MVA 2025-26 Pierre Gaillard, Rémy Degenne

Sequential Learning

Home Assignment

This homework should be uploaded by **Friday, March 20, 2026** on the website

`https://sequential-learning.github.io/`

The password to upload is `mva2026`. The penalty scale is minus two points (on the final grade over 20 points) for every day of delay. The homework should consist of a single pdf file, together with an optional single code file. If the upload does not succeed (for some reason), send it by email to pierre.gaillard@inria.fr and remy.degenne@inria.fr but only after you have tried the upload.
It can be done alone or in groups of two students. The code can be done in any language (`python`, `R`, `matlab`,...). The code file (e.g., notebook python, R) does not need to be delivered but the results and the figures must be included into the pdf report. The report can be written in English or in French.

All questions require a proper mathematical justification or derivation (unless otherwise stated), but most questions can be answered concisely in just a few lines. No question should require lengthy or tedious derivations or calculations.

Part 1. Blackwell approachability and Calibration

The goal of this homework is to use Online Gradient Descent to produce calibrated weather forecasts.

Blackwell Approachability via Online Gradient Descent

First, we define the notion of Blackwell approachability that generalizes regret minimization to multi-objective and unfolds as follows. Let $\mathcal{A}$ be a finite action set, and $(b_t)_{1\le t\le T}$ an arbitrary sequence of observations. At each round $t = 1,\dots,T$, the learner chooses $a_t \in \mathcal{A}$, the nature reveals $b_t \in \mathcal{B}$, the learner receives a vector payoff $g(a_t, b_t) \in [-G, G]^d$. We define the average payoff:

$$\bar{g}_T = \frac{1}{T}\sum_{t=1}^{T} g(a_t, b_t).$$

Let $C \subseteq \mathbb{R}^d$ be a closed convex set. We say that C is *approachable* if there exists a strategy such that $\inf_{x\in C} \left\|\bar{g}_T - x\right\|_2 \to 0$ as $T \to \infty$. By Blackwell's approachability theorem (admitted here) C is approachable if and only if

$$\forall \theta \in \mathbb{R}^d, \quad \min_{a\in\mathcal{A}} \sup_{b\in\mathcal{B}} \langle \theta, g(a,b)\rangle \le \sup_{c\in C} \langle \theta, c\rangle \,. \qquad (*)$$

1. Assume that at each round, each action $a \in \mathcal{A}$ is associated with a loss $\ell(a, b_t) \in \mathbb{R}$. By properly choosing $d \ge 1$, $C \subseteq \mathbb{R}^d$ and $g : \mathcal{A} \to \mathbb{R}^d$, show that approachability implies no-regret:

$$\sup_{a\in\mathcal{A}} \sum_{t=1}^{T} \ell(a_t, b_t) - \ell_t(a, b_t) = o(T) \qquad \text{as } T \to \infty \,.$$

We consider the target set $C = \{0\}$ and assume it is approachable. The goal is to design a strategy such that $\|\bar{g}_T\|_2 \to 0$. Define losses

$$\ell_t(\theta) = \langle \theta, -g(a_t, b_t) \rangle, \quad \theta \in \Theta = \{\theta \in \mathbb{R}^d : \|\theta\|_2 \leq 1\}.$$

Let $\theta_1 \in \Theta$ and $\theta_{t+1} = \mathrm{Proj}_\Theta(\theta_t + \eta_t g(a_t, b_t))$ be the iterates obtained by Online Gradient Descent with $\eta_t = \frac{1}{\sqrt{t}}$. At each round, the learner chooses

$$a_t \in \arg\min_{a \in \mathcal{A}} \sup_b \langle \theta_t, g(a, b) \rangle.$$

2. Show that $\langle \theta_t, g(a_t, b_t) \rangle \leq 0$.
3. Deduce that for any $\theta \in \Theta$, $\sum_{t=1}^T \langle \theta, g(a_t, b_t) \rangle \leq O(\sqrt{T})$.
4. Conclude that $\|\bar{g}_T\|_2 = O(T^{-1/2})$.

Application to Calibration

Let $N = 10$. Consider a sequence $y_1, \ldots, y_T \in \{0, 1\}$. At each round t, the player outputs a forecast $p_t \in \mathcal{P} = \left\{0, \frac{1}{N}, \frac{2}{N}, \ldots, 1\right\}$, which is intended to predict the probability that the next outcome to be 1. For instance, one may think of a weather forecaster who, each day, announces the probability of rain for the following day. The realized outcome is $y_t = 1$ if it rains and $y_t = 0$ otherwise.

The forecaster's performance is evaluated using the notion of *calibration*. Informally, calibration requires that for every forecast value $p \in \mathcal{P}$, the empirical frequency of rain

$$\rho_T(p) := \frac{\sum_{t=1}^T y_t \mathbb{1}\{p_t = p\}}{\sum_{t=1}^T \mathbb{1}\{p_t = p\}}$$

on the days when the forecaster predicted a probability p should itself be close to p. In other words, among all days where the forecast was, say, 0.7, it should rain on approximately 70% of those days.

The forecaster is ε-*calibrated* if for every $p \in \mathcal{P}$, almost surely

$$\Big(\frac{1}{T}\sum_{t=1}^T \mathbb{1}\{p_t = p\}\Big) |\rho_T(p) - p| \leq \varepsilon + o(1)$$

as $T \to \infty$.

Consider a randomized forecaster that outputs a probability vector $a_t \in \mathcal{A} := \Delta_{\mathcal{P}}$ and samples $p_t \sim a_t$. Define the payoff vectors with coordinates

$$g_i(a, y) = \mathbb{E}_{p \sim a}\big[\mathbb{1}\{p = i/N\}(y - p)\big] = a_i\big(y - \frac{i}{N}\big).$$

5. Show that $C = \{c \in \mathbb{R}^N : \|c\| \leq \varepsilon\}$ is approachable by proving Blackwell's condition for a well-chosen ε. What is the best ε that you can get?
6. Explain how the previous algorithm yields an ε-calibrated forecasting strategy (for the ε of the previous question) with rate $O(1/\sqrt{T})$.

Application to rain forecasts

Download rain data from ENS Paris Saclay (Gif-sur-Yvette).

```python
from meteostat import Daily, Point
from datetime import datetime

location = Point(48.709, 2.166)
start = datetime(2010, 1, 1)
end = datetime(2025, 12, 31)
data = Daily(grenoble, start, end).fetch()
data["rain"] = (data["prcp"] > 0).astype(int)
```

7. Apply the previous algorithm to produce rain forecasts in $\mathcal{P}$. Plot the evolution of the Brier score

$$\sum_{p \in \mathcal{P}} \Big(\rho_t(p) - p\Big)^2 \Big(\frac{1}{t} \sum_{s=1}^{t} \mathbb{1}\{p_s = p\}\Big)$$

as a function of t.

Part 2. Stochastic Best Arm Identification

In the best arm identification setting, an algorithm interacts with the environment in the standard bandit way: at each time, it selects an arm, then observes the corresponding reward. The goal of the algorithm is to find the arm with highest mean, as quickly as possible.

Each arm $k \in [K]$ has a reward distribution ν_k with mean μ_k. There is a unique best arm $k^* = \arg\max_k \mu_k$.

At each round $t = 1, \ldots, \tau$

- The player chooses an arm $k_t \in [K]$,
- The player observes a reward $X_t^{k_t} \sim \nu_{k_t}$, independent of all other rewards.

At the stopping time τ, the algorithm recommends an arm $\hat{k} \in [K]$

Setting 1: Best arm identification

A good algorithm should make few mistakes and stop quickly. With the notations of Figure 1, the probability of mistake of the algorithm on problem ν is $\mathbb{P}_\nu(\hat{k} \neq k^*)$. The possibly random time at which it stops and returns an answer is τ.

Notations: $N_t^k = \sum_{s=1}^t \mathbb{I}\{k_s = k\}$ is the number of times arm k was sampled up to time t. $\hat{\mu}_{t,k} = \frac{1}{N_t^k} \sum_{s=1}^t \mathbb{I}\{k_s = k\} X_s^{k_s}$ is the empirical mean of arm k.

Fixed Budget In *fixed budget* best arm identification, we are given a time T and set $\tau = T$. That is, the algorithm can sample a total number of T arms (T known in advance), then must stop and return an answer. We are interested in algorithms with low probability of mistake.

We suppose in this fixed budget section that the distributions of the arms all have support in $[0, 1]$.

1. *Uniform sampling.* The uniform sampling algorithm pulls all arms T/K times (we suppose that T is a multiple of K).

   (a) Prove that the probability of error of uniform sampling is at most $2\sum_{k=2}^{K}\exp(-\frac{T}{K}\frac{\Delta_k^2}{8})$, where $\Delta_k = \mu_{k^*} - \mu_k$.

```
Algorithm 1: Successive rejects
  Input: budget $T$, number of arms $K$.
  Let $A_1 = \{1,\dots,K\}$, $\overline{\log}(K) = \frac{1}{2} + \sum_{k=2}^{K}\frac{1}{k}$, $n_0 = 0$ and for $j \in \{1,\dots,K-1\}$,
      $$n_j = \left\lceil \frac{1}{\overline{\log}(K)} \frac{T-K}{K+1-j} \right\rceil .$$
  for each phase $j = 1, 2, \dots, K-1$ do
      For each $i \in A_j$, select arm $i$ during $n_j - n_{j-1}$ rounds.
      Let $A_{j+1} = A_j \setminus \arg\min_{k\in A_j} \hat{X}_{k,n_j}$, where $\hat{X}_{k,n_j}$ is the empirical mean of arm $k$ after $n_j$
        observations. (we only remove one element from $A_j$; if there is a tie, select randomly
        the arm to dismiss among the worst arms).
  end
```

2. The *Successive Rejects* algorithm is described in Algorithm 1. In each phase, it samples all arms uniformly, and at the end of a phase it discards the worse arm.

   (a) Give a bound on the probability that the best arm is discarded at the end of the first phase.

   (b) Let $B = \mathbb{I}\{\hat{k} \neq k^*\}$ be the Bernoulli random variable with value 1 if the algorithm makes a mistake and 0 otherwise. Its expectation on the bandit problem ν is $\mathbb{P}_\nu(\hat{k} \neq k^*)$. Suppose that we run n parallel experiments, and that in each experiment $i \in [n]$ we run successive rejects on the same bandit ν and compute the corresponding $B_i = \mathbb{I}\{\hat{k} \neq k^*\}$. Give a confidence interval for $\mathbb{P}_\nu(\hat{k} \neq k^*)$.

   (c) Implement successive rejects and uniform sampling. Plot the probability of error of both algorithms for $K = 20$ Bernoulli arms with $\mu_1 = 0.6$ and $\mu_k = 0.5$ for $k \geq 2$, for $T \in \{100, 500, 2000\}$. Plot confidence intervals and make sure to use enough experiments to get intervals with smaller width than the error probability.

Fixed Confidence In *fixed confidence* best arm identification, we consider only *δ-correct* algorithms, which satisfy $\mathbb{P}_\nu(\hat{k} \neq k^*) \leq \delta$ for all tuples of distributions in our model. We are then looking for such algorithms with minimal expected stopping time $\mathbb{E}_\nu[\tau]$.
All arm distributions in this section will be Gaussian with variance 1. All experiments will use 10 such arms, with means $(0.5, 0.6, 0.5, 0.4, \dots, 0.4)$. All experiments will use $\delta = 0.01$.
Stopping rule. Let $\hat{*}_t = \arg\max_{k\in[K]} \hat{\mu}_{t,k}$. All algorithms will use the same stopping rule: stop when

$$\inf_{k\in[K]\setminus\{\hat{*}_t\}} \frac{1}{2} \frac{(\hat{\mu}_{t,\hat{*}_t} - \hat{\mu}_{t,k})^2}{\frac{1}{N_t^{\hat{*}_t}} + \frac{1}{N_t^k}} > \log\frac{1}{\delta} + 3\log(1 + \log t) .$$

τ is the first time such that this condition is satisfied. This is a heuristic approximation of the GLRT stopping rule seen in the course (the quantity on the left is equal to $\inf_{\lambda: *_\lambda \neq \hat{*}_t} \sum_{k=1}^{K} N_t^k \frac{1}{2}(\hat{\mu}_{t,k} - \lambda_k)^2$).

1. In order to get a sub-linear regret, a regret minimization algorithms has to sample mostly the best arm. It could thus be transformed into a best arm identification algorithm.

   (a) Implement the UCB algorithm for regret minimization, which samples the arm $k_t = \arg\max \hat{\mu}_{t-1,k} + \sqrt{\frac{2\log t}{N_{t-1}^k}}$. Plot the mean over 100 experiments of the regret of UCB on the Gaussian bandit problem described above, for t from 1 to 10000.

   (b) Implement a best arm identification algorithm that samples like UCB, stops according to the rule presented above, and returns the most played arm. Implement another algorithm that samples all arms uniformly, stops according to the rule presented above, and returns the arm with highest empirical mean. On a box plot, compare the stopping times of both algorithms.

2. Regret minimization algorithms don't explore enough to be good identification methods. We can modify them to explore more: this is the idea of *Top-Two* algorithms. A Top-Two algorithm computes at each time t a *leader* $B_t \in [K]$ and a *challenger* $C_t \in [K] \setminus \{B_t\}$, and then with probability $\beta \in (0,1)$ it samples the leader ($k_t = B_t$), and it samples the challenger with probability $1 - \beta$. It stops according to the rule presented above, and recommends the arm with highest empirical mean. We will use $\beta = 1/2$.

   (a) Implement the TTUCB algorithm: its leader is the UCB arm $B_t = \arg\max \hat{\mu}_{t-1,k} + \sqrt{\frac{2\log t}{N_{t-1}^k}}$ and its challenger is the arm for which it is hardest to say that it is worse, $C_t = \arg\min_{k \neq B_t} \frac{1}{2} \frac{(\hat{\mu}_{t-1,B_t} - \hat{\mu}_{t-1,k})^2}{\frac{1}{N_{t-1}^{B_t}} + \frac{1}{N_{t-1}^k}}$.

   (b) We added an exploration mechanism, the challenger, and it is possible that using UCB for the leader is not necessary anymore: we could simply select the empirical best arm, without the exploration bonus of UCB. Implement the so-called EB-TC algorithm which uses $B_t = \arg\max \hat{\mu}_{t-1,k}$ and $C_t = \arg\min_{k \neq B_t} \frac{1}{2} \frac{(\hat{\mu}_{t-1,B_t} - \hat{\mu}_{t-1,k})^2}{\frac{1}{N_{t-1}^{B_t}} + \frac{1}{N_{t-1}^k}}$.

   (c) On a box plot, compare the stopping times of all 4 algorithms (UCB, uniform, TTUCB and EB-TC) and comment the results.

3. (Bonus question) Thompson Sampling is another regret minimization algorithm. Implement a Top Two algorithm which uses Thompson Sampling. Give a pseudo-code of your algorithm. Compare that algorithm to the others on various bandit problems.