TD – Multi-Armed Bandits

COURS D'APPRENTISSAGE, ECOLE NORMALE SUPERIEURE, OCTOBRE 2018

Aude Genevay (aude.genevay@ens.fr)

We consider the following Bernoulli K-armed bandit setting. Let $\mu \in [0,1]^K$. At each time $t \geq 1$, a learner chooses $I_t \in \{1,\dots,K\}$ and receives reward $X_t(I_t) \sim \mathcal{B}(\mu_{I_t})$. In all the experiments, we will consider a Bernoulli bandit with $K = 2$ arms and means $\mu_1 = 0.6$ and $\mu_2 = 0.5$. The goal of the learner is to maximize his cumulative reward $\sum_{t=1}^n X_t(I_t)$. To do so, the learner minimizes his cumulative regret defined as

$$R_n = n\mu^* - \mathbb{E}\left[\sum_{t=1}^n X_t(I_t)\right], \qquad \text{where} \quad \mu^* = \max_{1\leq i\leq K} \mu_i \,.$$

We are look for algorithms that achieve sublinear regret $R_n = o(n)$ so that the average reward of the learner tends to the optimal reward μ^*.

1. Prove that the regret satisfies $R_n = \sum_{i=1}^K \Delta_i \mathbb{E}\big[T_i(n)\big]$, where $T_i(n) = \sum_{t=1}^n \mathbb{1}_{I_t = i}$ and $\Delta_i = \mu^* - \mu_i$.
2. Implement a Bernoulli bandit environment in Python.

The Follow the leader algorithm (FTL)

The "Follow-the-Leader" algorithm (FTL) chooses each action once and subsequently chooses the action with the largest average observed so far. Ties should be broken randomly:

$$I_t \in \operatorname*{arg\,max}_{1\leq i\leq K} \left\{ \widehat{\mu}_i(t-1) \right\}, \qquad \text{where} \quad \widehat{\mu}_i(t-1) := \frac{1}{T_i(t-1)} \sum_{s=1}^{t-1} X_s(I_s) \mathbb{1}_{I_s = i} \,.$$

3. Implement FTL.
4. Using a horizon of $n = 100$, run 1000 simulations of your implementation of Follow-the-Leader on the Bernoulli bandit above and record the (random) regret, R_n, in each simulation.
5. Plot the results using a histogram and explain the result of the Figure.
6. Rerun the experiments until $n = 1000$ while saving the regrets for each $t = \{1,\dots,n\}$. Plot the average regret obtained in the 1000 simulations as a function of t.
7. Explain the plot. Do you think Follow the Leader is a good algorithm? Why/why not?

The Explore-then-Commit algorithm (ETC)

The Explore-then-Commit algorithm starts by exploring all arms m times (i.e., during the first mK rounds) before choosing the action maximizing $\widehat{\mu}_i(mK)$ for the remaining rounds. Formally, it chooses

$$I_t = \left\{ \begin{array}{ll} i & \text{if} \quad (t \mod K) + 1 = i \quad \text{and} \quad t \leq mK \\ \arg\max_i \left\{ \widehat{\mu}_i(mK) \right\} & \text{if} \quad t > mK \end{array} \right. . \qquad (1)$$

8. Implement ETC.

9. Using horizon of $n = 100$, run 1000 simulations for $m = \{1, 2, 5, 10, 15, 20, 25, 30, 40\}$ and plot the average regret as a function of m. What is the best choice of m?
10. For the different choices of m, rerun the experiment of question 7. Is Explore-Then-Commmit a good algorithm?
11. We consider the case $K = 2$, we assume without loss of generality that the first arm is optimal (i.e., $\mu_1 = \mu^*$). We assume that $n \geq 2m$.
    (a) Show that
    $$\mathbb{E}[T_i(n)] = m + (n - 2m)\mathbb{P}\big(\widehat{\mu}_i(2m) \geq \max_{j \neq i} \widehat{\mu}_i(2m)\big)$$
    (b) Show that
    $$\mathbb{P}\big(\widehat{\mu}_i(2m) \geq \max_{j \neq i} \widehat{\mu}_i(2m)\big) \leq \mathbb{P}\big(\widehat{\mu}_i(2m) - \mu_i - (\widehat{\mu}_1(2m) - \mu_1) \geq \Delta_i\big)$$
    (c) Using Chernoff's inequality prove that
    $$\mathbb{P}\big(\widehat{\mu}_i(2m) - \mu_i - (\widehat{\mu}_1(2m) - \mu_1) \geq \Delta_i\big) \leq \exp\left(-\frac{m\Delta_i^2}{4}\right) .$$
    (d) Conclude that
    $$R_n \leq m\Delta_2 + n\Delta_2 \exp\left(-\frac{m\Delta_2^2}{4}\right) .$$
    (e) Optimize the bound in m assuming n is large and show a bound of the form
    $$R_n \leq \Delta + \frac{\Box}{\Delta}$$
    where $\Box$ is a constant up to log factors. Do you recover a value close to the one obtained by the experiments?
    (f) Remark that the previous bound in unbounded when Δ_2 tends to zero. Show that the worst case bound is $R_n = O(\sqrt{n})$ regardless of the value of Δ.

The Upper-Confidence-Bound algorithm (UCB)

A drawback of ETC is that the optimal value of m depends on Δ which is unknown in advance. Furthermore, all arms are sampled the same number of rounds during the exploration stage. Furthermore, one would want to sample more the arms that are close to be optimal while very bad arms should be quickly detected and stopped being explored. These drawbacks are solved by the Upper-Confidence-Bound algorithm that assigns to each arm a value called upper-confidence bound that with high probability is an upper-bound of unknown mean of the arm. UCB chooses action

$$I_t \in \operatorname*{arg\,max}_{1 \leq i \leq K} \left\{ \widehat{\mu}_i(t-1) + \sqrt{\frac{4 \log n}{T_i(t-1)}} \right\} .$$

It is possible to show that $R_n \leq 3 \sum_{i=1}^{K} \Delta_i + \sum_{i:\Delta_i > 0} \frac{16 \log n}{\Delta_i}$. In the worst case, this implies also $R_n = O(\sqrt{n})$ up to log factors.

12. Implement UCB.
13. Rerun the experiment of question 7 for UCB and compare the cumulative regret of UCB with the ones obtained by ETC (for different m) and FTL.