


Data Science for Actuaries (ACT6100)

Arthur Charpentier

Supervisé # 2 (régularisation - GLM)

automne 2Q20

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

Régression logistique

On veut résoudre ici $(\hat{\beta}_0, \hat{\beta}) = \underset{\beta_0, \beta}{\operatorname{argmin}} \{-\log \mathcal{L}(\beta_0, \beta)\}$ où

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

Algorithm 1: IWLS (Iterated Weighted Least Squares)

- 1 initialization : $\beta_0^{(0)}, \beta^{(0)}$;
 - 2 **for** $t=1, 2, \dots$ **do**
 - 3 $\mathbf{p}^{(t)} : p_i^{(t)} \leftarrow (1 + e^{-\beta_0^{(t-1)} - \mathbf{x}_i^\top \beta^{(t-1)}})^{-1}$;
 - 4 $\boldsymbol{\omega}^{(t)} : \omega_i^{(t)} \leftarrow p_i^{(t)}(1 - p_i^{(t)})$;
 - 5 $\mathbf{z}^{(t)} \leftarrow \beta_0^{(t-1)} + \mathbf{X}^\top \beta^{(t-1)} + \boldsymbol{\omega}^{(t)-1}(\mathbf{y} - \mathbf{p}^{(t)})$;
 - 6 $\beta^{(t)} \leftarrow \underset{\beta}{\operatorname{argmin}} \{(\mathbf{z} - \beta_0 - \mathbf{X}\beta)^\top \boldsymbol{\omega}(\mathbf{z} - \beta_0 - \mathbf{X}\beta)\}$;
-

Régression logistique pénalisée

On veut résoudre ici

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

On adapte l'algorithme précédant

Algorithm 2: IPWLS (Penalized Weighted Least Squares)

```
1 initialization :  $\beta_0^{(0)}, \beta^{(0)}$ ;  
2 for  $t=1, 2, \dots$  do  
3    $\mathbf{p}^{(t)} : p_i^{(t)} \leftarrow (1 + e^{-\beta_0^{(t-1)} - \mathbf{x}_i^\top \beta^{(t-1)}})^{-1}$ ;  
4    $\omega^{(t)} : \omega_i^{(t)} \leftarrow p_i^{(t)}(1 - p_i^{(t)})$ ;  
5    $\mathbf{z}^{(t)} \leftarrow \beta_0^{(t-1)} + \mathbf{X}^\top \beta^{(t-1)} + \omega^{(t)-1}(\mathbf{y} - \mathbf{p}^{(t)})$ ;  
6    $\beta_0^{(t)}, \beta^{(t)} \leftarrow$   
    $\operatorname{argmin}\{(\mathbf{z} - \beta_0 - \mathbf{X}\beta)^\top \omega(\mathbf{z} - \beta_0 - \mathbf{X}\beta) + \lambda \|\beta\|_1\};$ 
```

Régression logistique pénalisée

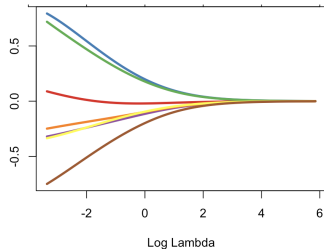
Algorithm 3: WLS LASSO (step 6)

```
1  $\alpha_0 \leftarrow \frac{\sum_i \omega_i^{(t)} \mathbf{z}_i^{(t)}}{\sum_i \omega_i^{(t)}}$  and  $\alpha_j \leftarrow \hat{\beta}_j^{(t-1)}$  for  $j = 1, 2, \dots, k$ ;  
2 for  $j=1, 2, \dots, k$  do  
3   for  $i=1, 2, \dots, n$  do  
4      $r_{ij} \leftarrow \mathbf{z}_i^{(t)} - \alpha_0 - \sum_{\ell} \alpha_{\ell} x_{i\ell}$   
5      $u_j^{(t)} \leftarrow \sum_i \omega_i^{(t)} r_{ij} x_{ij}$  and  $v_j^{(t)} \leftarrow \sum_i \omega_i^{(t)} x_{ij}^2$ ;  
6      $\alpha_j = \text{sign}(u_j^{(t)}) \left( \frac{|u_j^{(t)} - \lambda|}{v_j^{(t)}} \right)_+ ;$   
7  $\hat{\beta}_0^{(t)} \leftarrow \alpha_0$  and  $\hat{\beta}_j^{(t)} \leftarrow \alpha_j$ 
```

Régression Ridge avec R

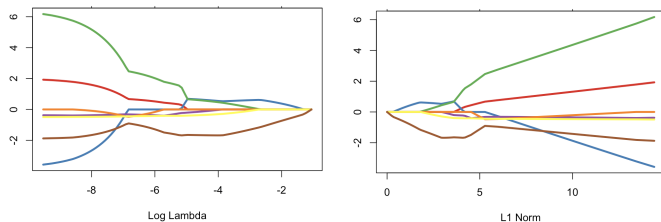
```
1 > myocarde=read.table(  
2   "http://freakonometrics.free.fr/saporta.csv",  
3   header=TRUE, sep=";")  
4 y = myocarde$PRONO=="1"  
5 X = myocarde[,1:7]  
6 for(j in 1:7) X[,j] = (X[,j]-mean(X[,j]))/sd(X[,j])  
7 X = as.matrix(X)
```

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



Régression LASSO avec R

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

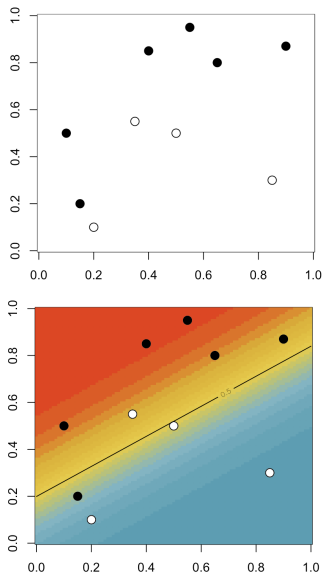


```
1 > round(cbind(glm_lasso$lambda, t(glm_lasso$beta)), 3)
2 Lambda  FRCAR  INCAR  INSYS  PRDIA  PAPUL  PVENT  REPUL
3 0.288    .      .      .      .      .      .      -0.244
4 0.263    .      0.025 .      .      .      .      -0.334
5 0.078    .      0.583 .      .      .      .      -1.069
6 0.071    .      0.609 0.005 .      .      .      -1.125
7 0.065    .      0.618 0.030 .      .      -0.007 -1.175
```

Régression logistique avec R

$y \in \{0, 1\}$ and two covariates x_1 and x_2

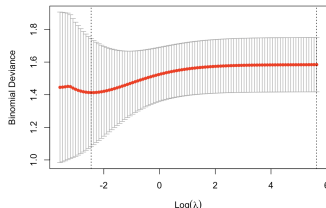
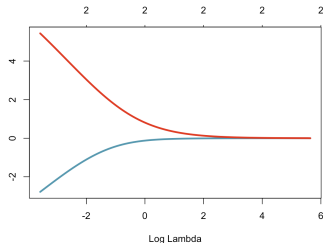
```
1 > x1 = c(.4,.55,.65,.9,.1,.35,.5,  
  .15,.2,.85)  
2 > x2 = c(.85,.95,.8,.87,.5,.55,  
  .5,.2,.1,.3)  
3 > y = c(1,1,1,1,1,0,0,1,0,0)  
4 > df = data.frame(x1=x1,x2=x2, y=as  
  .factor(z))  
5 > reg = glm(y~x1+x2,data=df, family  
  =binomial(link = "logit"))  
6 > summary(reg)  
7  
8 Coefficients:  
9      Estimate Std. z value Pr(>|z|)  
10 (Int) -1.706   1.999  -0.854  0.393  
11 x1      -5.489   5.360  -1.024  0.306  
12 x2       8.568   5.515   1.554  0.120
```



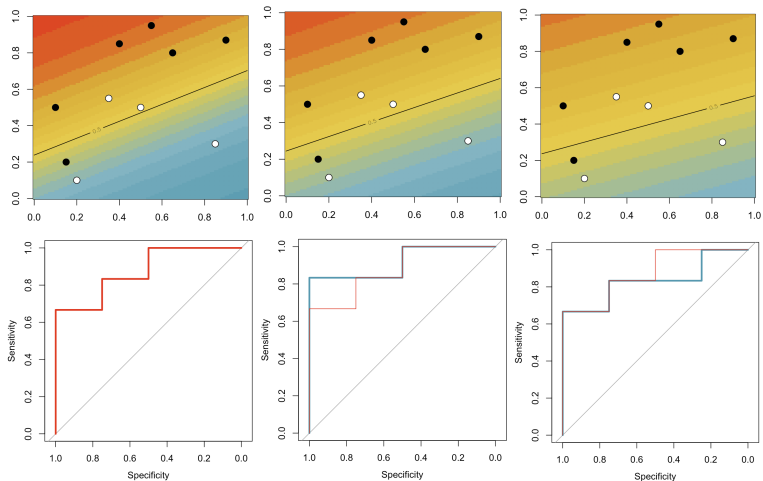
Régression Ridge avec R

$y \in \{0, \bullet\}$ and two covariates x_1 and x_2

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



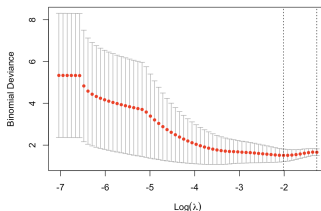
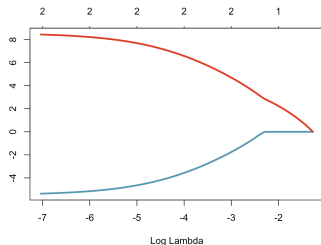
Régression Ridge avec R



Régression LASSO avec R

$y \in \{0, 1\}$ and two covariates x_1 and x_2

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



Régression LASSO avec R

