Data Science for Actuaries (ACT6100)

Arthur Charpentier

Supervisé # 2 (régularisation - GLM)

automne 2Q20

https://github.com/freakonometrics/ACT6100/

Régression logistique

On veut résoudre ici $(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) = \underset{\beta_0, \boldsymbol{\beta}}{\text{argmin}} \left\{ -\log \mathcal{L}(\beta_0, \boldsymbol{\beta}) \right\}$ où

$$\log \mathcal{L}(\beta_0, \boldsymbol{\beta}) = \sum_{i=1}^{n} \left(y_i(\beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta}) - \log[1 + \exp(\beta_0 + \boldsymbol{x}_i^\top \boldsymbol{\beta})] \right)$$

Algorithm 1: IWLS (Iterated Weighted Least Squares)

initialization : $\beta_0^{(0)}, \boldsymbol{\beta}^{(0)}$;

for t=1,2,... **do**

- $\boldsymbol{p}^{(t)} : p_i^{(t)} \leftarrow (1 + e^{-\beta_0^{(t-1)} - \boldsymbol{x}_i^\top \boldsymbol{\beta}^{(t-1)}})^{-1}$;
- $\boldsymbol{\omega}^{(t)} : \omega_i^{(t)} \leftarrow p_i^{(t)}(1 - p_i^{(t)})$;
- $\boldsymbol{z}^{(t)} \leftarrow \beta_0^{(t-1)} + \boldsymbol{X}^\top \boldsymbol{\beta}^{(t-1)} + \boldsymbol{\omega}^{(t)-1}(\boldsymbol{y} - \boldsymbol{p}^{(t)})$;
- $\boldsymbol{\beta}^{(t)} \leftarrow \text{argmin}\{(\boldsymbol{z} - \beta_0 - \boldsymbol{X}\boldsymbol{\beta})^\top \boldsymbol{\omega}(\boldsymbol{z} - \beta_0 - \boldsymbol{X}\boldsymbol{\beta})\}$;

Régression logistique pénalisée

On veut résoudre ici

$$(\widehat{\beta}_0, \widehat{\boldsymbol{\beta}}) = \underset{\beta_0, \boldsymbol{\beta}}{\text{argmin}} \{-\log \mathcal{L}(\beta_0, \boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1\}$$

On adapte l'alogrithme précédant

Algorithm 2: IPWLS (Penalized Weighted Least Squares)

---

initialization : $\beta_0^{(0)}, \boldsymbol{\beta}^{(0)}$;

for t=1,2,... **do**

$\quad \boldsymbol{p}^{(t)} : p_i^{(t)} \leftarrow (1 + e^{-\beta_0^{(t-1)} - \boldsymbol{x}_i^\top \boldsymbol{\beta}^{(t-1)}})^{-1}$;

$\quad \boldsymbol{\omega}^{(t)} : \omega_i^{(t)} \leftarrow p_i^{(t)}(1 - p_i^{(t)})$;

$\quad \boldsymbol{z}^{(t)} \leftarrow \beta_0^{(t-1)} + \boldsymbol{X}^\top \boldsymbol{\beta}^{(t-1)} + \boldsymbol{\omega}^{(t)-1}(\boldsymbol{y} - \boldsymbol{p}^{(t)})$;

$\quad \beta_0^{(t)}, \boldsymbol{\beta}^{(t)} \leftarrow$ $\text{argmin}\{(\boldsymbol{z} - \beta_0 - \boldsymbol{X}\boldsymbol{\beta})^\top \boldsymbol{\omega}(\boldsymbol{z} - \beta_0 - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1\}$;

Régression logistique pénalisée

Algorithm 3: WLS LASSO (step 6)

$\alpha_0 \leftarrow \dfrac{\sum_i \omega_i^{(t)} \boldsymbol{z}_i^{(t)}}{\sum_i \omega_i^{(t)}}$ and $\alpha_j \leftarrow \widehat{\boldsymbol{\beta}}_j^{(t-1)}$ for $j = 1, 2, \cdots, k$;

for j=1,2,...,k **do**

  for i=1,2,...,n **do**

    $r_{i,j} \leftarrow \boldsymbol{z}_i^{(t)} - \alpha_0 - \sum_\ell \alpha_\ell x_{i\ell}$

  $u_j^{(t)} \leftarrow \sum_i \omega_i^{(t)} r_{ij} x_{ij}$ and $v_j^{(t)} \leftarrow \sum_i \omega_i^{(t)} x_{ij}^2$;

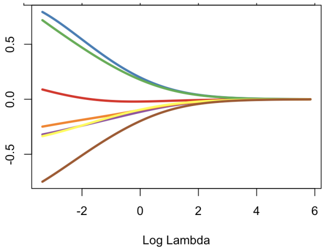
  $\alpha_j = \text{sign}(u_j^{(t)}) \left(\dfrac{|u_j^{(t)} - \lambda|}{v_j^{(t)}} \right)_+$;

$\widehat{\beta}_0^{(t)} \leftarrow \alpha_0$ and $\widehat{\boldsymbol{\beta}}_j^{(t)} \leftarrow \alpha_j$

Régression Ridge avec R

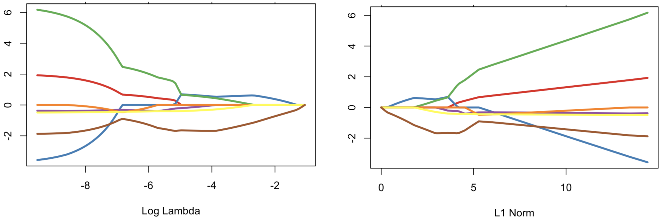
```
> myocarde=read.table(
  "http://freakonometrics.free.fr/saporta.csv",
  header=TRUE,sep=";")
y = myocarde$PRONO=="1"
X = myocarde[,1:7]
for(j in 1:7) X[,j] = (X[,j]-mean(X[,j]))/sd(X[,j])
X = as.matrix(X)
```

```
> library(glmnet)
> glm_ridge = glmnet(X, y, alpha=0,
    family="binomial")
> plot(glm_ridge,xvar="lambda",col=
   colrs,lwd=3)
```



Régression LASSO avec R

```
> library(glmnet)
> glm_lasso = glmnet(X, y, alpha=1, family="binomial")
> plot(glm_lasso,xvar="lambda",col=colrs,lwd=3)
```



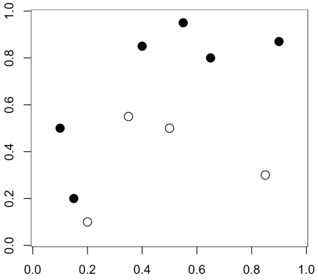

```
> round(cbind(glm_lasso$lambda,t(glm_lasso$beta),3)
Lambda FRCAR  INCAR INSYS  PRDIA  PAPUL  PVENT  REPUL
 0.288 .      .     .      .      .      .     -0.244
 0.263 .      0.025 .      .      .      .     -0.334
 0.078 .      0.583 .      .      .      .     -1.069
 0.071 .      0.609 0.005  .      .      .     -1.125
 0.065 .      0.618 0.030  .      .     -0.007 -1.175
```

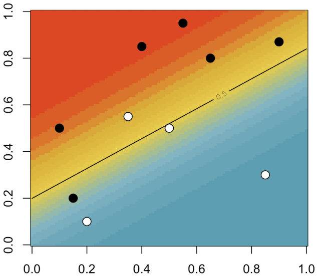
Régression logistique avec R

$y \in \{\circ, \bullet\}$ and two covariates x_1 and x_2

```
> x1 = c(.4,.55,.65,.9,.1,.35,.5,.15,.2,.85)
> x2 = c(.85,.95,.8,.87,.5,.55,.5,.2,.1,.3)
> y = c(1,1,1,1,1,0,0,1,0,0)
> df = data.frame(x1=x1,x2=x2, y=as.factor(z))
> reg = glm(y~x1+x2,data=df, family=binomial(link = "logit"))
> summary(reg)

Coefficients:
      Estimate Std. z value Pr(>|z|)
(Int)  -1.706  1.999  -0.854  0.393
x1     -5.489  5.360  -1.024  0.306
x2      8.568  5.515   1.554  0.120
```

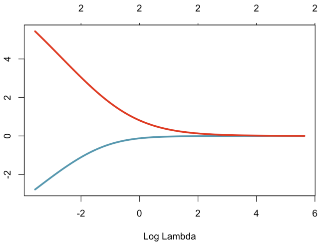


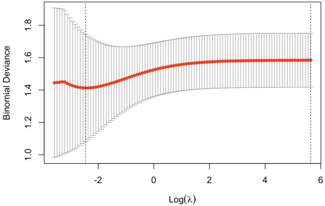


Régression Ridge avec R

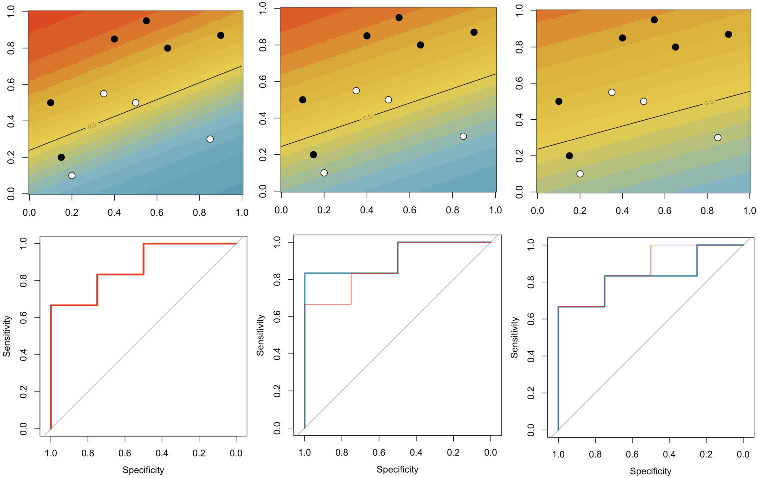
$y \in \{\circ, \bullet\}$ and two covariates x_1 and x_2

```
> library(glmnet)
X=as.matrix(cbind(df$x1,df$x2))
Y=df$y
ridge = glmnet(x = X,y = Y,family =
     "binomial",alpha=0,
    standardize = TRUE)
plot(ridge,col=clr2,lwd=3,xvar = "
    lambda")
cvfit = cv.glmnet(x=X, y=Y,alpha=0,
    family = "binomial")
plot(cvfit)
```





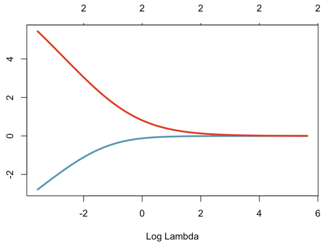
Régression Ridge avec R

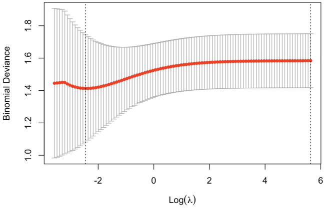


Régression LASSO avec R

$y \in \{\circ, \bullet\}$ and two covariates x_1 and x_2

```
> library(glmnet)
> X=as.matrix(cbind(df$x1,df$x2))
> Y=df$y
> lasso = glmnet(x = X,y = Y,family
    = "binomial", alpha=1,
   standardize = TRUE)
> plot(lasso,col=clr2,lwd=3,xvar =
    "lambda")
> cvfit = cv.glmnet(x=X, y=Y, alpha
   =1, family = "binomial")
> plot(cvfit)
```





Régression LASSO avec R