

1. Exercice Sheet

1. The Regret

Recall Definition 1.1.6 from the lecture. Suppose ν is a bandit model and $(\pi_t)_{t=1,\dots,n}$ a learning strategy. Then the regret is defined by

$$R_n(\pi) := nQ_* - \mathbb{E}_\pi \left[\sum_{t=1}^n X_t \right], \quad n \in \mathbb{N},$$

where $Q_* := \int_{-\infty}^{\infty} x P_{a_*}(dx)$ the expected reward of the best arm $a_* = \operatorname{argmax}_a Q_a$.

- a) Suppose a two-armed bandit with $Q_1 = 1$ and $Q_2 = -1$ and a learning strategy π given by

$$\pi_t = \begin{cases} \delta_1, & t \text{ even,} \\ \delta_2, & t \text{ odd.} \end{cases}$$

Calculate the regret $R_n(\pi)$.

- b) Define a stochastic bandit and a learning strategy such that the regret is 5 for all $n \geq 5$.
 c) Show for all learning strategies π that $R_n(\pi) \geq 0$ and $\limsup_{n \rightarrow \infty} \frac{R_n(\pi)}{n} < \infty$.
 d) Let $R_n(\pi) = 0$. Prove that π is deterministic, i.e. all π_t are almost surely constant and only chose the best arm.
 e) Suppose ν is a 1-subgaussian bandit model with k arms and $km \leq n$, then consider the explore then commit algorithm and recall the regret bound:

$$R_n \leq \underbrace{m \sum_{a \in \mathcal{A}} \Delta_a}_{\text{exploration}} + \underbrace{(n - mk) \sum_{a \in \mathcal{A}} \Delta_a \exp\left(-\frac{m\Delta_a^2}{4}\right)}_{\text{exploitation}}.$$

Assume now $k = 2$, such that $\Delta_1 = 0$ and $\Delta_2 = \Delta$ then we get

$$R_n \leq m\Delta + (n - m2)\Delta \exp\left(-\frac{m\Delta^2}{4}\right) \leq m\Delta + n\Delta \exp\left(-\frac{m\Delta^2}{4}\right).$$

Show that this upper bound is minimized for $m = \max\left\{1, \left\lceil \frac{4}{\Delta^2} \log\left(\frac{n\Delta^2}{4}\right) \right\rceil\right\}$.

2. Programming task: explore-then-commit Algorithm

The first algorithm from the lecture was the explore-then-commit algorithm. If you are not so familiar with programming, you can find a standard implementation of the algorithm in the Jupiter Notebook that we have provided on the website. Otherwise, you can also implement your own explore-then-commit algorithm.

- a) Implement the 10-armed bandit from Chapter 2.3 by Sutton and Barto. Therefore, you should first implement a general Gaussian bandit using the code snippet below:

Listing 1: Gaussian Bandit

```

class GaussianBanditEnv(Env):
    def __init__(self, mean_parameter, max_steps):
        """ create a multiarm bandit with 'len(p_parameter)' arms
        Args:
            mean_parameter (list): list containing mean
            parameter of gaussian bandit arms
            max_steps (int): number of total steps
            for the bandit problem
        """
        pass

    def step(self, action):
        """ play an action in the gaussian bandit modell
        Args:
            action (int): choosen arm
        Returns:
            list: new state, reward,
            done (bool if game is finished), info
        """
        pass

    def reset(self):
        """ reset all statistics to run a new game
        """
        pass

```

Please ignore the return "new state" in the step function for now, we will need this later for reinforcement learning.

Then simulate 10 standard Gaussian random variables representing the expected value Q_a of each arm and create the Gaussian 10-arm bandit.

- b) Play the bandit $n = 10000$ times with the explore-then-commit learning strategy for different choices of m and compute the corresponding rewards. Graph the regret in terms of different m . Which m minimizes regret?

3. *Programming task advanced: explore-then-commit

Suppose the 10-armed bandit from Excercise 2 and use the regret bound from Theorem 1.2.2 to find an optimal m that minimizes the regret. If you cannot find an analytic form for the

optimal m , use a numerical method to determine the optimal m . Plot the rewards and make a comparison to the previous exercise.