

# CH3 Scoring\_appli

Youming

2023-09-21

## CH3 Evaluation d'un score

### 1. Context et objectif

$x_1, x_2, \dots, x_n \in X$  sont des observations réparties en Group  $y_I \in \{1, \dots, G\}$  indique le groupe auquel appartient  $x_i$

### 2. Erreur de classement

- Le score  $s$  est de bonne qualité si la classe  $\hat{y}(x)$  à laquelle il affecte  $x$  correspond, en moyenne, à la vraie classe  $y$  de  $x$ . , autrement dire :  $S$  est un bon score si on moyenne le classe estimée de  $x$  :  $\hat{y}(x)$  coïncide avec  $y$  若  $\hat{y}$  与实际  $y$  符合, 则  $s$  质量好
- Or, si l'on ne sait rien du modèle qui a généré  $(x, y)$ , l'erreur de classement :  $E(x, y)(1_{\{\hat{y}(x)=y\}})$  ne peut pas être déterminée. Il existe cependant plusieurs façons de l'estimer et d'évaluer ainsi la qualité de  $s$ . 不过, 若不知道生成  $(x, y)$  的模型, erreur de classement **autrement dire : on souhaite que l'erreur de classement soit minimale.** 就不能被定论, 好在有以下方法可以评估  $s$   $\varepsilon = E[1_{\{\hat{y}(x) \neq y\}}]$

Deux problèmes :

- i) On ne connaît pas la loi du couple  $(x, y)$  ; on ne peut pas calculer  $\varepsilon$
- ii) quand on a la vraie loi du couple  $(x, y)$ ,  $\varepsilon$  dépend du classifieur à travers  $\hat{y}(x)$  et on ne peut pas la rendre arbitrairement proche de 0

Pour(ii) , il n'y a rien à faire, pour (i) on dispose plusieurs de méthodes d'estimation

### Erreur théorique

En analyse discriminante probabiliste, on modélise les données du groupe  $g$  par  $f_g(\bullet, \theta_g)$  et le poids du groupe est  $\pi_g$ . L'erreur théorique est alors :

$$\hat{\varepsilon}_{th} = \sum_{g=1}^G \pi_g \int_{\{x \in X; \hat{y}(x) \neq g\}} f_g(x, \hat{\theta}_g) dx$$

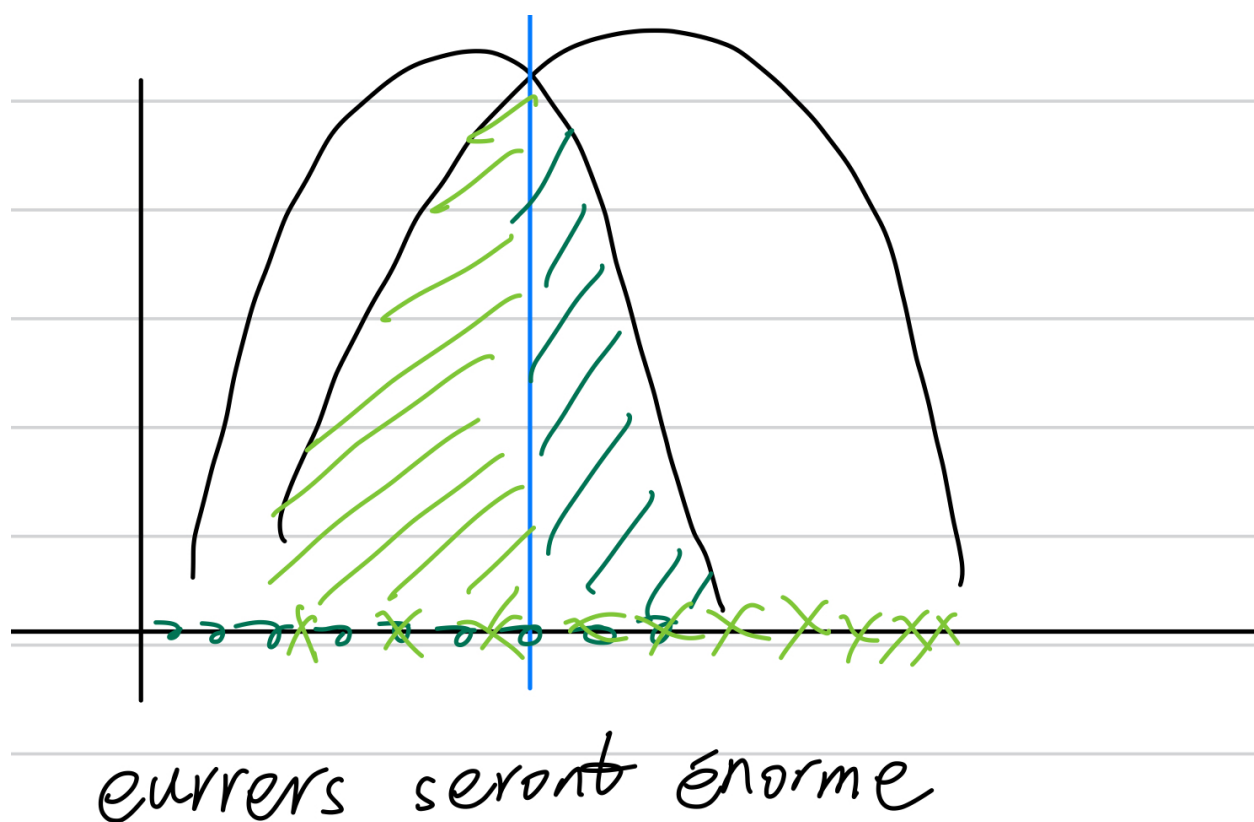


图 1: Erreur sera enorme

Justification : Voir CH2

L'estimateur  $\hat{\epsilon}$  possède deux inconvénients :

概率建模：在概率判别分析中，我们试图使用条件概率密度函数  $f_g(\bullet, \theta_g)$ ，对属于不同组（通常表示为 G）的数据进行建模。每个组都有一个权重  $\pi_g$ ，代表属于该组的数据比例。

(a) il hérite du biais du modèle postulé

(b) il est fréquent que les fonctions  $f_g$  ne permettent pas de calculer  $\hat{\epsilon}_{th}$  explicitement <sup>1</sup>

理论误差：理论误差  $\hat{\epsilon}_{th}$  是理论分类误差的度量。它被定义为每个组 g 的分类误差之和。总和的每个项都对应于密度函数  $f_g(x, \hat{\theta}_g)$  在数据集上的积分，在数据集上，模型错误地分配了相关的组 g，即  $\hat{y}(\infty) \neq g$ 。  
 $\hat{\epsilon}$  估计器的缺点：

(a) 假设模型的偏差： $\hat{\epsilon}_{th}$  估计器有一个主要缺点。它继承了假定模型的偏差，这意味着如果用于估计  $f_g$  密度的基本概率模型有偏差或不正确，那么  $\hat{\epsilon}_{th}$  也会有偏差，无法准确评估真实的分类误差。

(b) 计算困难：密度函数  $f_g$  通常不允许明确计算  $\hat{\epsilon}_{th}$ 。这意味着，在许多情况下，无法通过分析确定理论误差，从而限制了准确估计分类误差的能力。

总之，摘录解释了如何利用各组的条件概率密度来模拟理论误差，但指出由于假设模型的原因，这种估计方法可能存在偏差，而且由于密度函数的复杂性，在实践中可能难以计算。这是概率判别分析中常见的难题。

En probabilitiste, les erreurs de group G sont modélisés par la fonction de probabilité  $f_g(\cdot; \cdot_g)$  est on note  $T_i g$  le poids de la classe g

---

### Exercice 1 TD3 : <sup>2</sup>

On envisager deux modèle : le classifieur Gaussien Hétéroscédastique ( $\hat{\sigma}_1^2 \neq \hat{\sigma}_2^2$ )

Affectation de M. Li (découvert = 1.9)

Poid de classe:  $\hat{\pi}_1 = \hat{\pi}_2 = 0.5$

Centre de classe:  $\hat{\mu}_1 = 1,5$  et  $\hat{\mu}_2 = 2,5$

Variance:  $\hat{\sigma}_1^2 = 3$  et  $\hat{\sigma}_2^2 = 3,2$

x = décoissant

y = classe

两组分布表达式：

---

<sup>1</sup>méthode Monte Carlo, 随机抽样统计方法

<sup>2</sup>Rappelle : Loi normale  $f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$

$$(x|y=1) \sim f(x; \hat{\mu}_1, \hat{\sigma}_1^2) = \frac{1}{\sqrt{2\pi}} \frac{1}{\hat{\sigma}_1} \exp \left( -\frac{1}{2} \left( \frac{x - \hat{\mu}_1}{\hat{\sigma}_1} \right)^2 \right)$$

$$(x|y=2) \sim f(x; \hat{\mu}_2, \hat{\sigma}_2^2) = \frac{1}{\sqrt{2\pi}} \frac{1}{\hat{\sigma}_2} \exp \left( -\frac{1}{2} \left( \frac{x - \hat{\mu}_2}{\hat{\sigma}_2} \right)^2 \right)$$

概率表达式:

$$P(Li \in Class1 | x(Li) = 1, 9) = \frac{\hat{\Pi}_1 f_1(1, 9; \hat{\mu}_1, \hat{\sigma}_1^2)}{\sum_{j=1}^2 \hat{\Pi}_j f_j(1, 9; \hat{\mu}_j, \hat{\sigma}_j^2)} = \frac{\text{组 1}}{\text{两组}}$$

Sur R on a

```
library(stats)
pi1 = 0.5; pi2 = 0.5
mu1 = 1.5; mu2 = 2.5
sicar1 = 3; sicar2 = 3.2
pf1 = pi1 * dnorm(1.9, mean = mu1, sd = sqrt(sicar1))
pf2 = pi2 * dnorm(1.9, mean = mu2, sd = sqrt(sicar2))
t1 = pf1/(pf1+pf2)
print(t1)
```

```
## [1] 0.5154582
```

M.Li a 52% de class apportement à la classe 1, on affect à la class 1

**Estimation de l'erreur de classement :**

On détermine la fontiere de classement  $\gamma$  en réselment <12>

$$t1(r) = t2(r)$$

$$\hat{\pi}_1 \times f_1(r; \hat{\mu}_1, \hat{\sigma}_1^2) = \hat{\pi}_2 \times f_2(r; \hat{\mu}_2, \hat{\sigma}_2^2)$$

$$\hat{\pi}_1 \frac{1}{\sqrt{2\pi}} \frac{1}{\hat{\sigma}_1} \exp \left( -\frac{1}{2} \left( \frac{\lambda - \hat{\mu}_1}{\hat{\sigma}_1^2} \right) \right) = \hat{\pi}_2 \frac{1}{\sqrt{2\pi}} \frac{1}{\hat{\sigma}_2} \exp \left( -\frac{1}{2} \left( \frac{\lambda - \hat{\mu}_2}{\hat{\sigma}_2^2} \right) \right) \quad (*)$$

On résoudre à la mains et on résoudre avec R :

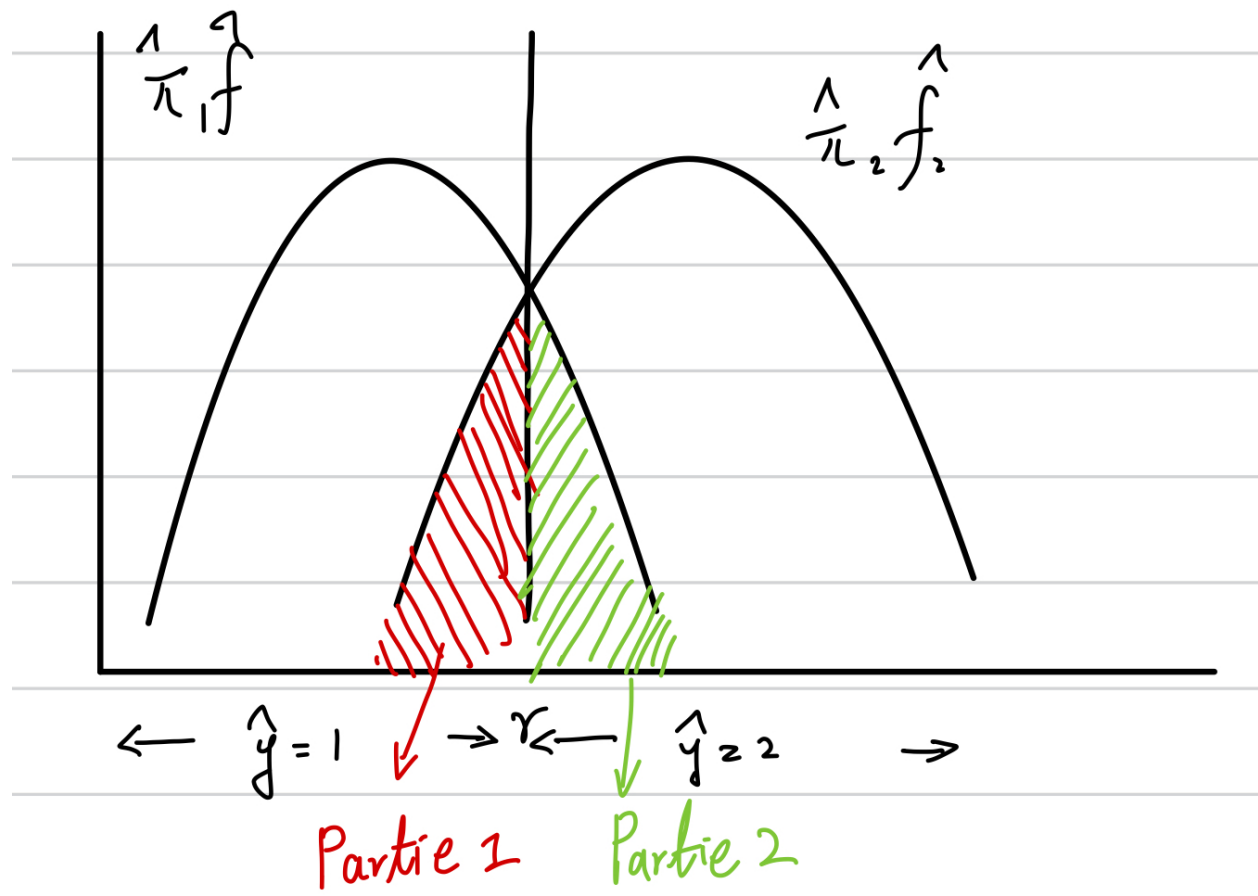
```
y = function(x){
  pi1*dnorm(x,mu1,sqrt(sicar1))-pi2*dnorm(x, mu2, sqrt(sicar2))
}
out = uniroot (y, lower=1.5, upper = 2.5)
```

```
gamma = out$root
print(gamma)
```

```
## [1] 2.091595
```

```
-> r = 2.09
```

$$\hat{\varepsilon}_{th} = \underbrace{\hat{\Pi}_1 \int_r^+ \infty f(x; \hat{\mu}_1; \hat{\sigma}_1^2) dx}_{Partie_1} + \underbrace{\hat{\Pi}_2 \int_r^+ \infty f(x; \hat{\mu}_2; \hat{\sigma}_2^2) dx}_{Partie_2}$$



$$\begin{aligned} \hat{\varepsilon}_{th} &= \hat{\Pi}_1 P(N|\hat{\mu}_1; \hat{\sigma}_1^2 > \gamma) + \hat{\Pi}_2 P(N|\hat{\mu}_2; \hat{\sigma}_2^2 > \gamma) \\ &= \hat{\Pi}_1 \left( 1 - f\left(\frac{\gamma - \hat{\mu}_1}{\hat{\sigma}_1} \right) \right) + \hat{\Pi}_2 \left( f\left(\frac{\gamma - \hat{\mu}_2}{\hat{\sigma}_2} \right) \right) \end{aligned}$$

Sur R,

```
Varpsilon = pi1 * (1-pnorm(2.03,mu1,sqrt(sicar1))) + pi2*pnorm(2.09,mu2,sqrt(sicar2))
print(Varpsilon)
```

## [1] 0.3945809

$\hat{\varepsilon} = 0.3880232$

ii) Classifier Gamma homosédastique : Même démarche avec

$$\hat{\sigma}_1^2 = \hat{\sigma}_2^2 = \hat{\Pi}_1 \times 3.0 + \hat{\Pi}_2 \times 3.2 = 3.1$$

Affectation de M.Li :

$$t_1(Li) = 0.51$$

On affecté M.Li à classe 1 avec 51% de classe erreur de classement  $\hat{\varepsilon}_{th} = 0.3882125$  et  $\gamma = 2$

---

### Eurreur apparente (ou de resubstitution)

L'erreur apparente esst calculée par restitution : les observations de  $D$  sont prises à la fois pour données d'apprentissage et données de test.

- Construire estimée un classifieur sur les donné de  $D$
- Pour chaque  $i \in \{1, \dots, n\}$  déterminer la classe estimée de  $x_i : \hat{y}(x_i)$
- Calculer l'eurreur apparente :

$$\hat{\varepsilon} = \#\{i; \hat{y}(x_i) \neq y_i\} / n$$

3

- $\hat{\varepsilon}_{ap}$  **sous estime l'erreur de classement (biais d'optimisme)**. Le biais de  $\hat{\varepsilon}_{ap}$  est d'autant plus grand que  $n$  est petit ou que le modèle est complexe (beaucoup de paramètre).

“Erreur apparentée” (或称重复估计误差) 这个概念通常用于衡量模型在训练数据集上的性能。它的主要特点是，它评估模型对于已经见过的数据的拟合程度，而不是对未见过的数据的泛化性能。在训练数据集上计算的误差通常会**低估模型的真实性能**，因为模型可能会过度拟合训练数据，无法很好地泛化到新数据，特别是在数据有限或模型复杂时。

因此，“erreur apparentée” 或重复估计误差并不是一种很好的模型性能评估方法，因为它不能提供模型在未见过的数据上的表现。为了更全面地评估模型的性能，通常还需要使用交叉验证或将数据集划分为训练集和测试集，以便评估模型在未见过的数据上的泛化能力。这可以帮助确定模型是否过度拟合了训练数据，以及模型在真实世界中的实际效果如何。

### Bootstrap

L'erreur estimée par bootstrap vise à réduire le biais de l'erreur apparente  $\hat{\varepsilon}_{ap}$  calculée sur  $D$ . Pour  $j = 1, \dots, N$  :

---

<sup>3</sup>这里的井号 # 表示数学符号中的计数符号，用于表示一个集合中元素的数量或计数。在这个上下文中， $\#\{i; \hat{y}(x_i) \neq y_i\}$  表示了满足条件  $\hat{y}(x_i) \neq y_i$  的数据点的数量。它计算了分类器在测试数据中错误分类的数据点数目。

- objectif : corrigé le biais de  $\hat{\varepsilon}_{ap}$
- étapes : pour  $i$  allant de 1 à  $N$  (chemin)
  - tirer de  $D$  un échantillon  $D_j$  de taille  $n$  comportant de possibles répétitions
  - construire le classifieur sur les données de  $D_j$  puis: (使用  $D_j$  上的数据来构建一个分类器, 然后: )
    - \* classer les données de  $D_j$  et déterminer le taux d'erreur  $\hat{\varepsilon}_{D_j}$
    - \* classer les données de  $D$  et déterminer le taux d'erreur  $\hat{\varepsilon}_D$
- calculer  $\beta_j = \hat{\varepsilon}_{D_j} - \hat{\varepsilon}_D$

L'erreur estimée par bootstrap est :  $\hat{\varepsilon}_{boot} = \hat{\varepsilon}_{ap} + \sum_{j=1}^N \beta_j / N$

Le terme  $\sum_{j=1}^N \beta_j / N$  est destiné à corriger le biais d'optimisme associé à l'erreur apparente  $\hat{\varepsilon}_{ap}$

### Leave one out

Le leave One Out ou validation croisée considère tout à tout chacun des points de  $D$  pour donnée de test et les autres points pour données d'apprentissage. 留一法是一种交叉验证的技巧, 它考虑将每一个数据点作为测试数据, 其余数据点作为训练数据。这意味着对于每个数据点  $x_i$ , 它都会被单独用作测试数据, 其余数据点用于训练。

Pour  $i = 1, \dots, n$  :

- construire le classifieur sur  $D$  privé de  $(x_i, y_i)$  构建一个分类器, 使用数据集  $D$  但不包括数据点  $(x_i, y_i)$ 。
- estimer la classe de  $x_i : y_i$  估计数据点  $x_i$  的类别, 表示为  $\hat{y}_i$ 。
- comparer  $\hat{y}_i$  à la vraie classe de  $x_i : y_i$  比较  $\hat{y}_i$  与数据点  $x_i$  的真实类别  $y_i$ 。

L'erreur de validation croisée (Cross Validation) est  $\hat{\varepsilon}_{CV} = \sum_{i=1}^n 1_{\{\hat{y}_i \neq y_i\}} / n$

交叉验证错误率  $\hat{\varepsilon}_{CV}$  是通过计算被分类器错误分类的数据点数量, 然后除以数据点总数  $n$ , 得到的错误率。具体公式为:  $\hat{\varepsilon}_{CV} = \sum_{i=1}^n 1_{\{\hat{y}_i \neq y_i\}} / n$ 。

留一法交叉验证的主要思想是通过反复单独测试每个数据点, 以获得对分类器性能的更准确估计。它是一种有效的方法, 特别适用于小型数据集或在评估分类器性能时需要减小估计偏差的情况。

### v-fold Cross Validation

受留一法启发, On établit d'abord une partition de  $\{1, \dots, n\}$  en  $K$  groupes de tailles similaires :  $I_1, \dots, I_K$ . Puis, pour  $k = \{1, \dots, K\}$  :

- on construit un classifieur sur  $\{(x_i, y_i); i \notin I_k\}$
- on calcule l'erreur de classement  $\hat{\varepsilon}_k$  sur  $\{(x_i, y_i); i \in I_k\}$

L'erreur v-fold Cross Validation est :  $\hat{\varepsilon}_{v\text{-fold } CV} = \sum_{k=1}^K \hat{\varepsilon}_k / K$ .

Lorsque  $K = n$ , le v-fold Crosse validation est un Leave One Out

v-fold 交叉验证是一种常用的模型评估方法，它允许将数据集分成多个子集，以多次评估模型性能，从而提供更准确的性能估计。这对于在小样本情况下或需要减小估计偏差的情况下非常有用。

## Les courbes ROC (receier opérating characteristic)

On suppose le donné répartis en deux classes

- Classe\_1 = sans risque
- Classe\_2 = à risque

Un indicidu  $x \in X$  est affect à l'une de classe par comparaison de non score à un seuil  $c$  :

- $\hat{y}(x) = 2$  ssi  $s(x) > c$
- $\hat{y}(x) = 1$  ssi  $s(x) \leq c$

Les observation se répartient en quatre catégories :

Etat	Explication	Equation
Faux Positifs	真 1 算 2	$FP = \#\{y_i = 1, \hat{y}(x_i) = 2\}$
Vrais Positifs	真 2 算 2	$VP = \#\{y_i = 2, \hat{y}(x_i) = 2\}$
Faux Négatif	真 2 算 1	$FN = \#\{y_i = 2, \hat{y}(x_i) = 1\}$
Vrais Négatif	真 1 算 1	$VN = \#\{y_i = 1, \hat{y}(x_i) = 1\}$

Nombre de négatif	$N = FP + VN$
Nombre de possitif	$P = VP + FN$
Taux de faux possitif	$TFP = FP/P$
Taux de vrais possitif	$TVP = VP/P$

**remarque** L'erreur apparence est :

$$\hat{\varepsilon}_{esp} = TFP \times \frac{N}{n} + |1 - TVP| \times \frac{P}{n}$$



## Justification

$$\begin{aligned}
 TFP \times \frac{N}{n} + |1 - TVP| \times \frac{P}{n} &= \frac{FP}{N} \times \frac{N}{n} + \left(1 - \frac{VP}{P}\right) \times \frac{P}{n} \\
 &= \frac{FP}{n} + \frac{P - VP}{P} \times \frac{P}{n} \\
 &= \frac{FR}{n} + \frac{FN}{n} \\
 &= \frac{FP + FN}{n} \\
 &= \frac{\#\{\text{mal classe}\}}{n} \\
 &= \hat{\varepsilon}_{ap}
 \end{aligned}$$

Le statistique TFP et TVP dépendent du seuil  $c$  auquel on compare les valeurs du score  $S$  calculés en l'échantillon de test

à chaque valeur de  $c$  xxxxxxxx par du couple (TFP, TVP)

Les courb ROC est l'ensemble des points  $\{(TFP_c; TVP_c); c \in s(X)\}$

On appelle avec (area under curve) l'aire sous la courbe Roc

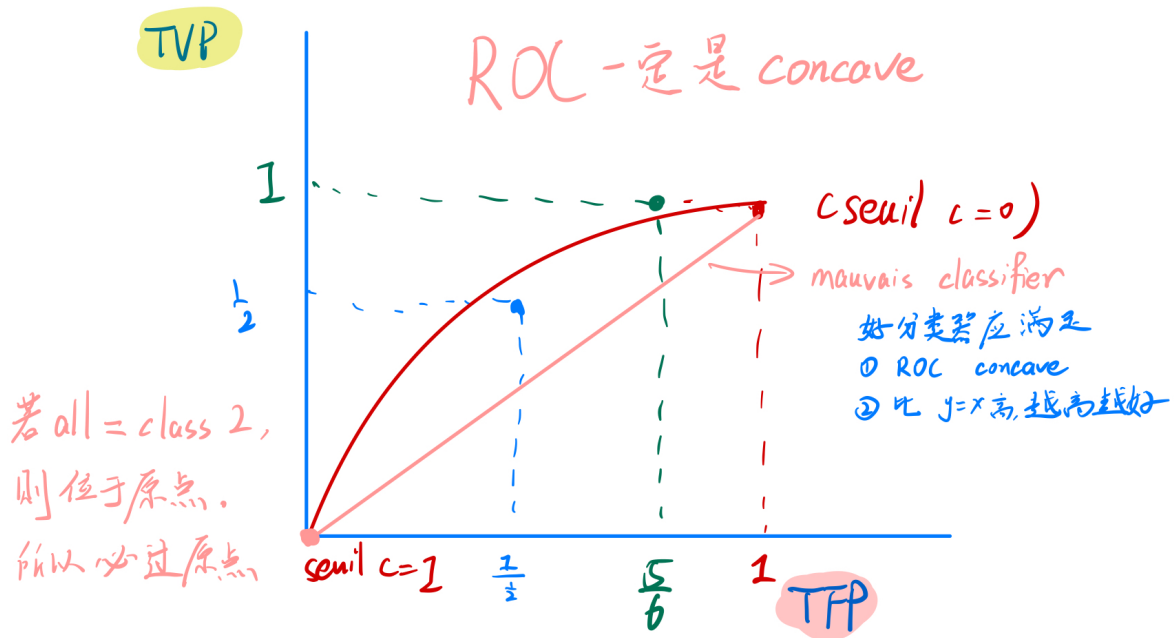
Exercice en considère les données de table 3 EX3 td 3

Tracer la courbe ROC du classification

Période	Taux de change en début de période €/S	Rentabilité du marché américain	Rentabilité du marché français			
1	1	0.67	10%	12%		
2	2	0.74	-2%	13%		
3	3	0.8	12%	-4%		
4	4	0.71	22%	5%		
5	5	0.76	5%	14%		
6	6	0.82				
Question 1						
Période	Taux de change en début de période €/S	Évolution du change €/S	Évolution du change \$/€	américain pour un investisseur français	Rentabilité du marché français pour un investisseur américain	
1	1	0.67				
2	2	0.74	10.45%	-9.44%	-0.41%	23.70%
3	3	0.8	8.11%	-7.50%	-9.35%	22.10%
4	4	0.71	-11.25%	12.68%	26.20%	-14.80%
5	5	0.76	7.04%	-6.58%	13.97%	12.89%
6	6	0.82	7.89%	-7.32%	-2.68%	23.00%

$\hat{1}$	$\hat{2}$
1	2
2	2
4	4
6	4
TFP = 2/3	
TVP = 1/2	

Seuille  $c = 1/2$



## Choix de modèle

On peut comparer les modèles paramétrique ou semi-paramétrique par des critères d'information. Tel que AIC, BIC (voir CH2)

Cependant, un critère de choix de modèle réalise un objectif (AIC estime la déviance d'un modèle, BIC estime la probabilité d'un modèle conditionnellement aux données, etc)

L'objectif du critère de choix de modèle peut être éloigné de celui de la classification. **Autrement dit, un modèle peut être bon du point de vue du critère qu'il optimise mais ne pas conduire à un classifieur satisfaisant.**

étape : établir une partition de  $\{1, \dots, n\}$  en  $K$  classes de

```
#x = read.table(file = "http://alexandre.lourme.free.fr/M2IREF/SCORING/BANKNOTE" , sep=',', dec='.')
```

On considère trois modèles :

- AD Gaussienne homosédastique
- AD Gaussienne hétérosédastique
- Regression logistique

Déterminer sous classe de modèle

- l'erreur LOO
- L'erreur apparente
- La classe de nouveau billet : Length = 214,90 Right = 129,96 Left = 130,12

$$\hat{\varepsilon}_{boost} = \left( \frac{1}{N} \sum_{i=1}^N \beta_i \right) + \hat{\varepsilon}_{op}$$

> V fold cross valuation Principe : le même que  $L_o$  mesure à chaque estimation stochastique plusieurs port de données d'apprentissage

etape : établir une partition de  $\{1, \dots, n\}$  en  $t$  class de taille  $I_1, \dots, I_K$  peu d'allant de 1 à  $K$

continuer un classifier sur  $D1\{(); I_k\}$  Estimer le taux d'erreur sur  $\{(x, z); i \in I_k\}$  :  $\hat{\varepsilon}$

Erreur V-fold cross valuation est

$$\hat{\varepsilon}_{V-fold} = \sum_{k=1}^k \hat{\varepsilon}_k / k$$

## EX2 TD3 Table 2 indique le flux et la solvabilité de dix clients bancaires

client	1	2	3	4	5	6	7	8	9	10
flux	2	4	6	5	4	7	8	6	5	6
solvable	N	N	N	N	O	O	O	O	O	O

Estimez l'erreur de classement d'un classifieur gaussien (i) par résubstitution (ii) par bootstrap (iii) par Leave One Out (iv) par v-fold Cross Validation.

*# une fonction qui calcule l'erreur de classement du classifieur gaussien homoscédastique*  
`library(Rmixmod)`

## Loading required package: Rcpp

## Rmixmod v. 2.1.8 / URI: [www.mixmod.org](http://www.mixmod.org)

```
ergauss <- function(train,trainlab,test,testlab){
  trainlab=as.factor(trainlab)
  testlab=as.factor(testlab)
  yourmodel=mixmodGaussianModel(listModels=c("Gaussian_pk_L_C"))
  out <- mixmodLearn(train, knownLabels=trainlab, models = yourmodel)
  pred <- mixmodPredict(data = test, classificationRule = out["bestResult"])
  erg = 1-sum(pred[5]==as.numeric(testlab))/length(test) # taux d'erreur sur les données de t
}

# les données
mydata <- data.frame(
  flux=c(2,4,6,5,4,7,8,6,5,6), solvable=c('N','N','N','N','O','O','O','O','O','O')
)
attach(mydata)
n=nrow(mydata)
#####
```

```

# estimation de l'erreur par resubstitution (erreur apparente)
#####
epsilon_ap=ergauss(flux,solvable,flux,solvable)
cat('epsilon_ap : ',epsilon_ap,'\n')

## epsilon_ap : 0.3

## epsilon_ap : 0.3

#####
# estimation de l'erreur par Leave One Out (ou Cross Validation)
#####
comp=0
for (i in 1:n){ comp=comp+ergauss(flux[-i],solvable[-i],flux[i],solvable[i])
}
epsilon_cv=comp/n
cat('epsilon_cv : ',epsilon_cv,'\n')

## epsilon_cv : 0.6

## epsilon_cv : 0.6

#####
# estimation de l'erreur par v-fold Cross Validation
#####
K=5 # nbre de sous échantillons
epsilonlist <- NULL
list=sample(1:n,n,replace=FALSE)
for (j in 1:K){
  train <- mydata[-c(((2*j-1):(2*j))),] # échantillon d'apprentissage
  test <- mydata[c(((2*j-1):(2*j))),] # échantillon de test
  epsilonlist[j] <- ergauss(train$flux,train$solvable,test$flux,test$solvable)
}
epsilon_vfoldcv = mean(epsilonlist)
cat('epsilon_vfoldcv : ',epsilon_vfoldcv,'\n')

```

```
## epsilon_vfoldcv : 0.7
## epsilon_vfoldcv : 0.7

#####
# estimation de l'erreur par bootstrap
#####
beta <- NULL
N <- 20
for (j in 1:N){
  select <- sample(1:n,n,replace=TRUE)
  locdata <- mydata[select,] # n-échantillon tiré des données avec de possibles répétitions
  epsilonj=ergauss(locdata$flux,locdata$solvable,locdata$flux,locdata$solvable)
  epsilon=ergauss(locdata$flux,locdata$solvable,mydata$flux,mydata$solvable)
  beta[j]=epsilonj-epsilon
}
epsilon_boot = epsilon_ap + mean(beta,na.rm=TRUE)
cat('epsilon_boot : ',epsilon_boot,'\n')

## epsilon_boot : 0.24
```

## EX3 TD3

Tale 3 donne la classe vraie et les probabilités conditionnelles obtenues sur un échantillon de test de dix sujets grâce à in classifieur basé sur un modèle paramétrique <sup>4</sup>

client	1	2	3	4	5	6	7	8	9	10
classe vraie	2	2	2	2	1	1	1	1	1	1
$t_1$	0.27	0.18	0.56	0.61	0.09	0.27	0.96	0.34	0.69	0.13
$t_2$	0.73	0.82	0.44	0.39	0.91	0.73	0.04	0.66	0.31	0.87

Tracez la courbe ROC du classifieur et calculez AUC.

```
# les données
t <- data.frame( t1=c(0.27,0.18,0.56,0.61,0.09,0.27,0.96,0.34,0.69,0.13) )
t$t2=1-t$t1 # probabilités d'appartenir à la classe 2
```

<sup>4</sup>Table 3 : partition (classe rais) et probabilités conditionnelles ( $t_1$  : probabilité d'appartenir à la classe 1,  $t_2$  : probabilité d'appartenir à la classe 2) obtenues sur un échantillon de test de dix clients par un classifieur paramétriques 表 3: 参数分类器对 10 个客户测试样本进行分区 (类别 rais) 和条件概率 ( $t_1$  : 属于类别 1 的概率,  $t_2$  : 属于类别 2 的概率) 得出的结果。

```

compare <- data.frame( verite=c(2,2,2,2,1,1,1,1,1,1))
compare$decision=compare$verite
TFP <- TVP <- NULL
for (ii in 0:10){
  compare$decision[t$t2>ii/10]=2
  compare$decision[t$t2<=ii/10]=1
  verite <- factor(compare$verite,levels=c(1:2))
  decision <- factor(compare$decision,levels=c(1:2))
  confusion <- table(verite,decision)
  TFP[ii]=confusion[1,2]/(confusion[1,1]+confusion[1,2])
  TVP[ii]=confusion[2,2]/(confusion[2,1]+confusion[2,2])
  cat('-----','\n')
  cat('seuil pour la classe 2 :',ii/10,'\n')
  print(confusion)
  cat('TFP : ',TFP[ii],'\n')
  cat('TVP : ',TVP[ii],'\n')
}

```

```

## -----
## seuil pour la classe 2 : 0
##      decision
## verite 1 2
##      1 0 6
##      2 0 4
## TFP :
## TVP :
## -----
## seuil pour la classe 2 : 0.1
##      decision
## verite 1 2
##      1 1 5
##      2 0 4
## TFP : 0.8333333
## TVP : 1
## -----
## seuil pour la classe 2 : 0.2
##      decision
## verite 1 2
##      1 1 5
##      2 0 4

```

```

## TFP : 0.8333333
## TVP : 1
## -----
## seuil pour la classe 2 : 0.3
##      decision
## verite 1 2
##      1 1 5
##      2 0 4
## TFP : 0.8333333
## TVP : 1
## -----
## seuil pour la classe 2 : 0.4
##      decision
## verite 1 2
##      1 2 4
##      2 1 3
## TFP : 0.6666667
## TVP : 0.75
## -----
## seuil pour la classe 2 : 0.5
##      decision
## verite 1 2
##      1 2 4
##      2 2 2
## TFP : 0.6666667
## TVP : 0.5
## -----
## seuil pour la classe 2 : 0.6
##      decision
## verite 1 2
##      1 2 4
##      2 2 2
## TFP : 0.6666667
## TVP : 0.5
## -----
## seuil pour la classe 2 : 0.7
##      decision
## verite 1 2
##      1 3 3

```

```

##      2 2 2
## TFP : 0.5
## TVP : 0.5
## -----
## seuil pour la classe 2 : 0.8
##      decision
## verite 1 2
##      1 4 2
##      2 3 1
## TFP : 0.3333333
## TVP : 0.25
## -----
## seuil pour la classe 2 : 0.9
##      decision
## verite 1 2
##      1 5 1
##      2 4 0
## TFP : 0.1666667
## TVP : 0
## -----
## seuil pour la classe 2 : 1
##      decision
## verite 1 2
##      1 6 0
##      2 4 0
## TFP : 0
## TVP : 0
## -----
## seuil pour la classe 2 : 0
##      decision
## verite 1 2
## 106
## 204 ## TFP :
## TVP :
## -----
## seuil pour la classe 2 : 0.1
##      decision
## verite 1 2
## 115

```



```

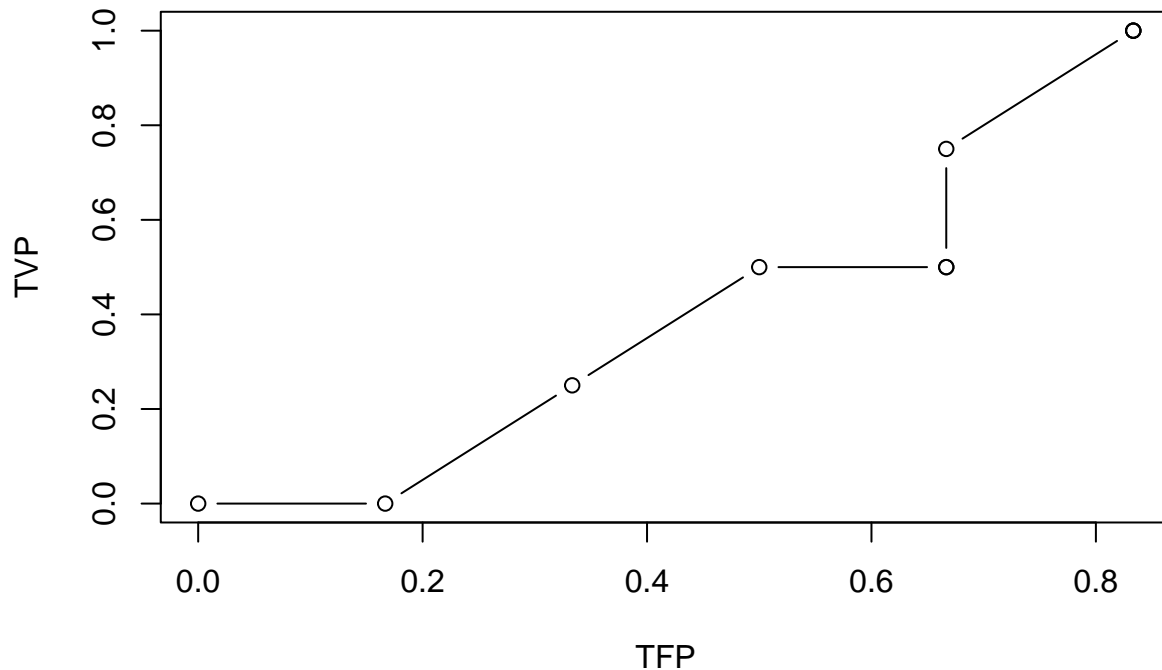
## 204
## TFP : 0.8333333
## TVP : 1
## -----
## seuil pour la classe 2 : 0.2
##      decision
## verite 1 2
## 115
## 204
## TFP : 0.8333333
## TVP : 1
## -----
## seuil pour la classe 2 : 0.3
##      decision
## verite 1 2
## 115
## 204
## TFP : 0.8333333
## TVP : 1
## -----
## seuil pour la classe 2 : 0.4
##      decision
## verite 1 2
## 124
## 213
## TFP : 0.6666667
## TVP : 0.75
## -----
## seuil pour la classe 2 : 0.5
##      decision
## verite 1 2
## 124
## 222
## TFP : 0.6666667
## TVP : 0.5
## -----
## seuil pour la classe 2 : 0.6
##      decision
## verite 1 2

```

```

## 124
## 222
## TFP : 0.6666667
## TVP : 0.5
## -----
## seuil pour la classe 2 : 0.7
##      decision
## verite 1 2
## 133
## 222 ## TFP : 0.5 ## TVP : 0.5 ## -----
## seuil pour la classe 2 : 0.8
##      decision
## verite 1 2
## 142
## 231
## TFP : 0.3333333
## TVP : 0.25
## -----
## seuil pour la classe 2 : 0.9
##      decision
## verite 1 2
## 151
## 240
## TFP : 0.1666667
## TVP : 0
## -----
## seuil pour la classe 2 : 1
##      decision
## verite 1 2
## 160
## 240 ## TFP : 0
## TVP : 0
plot(TFP,TVP,'b',xlab='TFP',ylab='TVP')

```



```
library(zoo)
```

```
##
```

```
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      as.Date, as.Date.numeric
```

```
id <- order(TFP)
```

```
AUC <- sum(diff(TFP[id])*rollmean(TVP[id],2)) # area under ROC curve print(AUC)
```

```
print(AUC)
```

```
## [1] 0.3125
```

```
## [1] 0.3125
```

## EX4 TD3

M. Shan souhaite modéliser les données de Tables 2 par un modèle gaussien à deux classes. Lequel du modèle hétéroscédastique ou homoscédastique est le meilleur selon BIC?

```
# BIC du modèle gaussien hétéroscédastique avec Rmixmod
```

```
library(Rmixmod)
```

```
model=mixmodGaussianModel(listModels=c("Gaussian_pk_Lk_Ck")) # modèle gaussien hétéroscédasti
```

```
out <- mixmodLearn(flux,knownLabels=as.factor(solvable),models=model,criterion=c('BIC'))
```

```
bic=-out[9][[1]][3]/2
```

```
cat('BIC : ',bic,'\n')
```

```
## BIC : -29.77396
```

```
## BIC : -29.77396
```

```
# BIC du modèle gaussien hétéroscédastique à la main
n=nrow(mydata)
n1=sum(solvable=='O') # nbre de solvables
pi1=n1/n # proportion de solvables
m1=mean(flux[solvable=='O']) # moyenne du flux chez les solvables
v1=var(flux[solvable=='O'])*(n1-1)/n1 # variance du flux chez les solvables
n2=sum(solvable=='N') # nbre de non solvables
pi2=n2/n # proportion de non solvables
m2=mean(flux[solvable=='N']) # moyenne du flux chez les non solvables
v2=var(flux[solvable=='N'])*(n2-1)/n2 # variance du flux chez les non solvables
loglik = 0
for (i in 1:n){
  loglik=loglik+
    (solvable[i]=='O')*(log(pi1)+log(dnorm(flux[i],m1,sqrt(v1))))+
    (solvable[i]=='N')*(log(pi2)+log(dnorm(flux[i],m2,sqrt(v2))))
}
eta=1+2+2
bic=loglik-eta/2*log(n)
cat('BIC : ',bic,'\n')
```

```
## BIC : -29.77396
```

```
## BIC : -29.77396
```

## Modèle gaussien homoscédastique

```
# BIC du modèle gaussien homoscédastique avec Rmixmod
library(Rmixmod)
model=mixmodGaussianModel(listModels=c("Gaussian_pk_L_C")) # modèle gaussien homoscédastique
out <- mixmodLearn(flux,knownLabels=as.factor(solvable),models=model,criterion=c('BIC'))
bic=-out[9][[1]][3]/2
cat('BIC : ',bic,'\n')
```

```
## BIC : -28.66772
```

```
## BIC : -28.66772
```

```
# BIC du modèle gaussien homoscédastique à la main
n=nrow(mydata)
```

```

m=pi1*m1+pi2*m2 # moyenne du flux
v=pi1*v1+pi2*v2 # estimateur MV de la variance du flux
loglik = 0
for (i in 1:n){loglik=loglik+
  (solvable[i]=='0')*(log(pi1)+log(dnorm(flux[i],m1,sqrt(v))))+
  (solvable[i]=='N')*(log(pi2)+log(dnorm(flux[i],m2,sqrt(v))))
}
eta=1+2+1
bic=loglik-eta/2*log(n)
cat('BIC : ',bic,'\n')

## BIC : -28.66772
## BIC : -28.66772

```