# 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 $\ell(\hat{y}, y) = \mathbb{1}[\hat{y} \neq y]$

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

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

$\exists f \in \mathcal{F}$ such that $f$ omits zero error

# ERM Algorithm (Empirical Risk Minimization)

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

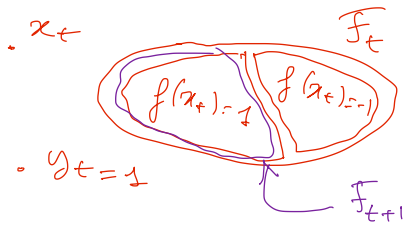
**Algorithm**

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** $\mathcal{F}_{t+1} = \{f \in \mathcal{F}_t : f(x_t) = y_t\}$

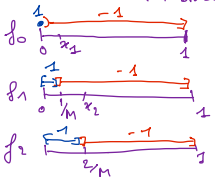
# Failure of ERM

study a worst case scenario

$\mathcal{X} = [0,1]$, $\mathcal{Y} = \{-1, 1\}$

$\mathcal{F} = \{f_0, f_1, \dots, f_M\}$

$f_i(x) = \begin{cases} 1 \text{ if } x \le \frac{i}{M} \\ -1 \text{ otherwise} \end{cases}$




Assumption: among all valid $f \in \mathcal{F}_t$, pick the first one.

Simulate an adversarial environment (the correct classifier was $f_M$)

1). $t = 1$

$x_1 = \frac{1}{M}$, $y_1 = 1$ (hidden to the learner)

- $\mathcal{F}_1 = \mathcal{F}$, so we choose $f = f_0$
- $\hat{y}_1 = f_0(x_1) \neq y_1$

$\Rightarrow$ error !! $\ell(\hat{y}_1, y_1) = 1$

- $\mathcal{F}_2 = \mathcal{F} \setminus \{f_0\} = \{f_1, f_2, f_3 \dots\}$

2). $t = 2$, $x_2 = \frac{2}{M}$, $y_2 = 1$

- we pick $f_1$
- $\hat{y}_2 = f_1(x_2) = -1 \Rightarrow$ error !! $\ell(\hat{y}_2, y_2) = 1$

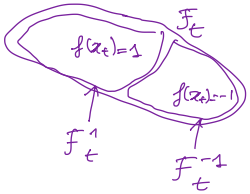
$\vdots$

$M-1$) error

$\Rightarrow$ cumulated loss $= \sum_{t=1}^{M-1} \ell(\hat{y}_t, y_t) = M-1$ (worst)

# 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

$$\mathcal{F}_1 = \{f_{CNN}, f_{MeteoFrance}, \dots\}$$

| 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 | Sunny |
| 2 | High | High | Rainy | Sunny |
| 3 | Low | Low | Sunny | Sunny |
| 4 | … | … | | … |

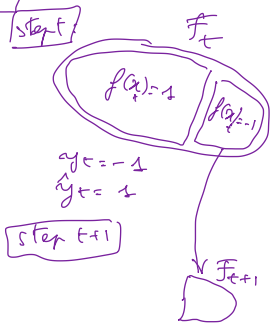
$\mathcal{F}_2 = \{f_{CNN}, f_{WC}, f_{AW}\}$

$\mathcal{F}_3 = \{f_{CNN}\}$

# Halving Analysis

**Thm:** in the two class setting, let $\ell_t = \mathbb{1}[\hat{y}_t \neq y_t]$ (cumulated loss)

$$\sum_{t=1}^{T} \ell_t \leq \ln_2 |\mathcal{F}|$$




**proof sketch:**

- let $\Omega_t = |\mathcal{F}_t|$
- If the prediction $\hat{y}_t$ is incorrect at time t ($\ell_t = 1$) then, $\Omega_{t+1} \leq \frac{\Omega_t}{2}$
- $\Omega_1 = |\mathcal{F}|$
- $\Omega_t \geq 1$ for all t by realizability assumption.

$$1 \leq \Omega_{T+1} \leq \Omega_1 \times 2^{-\sum_{t=1}^{T} \ell_t}$$

$$\Rightarrow \ln_2 \left( \Omega_1 \cdot 2^{-\sum_t \ell_t} \right) \geq 0$$

$$\ln_2 |\mathcal{F}| \geq \sum_{t=1}^{T} \ell_t$$

# 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 $T$

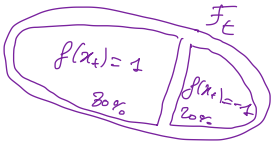
- Receive $x_t$
- **Draw** $f_t \sim P_t$
- **Predict** $\hat{y}_t = f_t(x_t)$
- Receive the true label $y_t$, and my prediction costs me $\ell(\hat{y}_t, y_t)$
- **Update** $P_{t+1}$

# Algorithme Randomisé dans le cas réalisable

- Let $\mathcal{F}$ be a family of classifiers
- I choose $P_t = Unif(\mathcal{F}_t)$
- 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

**Thm:** let $\ell_t = \mathbb{1}[y_t \neq \hat{y}_t]$

$$\mathbb{E}\left[\sum_{t=1}^{T} \ell_t\right] \leq \ln |\mathcal{F}|$$

$\mathbb{E}$ : expectation over $f_t \sim \text{unif}(\mathcal{F}_t)$

proof: $\Omega_t = |\mathcal{F}_t|$

note: $\mathbb{E}\left[\sum_{t=1}^{T} \ell_t\right] = \sum_{t=1}^{T} P(\ell_t = 1)$

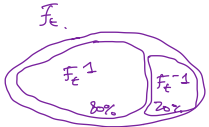
$$P(\ell_t = 0) = P_{f \sim \text{unif}(\mathcal{F}_t)}(f(x_t) = y_t) = \frac{\Omega_{t+1}}{\Omega_t}$$

$$1 \leq \Omega_{t+1} = \Omega_t \times P(\ell_t = 0) = \mathbb{E}[1 - \ell_t] \cdot \Omega_t$$

$$= \Omega_1 \times \prod_{t=1}^{T} \mathbb{E}[1 - \ell_t]$$

log

$$0 \leq \ln |\mathcal{F}| + \sum_{t=1}^{T} \ln \mathbb{E}[1 - \ell_t] \leq \ln |\mathcal{F}| - \sum_{t=1}^{T} \mathbb{E}[\ell_t]$$




- $P(y_t = 1) = 80\%$
  $P(y_t = -1) = 20\%$
- assume $y_t = 1$
  $P(\ell_t = 0) = 80\% = \frac{|\mathcal{F}_t^{-1}|}{|\mathcal{F}_t|} = \frac{|\mathcal{F}_{t+1}|}{|\mathcal{F}_t|}$
- assume $y_t = -1$
  $P(\ell_t = 0) = 20\% = \frac{|\mathcal{F}_t^{-1}|}{|\mathcal{F}_t|} = \frac{|\mathcal{F}_{t+1}|}{|\mathcal{F}_t|}$

# 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 $\infty$
- 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"

proof sketch: $\mathcal{Y} = \{-1, 1\}$ $\quad F = \{f_1, f_{-1}\}$ with $f_1(x) = 1$, $f_{-1}(x) = -1$, $\forall x \in \mathcal{X}$

- Our learning algo is $\mathcal{A}(x_1, y_1, x_2, y_2 \dots x_t) \longrightarrow \hat{y}_t$
- Because $\mathcal{A}$ is deterministic, the environment can simulate $\mathcal{A}(\dots)$
- let $y_t = -\hat{y}_t$ (Malicious environment)

$$\sum_{t=1}^{T} \ell(\hat{y}_t, y_t) = T \qquad \min_{f \in F} \sum_{t=1}^{T} \ell(f(x_t), y_t) \leq T/2$$

| $t$ | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| $y_t$ | 1 | -1 | -1 | 1 | -1 | 1 |
| $f_1$ | 1 | (1) | (1) | 1 | (1) | 1 → 3 errors |
| $f_{-1}$ | (-1) | -1 | -1 | (-1) | 1 | (-1) → 3 errors |

$$\frac{1}{T} \text{Regret}_T \geq \frac{1}{T}\left(T - \frac{T}{2}\right) \geqslant \frac{1}{2}$$

So $\frac{1}{T}\text{Regret} \not\to 0$, thus $\mathcal{A}(\dots)$ is not 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** $P_{t+1}$

Assume $\ell$ is the 0/1 loss
- at the $t$, if $f$ predicts correctly.
$w_{f,t+1} = w_{f,t} \times e^{-\beta \times 0} = w_{f,t}$
- if $f$ makes a mistake,
$w_{f,t+1} = w_{f,t} \times e^{-\beta}$

# Analyzing Hedge

**Thm**

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

$\frac{1}{T}\mathbb{E}[regret_T] \leqslant c\sqrt{\frac{2 \ln |\mathcal{F}|}{T}}$

$\frac{1}{T}\mathbb{E}[regret_T] \leq UB$

# 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 $\mathbb{E}[Regret_T] \leqslant UB$ 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

The online learner generates $f_1 .. f_T$ (one classifier per timestep)

$\bar{f} \sim \text{Unif}(f_1 .. f_T)$

Study the true expected risk of $\bar{f}$. Assumption : $S = (x_1, y_1) .. (x_T, y_T)$ drawn i.i.d from $P$

$$\mathop{\mathbb{E}}_{\bar{f}} R(\bar{f}) = \frac{1}{T} \sum_{t=1}^{T} R(f_t) = \frac{1}{T} \sum_{t=1}^{T} \mathop{\mathbb{E}}_{x_t, y_t \sim P} \ell(f_t(x_t), y_t)$$

$$= \mathop{\mathbb{E}}_{S} \left[ \frac{1}{T} \sum_{t=1}^{T} \ell(f_t(x_t), y_t) \right]$$

$$\leq \mathop{\mathbb{E}}_{S} \left[ \min_{f \in F} \frac{1}{T} \sum_{t=1}^{T} \ell(f(x_t), y_t) + \frac{UB}{T} \right]$$

$$\leq \min_{f \in F} \mathop{\mathbb{E}}_{S} \left[ \frac{1}{T} \sum_{t=1}^{T} \ell(f(x_t), y_t) \right] + \frac{UB}{T} \quad \text{(by Jensen inequality)}$$

$$\underbrace{\min_{f \in F} \mathop{\mathbb{E}}_{S} \left[ \frac{1}{T} \sum_{t=1}^{T} \ell(f(x_t), y_t) \right]}_{R(f_F)}$$

$$\mathbb{E}[\text{Regret}_T] \leq UB$$

$$\Updownarrow$$

$$\mathbb{E}\left[ \sum_{t=1}^{T} \ell(f_t(x_t), y_t) - \min_{f \in F} \sum_{t=1}^{T} \ell(f(x_t), y_t) \right] \leq UB$$

## 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 $\Theta$ 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** $\theta_t = \arg\min_{\theta \in \Theta} \sum_{k=1}^{t-1} \ell_k(\theta) + \lambda C(\theta)$
- **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

For simplicity, assume the loss function $\ell_t(\theta)$ is linear in $\theta$

So we can write $\ell_t(\theta) = g_t^T \theta$ for some $g_t \in \mathbb{R}^d$

Assume $\Theta = \mathbb{R}^d$

R-ERM: $\theta_{t+1} = \arg\min_\theta \left\{ \sum_{k=1}^{t} \ell_k(\theta) + \lambda C(\theta) \right\}$

$L_{t+1}(\theta)$

pick $C(\theta) = \|\theta\|_2^2$

exercise: 1) write the optimality condition for $\theta_{t+1}$

2) ——————— $\theta_t$

3) link them

$$\nabla L_{t+1}(\theta_{t+1}) = \sum_{k=1}^{t} g_k + 2\lambda \theta_{t+1} = 0$$

$$\Rightarrow \theta_{t+1} = -\frac{1}{2\lambda} \sum_{k=1}^{t} g_k$$

$$\nabla L_t(\theta_t) = \sum_{k=1}^{t-1} g_k + 2\lambda \theta_t = 0$$

$$\Rightarrow \theta_t = -\frac{1}{2\lambda} \sum_{k=1}^{t-1} g_k \quad \text{and} \quad \sum_{k=1}^{t-1} g_k = -2\lambda \theta_t$$

$$\theta_{t+1} = -\frac{1}{2\lambda} g_t - \frac{1}{2\lambda} \sum_{k=1}^{t-1} g_k$$

$$\boxed{\theta_{t+1} = \theta_t - \frac{1}{2\lambda} g_t} \Leftrightarrow \underbrace{\boxed{\theta_{t+1} = \theta_t - \frac{1}{2\lambda} \nabla_\theta \ell_t(\theta_t)}}_{\text{SGD}}$$

1) If $\nabla_\theta \ell_t$ is small, then, the algo is stable: $\theta_{t+1}$ is close to $\theta_t$

# Lemme "Be The Leader (BTL)"

**Lemma**

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

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

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

# Stability of R-ERM

**Lemma**

*If $\ell_t$ is convex and $\rho$-Lipschitz, then* $\|\theta_{t+1} - \theta_t\| \leqslant \frac{\rho}{\lambda}$ with $C(\theta) = \|\theta\|_2^2$

# Regret of R-ERM

## Theorem

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

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

$\frac{1}{T} Regret \leq \frac{2 W_2 \rho}{\sqrt{T}}$