Sequential Learning **MVA 2020**

Homework — March 6

Instructor: Pierre Gaillard *Student: Céline Moucer, Antoine Moulin*

This homework is due by **Friday March 6, 2020**. It is to be returned by email to pierre.gaillard@inria.fr **as a pdf file** (not a jupyter notebook). The code can be done in any langage (`python`, `R`, `matlab`,...) but the results and the figures must be included into the pdf report.

Most 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. Link between online learning and game theory

We consider the sequential version of a two-player zero-sum games between a player and an adversary.

Let $L \in [-1,1]^{M\times N}$ be a loss matrix.
At each round $t = 1, \dots, T$
- The player choose a distribution $p_t \in \Delta_M := \{p \in [0,1]^M, \sum_{i=1}^M p_i = 1\}$
- The adversary chooses a distribution $q_t \in \Delta_N$
- The actions of both players are sampled $i_t \sim p_t$ and $j_t \sim q_t$
- The player incurs the loss $L(i_t, j_t)$ and the adversary the loss $-L(i_t, j_t)$.

Setting 1: Setting of a sequential two-player zero sum game

1. Recall M, N and a loss matrix $L \in [-1,1]^{M\times N}$ that corresponds to the game "Rock paper scissors"[1].

Solution: For the game "Rock paper scissors", the player and the adversary have three possible actions, so:

$$\boxed{N = M = 3}$$

and the loss is (in a basis "Rock paper scissors"):

$$\boxed{L = \begin{pmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{pmatrix}}$$

[1] This is a common game where two players choose one of 3 options: (Rock, Paper, Scissors). The winner is decided according to the following: Rock crushes scissors, Paper covers Rock, Scissors cuts paper

Full information feedback In this part, we assume that both players know the matrix L in advance and can compute $L(i, j)$ for any (i, j).

2. *Implementation of* `EWA`.

   (a) In order to implement the exponential weight algorithm, you need a way to sample from the exponential weight distribution. Implement the function `rand_weighted` that takes as input a probability vector $p \in \Delta_M$ and uses a single call to `rand()` to return $X \in [M]$ with $P(X = i) = p_i$.

Solution: See code.

   (b) Define a function `EWA_update` that takes as input a vector $p_t \in \Delta_M$ and a loss vector $\ell_t \in [-1, 1]^M$ and return the updated vector $p_{t+1} \in \Delta_M$ defined for all $i \in [M]$ by

$$p_{t+1}(i) = \frac{p_t(i) \exp(-\eta \ell_t(i))}{\sum_{j=1}^{M} p_t(j) \exp(-\eta \ell_t(j))} .$$

Solution: See code.

3. *Simulation against a fixed adversary.* Consider the game "Rock paper scissors" and assume that the adversary chooses $q_t = (1/2, 1/4, 1/4)$ and samples $j_t \sim q_t$ for all rounds $t \geq 1$.

   (a) What is the loss $\ell_t(i)$ incurred by the player if he chooses action i at time t? Simulate an instance of the game for $t = 1, \ldots, T = 100$ for $\eta = 1$.

Solution: The loss incurred by the player if he chooses action i at time t is:

$$\boxed{\ell_t(i) = L(i, j_t)}$$

   (b) Plot the evolution of the weight vectors $p_1, p_2, \ldots, p_T$. What seems to be the best strategy against this adversary?

Solution: From this figure, we can conclude that the best strategy against this adversary is to play

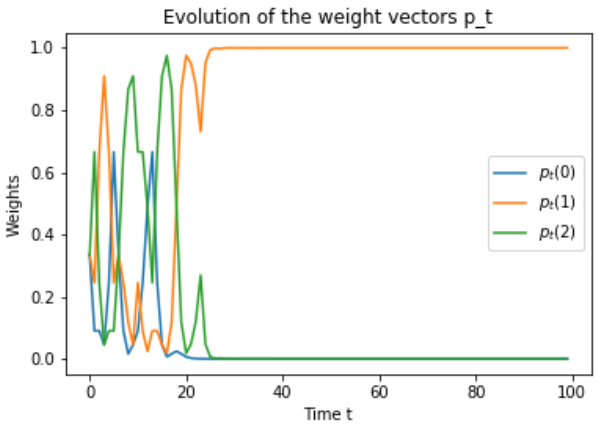


Figure 1: Evolution of the weight vectors from the EWA algorithm for $\eta = 1$ and $T = 100$, as a function of t.

the action 1 all the time. It is indeed quite coherent with the game, since the action 1 (paper) wins over the action 0 (rock), which is the most played since the adversary chooses it half of the time ($q_t = (1/2, 1/4, 1/4)$).

(c) Plot the average loss $\bar{\ell}_t = \frac{1}{t}\sum_{s=1}^{t} \ell(i_s, j_s)$ as a function of t.

Solution: We notice that the average loss tends to be negative, meaning the player is winning in

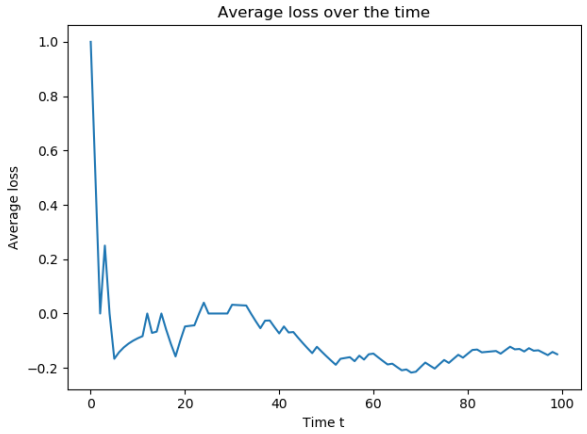


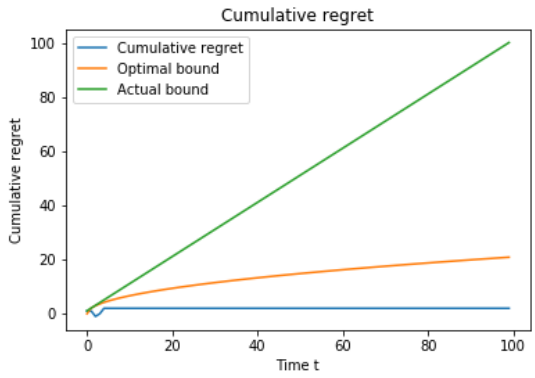
Figure 2: Average loss for EWA algorithm for $\eta = 1$ and $T = 100$, as a function of t.

the long run. It shows that the EWA strategy is working against a non-adaptative adversary. The limit seems to be -0.2, which corresponds to the expected loss when the probability distributions are $p = (0, 1, 0)$ and $q = (1/2, 1/4, 1/4)$.

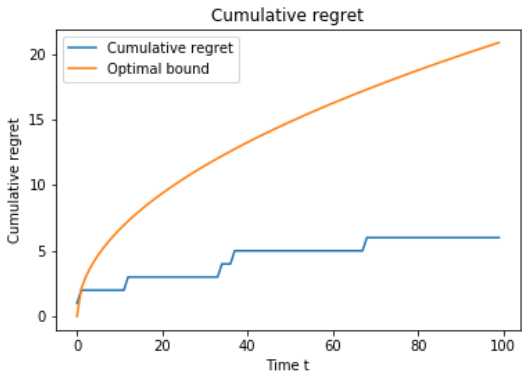
(d) Plot the cumulative regret.

Solution: Let us plot the cumulative regret compared to:

- The optimal bound (for the optimal η): $f(t) = 2\sqrt{tlog(3)}$.
- The actual bound: $f(t) = \eta t + \frac{log(3)}{\eta}$.



(a) Cumulative regret of the EWA algorithm for $\eta = 1$ and $T = 100$, as a function of t, compared with the real bound and its optimal one.



(b) Cumulative regret of the EWA algorithm for $\eta = 1$ and $T = 100$, as a function of t, compared with the optimal bound.

Figure 3: Cumulative regret of the EWA.

The cumulative regret is below the actual and the optimal bound, even with a non optimal value of η.

(e) To see if the algorithm is stable, repeat the simulation $n = 10$ times and plot the average loss $(\bar{\ell}_t)_{t\geq 1}$ obtained in average, in maximum and in minimum over the n simulations.

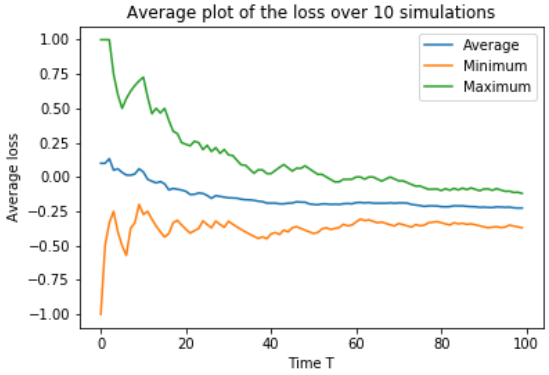


Figure 4: Average loss for EWA over 10 simulations for $\eta = 1$ and $T = 100$, as a function of t.

Solution: The plot is given in figure 4 and shows that the algorithm is stable: the minimum and the maximum loss are both concentrating around the average loss, what indicates a low variance average loss when t increases.

(f) Repeat one simulation for different values of learning rates $\eta \in \{0.01, 0.05, 0.1, 0.5, 1\}$ and plot the final regret as a function of η. What are the best η in practice and in theory.

Solution:

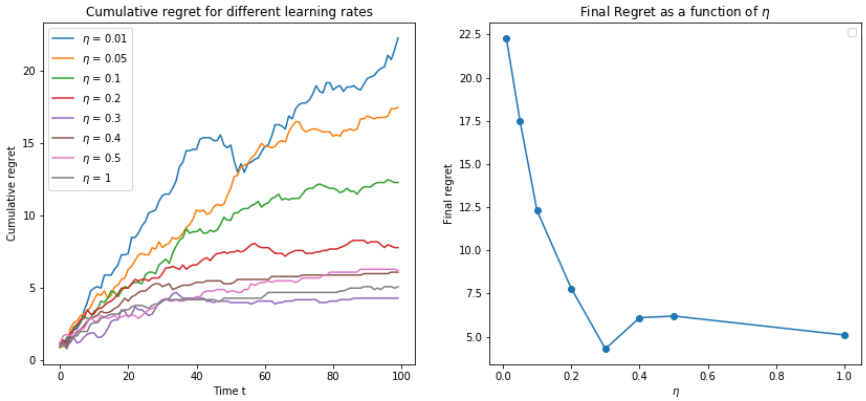


Figure 5: Average loss for EWA for different values of *eta* and $T = 100$, as a function of t.

The best values for η are obtained for $\eta \in [0.2, 1]$. We performed the simulation for T=100, and for T=1000, since the final regrets are used to compare η. In theory, the best value for η is achieved with:

$$\eta = \sqrt{\frac{log(M)}{T}}$$

Thus, $\eta \approx 0.03$ if $T = 1000$, and $\eta \approx 0.1$ for $T = 100$. However, we reach better results for higher values of η. Indeed, the plot in figure 9 showed that the optimal bound is not tight. We could explain this phenomenon with the quite simple structure of the problem. Lower values of η allow to explore more first. But here, since the action is not adaptative, higher values of η only allows to converge quicker to action 1 (question (3.b)), and keep the loss slow in the first plays.

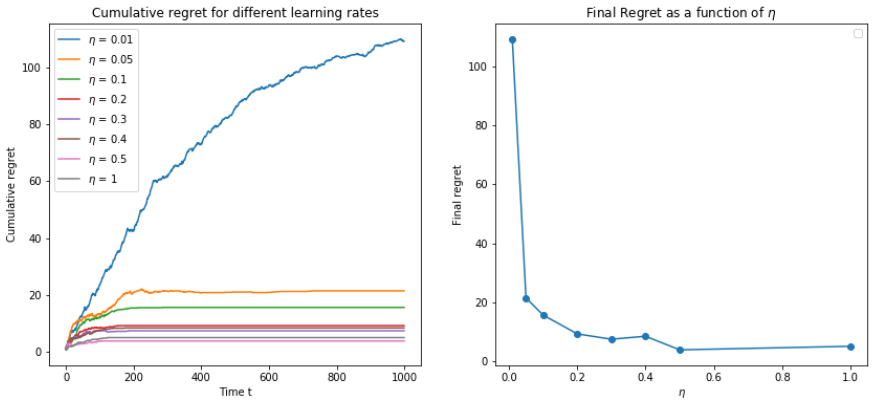


Figure 6: Average loss for EWA for different values of *eta* and $T = 1000$, as a function of t.

4. *Simulation against an adaptive adversary.* Repeat the simulation of question 3) when the adversary is also playing EWA with learning parameters $\eta = 0.05$.

   (a) Plot $\frac{1}{t}\sum_{s=1}^{t} \ell(i_s, j_s)$ as a function of t.

Solution: In this question, the adversary becomes adaptative with EWA updates and $\eta_2 = 0.05$, the player still follows EWA updates with $\eta_1 = 0.05$.

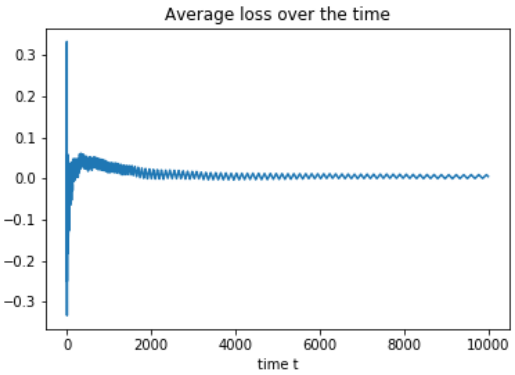


Figure 7: Average loss for an adaptative EWA adversary as a function of t, with $T = 10000$, $\eta_1 = 0.05$, and $\eta_2 = 0.05$.

The average loss converges to 0. This is coherent with the course (if the adversary follows a regret minimization algorithm, the average loss converges almost surely). Besides, since the losses are symmetric for the player and its adversary, the value of the limit is also coherent.

> It is possible to show that if both players play according to a regret minimizing strategy the cumulative loss of the player converges to the value of the game
>
> $$V = \min_{p \in \Delta_M} \max_{q \in \Delta_q} p^\top L q \,.$$

   (b) Define $\bar{p}_t = \frac{1}{t}\sum_{s=1}^{t} p_s$. Plot in log scale $\|\bar{p}_t - (1/3, 1/3, 1/3)\|_2$ as a function of $t = 1, \ldots, 10\,000$.

Solution:

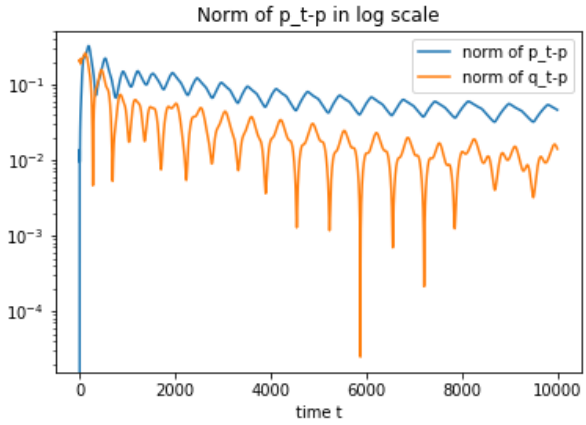


Figure 8: Plot of $||\bar{p}_t - (1/3, 1/3, 1/3)||_2$ and $||\bar{q}_t - (1/3, 1/3, 1/3)||_2$, for $\eta = 0.05$ and $T = 10000$

The norm $\|\bar{p}_t - (1/3, 1/3, 1/3)\|_2$ converges to 0, and oscillates while decreasing, and so does $\|\bar{q}_t - (1/3, 1/3, 1/3)\|_2$. This is again coherent with the Nash equilibrium : both player follows a minimization regret algorithm, and thus tend to have a uniform action.

> It is possible to show that $(\bar{p}_t, \bar{q}_t)_{t\geq 1}$ converges almost surely to a Nash equilibrium of the game. This means that if $p \times q$ is a Nash equilibrium, none of the players should change is strategy if the other player does not change hers.

Bandit feedback Now, we assume that the players do not know the game in advance but only observe the performance $L(i_t, j_t)$ (that we assume here to be in $[0, 1]$) of the actions played at time t. They need to learn the game and adapt to the adversary as one goes along.

5. *Implementation of* ***EXP3***. Since both players are symmetric, we focus on the first player.

   (a) Implement the function `estimated_loss` that takes as input the action $i_t \in [M]$ played at round $t \geq 1$ and the loss $L(i_t, j_t)$ suffered by the player and return the vector of estimated loss $\widehat{\ell}_t \in \mathbb{R}_+^M$ used by `EXP3`.

Solution: See the code.

   (b) Implement the function `EXP3_update` that takes as input a vector $p_t \in \Delta_M$, the action $i_t \in [M]$ played by the player and the loss $L(i_t, j_t)$ and return the updated weight vector $p_{t+1} \in \Delta_M$.

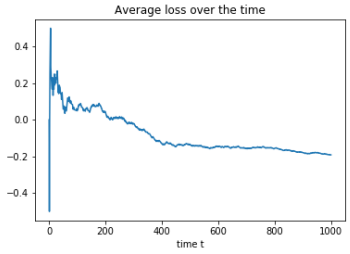
Solution: See the code.

6. Repeat Questions 3.a) to 3.f) with `EXP3` instead of `EWA`.

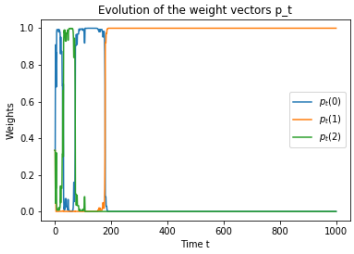
Solution: When using `EXP3` instead of `EWA`, we observe that even though the algorithm takes more time to learn the optimal strategy, we obtain similar results for questions 3.a) to 3.f). Indeed, although the information is now incomplete, the game is simple enough so the algorithm can learn to counter a fixed adversary and have a low regret. To see the plots, one can just run the script `main.py`. Here are some of them in figure 9.

We observe the same behavior for the weight vectors and the average loss as for the EWA algorithm. The cumulative regret is again under the optimal bound even if the η value is not the optimal one.

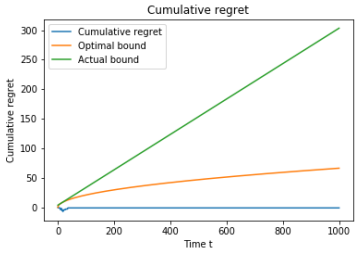
This comparison highlights the differences between EWA and EXP3. Indeed, the best cumulative regret seems to be much higher than on figure 6 for EWA, and the value of η more discriminant.



(a) Cumulative regret of EXP3, for $\eta = 1, T = 1000$, as a function of t, compared with the real and optimal bounds.



(b) Evolution of the weight vectors for EXP3, $\eta = 1$ and $T = 1000$, as a function of t.



(c) Average loss for EXP3 over 10 simulations, for $\eta = 1$ and $T = 1000$, as a function of t

Figure 9

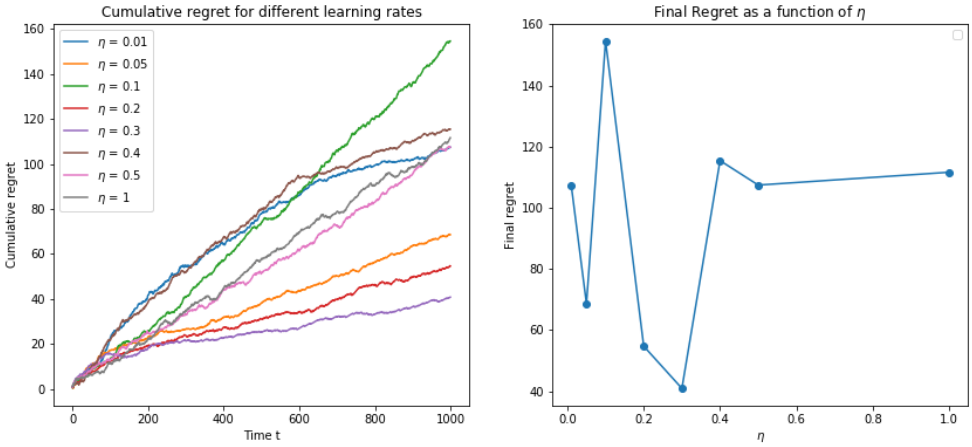


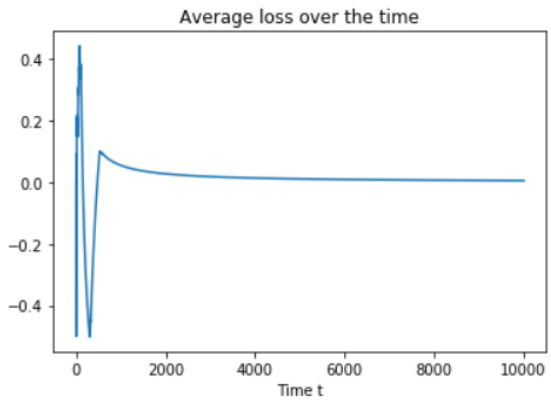
Figure 10: Average loss for EXP3 as a function of t, with η varying on a grid, and $T = 1000$.

Even if the best η is again around 0.3, small variations around this values induces big differences in the cumulative regret. Thus, EXP3 is less stable than EWA when considering the optimal η value.

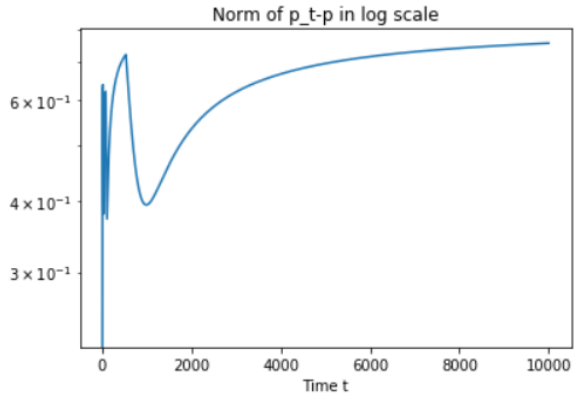
7. Repeat Question 4.a) and 4.b) with `EXP3` instead of `EWA`.

Solution: As previously, we can obtain a loss of 0, but `EXP3` is less stable than `EWA`. Indeed, although we have a loss of 0 as shown in figure 14a, the strategy obtained is not the uniform distribution but a constant one: both the player and the adversary always choose the same action.

As a result, when we plot the distance between the player's distribution and the uniform distribution, it does not tend to 0, as shown in figure 14b.



(a) Average loss as a function of t when playing `EXP3` vs `EXP3`.



(b) Distance between the player's distribution and the uniform distribution.

Figure 11

Optional extentions The following questions are optional.

8. Repeat Question 4.a) when the adversary is playing a `UCB` algorithm. Who wins between `UCB` and `EXP3`?

Solution:

9. In the last lecture, we see that EXP3 has a sublinear expected regret. Yet, as shown by question 6.e), it is extremely unstable with a large variance. Implement `EXP3.IX` (see Chapter 12 of [2]) a modification of `EXP3` that controls the regret in expectation and simultaneously keeps it stable. Repeat question 3.e) with `EXP3.IX`.

Solution:

10. Try different games (not necessarily zero-sum games). In particular, how these algorithms behave for the prisoner's dilemna (see wikipedia)? The prisoner's dilemna is a two-player games that shows why two completely rational individuals might not cooperate, even if it appears that it is in their best interests to do so. The losses matrices are:

$$L^{(player)} = \begin{pmatrix} 1 & 3 \\ 0 & 2 \end{pmatrix} \quad \text{and} \quad L^{(adversary)} = \begin{pmatrix} 1 & 0 \\ 3 & 2 \end{pmatrix}.$$

Solution:

Part 2. Theory – Sleeping experts

The classical definition of regret compares the performance of an algorithm with the performance of the best "constant" action. But in some applications, some actions may be sometimes unavailable. The purpose of this exercise is to deal with this issue.

We consider the following full-information setting with a finite decision set $\mathcal{X} := \{1, \dots, K\}$. At each time $t \geq 1$, a subset of active decisions $A_t \subseteq \mathcal{X}$ is available, the other decisions are sleeping (or inactive) and cannot be chosen; the player chooses a distribution p_t over active decision A_t (i.e., $\sum_{j \in A_t} p_t(j) = 1$ and $p_t(k) = 0$ for $k \notin A_t$) and observes the loss $\ell_t(k) \in [0, 1]$ of all decisions in A_t.

The sleeping regret is defined

$$R_T(k) := \sum_{t=1}^{T} \big(p_t \cdot \ell_t - \ell_t(k)\big) \mathbb{1}\{k \in A_t\}\,, \qquad \text{(Sleeping regret)}$$

with respect to decision $k \in \mathcal{X}$, where $p_t \cdot \ell_t = \sum_{j \in A_t} p_t(j)\ell_t(j)$ is the loss of the player.

11. **The prod algorithm** Here, we consider the case where all experts are active $A_t = \mathcal{X}$ for all $t \geq 1$. Let $0 \leq \eta(1), \dots, \eta(K) \leq 1/2$ be K parameters. We define the weights

$$p_t(k) = \frac{\eta(k) w_t(k)}{\sum_{j=1}^{K} \eta(j) w_t(j)} \qquad \text{where} \quad \begin{aligned} w_t(k) &= \prod_{s=1}^{t-1} \Big(1 + \eta(k)\big(p_s \cdot \ell_s - \ell_s(k)\big)\Big) \quad \text{if } t \geq 2 \\ w_1(k) &= 1\,, \end{aligned} \qquad (*)$$

for all $k \in \mathcal{X}$ and $t \geq 1$.

(a) Prove that $\log(1+x) \geq x - x^2$ for $x \geq -1/2$.

Solution: Let $g : x \mapsto \log(1+x) - x + x^2$ defined on $\left[-\frac{1}{2}, +\infty\right)$. This function is differentiable and for $x \geq -\frac{1}{2}$:

$$g'(x) = \frac{1}{1+x} - 1 + 2x = \frac{x\,(1+2x)}{1+x}$$

g' is negative for $x \in \left(-\frac{1}{2}, 0\right)$, null in $x \in \left\{-\frac{1}{2}, 0\right\}$ and positive for $x > 0$.

Hence, for all $x \geq -\frac{1}{2}$, $g(x) \geq g(0)$, *i.e.* $g(x) \geq 0$. Thus,

$$\boxed{\forall x \geq -\frac{1}{2},\ \log(1+x) \geq x - x^2}$$

(b) Denoting $W_t = \sum_{k=1}^{K} w_t(k)$. Prove that for all $k \in \mathcal{X}$

$$\log W_{T+1} \geq \eta(k) \sum_{t=1}^{T} \big(p_t \cdot \ell_t - \ell_t(k)\big) - (\eta(k))^2 \sum_{t=1}^{T} \big(p_t \cdot \ell_t - \ell_t(k)\big)^2$$

Solution: Let $k \in \mathcal{X}$. As the weights are positive and that log is nondecreasing, we have:

$$\log W_{T+1} = \log\left(\sum_{i=1}^{K} w_{T+1}(i)\right) \geq \log\left(w_{T+1}(k)\right)$$

First, since $0 \leq \eta(k) \leq \frac{1}{2}$ and since the loss is positive, we have :

$$\eta(k)\left(p_s \cdot \ell_s - \ell_s(k)\right) \geq -\ell_s(k)\eta(k) \geq -\eta(k) \geq -\frac{1}{2}$$

Besides, using the definition of the weights and the previous question :

$$\begin{aligned} \log\left(w_{T+1}(k)\right) &= \log\left[\prod_{s=1}^{T} \left(1 + \eta(k)\left[p_s \cdot \ell_s - \ell_s(k)\right]\right)\right] \\ &= \sum_{s=1}^{T} \log\left(1 + \eta(k)\left[p_s \cdot \ell_s - \ell_s(k)\right]\right) \\ &\geq \sum_{s=1}^{T} \Big(\eta(k)\left[p_s \cdot \ell_s - \ell_s(k)\right] - \eta(k)^2\left[p_s \cdot \ell_s - \ell_s(k)\right]^2\Big) \\ &= \eta(k) \sum_{s=1}^{T} \left(p_s \cdot \ell_s - \ell_s(k)\right) - \eta(k)^2 \sum_{s=1}^{T} \left(p_s \cdot \ell_s - \ell_s(k)\right)^2 \end{aligned}$$

Thus,

$$\boxed{\forall k \in \mathcal{X},\, \log W_{T+1} \geq \eta(k) \sum_{s=1}^{T} (p_s \cdot \ell_s - \ell_s(k)) - \eta(k)^2 \sum_{s=1}^{T} (p_s \cdot \ell_s - \ell_s(k))^2}$$

(c) Show that $W_{t+1} = W_t$ for all $t \geq 1$. What is the value of $\log(W_{T+1})$?

Solution: Let $t \geq 1$. Using the definition of the weights, we have:

$$\begin{aligned} W_{t+1} &= \sum_{k=1}^{K} w_{t+1}(k) \\ &= \sum_{k=1}^{K} w_t(k) \left[1 + \eta(k) \left(p_t \cdot \ell_t - \ell_t(k)\right)\right] \\ &= W_t + (p_t \cdot \ell_t) \sum_{k=1}^{K} w_t(k)\eta(k) - \sum_{k=1}^{K} w_t(k)\eta(k)\ell_t(k) \end{aligned}$$

We denote $\mathbf{w}_t = (w_t(1), \ldots, w_t(K))^\top$ and $\eta = (\eta(1), \ldots, \eta(K))^\top$. By definition of $p_t(k)$ for all k,

$$\begin{aligned} W_{t+1} &= W_t + (p_t \cdot \ell_t) \left(\mathbf{w}_t^\top \eta\right) - \sum_{k=1}^{K} \left(\mathbf{w}_t^\top \eta\right) p_t(k)\ell_t(k) \\ &= W_t + (p_t \cdot \ell_t) \left(\mathbf{w}_t^\top \eta\right) - (p_t \cdot \ell_t) \left(\mathbf{w}_t^\top \eta\right) \\ &= W_t \end{aligned}$$

Hence,

$$\boxed{\forall t \geq 1,\, W_{t+1} = W_t}$$

We can deduce $W_{T+1} = W_1 = K$ and:

$$\boxed{\log W_{T+1} = \log K}$$

(d) Assuming $\eta(k)$ are well-optimized, show the regret bound for all arms $k \in [K]$

$$\sum_{t=1}^{T} p_t \cdot \ell_t - \ell_t(k) \leq 2\sqrt{(\log K) \sum_{t=1}^{T} \left(p_t \cdot \ell_t - \ell_t(k)\right)^2}\,.$$

Solution: Let $k \in \mathcal{X}$. Using the two previous questions,

$$\eta(k) \sum_{s=1}^{T} (p_s \cdot \ell_s - \ell_s(k)) \leq \log K + \eta(k)^2 \sum_{s=1}^{T} (p_s \cdot \ell_s - \ell_s(k))^2$$

We assume $\eta(k) \neq 0$:

$$\sum_{s=1}^{T} (p_s \cdot \ell_s - \ell_s(k)) \leq \frac{1}{\eta(k)} \log K + \eta(k) \sum_{s=1}^{T} (p_s \cdot \ell_s - \ell_s(k))^2$$

Let $\alpha \in \mathbb{R}$ and $g : x \mapsto \frac{1}{x}\log K + \alpha x$. g is differentiable and for all $x \neq 0$ $g'(x) = \alpha - \frac{1}{x^2}\log K$. g' is null in $x^* = \sqrt{\frac{\log K}{\alpha}}$, negative before and then positive, which shows this is a minimum. We have $g(x^*) = 2\sqrt{\alpha \log K}$. Thus,

$$\boxed{\forall k \in \mathcal{X},\ \sum_{t=1}^{T}(p_t \cdot \ell_t - \ell_t(k)) \leq 2\sqrt{(\log K)\sum_{t=1}^{T}(p_t \cdot \ell_t - \ell_t(k))^2}}$$

12. **Sleeping experts** Now, we assume that some decisions are sometimes not possible (sleeping), i.e., $A_t \subsetneq \mathcal{X}$ for some $t \geq 1$. The idea is to use Algorithm $(*)$ with past modified losses

$$\tilde{\ell}_t(k) := \begin{cases} \ell_t(k) & \text{if } k \in A_t \\ p_t \cdot \ell_t = \sum_{k \in A_t} p_t(k)\ell_t(k) & \text{if } k \notin A_t \end{cases},$$

i.e., by assigning the loss of the algorithm $p_t \cdot \ell_t$ to all inactive decisions $k \notin A_t$. The algorithm outputs weights $\tilde{p}_t(k)$ and $\tilde{w}_t(k)$ obtained by replacing $\ell_t(k)$ with $\tilde{\ell}_t(k)$ in Equation $(*)$. This vector is then used to form another weight vector

$$p_t(k) = \frac{\tilde{p}_t(k)\mathbb{1}_{k \in A_t}}{\sum_{j=1}^{K}\tilde{p}_t(j)\mathbb{1}_{j \in A_t}}$$

which has non zero weights only on active arms A_t.

(a) Show that the instantaneous regret on the modified losses equals the sleeping regret on the original rewards; i.e. for all $t \geq 1$, and all $k \in \mathcal{X}$

$$\tilde{p}_t \cdot \tilde{\ell}_t - \tilde{\ell}_t(k) = \big(p_t \cdot \ell_t - \ell_t(k)\big)\mathbb{1}_{k \in A_t}\,.$$

Solution: Let $t \geq 1$. We first note that as p_t has nonzero weights only on active arms, we have:

$$p_t \cdot \ell_t = \sum_{k \in A_t} p_t(k)\ell_t(k) = \sum_{k=1}^{K} p_t(k)\ell_t(k)\mathbb{1}_{k \in A_t} = \sum_{k=1}^{K} p_t(k)\ell_t(k)$$

Let $k \in \mathcal{X}$. By definition of $\tilde{p}_t$ and $\tilde{\ell}_t$:

$$\begin{aligned}
\tilde{p}_t \cdot \tilde{\ell}_t &= \sum_{i=1}^{K}\tilde{p}_t(i)\tilde{\ell}_t(i) \\
&= \sum_{i=1}^{K}\tilde{p}_t(i)\left(\ell_t(i)\mathbb{1}_{i \in A_t} + (p_t \cdot \ell_t)\,\mathbb{1}_{i \in A_t^c}\right) \\
&= \sum_{i=1}^{K}\left(\sum_{j=1}^{K}\tilde{p}_t(j)\mathbb{1}_{j \in A_t}\right)p_t(i)\ell_t(i) + (p_t \cdot \ell_t)\sum_{i=1}^{K}\tilde{p}_t(i)\mathbb{1}_{i \in A_t^c} \\
&= (p_t \cdot \ell_t)\left(\sum_{j=1}^{K}\tilde{p}_t(j)\mathbb{1}_{j \in A_t} + \sum_{i=1}^{K}\tilde{p}_t(i)\mathbb{1}_{i \in A_t^c}\right) \\
&= p_t \cdot \ell_t
\end{aligned}$$

Besides, $\tilde{\ell}_t(k) = \ell_t(k)\mathbb{1}_{k \in A_t} + (p_t \cdot \ell_t)\,\mathbb{1}_{k \in A_t^c}$. Thus,

$$\boxed{\forall k \in \mathcal{X},\ \tilde{p}_t \cdot \tilde{\ell}_t - \tilde{\ell}_t(k) = (p_t \cdot \ell_t - \ell_t(k))\,\mathbb{1}_{k \in A_t}}$$

(b) Conclude that $R_T(k) \leq 2\sqrt{(\log K)T_k}$ where $T_k = \sum_{t=1}^{T} \mathbb{1}\{k \in A_t\}$ is the number of times arm k is active.

Solution: Let us recall the definition of $\tilde{\ell}_t(k)$

$$\tilde{\ell}_t(k) := \begin{cases} \ell_t(k) & \text{if } k \in A_t \\ p_t \cdot \ell_t = \sum_{k \in A_t} p_t(k)\ell_t(k) & \text{if } k \notin A_t \end{cases},$$

Since $l_t(k) \in [0,1]$, $\tilde{\ell}_t(k) \in [0,1]$, and since $\tilde{p}_t$ and $\tilde{w}_t$ are constructed as in 11, we can apply the results of question 11. Besides, we have $(p_t \cdot \ell_t - \ell_t(k)) \leq 1$. Thus, $\forall k \in \mathcal{X}$:

$$\begin{aligned} \sum_{t=1}^{T} \left(\tilde{p}_t \cdot \tilde{\ell}_t - \tilde{\ell}_t(k)\right) &\leq 2\sqrt{(\log K) \sum_{t=1}^{T} \left(p_t \cdot \tilde{\ell}_t - \tilde{\ell}_t(k)\right)^2} \\ &\leq 2\sqrt{(\log K) \sum_{t=1}^{T} \left((p_t \cdot \ell_t - \ell_t(k))\, \mathbb{1}_{k \in A_t}\right)^2} \\ &\leq 2\sqrt{(\log K) \sum_{t=1}^{T} \mathbb{1}_{k \in A_t}} \\ &\leq 2\sqrt{(\log K)\, T_k} \end{aligned}$$

Thus :

$$\boxed{\forall k \in \mathcal{X}, \sum_{t=1}^{T} \left(p_t \cdot \tilde{\ell}_t - \tilde{\ell}_t(k)\right) \leq 2\sqrt{(\log K)\, T_k}}$$

Part 3. Experiments – predict votes of surveys

In these experiments, we will apply online convex optimization algorithms to pairwise comparison datasets. Comparison data arises in many different applications such as sport competition, recommender systems or web clicks. We consider the following sequential setting. Let $\mathcal{Z} = \{1, \ldots, N\}$ be a finite set of items (for example football teams in a competition).

At each iteration $t \geq 1$,

- the learner receives the labels of two items that are competing $z_t = (z_t(1), z_t(2)) \in \mathcal{Z}^2$
- the learner predicts $\hat{y}_t \in (0,1)$ the probability of victory of item $z_t(1)$.
- the environment reveals the result of the match $y_t = 1$ if item $z_t(1)$ wins the match and $y_t = 0$ otherwise (if team $z_t(2)$ wins).

The learner aims at minimizing his cumulative loss: $\hat{L}_T = \sum_{t=1}^{T} \ell(\hat{y}_t, y_t)$, where $\ell(\hat{y}_t, y_t) = (1 - \hat{y}_t)y_t + \hat{y}_t(1 - y_t)$.

13. Justify the choice of ℓ.

Solution: When dealing with a classification problem like this one, a natural choice would be to minimize the cross-entropy. However, because the definition involves a logarithm, this loss may not be smooth enough for the algorithms we want to use, such as the Online Gradient Descent which requires the loss to have a bounded gradient. Hence, we use a similar but smoother loss, as defined above, to solve this classification problem.

Datasets We consider two datasets from [3] that contain surveys of civic comparisons (can be download at `http://pierre.gaillard.me/teaching/mva`). Each dataset consists of two files of $T = 15\,000$ rows corresponding to votes:

- ideas dataset: the participants are suggested two politic ideas such as ('free beer' vs 'free ice cream') and are asked to vote for the best.
- politicians dataset: the participants are asked which political figure within a pair such as ('Obama' vs 'Goldman Sachs') had "the worse year in Washington."

The datasets contain two files:

- `ideas-id.csv` (resp. `politicans-id.csv`) that contains id and text of the ideas (resp. political figures).
- `ideas-votes.csv` (resp. `politicians-votes.csv`) that contains the id of the two competing ideas (resp. political figures) in z1 and z2 and a column y which is 1 if the participant voted for z1 and 0 otherwise.

The goal of the learner is to sequentially predict the results of the votes minimizing the number of mistakes.

14. Implement the Exponentially Weighted Average Forecaster (EWA) and Online Gradient Descent (OGD) (and optionally the Prod forecaster of question 1) with parameter $\eta > 0$ that at each round $t \geq 1$ take a finite set of predictions $f_t(1), \dots, f_t(K) \in [0,1]^K$ and forecast $\hat{y}_t = \sum_{k=1}^K p_t(k) f_t(k) \in [0,1]$ the probability that idea 1 wins the vote. [2]

    For the euclidean projection onto the simplex, see [1].

Solution: See the code for the implementation details.

15. We consider the sleeping strategies indexed by $k \in \{1, \dots, 2N\}$ that predict for $1 \leq k \leq N$

$$f_t(k) = \begin{cases} 1 & \text{if } k = z_t(1) \\ 0 & \text{if } k = z_t(2) \\ \emptyset & \text{otherwise} \end{cases} \qquad \text{and} \qquad f_t(k+N) = \begin{cases} 0 & \text{if } k = z_t(1) \\ 1 & \text{if } k = z_t(2) \\ \emptyset & \text{otherwise} \end{cases},$$

    where $\emptyset$ means that the strategy is sleeping. Basically, $f_t(k)$ (resp. $f_t(k+N)$) predicts always the victory (resp. loss) of the idea k during the votes. Remark that the sleeping trick of question 2) works for any algorithm so that we might replace $\emptyset$ with the prediction of the algorithm itself $\hat{y}_t$. Run the two algorithms of the preceding question (EWA, OGD) with these predictions $f_t(1), \dots, f_t(K) \in [0,1]^K$.

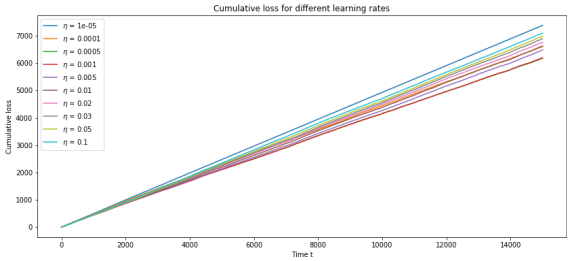
    (a) Plot the cumulative loss of the algorithms at $T = 15\,000$ according to different values of $\eta \in (0, 1/2)$ chosen in a grid.

        In this question, we plot the cumulative loss for different values of η, computes with the OGD update and the EWA update.

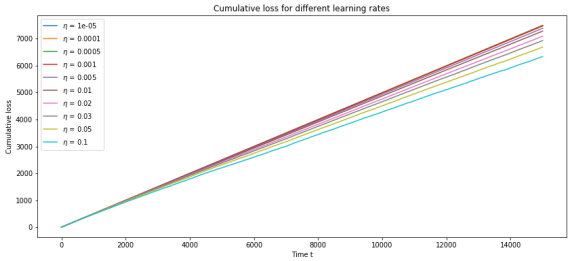
Solution:

In both case, the best cumulative loss is obtained for $\eta = 0.001$. However, it is not the value of $\eta = \sqrt{\frac{log(K)}{T}} = 0.02$ for which the regret has an optimal bound. Lower and higher values for η increases the cumulative loss.

[2] This question does not require any answer in the final report. $f_t(1), \dots, f_t(K)$ are prediction of experts that are inputs, they will be defined explicitely in the next question.



(a) Cumulative loss when playing OGD for different η values

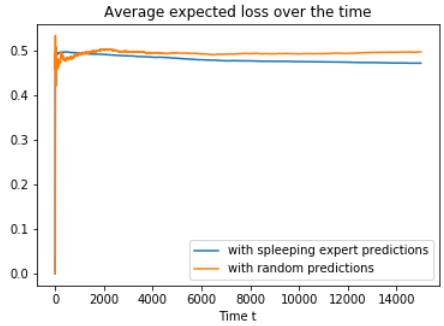


(b) Cumulative loss when playing EWA for different η values

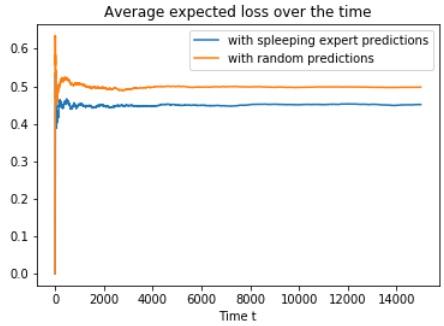
Figure 12

(b) Plot the average expected loss of the algorithms $(1/t)\hat{L}_t$ according to the number of rounds $t = 1, \dots, T$ (i.e., number of votes) for well-chosen values of η (justify the choice). Do the algorithms beat random predictions? Based on the observation of question a), we choose $\eta = 0.001$ for the next plots.

Solution:



(a) Average loss when playing EWA for $\eta = 0.001$, compared with random predictions



(b) Average loss when playing OGD for $\eta = 0.001$, compared with random predictions

Figure 13

EWA produces a slightly better prediction when compared with the average loss of the random prediction. Its accuracy is around 62%. OGD produces a much better solution in terms of average loss, from the beginning, and that has an accuracy around 60 %.

(c) At each round $t \geq 1$, assume that the algorithms predict the vote $\hat{Y}_t = 1$ with probability $\hat{y}_t$ and 0 otherwise. For each algorithm (for the η chosen in question 6(a)), plot its true average loss

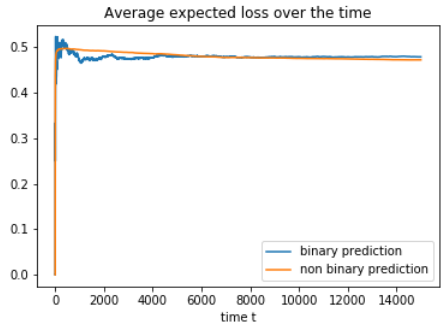
$$\frac{1}{t}\sum_{s=1}^{t} \mathbb{1}_{\hat{Y}_s \neq y_s},$$

according to $t = 1, \dots, T$.

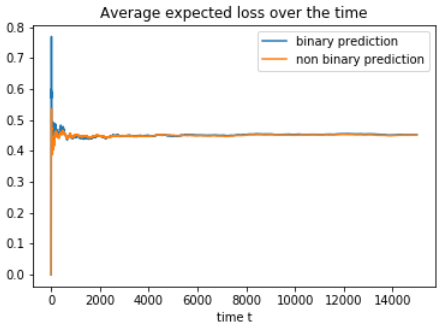
Solution:

Even if the true average is a dirac, which is non convex and whose gradient is not bounded, its loss is almost the same as the loss from question 14), for both updates.

16. (optional) Explore different ideas to improve the final performance. For example, you can add new sleeping strategies to be combined or you can perform OGD or EG to estimate the



(a) Average loss when playing EWA for $\eta = 0.001$, compared with binary predictions



(b) Average loss when playing OGD for $\eta = 0.001$, compared with binary predictions

Figure 14

best Bradley Terry model (https://en.wikipedia.org/wiki/Bradley-Terry_model) on the fly,...

References

[1] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279. ACM, 2008.

[2] Tor Lattimore and Csaba Szepesvári. Bandit algorithms. *preprint*, page 28, 2018.

[3] Matthew J Salganik and Karen EC Levy. Wiki surveys: Open and quantifiable social data collection. *PloS one*, 10(5):e0123483, 2015.