Prediction of individual sequences : homework

Lucas CLARTE

06 mars 2020

Part 1 - Link between online learning and game theory

All the code is written in the Python script `homework.py`.

1. For the game "Rock Paper scissors", we define the actions rock = 1, paper = 2, scissors = 3. The loss matrix L is as follows

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

2. (a) We implement the function `rand_weighted(p)` that samples $\imath \in [|M|]$ with respect to the distribution $p \in \Delta_M$. In Python, the function is simply

```
def rand_weighted(p) :
        return np.argwhere(np.cumsum(p) > np.random.rand())[0, 0]
```

In this function, we build an array `c = np.cumsum(p)` that contains the cumulative probabilities of p i.e $c_i = p_1 + ... + p_i$. We then sample a number $x \in [0, 1)$ and determine the first index i in the cumulative array c such that $c_i > x$. It is easy to show that for all j, $i = j$ with probability p_j.

2. (b) To implement the function `EWA_update(p, l)` we simply multiply the vectors p and $\exp(-\eta l_t(i))$ component-wise and then normalize the new vector.

3. (a.)) We simulate EWA against a fixed adversary with strategy $q = (0.5, 0.25, 0.25)$, in the function `EWA(L, T, eta, q`. This function runs T iterations. At iteration t, we sample an action j_t from q using `rand_weighted` and update the weights p_t with the function `EWA_update` called with the loss l_t. The loss $l_t(i)$ of the player if he choses the action i is equal to $L(i, j_t)$.

3. (b) As asked, we simulate the game with $T = 100, \eta = 1.0$ and plot the weight vectors $p_1, ..., p_T$ in the figure (**??**) . We see that the best strategy is $p = (0.0, 1.0, 0.0)$. We can prove it rigorously. Indeed, consider the strategy $p = (x, y, z)$ that minimizes the average loss $l(p, q) = \mathbb{E}_{i \sim p, j \sim q}(L(i, j))$. We write

$$\begin{aligned} l(p, q) &= x \times (0.25 - 0.25) + y \times (0.25 - 0.5) + z \times (0.5 - 0.25) \\ &= -0.25 \times y + 0.25 \times z \end{aligned}$$

The optimal value of p that satisifies the constraint $p \in \Delta_3$ and minimizes the above expression is $x = 0, y = 1, z = 0$ which is what we wanted.

3. (c) We plot the average loss $\bar{l}_t = \frac{1}{t} \sum_{1 \leqslant s \leqslant t} l(i_s, j_s)$ as a function of t, and obtain the figure (**??**). The figure shows 10 different runs of the simulation.

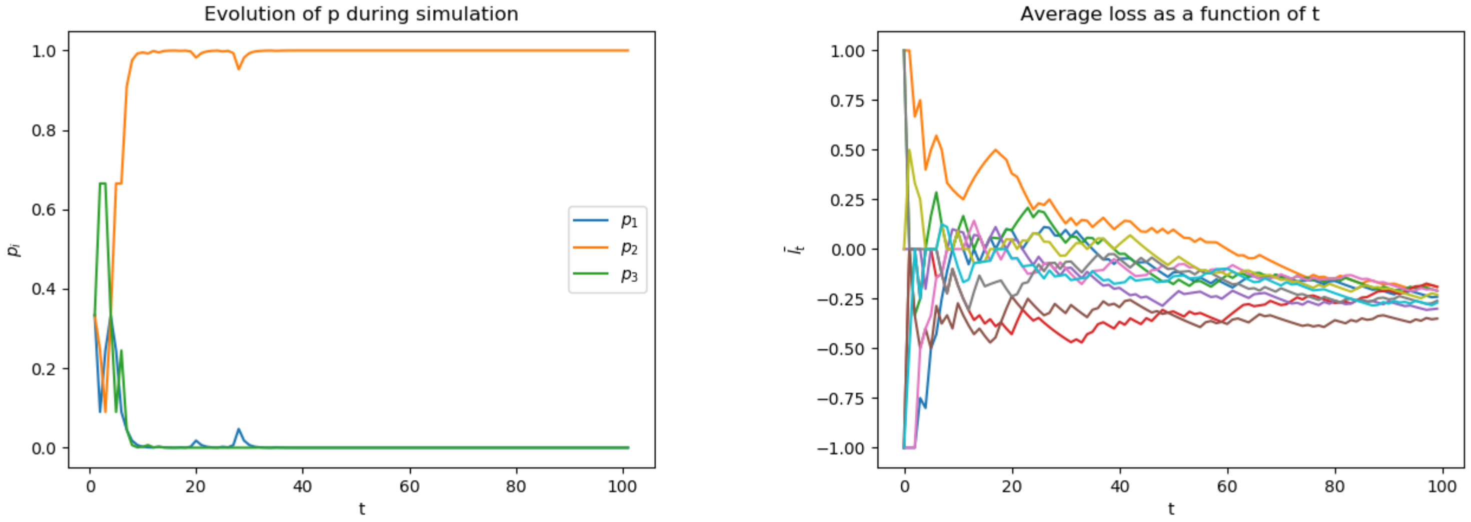


Figure 1: Left : plot of the player's stragegy p with `EWA`, when opponent has a fixed strategy (0.5, 0.25, 0.25). Right : plot of the average loss as a function of t for 10 simulations of the game with `EWA`

3. (d) Here, the cumulative regret is defined by

$$R_t = \sum_{s=1}^{t} L(i_s, j_s) - \min_i \sum_{s \leqslant t} L(i, j_s) \tag{1}$$

Figure (**??**) shows the cumulative regret for 10 trials with T = 100. We observe experimentally that for each trial, after a certain time the cumulative regret is constant.

3. (e) To estimate the stability, n = 10 simulations are executed, and we plot in figure (**??**) the n average losses as well as the minimum, maximum and mean losses. At first glance, the `EWA` seems to be stable. Indeed, the maximum and minimum both seem to converge towards the value $l_\infty = -0.25$.

3. (f) In theory, the best learning rate η with `EWA` is $\eta_{\text{EWA}} = \sqrt{2ln(K)/T}$ with $K = 3$ here and $T = 100$, so we obtain $\eta_{\text{EWA}} \simeq 0.15$. We see that in practice, the best learning rate is not necessarily equal to η_{EWA}.

4. *Simulation against an adaptive adversary* In this question, the player still uses `EWA` with a learning rate $\eta = 1.0$ while the opponent uses a learning rate $\eta = 0.05$

4.(a) When the adversary uses `EWA` like the player, we observe that the average loss seems to converge towards 0 which is the value of the game.

4. (b)

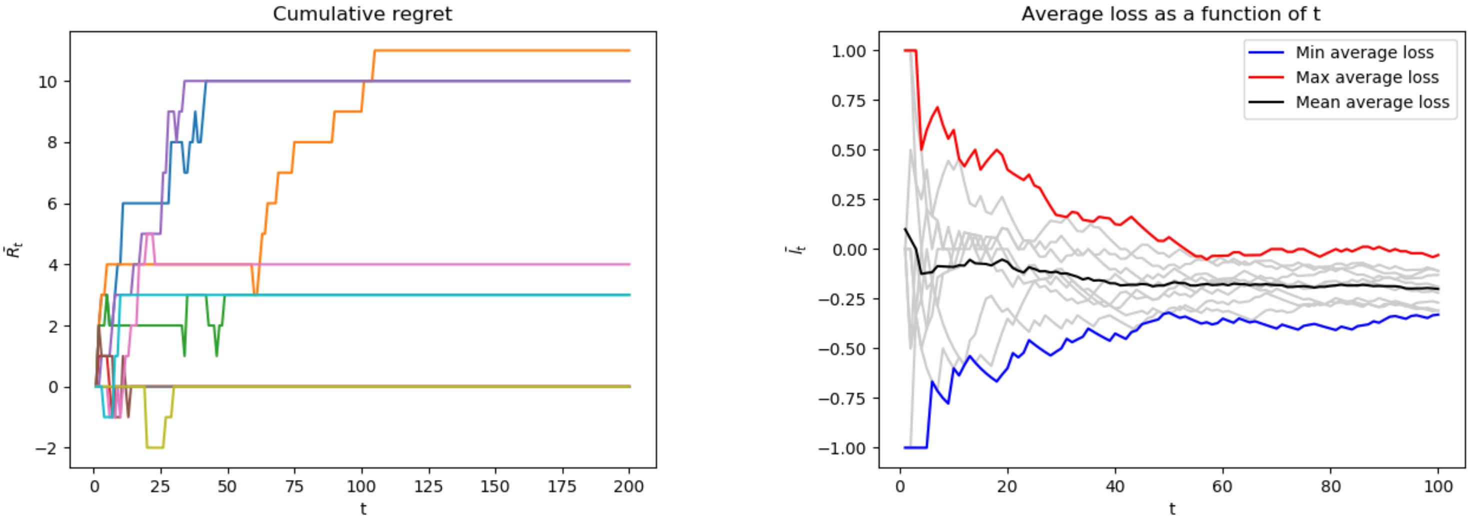


Figure 2: Left : Cumulative regret .Right : average, minimum and maximum of the average loss $\bar{l}_t$ for n = 10 simulation trials. The n average losses are plotted in grey.

Bandit feedback We now assume that the player doesn't have access to L but only the incured loss $L(i_t, j_t)$. We will here implement the algorithm `Exp3` where at each time t, the action i_t is selected with a probability

$$\mathbb{P}(i_t = i) = \frac{\exp\left(-\eta \hat{l_t}(i)\right)}{\sum_j \exp\left(-\eta \hat{l_t}(j)\right)} \tag{2}$$

where $\hat{l_t}(i)$ is the *estimated loss* of the action i, i.e the average loss incured by the player when the action i was selected.

5.(a) To implement the function `estimated_loss`, we maintain a vector $C \in \mathbb{N}^M$ where at each time, C_i is the number of times the action i has been played. We also maintain the vector $S \in \mathbb{R}^M$ where $S_i = \sum_i L(i_t, j_t)\mathbf{1}(i_t = i)$ is the sum of the losses occured by the player every time they chose the action i. The function `estimated_loss(i)` simply returns S_i/C_i.

5. (b)

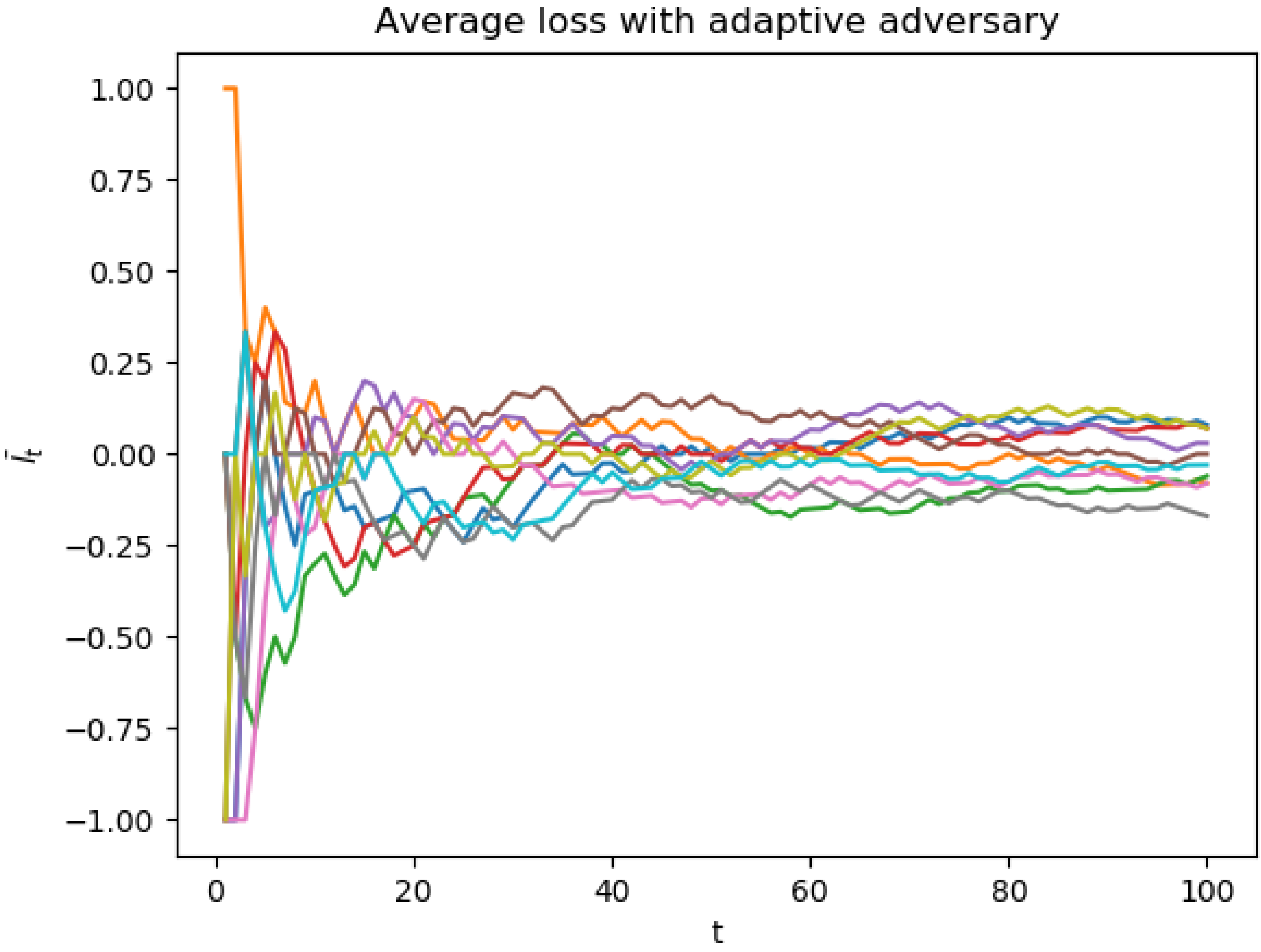
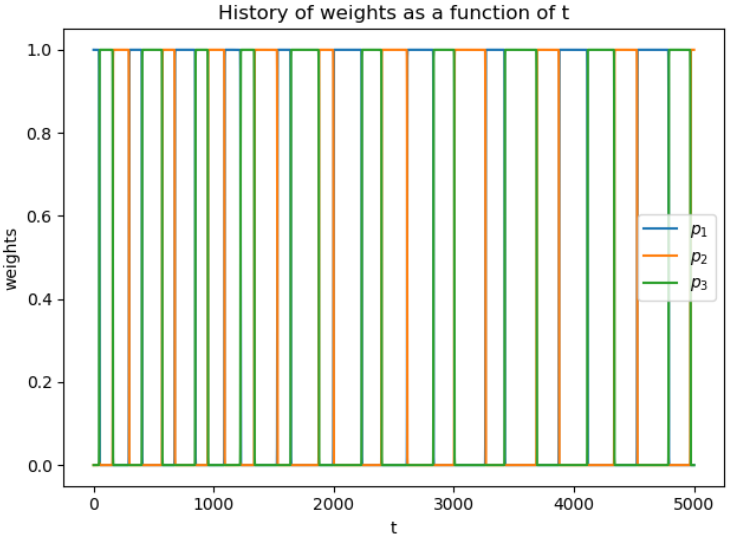
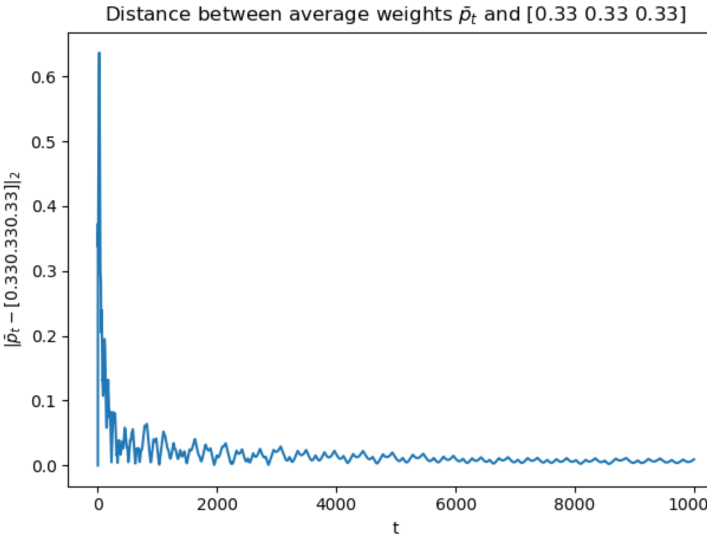


Figure 3: Average loss for n = 10 simulations with adaptive adversary



History of weights as a function of t



Distance between average weights $\bar{p}_t$ and [0.33 0.33 0.33]