Fondamentaux de l'Apprentissage Automatique

Lecturer: Yann Chevaleyre
Scribe: Amigo Joseph

Lecture n°6 #
02/11/2023

1 Online Learning in the realizable case

1.1 Online learning protocol

The general steps of an online learning protocol look like the following :

Let $S = (x_1, y_1) \ldots (x_N, y_N) \sim \mathcal{P}^N$ be dataset.
For $t = 1$ to T do :

1. The environment chooses x_t, y_t, and reveals x_t to the learner
2. The learner predicts $\hat{y}_t$
3. The environment reveals y_t
4. The learner endures the cost $\ell(\hat{y}_t, y_t)$

In the following let's define a **finite** set of functions $\mathcal{F}$. We will assume that we are in the **realizable case**, meaning that there exists in $\mathcal{F}$ a classifier that commits zero error on the dataset S.

1.2 Empirical Risk Minimization (ERM) online algorithm

The ERM algorithm is the following online algorithm :
Define $\mathcal{F}_1 = \mathcal{F}$ and do :
For $t = 1$ to T, do :

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

1.3 Failure of ERM

With an ERM online algorithm, it can be that we make almost only mistakes (in the realizable case we have a perfectly correct classifier, so with the algorithm above, at least one prediction will be correct). Let's study this failure case in detail.

Let's define $\mathcal{X} = [0, 1]$, $\mathcal{Y} = \{-1, 1\}$ and $\mathcal{F} = \{f_0, f_1, \ldots, f_M\}$, where $\mathrm{f_i(x)} = \begin{cases} 1 & \text{if } x \leq \frac{i}{M} \\ -1 & \text{otherwise} \end{cases}$.

Let's simulate the online ERM algorithm on the dataset $S = ((\frac{1}{M}, 1), \ldots, (\frac{M-1}{M}, 1))$ with the assumption that among all valid $f \in \mathcal{F}_t$, we pick the first one. **Note that f_M is the perfect predictor on S, so we're indeed in the realizable case**. Also, note that it is enumerated last.

— $t = 1$

— $x_1 = \frac{1}{M}, y_1 = 1$
— $\mathcal{F}_1 = \mathcal{F}$, so we choose $f = f_0$ (remember the assumption "among all valid $f \in \mathcal{F}_t$, we pick the first one")
— $\hat{y}_1 = f_0(x_1) = -1 \neq y_1$ $(x_1 = \frac{1}{M} > \frac{0}{M}) \implies \ell(\hat{y}_1, y_1) = 1$
— $\mathcal{F}_2 = \mathcal{F} \setminus \{f_0\}$ (in our settings there is only f_0 that makes a mistake for $(x_1, y_1) = (\frac{1}{M}, 1)$)
...
— $t = M - 1$
— $x_{M-1} = \frac{M-1}{M}, y_{M-1} = 1$
— we pick f_{M-1} (remember the assumption "among all valid $f \in \mathcal{F}_t$, we pick the first one")
— $\hat{y}_{M-1} = f_{M-2}(x_2) = -1 \neq y_{M-1} \implies \ell(\hat{y}_{M-1}, y_{M-1}) = 1$

Finally we get the cumulated loss $= \sum_{t=1}^{M-1} \ell(\hat{y}_t, y_t) = M - 1$.

1.4 Halving algorithm

Let's define $\mathcal{F}_1 = \mathcal{F}$. The halving algorithm is the following online algorithm :
For $t = 1$ to T, do :

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

1.5 Analysis of halving

Theorem 1. *With the halving algorithm, in the two-class setting, let* $l_t = \mathbb{1}_{[\hat{y}_t \neq y_t]}$, *then* $\sum_{t=1}^{T} l_t \leqslant \ln_2 |\mathcal{F}|$.

Proof of theorem 1. Let $\Omega_t = |\mathcal{F}_t|$,
— If the prediction $\hat{y}_t$ is incorrect at time $(l_t = 1)$ then, $\Omega_{t+1} \leqslant \frac{\Omega_t}{2}$
— $\Omega_1 = |\mathcal{F}|$
— $\Omega_t \geqslant 1$ for all t by realizalility assumption.

Therefore :

$$
\begin{aligned}
1 \leqslant \Omega_{T+1} &\leqslant \Omega_1 \times 2^{-\sum_{t=1}^{T} l_t} \\
&\Rightarrow \ln_2\left(\Omega_1 \times 2^{-\sum_{t=1}^{T} l_t}\right) \geqslant 0 \\
&\Rightarrow \ln_2 |\mathcal{F}| \geqslant \sum_{t=1}^{T} l_t
\end{aligned}
$$

□

1.6 Generic randomized algorithm

Let $\mathcal{F}$ be a family of classifiers, and let P_t be a distribution over $\mathcal{F}$. A generic randomized algorithm looks like the following :

For $t = 1$ to T, do :

1. Receive x_t
2. Draw $f_t \sim P_t$
3. Predict $\hat{y}_t = f_t\left(x_t\right)$
4. Receive the true label y_t, and my prediction costs me $\ell\left(\hat{y}_t, y_t\right)$
5. Update P_{t+1}

1.7 Uniform case

Let's study a particular randomized algorithm. Let $\mathcal{F}$ be a family of classifiers. Choose $P_t = \text{Unif}\left(\mathcal{F}_t\right)$.

Let's define $\mathcal{F}_1 = \mathcal{F}$.

For $t = 1$ to T, do :

1. Receive x_t
2. Draw $f_t \sim P_t$
3. Predict $\hat{y}_t = f_t\left(x_t\right)$
4. Receive the true label y_t, and my prediction costs me $\ell\left(\hat{y}_t, y_t\right)$
5. Update $\mathcal{F}_{t+1} = \{f \in \mathcal{F}_t : f\left(x_t\right) = y_t\}$ and P_{t+1}

1.8 Analysis of the uniform randomized algorithm

Theorem 2. *With the uniform randomized algorithm, in the two-class setting, let* $l_t = \mathbb{1}_{[\hat{y}_t \neq y_t]}$, *then*

$$\mathbb{E}_{f_1,\dots,f_t \sim \mathcal{U}(\mathcal{F}_1),\dots,\mathcal{U}(\mathcal{F}_t)}\left[\sum_{k=1}^{T} l_t\right] \leq \ln|\mathcal{F}|$$

Proof of theorem 2. Let $\Omega_t = |\mathcal{F}_t|$, we have :

$$\begin{aligned}
\mathbb{E}_{f_1,\dots,f_t \sim \mathcal{U}(\mathcal{F}_1),\dots,\mathcal{U}(\mathcal{F}_t)}\left[\sum_{k=1}^{T} l_t\right] &= \sum_{k=1}^{T} \mathbb{E}_{f_1,\dots,f_t \sim \mathcal{U}(\mathcal{F}_1),\dots,\mathcal{U}(\mathcal{F}_t)}[l_t] \\
&= \sum_{k=1}^{T} \mathbb{P}(l_t = 1)
\end{aligned}$$

And :

$$
\begin{aligned}
\mathbb{P}(l_t = 0) &= \mathbb{P}(\mathbb{1}_{[f_t(x_t) \neq y_t]} = 0) \\
&= \mathbb{P}(f_t(x_t) = y_t) \\
&= \mathbb{P}(f_t \in \{f \in \mathcal{F}_t : f(x_t) = y_t\}) \\
&= \mathbb{P}(f_t \in \mathcal{F}_{t+1}) \\
&= \frac{|\mathcal{F}_{t+1}|}{|\mathcal{F}_t|} \\
&= \frac{\Omega_{t+1}}{\Omega_t} \\
\Omega_{t+1} &= \Omega_t \times \mathbb{P}(l_t = 0) \\
&= \Omega_{t-1} \times \mathbb{P}(l_{t-1} = 0) \times \mathbb{P}(l_t = 0) \\
&\dots \\
&= \Omega_1 \prod_{k=1}^{t} \mathbb{P}(l_k = 0)
\end{aligned}
$$

Furthermore :

$$
\begin{aligned}
1 \leq \Omega_{t+1} &\leq \Omega_1 \prod_{k=1}^{t} \mathbb{P}(l_k = 0) \\
0 &\leq \ln(\Omega_1) + \sum_{k=1}^{t} \ln(\mathbb{P}(l_k = 0)) \\
0 &\leq \ln(\Omega_1) + \sum_{k=1}^{t} \ln(1 - \mathbb{P}(l_k = 1))
\end{aligned}
$$

And because $\forall x \in [0, 1[, \ln(1 - x) \leq -x$:

$$
\begin{aligned}
0 &\leq \ln(\Omega_1) - \sum_{k=1}^{t} \mathbb{P}(l_k = 1) \\
\mathbb{E}\Big[\sum_{k=1}^{t} l_k\Big] &\leq |\mathcal{F}| \qquad (\forall t)
\end{aligned}
$$

□

2 Online Learning in the non-realizable case

2.1 Regret

— The cumulated loss $\sum_{t=1}^{T} \ell\left(f_t\left(x_t\right), y_t\right)$ can tend to ∞
— So we look at the cumulated regret : $\text{Regret}_T = \sum_{t=1}^{T} \ell\left(f_t\left(x_t\right), y_t\right) - \min_{f \in \mathcal{F}} \sum_{t=1}^{T} \ell\left(f\left(x_t\right), y_t\right)$
— We compare it to the best classifier who would know the samples in advance
— An algorithm is "no regret" if $\frac{1}{T}\,\text{Regret}_T \to 0$ when $T \to \infty$
— Note : For a randomized learner, we look at the expected regret $\mathbb{E}\left[\text{Regret}_T\right]$

2.2 Failure of ERM in the non-realizable case

Theorem 3. *With the 0/1 loss, neither ERM nor any deterministic algorithm is "no regret".*

Proof of theorem 3 : Given :

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

Our learning algorithm is $\mathcal{A}(x_1, y_1, x_2, y_2, \ldots, x_t) \to \hat{y}_t$

Because $\mathcal{A}$ is deterministic, the environment can simulate $\mathcal{A}(\ldots)$. Let's take :

$$y_t = -\hat{y}_t \quad \text{(Malicious environment)}$$

Then :

$$\sum_{t=1}^{T} l\{y_t \neq \hat{y}_t\} = T$$

Let's define $(i_1, \ldots, i_T) \in \{-1, 1\}^T$ such that $f_{i_1}(x_1) = f_{\hat{y}_1}(x_1) = \hat{y}_1, \ldots, f_{i_T}(x_T) = f_{\hat{y}_T}(x_T) = \hat{y}_T$. Suppose :

$$\min_{f \in \mathcal{F}} \sum_{t=1}^{T} l\{y_t \neq f(x_t)\} > T/2$$

And without loss of generality suppose that f_1 is the function that meets this minimum. But in our settings every time f_1 makes a mistake, f_{-1} is correct. This means that f_{-1} made **strictly** less that $T - \frac{T}{2}$ mistakes (maximum number of mistakes for a classifier in our settings is T, and if $\sum_{t=1}^{T} l\{y_t \neq f_1(x_t)\} > T/2$ then $\sum_{t=1}^{T} l\{y_t = f_{-1}(x_t)\} > T/2$). But that's fewer errors than f_1, which is supposed to make the fewest errors. **Absurd**.

Hence :

$$\min_{f \in \mathcal{F}} \sum_{t=1}^{T} l\{y_t \neq f(x_t)\} \leq T/2$$

Therefore, the regret :

$$\frac{1}{T}\text{Regret} \geq \frac{1}{T}(T - T/2) \geq \frac{1}{2}$$

Since the regret is a constant strictly greater than 0, $\mathcal{A}(\ldots)$ is not no-regret.

□

2.3 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 $\beta > 0$

With this let's define the following algorithm which is called the Hedge algorithm :

Define $\mathcal{F}_1 = \mathcal{F}$, for $t = 1$ to T, do :

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

We have the following theorem for hedge :

Theorem 4. $\mathbb{E}[\ \textit{Regret}\ ] \leqslant c\sqrt{2T \ln |\mathcal{F}|}$

2.4 No regret implies PAC

Up to now, x_t and y_t were drawn from an arbitrarily distribution. Let's now analyse the case where x_t and y_t are drawn from a distribution p.

In this case, any no-regret algorithm is PAC (probably approximately correct).

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 $E[Regret] \leq UB$ then :

$$E[R(\bar{f})] \leq R(f_F) + \frac{1}{T}UB$$

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

If the online learner generates $f_1 \dots f_T$ (one classifier per timestep),

Study the true expected risk of f_g. Assumption : $(z_1, y_1) \dots (z_T, y_T)$ drawn i.i.d. from $\mathcal{R}$.

Given :

$$\begin{aligned}
\mathcal{R}(f_g) &= \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{z,y\sim p} l(f_g(z_t), y_t) \\
&= \mathbb{E}S\left[\frac{1}{T}\sum t = 1^T l(f_g(z_t), y_t)\right] \\
&\leq \mathbb{E}S\left[\min f \in \mathcal{F}\frac{1}{T}\sum_{t=1}^{T} l(f(z_t), y_t)\right] + \frac{UB}{T} \\
&\leq \min_{f\in\mathcal{F}} \mathbb{E}S\left[\frac{1}{T}\sum t = 1^T l(f(z_t), y_t)\right] + \frac{UB}{T} \quad \text{(by Jensen's inequality)} \\
&\leq UB
\end{aligned}$$

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

3.1 ERM Case

- 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.

We can define an ERM Algorithm - also named Follow The Leader (FTL) :
For $t = 1$ to T, do :
— 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".
We can define another algorithm, named Follow The Regularized Leader (FTRL) :
— 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.
Define $\mathcal{F}_1 = \mathcal{F}$, then for $t = 1$ to T, do :
— Receive x_t
— Choose $\theta_t = \arg\min_{\theta\in\Theta} \sum_{k=1}^{t-1} l_k(\theta) + \lambda C(\theta)$
— Predict $\hat{y}_t = f_t(x_t)$
— Receive the label y_t, and my prediction costs $\ell(\hat{y}_t, y_t)$
Often, $C(\theta) = \|\theta\|_2^2$.

3.2 R-ERM with linear losses, SGD and Mirror Descent

For simplicity, assume the loss function $l_t(\theta)$ is linear in θ, so we can write $l_t(\theta) = g_t^\top \theta$ for some $g_t \in \mathbb{R}^d$, assuming $\Theta = \mathbb{R}^d$.
R-ERM : $\theta_{t+1} = \arg\min \theta \in \Theta \left\{ \left[\sum k = 1^t l_k(\theta) \right] + \lambda \mathrm{C}(\theta) \right\}$
Pick $C(\theta) = \|\theta\|_2^2$

Exercise :
1) write the optimality condition for θ_{t+1}
2) write the optimality condition for θ_t
3) link them

Let's define $\nabla L_{t+1}(\theta) = \sum_{k=1}^{t} l_k(\theta) + 2\lambda \mathrm{C}(\theta)$

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

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

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

$$\theta_{t+1} = \theta_t - \frac{1}{2\lambda} g_t$$

Alternatively, if using a gradient term :

$$\theta_{t+1} = \theta_t - \frac{1}{2\lambda}\nabla_\theta l_t(\theta_t) \,\mathrm{SGD}$$

If $\nabla_\theta \ell_t$ is small, then the algorithm is stable : θ_{t+1} is close to θ_t.

3.3 Lemme "Be The Leader (BTL)"

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

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

- 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.*

3.4 Stability of R-ERM

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

3.5 Regret of R-ERM

Theorem 5. *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 :*

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