Problem Set 5 — *Due Friday, November 30, before class starts*

For the Exercise Sessions on Nov 16 and 23

Last name	First name	SCIPER Nr	Points

Problem 1: KL Divergence

Compute the KL Divergence of two scalar Gaussians $p(x) = \mathcal{N}(\mu_1, \sigma_1^2)$ and $q(x) = \mathcal{N}(\mu_2, \sigma_2^2)$.

Problem 2: Hoeffding's Lemma

Prove Lemma 7.4 in the lecture notes. In other words, prove that if X is a zero-mean random variable taking values in $[a, b]$ then

$$\mathbb{E}[e^{\lambda X}] \leq e^{\frac{\lambda^2}{2}[(a-b)^2/4]}.$$

Expressed differently, X is $[(a-b)^2/4]$-subgaussian.

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 $t = 1, \cdots, ,$ 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: Upper 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.

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

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?