Stochastic Multi-armed bandits
Practical Session

1. TP Part A (40min)

- Use implementation of bandit, regret, visualization.

  You are given a python project with 4 parts.

  – Algorithms: code of various bandit strategies
  – Environments: code of the stochastic bandit environment
  – Experiments: code to generate numerical experiments
  – Figures: plots output by the experiments

  A numerical experiment running a bandit strategy on a bandit problem has the following form

```
for t in range(0,timeHorizon):
    arm = learner.chooseArmToPlay()
    reward,expectedInstantaneousRegret=bandit.GenerateReward(arm)
    learner.receiveReward(arm,reward)
```

  See e.g. in file `Experiments_MakeBanditExperiments`, function `OneBanditOneLearnerOneRun`.

  1. Take a look at the file `Experiments_Demo`, and go over each function that is called.
  2. Run the file and observe the result of a single experiment with given time horizon T.
  3. Create a novel method in the file `Experiments_MakeBanditExperiments` in order to collect data over N independent runs instead of a single one. We are especially interested in getting the cumulative regret $R_{t,n}$ after $t = 1, \dots, T$ steps in run $n = 1, \dots, N$
  4. Create novel method to visualize
     (a) the histogram of the values $R_{T,n}, n = 1, \dots, N$, and
     (b) the average regret $\frac{1}{N}\sum_{n=1}^{N} R_{t,n}$ as a function of $t = 1, \dots, , T$.

  (bonus) Show also the error bars, or quantiles.

- Implement lower bounds.

  The asymptotic lower bound for Bernoulli configurations is given by

$$\liminf_{T\to\infty} \frac{\mathfrak{R}_T}{\log(T)} \geqslant \sum_{a\in\mathcal{A}} \frac{\mu_\star(\nu) - \mu_a(\nu)}{\mathtt{kl}(\mu_a(\nu), \mu_\star(\nu))} \text{ where } \mathtt{kl}(x, y) = x\log(\frac{x}{y}) + (1 - x)\log(\frac{1 - x}{1 - y}).$$

  Plot the function $T \to \sum_{a\in\mathcal{A}} \frac{\mu_\star(\nu)-\mu_a(\nu)\log(T)}{\mathtt{kl}(\mu_a(\nu),\mu_\star(\nu))}$. Later compare it with the regret of bandit strategies (e.g; FTL, UCB). What do you observe?

- Implement UCB variants:

  When argmax is not unique, choose arm with lowest number of pulls, if many of them, choose randomly amongst them.

Algorithm 1 Upper Confidence Bound strategy

1: choose $a_t = \operatorname*{argmax}_{a \in \mathcal{A}} \mu^+_{a,t-1}$ where $\mu^+_{a,t} = \tilde{\mu}_{a,t} + \sqrt{\frac{\log(1/\delta_t)}{2N_a(t)}}$ with $\delta_t = t^{-2}(t+1)^{-1}$.

- Compare strategies, including the asymptotic lower bound

  Compare the histogram of the cumulative regrets for FTL and UCB on a simple Bernoulli arm problem obtained using sufficiently many runs. Do you think that FTL is a "safe" strategy?

  Study the influence of the minimum gap, of the choice of δ_t, etc.

PAUSE 5 min

2. TP Part B (40min)

- Implement KL-UCB optimization for Bernoulli.

  The `KL-ucb` strategy is inspired from the regret lower bounds. The idea is to identify a most confusing distribution for each arm a, and decide to play arm a that enables to removes the one with highest mean from the set of plausible distributions. Another way to see this is by considering arms for which $N_t(a)$ is currently too small for the algorithm to be uniformly-good on $\mathcal{D}$, and pull those arms. See **?**.

  We now provide the construction of the `KL-ucb` strategy for a set of bandit configurations $\mathcal{D}$, that can be traced at least back to **?**. At each round t, an upper bound $U_a(t)$ is associated with the expectation μ_a of the distribution ν_a of each arm, then an arm a_{t+1} with highest upper bound is played.

Algorithm 2 The `KL-ucb` algorithm for unstructured $\mathcal{D}$.

Parameters: A set $\mathcal{D}$ of bandit configurations, a non-decreasing function $f : \mathbb{N} \to \mathbb{R}$

Initialization: Pull each arm of $\{1, \ldots, K\}$ once

for each round $t+1$, where $t \geqslant K$:

compute for each arm a the quantity

$$U_a(t) \quad = \quad \sup\left\{\mu_a(\nu) : \ \nu \in \mathcal{D}, \ \forall a' \in \mathcal{A} \setminus \{a\}, \ \nu_{a'} = \widehat{\nu}_{\mathcal{D},a'}(t) \quad \text{and} \quad N_a(t) \leqslant \frac{f(t)}{\mathtt{KL}\Big(\widehat{\nu}_{\mathcal{D},a}(t), \ \nu_a\Big)}\right\}$$

where $\quad \widehat{\nu}_{\mathcal{D},a}(t) = (\Pi_{\mathcal{D}}(\widehat{\nu}(t)))_a$is the projection of the the empirical distribution on the family $\mathcal{D}$.

Pull an arm $\quad a_{t+1} \in \operatorname*{argmax}_{a \in \mathcal{A}} U_a(t)$.

  The function `klucbBern(mu, f(t)/ N_t(a), precision)` from the file `Algorithms_kullback` computes the upper-confidence index of KL-UCB at precision `precision`. Look also at the function `maxEV(p, V, klMax)`. The theory suggests $f(t) = \log(t) + \xi \log\log(t)$ with $\xi = 3$, or more recently (see **?**) with $\xi = 0$.

  Make experiments using different values of ξ.

- Implement TS sampling for Bernoulli.

Consider a bandit problem with A arms that are Bernoulli distributions with means $\theta_1, \ldots, \theta_A \in [0, 1]$ The UCB algorithm uses confidence intervals on the unknown mean of each arm to make its decision. In a Bayesian view on the MAB, the θ_a are no longer seen as unknown parameters but as (independent) random variables following a uniform distribution. The posterior distribution on the arm a at time t of the bandit game is the distribution of θ_a conditional to the observations from arm a gathered up to time t. Each sample from arm a leads to an update of this posterior distribution.

Prior distribution $\quad \theta_a \sim U([0; 1])$

Posterior distribution $\quad \theta_a | X_1, \ldots, X_{N_a(t)} \sim \pi_a(t) \stackrel{\text{def}}{=} \text{Beta}(S_a(t) + 1, N_a(t) - Sa(t) + 1)$

where $X_1, \ldots, X_{N_a(t)}$ are the rewards from arm a gathered up to time t, and $S_a(t)$ is the number of rewards equal to 1 received until time t when pulling arm a.

We now present the TS strategy for Bernoulli arms. See **??**.

Algorithm 3 The Thompson sampling algorithm for Bernoulli distributions

---

Initialization: Pull each arm of $\{1, \ldots, K\}$ once

for each round $t + 1$, where $t \geqslant K$:

For each arm a, draw $\theta_a(t) \sim \pi_a(t)$ Pull arm $a_{t+1} \in \underset{a \in \mathcal{A}}{\text{argmax}}\ \theta_a(t)$, then update $\pi_a(t)$.

Implement this strategy: You can make use of the `beta` distribution from the library `scipy.stats`, and the method `rvs()`.

- Compare Strategies

  Compare the regret of TS, `KL-ucb` and UCB strategies on some easy of difficult Bernoulli bandit problems.

  Compare on other bandits problems, such as Gaussian bandit with standard deviation $\sigma = 1/2$.

We specify $\mathcal{D}$ (e.g Bernoulli distributions, or Gaussian distributions, etc.): one `KL-ucb` strategy and one TS strategy for each $\mathcal{D}$.

PAUSE 10 min

3. TP Part C (40min)

Main

- Implement UCB-Laplace:

  The UCB-strategy is dereived from a combination of Hoeffding inequality and crude union bounds. Using The peeling method, or the Laplace method instead, we obtain different strategies.

$$\begin{aligned}
\text{(UCB-peeling)} \quad \mu^+_{a,t} &= \tilde{\mu}_{a,t} + \sqrt{\frac{\alpha}{2N_t(a)} \log\left(\lceil \frac{\log(t)}{\log(\alpha)} \rceil \frac{1}{\delta}\right)}, \text{for } \alpha > 1, \delta \simeq 0 \\
\text{(UCB-Laplace)} \quad \mu^+_{a,t} &= \tilde{\mu}_{a,t} + \sqrt{\frac{(1 + \frac{1}{N_a(t)})}{2N_t(a)} \log\left(\sqrt{N_t(a) + 1}/\delta\right)}, \text{for } \delta \simeq 0
\end{aligned}$$

Choose $\delta = 0.01$ and compare these strategies to UCB, `KL-ucb` and TS.

- Bandits for Gaussian, for Poisson

  Generate bandit environments that are no longer using Bernoulli distributions only, but also Gaussian, etc.

- Implement BESA:

  The best-empirical subsampled arm (BESA) strategy introduced in the paper **?**. It is a very simple strategy that proceeds as follows:

  For 2 arms: If arm a has been pulled 3 times at time t, and arm b has been pulled 10 times, the algorithm sub-samples 3 observations out of the 10 of arm b, then compares the empirical mean built from b with these 3 samples, to the empirical mean built from a. The chosen arm is the one with the highest such empirical mean, and is called the "winner".

  Formally, the winner of the tournament between arm a_1 and a_2 is

  $$\operatorname*{argmax}_{a \in \{a_1, a_2\}} \tilde{\mu}^{\mathsf{sub}}_{a,t}\left(\min_a N_a(t)\right)$$

  where $\tilde{\mu}^{\mathsf{sub}}_{a,t}(n)$ denotes the empirical mean built from n sub-samples chosen uniformly at random without replacement amongst the $N_t(a)$ rewards that are available for arm a at time t. See **?** for further details. For A arms, it uniformly randomly shuffles the arm at time t, then organize a pairwise tournament between the arm: each arm competes another one, then we proceed similarly using the $A = 2$ winners, and proceed similarly until there is a single winner.

  Implement BESA for 2-arms only, and compare it against other strategies, on Bernoulli arms, then Gaussian arms, the others (Poisson, Exponential, etc). Is there a strategy that approximately dominates all others on all problems?

  Bonus: Implement the SDA strategy, and compare its regret.

References