MVA 2022-23 Pierre Gaillard, Rémy Degenne

Sequential Learning
Home Assignment

This homework should be uploaded by **Friday, March 17, 2023** as a pdf file on the website

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

The password to upload is `mva2023`. The penalty scale is minus two points (on the final grade over 20 points) for every day of delay. The homework can be done alone or in groups of two students. The code can be done in any langage (`python`, `R`, `matlab`,. . .) and should not be returned but the results and the figures must be included into the pdf report.

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. Bandit convex optimization

We consider the following setting of online zero-order convex optimization. Let $\Theta \subseteq \mathbb{R}^d$ be a convex decision set that contains the unit ball and with diameter $D := \sup_{\theta,\theta' \in \Theta} \|\theta - \theta'\|$. First, the environment chooses a sequence of convex and G-Lipschitz loss functions $\ell_t : \Theta \to [-1, 1]$, initially hidden from the learner. At each round $t \geq 1$, the learner is asked to choose an action $\theta_t \in \Theta$ and observes its loss $\ell_t(\theta_t)$. We consider the following algorithm.

```
Algorithm 1: OGD without gradients
  Input: decision set $\Theta$, $\delta > 0$ and $\eta > 0$
  Initialization: $\hat{\theta}_1 = 0$
  for $t$=1,. . . ,$T$ do
      Draw $u_t \in \mathbb{S}_1$ uniformly at random and set $\theta_t = \hat{\theta}_t + \delta u_t$
      Play $\theta_t$ and observes $\ell_t(\theta_t)$
      Update $\hat{\theta}_{t+1} = \mathrm{Proj}_{\Theta_\delta}\left[\hat{\theta}_t - \frac{d\eta}{\delta}\ell_t(\theta_t)u_t\right]$ where $\mathrm{Proj}_{\Theta_\delta}$ is the Euclidean projection on $\Theta_\delta := \{\theta, \frac{\theta}{1-\delta} \in \Theta\}$.
  end
```

1. Is the adversary oblivious or adaptive in this setting?

2. Write the definition of the regret R_T.

3. Define the δ-smoothed version of $\ell_t(\theta)$ by $\hat{\ell}_t(\theta) := \mathbb{E}_v[\ell_t(\theta + \delta v)]$ where v is uniformly sampled on the unit ball.
   (a) Show that when $d = 1$,
   $$\mathbb{E}_{u_t}\left[\frac{d}{\delta}\ell_t(\hat{\theta}_t + \delta u_t)u_t\right] = \nabla\hat{\ell}_t(\hat{\theta}_t)\,.$$
   We admit that the results holds for all $d \geq 1$ (this can be proved by using Stoke's theorem).
   (b) Show that for any $\theta \in \Theta_\delta$,
   $$\left|\hat{\ell}_t(\theta) - \ell_t(\theta)\right| \leq G\delta.$$

4. Define $h_t(\theta) = \hat{\ell}_t(\theta) + \langle \xi_t, \theta \rangle$, where $\xi_t = \frac{d}{\delta}\ell_t(\theta_t)u_t - \nabla \hat{\ell}_t(\hat{\theta}_t)$.
   (a) What would be the updates of Online Gradient Descent applied to the sequence of losses h_t?
   (b) Show that for any $\theta^*_\delta \in \Theta_\delta$
   $$\sum_{t=1}^{T} h_t(\hat{\theta}_t) - h_t(\theta^*_\delta) \le \eta \frac{d^2}{\delta^2} T + \frac{D^2}{\eta} .$$
   (c) Deduce that for any $\theta^*_\delta \in \Theta_\delta$
   $$\sum_{t=1} \mathbb{E}\big[\hat{\ell}_t(\hat{\theta}_t)\big] - \hat{\ell}_t(\theta^*_\delta) \le \eta \frac{d^2}{\delta^2} T + \frac{D^2}{\eta} .$$

5. Conclude that
$$\mathbb{E}\big[R_T\big] \le \eta \frac{d^2}{\delta^2} T + \frac{D^2}{\eta} + 4\delta D G T .$$

6. What regret do we obtain by optimizing the parameters δ and η?

7. In this question, we perform some simulations to assess the empirical performance of the algorithm and compare it with standard Online Gradient Descent. Let $d, T \ge 1$. We define $\theta^*_i = \frac{1}{2i}$, and let for $t = 1, \dots, T$, $x_t \sim \mathcal{N}(0, I_d)$ and $y_t = \langle \theta^*, x_t \rangle + \epsilon_t$, with $\epsilon_t \sim \mathcal{N}(0, .1)$. At each round, the learner is asked to form $\theta_t \in \mathbb{R}^d$ and is evaluated with the loss $\ell_t(\theta) = (\langle \theta, x_t \rangle - y_t)^2$.
   (a) Let $d = 2$. Implement OGD with and without gradients for $\Theta = \{\|\theta\| \le 1\}$. Plot the cumulative regrets obtained for theoretical values of η and δ for $n = 1, \dots, 1000$. Add standard deviation obtained over 100 runs of the experiment to the plots.
   (b) Fix $n = 1000$. Plot the regret with standard deviation as a function of $d = 1, \dots, 10$. What do you observe?

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 $t = 1, \dots, \tau$
– 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 $\mathbb{P}_\nu(\hat{k} \ne k^*)$. 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.

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 2: 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 2. 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 $\mu_1 = 0.5$ and $\mu_k = 0.4$ 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 $(0.5, 0.4, 0.4, 0.3, \dots, 0.3)$. 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, $C_t = \arg\min_{k \neq B_t} \frac{1}{2} \frac{(\hat{\mu}_{t,B_t} - \hat{\mu}_{t,k})^2}{\frac{1}{N_t^{B_t}} + \frac{1}{N_t^k}}$.
   (b) We added an exploration mechanism, the challenger, and it could be that using UCB for the leader is not necessary anymore. Implement the so-called EB-TC algorithm which uses $B_t = \arg\max \hat{\mu}_{t-1,k}$ and $C_t = \arg\min_{k \neq B_t} \frac{1}{2} \frac{(\hat{\mu}_{t,B_t} - \hat{\mu}_{t,k})^2}{\frac{1}{N_t^{B_t}} + \frac{1}{N_t^k}}$.
   (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.