Predictions of Individual Sequences

Home Assignment

This homework is due by **Friday March 1, 2018**. It is to be returned by email to pierre.gaillard@inria.fr as a pdf file. The code can be done in any langage (`python`, `R`, `matlab`,...) 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. 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.

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

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

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

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

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

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

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

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

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

2. **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 $k \in \mathcal{X}$

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

(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.

Part 2. 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 $\mathcal{Z} = \{1, \dots, N\}$ 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 $\ell(\hat{y}_t, y_t) = (1 - \hat{y}_t)y_t + \hat{y}_t(1 - y_t)$.

3. Justify the choice of ℓ.

Datasets We consider two datasets from [2] 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.

4. 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.

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

5. 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.

   (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?

   (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

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

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

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