

# STA212

Anthony Kalaydjian - Mathieu Occhipinti

2023-04-29

```
rm(list=ls())
setwd(getwd())
library(ggplot2)
set.seed(150)
```

## Exercice 1 : Modélisation probabiliste

(a)

D'après le cours, on a :

$$\begin{aligned} R(g^*) &= E_{X,Y} [1 \{g^*(X) \neq Y\}] \\ &= E_X [E_{Y|X} [1 \{g^*(X) \neq Y\}]] \\ &= E_X \left[ \frac{1}{2} - \left| \eta(X) - \frac{1}{2} \right| \right] \end{aligned}$$

$$\begin{aligned} \text{Donc } R(\hat{g}_n) - R(g^*) &= E_X \left[ \frac{1}{2} - \left| \hat{\eta}(X) - \frac{1}{2} \right| \right] - E_X \left[ \frac{1}{2} - \left| \eta(X) - \frac{1}{2} \right| \right] \\ &= E_X \left[ \left| \eta(X) - \frac{1}{2} \right| - \left| \hat{\eta}(X) - \frac{1}{2} \right| \right] \end{aligned}$$

Où

$$\begin{cases} |a| - |b| \leq ||a| - |b|| \leq |a - b| \leq 2|a - b| \\ (a, b) \in \mathbb{R}^2 \end{cases}$$

Donc

$$\boxed{R(\hat{g}_n) - R(g^*) \leq 2E_X [|\eta(X) - \hat{\eta}(X)|]}$$

(b)

Le résultat précédent nous indique que pour toute règle de classification empirique,  $\hat{g}_n$  issue de l'estimateur  $\hat{\eta}$  de  $\eta$ , son risque associé est borné par le risque minimal issu de la règle de Bayes  $R(g^*)$  auquel on ajoute un terme d'erreur d'estimation de  $\eta$ .

## Exercice 2 : Classification multi-classes

(a)

Soit  $\mathcal{Y} = \{1, \dots, K\}$ .

On cherche à classer les individus de  $\mathcal{X}$  selon leur classe sur  $\mathcal{Y}$ .

Soit  $g^*$  l'estimateur de risque minimal associé à cette classification.

$$\begin{aligned} R(g^*) &:= \mathbb{E} [1_{\{g^*(X) \neq Y\}}] \\ &= \mathbb{E}_X \mathbb{E}_{Y|X=x} [1_{\{g^*(x) \neq Y\}}] \\ &= 1 - \mathbb{E}_X \mathbb{E}_{Y|X=x} [1_{\{g^*(x) = Y\}}] \\ &= 1 - \mathbb{E}_X \mathbb{P} [g^*(x) = Y] \\ R(g^*) &= 1 - \mathbb{E}_X \left[ \sum_{i=1}^K \eta_i(x) 1_{\{g^*(x)=i\}} \right] \end{aligned}$$

Ce risque est minimal lorsque le terme dans l'espérance est maximal.

Donc :

$$\boxed{g^*(x) = \operatorname{argmax}_{i \in \mathcal{Y}} \eta_i(x)}$$

(b)

D'après le calcul précédent, et en utilisant les définitions de  $g^*$  et de  $g$ :

$$\begin{aligned} R(g) - R(g^*) &= 1 - \mathbb{E}_X \left[ \sum_{i=1}^K \eta_i(x) 1_{\{g(x)=i\}} \right] - 1 + \mathbb{E}_X \left[ \sum_{i=1}^K \eta_i(x) 1_{\{g^*(x)=i\}} \right] \\ &= \mathbb{E}_X \left[ \sum_{i=1}^K \eta_i(x) 1_{\{g^*(x)=i\}} \right] - \mathbb{E}_X \left[ \sum_{i=1}^K \eta_i(x) 1_{\{g(x)=i\}} \right] \\ &= \mathbb{E}_X \left[ \max_{i \in \mathcal{Y}} \eta_i(x) - \mathbb{E}_X \eta_{g(x)} \right] \\ &= \mathbb{E}_X \left[ \max_{i \in \mathcal{Y}} \eta_i(x) - \eta_{g(x)} \right] \end{aligned}$$

Donc :

$$\boxed{R(g) - R(g^*) = \mathbb{E}_X \left[ \max_{i \in \mathcal{Y}} \eta_i(x) - \eta_{g(x)} \right]}$$

(c)

On considère maintenant la règle de classification  $\hat{g}_n$  approchée de  $g$  et issue de  $\hat{\eta}_i$  un estimateur de  $\eta_i$  obtenu à partir des données.

La règle de classification est donc la suivante :

$$\hat{g}_n(x) = \operatorname{argmax}_{j \in \mathcal{Y}} \hat{\eta}_j(x)$$

$$\begin{aligned} R(\hat{g}_n) - R(g^*) &= \mathbb{E}_X \left[ \max_{i \in \mathcal{Y}} \eta_i(x) - \eta_{\hat{g}_n(x)} \right] \\ &= \mathbb{E}_X \left[ \max_{i \in \mathcal{Y}} \eta_i(x) - \eta_{g_n(x)} \right] \end{aligned}$$

### Exercice 3 : Implémentation d'un perceptron (origine des SVM)

(a)

Supposons que l'on soit à l'itération  $t$  de l'algorithme, et tel que  $m(\theta^t) \neq \emptyset$ .

Soit alors  $i \in m(\theta^t)$

D'une part:

$$\begin{aligned} \langle \theta^{t+1}, \theta^* \rangle &= \langle \theta^t + y_i x_i, \theta^* \rangle \\ &= \langle \theta^t, \theta^* \rangle + y_i \langle x_i, \theta^* \rangle \\ &\geq \langle \theta^t, \theta^* \rangle + \|\theta^*\|_2 \delta \end{aligned}$$

Par Cauchy-Schwarz:

$$\begin{aligned} \|\theta^{t+1}\|_2 \|\theta^*\|_2 &\geq \langle \theta^{t+1}, \theta^* \rangle \geq \langle \theta^t, \theta^* \rangle + \|\theta^*\|_2 \delta \\ &\geq \langle \theta^{t-1}, \theta^* \rangle + 2\|\theta^*\|_2 \delta \\ &\geq \dots \\ &\geq \langle \theta^0, \theta^* \rangle + t\|\theta^*\|_2 \delta \\ \|\theta^{t+1}\|_2 \|\theta^*\|_2 &\geq t\|\theta^*\|_2 \delta \end{aligned}$$

Donc:

$$\boxed{\|\theta^{t+1}\|_2 \geq t\delta} \tag{1}$$

D'autre part:

$$\begin{aligned} \|\theta^{t+1}\|^2 &= \|\theta^t + y_i x_i\|^2 \\ &= \|\theta^t\|^2 + 2y_i \langle \theta^t, x_i \rangle + \|y_i x_i\|^2 \\ &\leq \|\theta^t\|^2 + \|x_i\|^2 \\ &\leq \|\theta^t\|^2 + R^2 \\ &\leq \|\theta^{t-1}\|^2 + 2R^2 \\ &\leq \dots \\ &\leq \|\theta^0\|^2 + tR^2 \\ \|\theta^{t+1}\|^2 &\leq tR^2 \end{aligned}$$

Donc:

$$\boxed{\|\theta^{t+1}\|^2 \leq tR^2} \quad (2)$$

Finalement, d'après 1 et 2, on a :

$$t^2\delta^2 \leq \|\theta^{t+1}\|^2 \leq tR^2$$

Donc  $\boxed{t \leq \frac{R^2}{\delta^2}}$

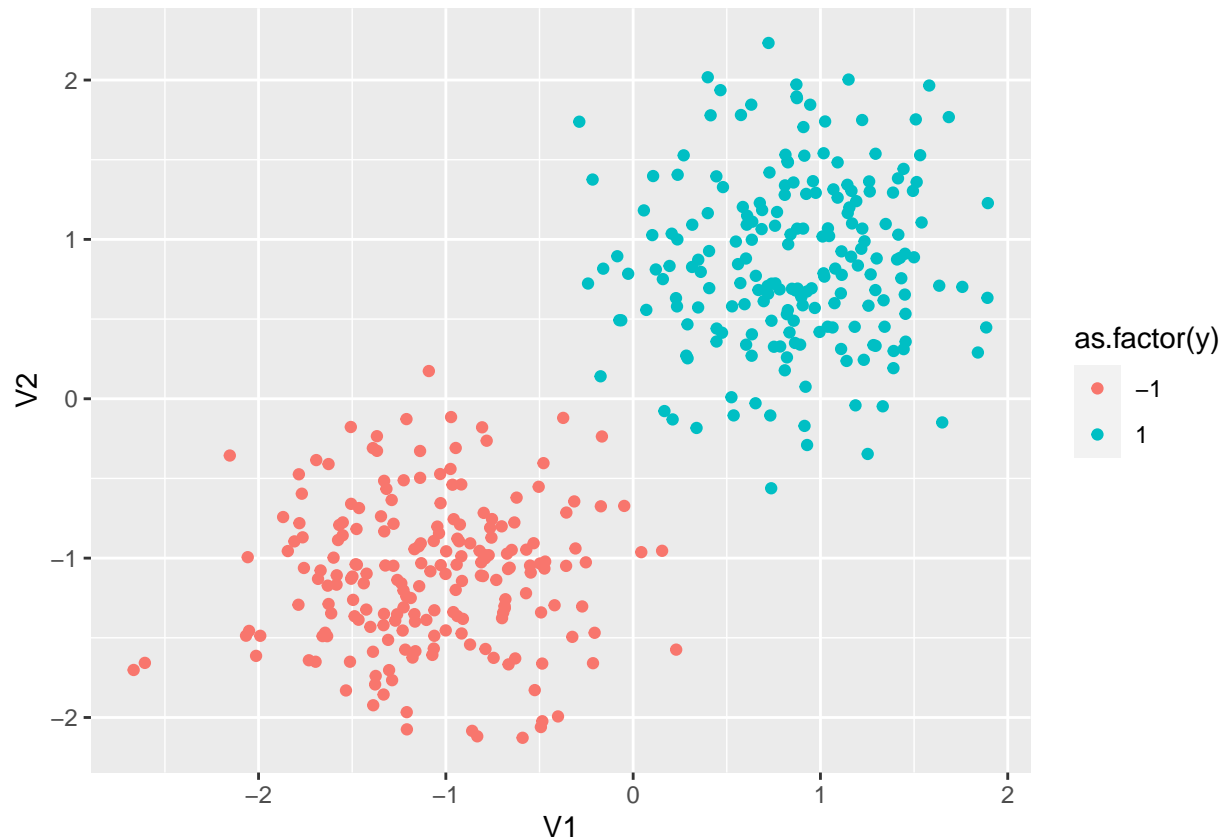
Ainsi, on a montré que si  $m(\theta^t) \neq \emptyset$ , alors  $t \leq \frac{R^2}{\delta^2}$

Donc, au delà de  $T = \frac{R^2}{\delta^2}$  itérations,  $m(\theta^T)$  sera vide et l'algorithme aura donc convergé.

(b)

Importation des données :

```
load(file="X_y.rda")
df <- as.data.frame(cbind(X, y))
names(df) <- c("V1", "V2", "V3", "y")
plt1 <- ggplot(data=df) + aes(x=V1, y=V2, z=y, color=as.factor(y)) + geom_point()
plt1
```



La variable V3 est une variable d'intercept.

## Algorithme perceptron

```
perceptron <- function(X, y){
  theta <- c(0, 0, 0)
  n <- nrow(X)
  m <- seq(1, n)
  counter <- 0

  while (length(m) != 0){
    #sample a random item from m
    index = sample(m, 1)

    #update theta
    theta <- theta + y[index]*X[index,]

    #calculate the new m
    temp <- sapply(X=seq(1, n), FUN=function(k) theta%*%X[k,])
    criterion <- y*temp
    m <- which(criterion<0)
    counter <- counter + 1
  }
  return(list(theta=theta, count=counter))
}

res <- perceptron(X, y)
theta.star <- res$theta
count.star <- res$count

theta.star
```

```
## [1] 3.438710 4.537851 1.000000
```

```
count.star
```

```
## [1] 5
```

L'algorithme converge en 5 itérations, et nous trouve la valeur de  $\theta^* = (3.438710, 4.537851, 1.000000)^T$ .

## plot

Le plot montre que l'on a bien séparé les deux classes à l'aide de notre hyperplan.

```
plt1 + geom_abline(intercept=-theta.star[3]/theta.star[2], slope=-theta.star[1]/theta.star[2])
```

