear regret.

Problem 3: Epsilon-Greedy Algorithm

Recall our original *explore-then-exploit* strategy. We had a fixed time horizon n. For some m, a function of n and the gaps $\{\Delta_k\}$, we explore each of the K arms m times initially. Then we pick the best arm according to their empirical gains and play this arm until we reach round n. We have seen that this strategy achieves an asymptotic regret of order $\ln(n)$ if the environment is fixed and we think of n tending to infinity but a worst-case regret of order $\sqrt{n}$ if we use the gaps when determining m and of order $n^{\frac{2}{3}}$ if we do not use the gaps in order to determine m.

Here is a slightly different algorithm. Let $\epsilon_t = t^{-\frac{1}{3}}$. For each round $t = 1, \ldots,$ toss a coin with success probability ϵ_t. If success, then explore arms uniformly at random. If not success, then pick in this round the arm that currently has the highest empirical average.

Show that for this algorithm the expected regret at *any* time t is upper bounded by $t^{\frac{2}{3}}$ times terms in t and K of lower order. This is a similar to the worst-case of the explore-then-exploit strategy but here we do not need to know the horizon a priori. Assume that the rewards are in $[0, 1]$.

Problem 4: These Bandits - Exp3

Consider an adversarial bandit setting with K bandits, where the rewards are arbitrary numbers $x_{t,k} \in [0, 1]$ (t stands for the time index and runs from 1 to n and k is the index of the bandit, which goes from 1 to K). You are the adversary, in charge of designing the rewards. You know that the policy that is used is the exp3 algorithm.

Your task is to fill in the numbers. You are given the constraint that the "average" value of all rewards must be $\frac{1}{2}$, where the "average" means the sum over all $n \times K$ entries divided by $n \times K$.

Your aim is to make the expected reward (not regret) of the player as small as possible.

(i) In general, what is the expected reward the player gets in this adversarial setting when using the exp3 algorithm? State the reward normalized by the time n. We are only interested in the first order term, i.e., the constant, and not higher order terms that vanish with n.

(ii) Explain how you fill in the numbers to minimize the expected reward and compute this reward. As before, our interest is in the first order term.

Problem 5: Thompson Sampling with Bernoulli Losses

This problem deals with a Bayesian approach to multi-arm bandits. Although we will not pursue this facet in the current problem, the Bayesian approach is useful since within this framework it is relatively easy to incorporate prior information into the algorithm.

Assume that we have K bandits, and that bandit k outputs a $\{0,1\}$-valued Bernoulli random variable with parameter $\theta_k \in [0,1]$. Let π be the uniform prior on $[0,1]^K$, i.e., the uniform prior on the set of all parameters $\theta = (\theta_1, \cdots, \theta_K)$. Let

$$T_k^1(t) = |\{\tau \leq t : A_\tau = k; Y_\tau = 1\}|,$$
$$T_k^0(t) = |\{\tau \leq t : A_\tau = k; Y_\tau = 0\}|.$$

In words, $T_k^1(t)$ is the number of times up to and including time t that we have chosen action k and the output of arm k was 1 and similarly $T_k^0(t)$ is the number of times up to and including time t that we have choses action k and the output of the arm k was 0.

The goal is to find the arm with the highest parameter, i.e., the goal is to determine

$$k^* = \text{argmax}_k \theta_k.$$

In the Bayesian approach we proceed as follows. At time time t:

1. Compute for each arm k the distribution $p(\theta_k(t)|T_k^1(t-1), T_k^0(t-1))$.
2. Generate samples of these parameters according to their distributions.
3. Pick the arm j with the largest sample.
4. Observe the output of the j-th arm, call it $Y_j(t)$, and update the counters T_j^1 and T_j^0 accordingly.

Show that this algorithm "works" in the sense that eventually it will pick the best arm. More precisely, show the following two claims.

1. Show that $p(\theta_k(t)|T_k^1(t-1), T_k^0(t-1))$ is a Beta distributed and determine α and β.
2. Show that as t tends to infinity the probability that we choose the correct arm tends to 1. [HINT: To simplify your life, you can assume that for every arm k, $T_k^1(t-1) + T_k^0(t-1) \stackrel{t\to\infty}{\to} \infty$.]

NOTE: Recall that the density of the Beta distribution on $[0,1]$ with parameters α and β is equal to

$$f(x;\alpha,\beta) = \text{constant } x^{\alpha-1}(1-x)^{\beta-1}.$$

Further, the expected value of $f(x;\alpha,\beta)$ is $\frac{\alpha}{\alpha+\beta}$ and its variance is $\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$.