Online Learning

Y. Chevaleyre

M2 IASD - Univ. Dauphine - PSL

November 2, 2023

Outline

Introduction and Learning Protocols

Standard setting (Batch)

Protocole

- The learner receives $S = (x_1, y_1) \dots (x_N, y_N) \sim \mathcal{P}^N$
- The learner generates f_S (with ERM, ERM régularisé, ...)

Objectif: minimise $\hat{R}(f_S) = \frac{1}{N} \sum_{i=1}^{N} \ell(f_S(x_i), y_i)$, or (ideally) minimise $R(f_S)$

Online Learning Protocol

Protocol

For $t = 1$ to T

- The environment chooses x_t, y_t, and reveals x_t to the learner
- **The learner predicts $\hat{y}_t$**
- The environment reveals y_t
- The learner endures the cost $\ell(\hat{y}_t, y_t)$

- Notes:
  - ex: mails SPAMs detection
  - The environment can produce arbitrary couples x_t, y_t (not i.i.d)
  - Study of worst case = zero sum two player game
  - We can have $T = \infty$
  - To simplify, we will study the realizable case with $\ell(\hat{y}, y) = 1[\hat{y} \neq y]$

Objective: minimise $\sum_{t=1}^{T} \ell(\hat{y}_t, y_t)$ the cumulated loss

Realizable case with 0/1 loss and finite $\mathcal{F}$

ERM Algorithm (Empirical Risk Minimization)

- Let $\mathcal{F}$ be a family of classifiers.

Algorithm

For $t = 1$ to T

- Receive x_t
- **Choose arbitrarily** $f_t \in \mathcal{F}$ **among those who perfectly classify previous data (zero error)**
- **Predict** $\hat{y}_t = f_t(x_t)$
- Receive the true label y_t, and my prediction costs $\ell(\hat{y}_t, y_t)$

Algorithme ERM

- Alternative formulation

Algorithme

$\mathcal{F}_1 = \mathcal{F}$

For $t = 1$ to T

- Receive x_t
- **Choose arbitrarily** $f_t \in \mathcal{F}_t$
- **Predict** $\hat{y}_t = f_t(x_t)$
- Receive the true label y_t, and my prediction costs me $\ell(\hat{y}_t, y_t)$
- **Update** $\mathcal{F}_{t+1} = \{f \in \mathcal{F}_t : f(x_t) = y_t\}$

Failure of ERM

Halving Algorithm

- Let $\mathcal{F}$ be a family of classifiers

Algorithm

$\mathcal{F}_1 = \mathcal{F}$
For $t = 1$ to T

- Receive x_t
- Let $\mathcal{F}_t^k = \{f \in \mathcal{F}_t : f(x) = k\}$, for all $k \in \mathcal{Y}$
- **Predict** $\hat{y}_t = \arg\max_{k \in \mathcal{Y}} \left|\mathcal{F}_t^k\right|$
- Receive the true label y_t, and my prediction costs me $\ell(\hat{y}_t, y_t)$
- **Update** $\mathcal{F}_{t+1} = \{f \in \mathcal{F}_t : f(x_t) = y_t\}$

*prob*lem: computational cost of prediction.

Running Halving

$$\mathscr{F} = \{f_{CNN}, f_{MeteoFrance}, \ldots\}$$

Temperature	Air pressure	CNN	Weather Chann	Meteo France	Accu Weather
High	High	Sunny	Rainy	Rainy	Rainy
High	Low	Sunny	Sunny	Rainy	Sunny
Low	High	Rainy	Rainy	Rainy	Rainy
Low	Low	Sunny	Sunny	Rainy	Sunny

Examples given to Halving

iteration	Temperature	Air pressure	$\hat{y}_t$	y_t
1	High	Low		Sunny
2	High	High		Sunny
3	Low	Low		Sunny
4	...	...		...

Halving Analysis

Generic Randomized Algorithm

- Let $\mathcal{F}$ be a family of classifiers, let P_t be a distribution over $\mathcal{F}$.

Algorithm

For $t = 1$ to T

- Receive x_t
- **Draw** $f_t \sim P_t$
- **Predict** $\hat{y}_t = f_t(x_t)$
- Receive the true label y_t, and my prediction costs me $\ell(\hat{y}_t, y_t)$
- **Update** P_{t+1}

Algorithme Randomisé dans le cas réalisable

- Let $\mathcal{F}$ be a family of classifiers
- I choose $P_t = Unif\left(\mathcal{F}_t\right)$
- As in the naïve algorithm, I have $\mathcal{F}_{t+1} = \{f \in \mathcal{F}_t : f(x_t) = y_t\}$

Algorithm

$\mathcal{F}_1 = \mathcal{F}$
For $t = 1$ to T

- Receive x_t
- **Draw** $f_t \sim P_t$
- **Predict** $\hat{y}_t = f_t(x_t)$
- Receive the true label y_t, and my prediction costs me $\ell(\hat{y}_t, y_t)$
- **Update** $\mathcal{F}_{t+1}$ **and** P_{t+1}

Analysing the randomized algorithm

Non realizable case with finite $\mathcal{F}$

Regret notion

- The cumulated loss $\sum_{t=1}^{T} \ell(f_t(x_t), y_t)$ can tend to ∞
- So we look at the *cumulated regret:*

$$Regret_T = \sum_{t=1}^{T} \ell(f_t(x_t), y_t) - \min_{f \in \mathcal{F}} \sum_{t=1}^{T} \ell(f(x_t), y_t)$$

- We compare it to the best classifier who would know the samples *in advance*
- An algorithm is "no regret" if $\frac{1}{T} Regret_T \to 0$ when $T \to \infty$
- Note: For a randomized learner, we look at the expected regret $\mathbb{E}[Regret_T]$

Failure of ERM in the non realizable case

Thm

With the 0/1 loss, neither ERM nor any deterministic algorithm is “no regret”

Randomized Algorithm in the non realizable case

- This algorithm works for any bounded loss $\ell(\cdot,\cdot) \leqslant c$
- Let $\beta \in]0,1[$. Choose $P_t(f) = \frac{1}{\Omega_t} w_{f,t}$ with $\Omega_t = \sum_{f \in \mathcal{F}} w_{f,t}$
  - $w_{f,1} = 1$
  - $w_{f,t+1} = w_{f,t} e^{-\beta \ell(f(x_t),y_t)}$ for some constant $\beta > 0$

Hedge Algorithm

$\mathcal{F}_1 = \mathcal{F}$
For $t = 1$ to T

- Receive x_t
- **Draw** $f_t \sim P_t$
- **Predict** $\hat{y}_t = f_t(x_t)$
- Receive the true label y_t, and my prediction costs me $\ell(\hat{y}_t, y_t)$
- **Update** $\mathcal{F}_{t+1}$ **and** P_{t+1}

Analyzing Hedge

Thm

$\mathbb{E}\left[Regret\right] \leqslant c\sqrt{2T\ln|\mathcal{F}|}$

From Online to Batch: No regret implies PAC

- Up to now, x_t, y_t was arbitrary. What if x_t, y_t is drawn i.i.d. from P ?
- In that case, any no-regret algorithm will give a PAC-learning algorithm !
- **Assumption:** $S = (x_t, y_y)_{t=1}^{T}$ is drawn from P^T. After running a no-regret algorithm, we return $\bar{f}$, a function drawn at random from $f_1 \dots f_T$.
- **Proposition:** If an online learner guarantees that $Regret_T \leqslant UB$ then

$$\mathbb{E}\left[R\left(\bar{f}\right)\right] \leqslant R\left(f_{\mathcal{F}}\right) + \frac{1}{T} UB$$

- **Corollary:** The majority classifier (over the set $f_1 \dots f_T$) is a PAC-learner.

From Online to Batch: No regret implies PAC

Online Learning with infinite $\mathcal{F}$ for a convex loss

ERM

- **Assumptions:** $f \in \mathcal{F}$ is represented by a vector $\theta \in \Theta \subseteq \mathbb{R}^d$ (as for logistic regression. E.g. $f(x) = \theta^\top x$). The set Θ is convex. We define $\ell_t(\theta) = \ell(f_\theta(x_t), y_t)$ convex loss.

ERM Algorithm - also named Follow The Leader (FTL)

For $t = 1$ to T

- Receive x_t
- **Choose** $\theta_t = \arg\min_{\theta \in \Theta} \sum_{k=1}^{t-1} \ell_k(\theta)$
- **Predict** $\hat{y}_t = f_t(x_t)$
- Receive the label y_t, and my prediction costs $\ell(\hat{y}_t, y_t)$

ERM fails as before because it is "unstable"

Regularized ERM

- **Assumptions:** $f \in \mathcal{F}$ is represented by a vector $\theta \in \Theta \subseteq \mathbb{R}^d$. $\ell_t(\theta) = \ell(f_\theta(x_t), y_t)$ is a convex loss.

Algorithme R-ERM - also named Follow The Regularized Leader (FTRL)

$\mathcal{F}_1 = \mathcal{F}$
For $t = 1$ to T

- Receive x_t
- **Choose** $\theta_t = \arg\min_{\theta \in \Theta} \sum_{k=1}^{t-1} \ell_t(\theta) + \lambda C(\theta)$
- **Predict** $\hat{y}_t = f_t(x_t)$
- Receive the label y_t, and my prediction costs $\ell(\hat{y}_t, y_t)$

- Often, $C(\theta) = \|\theta\|_2^2$

R-ERM with linear losses, SGD and Mirror Descent

Lemme "Be The Leader (BTL)"

Lemma

Let $\theta^* = \arg\min_\theta \sum_{t=1}^T \ell_t(\theta)$. *With R-ERM, we get*

$$\sum_{t=1}^{T} \left(\ell_t\left(\theta_t\right) - \ell_t\left(\theta^*\right)\right) \leqslant \lambda \|\theta^*\|_2^2 + \sum_{t=1}^{T} \left(\ell_t\left(\theta_t\right) - \ell_t\left(\theta_{t+1}\right)\right)$$

- *This lemma shows that if* θ_t *is stable and* ℓ_t *is "smooth" in some way, the regret of de R-ERM is low.*

Stability of R-ERM

Lemma

If ℓ_t is convex and ρ-Lipschitz, then $\|\theta_{t+1} - \theta_t\| \leqslant \frac{\rho}{\lambda}$

Regret of R-ERM

Theorem

Let ℓ_t, convex differentiable loss. Let $\theta^* = \arg\min_\theta \sum_{t=1}^{T} \ell_t(\theta)$. Si $\|\theta^*\|_2 \leqslant W_2$, if ℓ_t is ρ-Lipschitz, then with $\lambda = \frac{L\sqrt{T}}{W_2}$ we get:

$$Regret_T = \sum_{t=1}^{T} \left(\ell_t\left(\theta_t\right) - \ell_t\left(\theta^*\right)\right) \leqslant 2W_2\rho\sqrt{T}$$