Université de Montpellier Année 2024-2025
HAX912X - Modèles linéaires généralisés

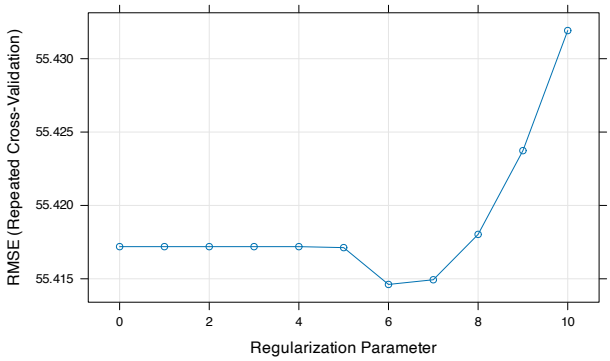
Examen de contrôle continu

Durée 1h - Sans document
Jeudi 3 octobre 2024

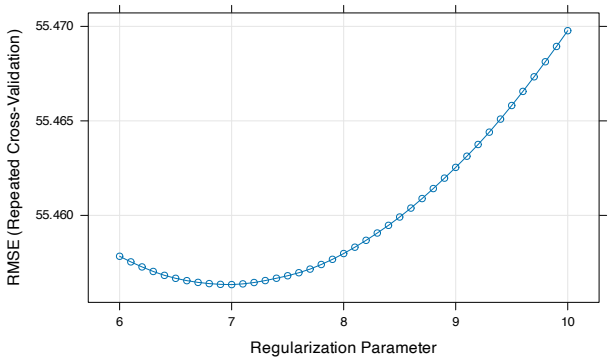
Exercice 1 (10 pts)

Décrire et commenter les résultats obtenus à l'aide du code R ci-dessous.

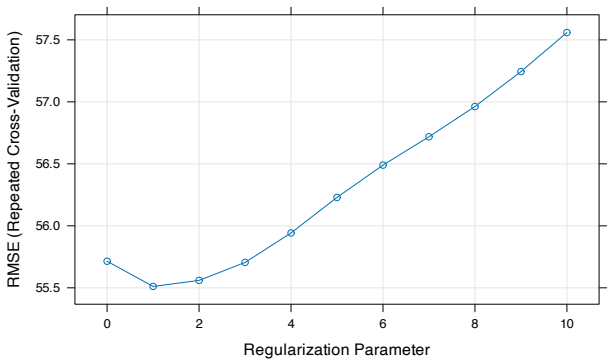
```
data(diabetes)
dim(diabetes$x)
# [1] 442  10
x <- diabetes$x
y <- diabetes$y

colMeans(diabetes$x)
#           age           sex           bmi           map            tc           ldl
# -3.587816e-16  9.321980e-17 -7.993543e-16  1.360463e-16 -9.117895e-17  1.263679e-16
#           hdl           tch           ltg           glu
# -4.505571e-16  3.834131e-16 -3.814488e-16 -3.428522e-16

diag(var(diabetes$x))
#         age         sex         bmi         map          tc         ldl         hdl
# 0.002267574 0.002267574 0.002267574 0.002267574 0.002267574 0.002267574 0.002267574
#         tch         ltg         glu
# 0.002267574 0.002267574 0.002267574

as.vector(createDataPartition(1:20, p = 0.8, list = FALSE))
# [1]  1  3  4  5  6  7  9 10 12 13 14 15 17 18 19 20

set.seed(123)
trainIndex <- createDataPartition(y, p = 0.8, list = FALSE)
x_train <- x[trainIndex, ]
x_test <- x[-trainIndex, ]
y_train <- y[trainIndex]
y_test <- y[-trainIndex]

train_control <- trainControl(method = "repeatedcv", number = 5, repeats = 10)

model1 <- train(
  x_train, y_train,
  method = "glmnet",
  tuneGrid = expand.grid(alpha = 0, lambda = seq(0, 10, by = 1)),
  trControl = train_control
)
plot(model1)
```



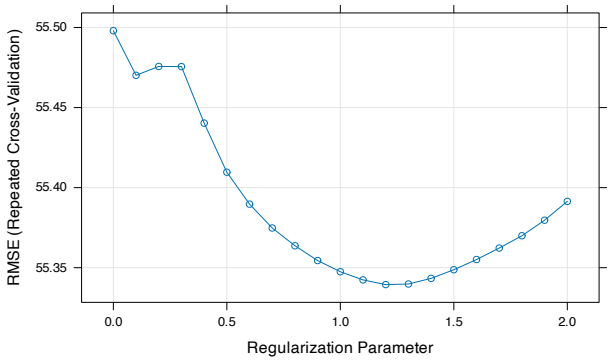

```
model1 <- train(
  x_train, y_train,
  method = "glmnet",
  tuneGrid = expand.grid(alpha = 0, lambda = seq(6, 10, by = 0.1)),
  trControl = train_control
)
plot(model1)
```




```
model1$bestTune$lambda
# [1] 7
model1$results[model1$results$lambda==model1$bestTune$lambda,]
#    alpha lambda     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
# 11     0      7 55.45634 0.4748846 45.12797 3.486439 0.07605376 3.179491

model2 <- train(
  x_train, y_train,
  method = "glmnet",
  tuneGrid = expand.grid(alpha = 1, lambda = seq(0, 10, by = 1)),
  trControl = train_control
)
plot(model2)
```




```
model2 <- train(
  x_train, y_train,
  method = "glmnet",
  tuneGrid = expand.grid(alpha = 1, lambda = seq(0, 2, by = 0.1)),
  trControl = train_control
)
plot(model2)
```




```
model2$bestTune$lambda
# [1] 1.2
model2$results[model2$results$lambda==model2$bestTune$lambda,]
#    alpha lambda     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
# 13     1    1.2 55.33944 0.4743875 45.07996 3.610067 0.07401547 3.271096

pred1 <- predict(model1, newdata = x_test)
pred2 <- predict(model2, newdata = x_test)

res1 <- sqrt(mean((pred1 - y_test)^2))
res1
# [1] 52.97107
res2 <- sqrt(mean((pred2 - y_test)^2))
res2
# [1] 53.00034
```

Correction

Cet exercice vaut un total de 10 points, chacun des points suivants vaut 2 points :

- Le jeu de données considéré comprend $n = 442$ individus et 10 prédicteurs qui sont déjà normalisés. Le jeu de données est découpé en deux parties : un échantillon d'apprentissage comprenant 80% des données et un échantillon de test.
- Un premier modèle (model 1), qui incorpore toutes les variables et utilise la régularisation Ridge, est mis en œuvre. Le paramètre de régularisation associé est sélectionné par validation croisée en 5 ensembles, répétée 100 fois, en utilisant le critère RMSE.
- Pour trouver la meilleure valeur du paramètre de régularisation, un focus est effectué sur les valeurs comprises entre 6 et 10. La meilleure valeur est 7 et le RMSE estimé est d'environ 55,46.
- Un deuxième modèle (model 2), qui incorpore toutes les variables et utilise la régularisation Lasso, est mis en œuvre. Le paramètre de régularisation associé est sélectionné par validation croisée en 5 ensembles, répétée 100 fois, en utilisant le critère RMSE. Pour trouver la meilleure valeur du paramètre de régularisation, un focus est effectué sur les valeurs comprises entre 0 et 2. La meilleure valeur est 1,2 et le RMSE estimé est d'environ 55,33.
- Comme pour la validation croisée, les résultats du modèle 1 et du modèle 2 sont très similaires sur un ensemble de données de test.

Exercice 2 (5 pts)

On considère le modèle de régression suivant, pour tout $i = 1, \dots, n$ avec $n = 2k$ et k un entier strictement positif

$$y_i = \beta_1 + \beta_2 x_i + u_i$$

avec $\mathbb{E}(u_i) = 0$, $\mathbb{V}(u_i) = 1$ si $i = 1, \dots, k$ et $\mathbb{V}(u_i) = 2$ si $i = k+1, \dots, n$, $\mathbb{C}(u_i, u_j) = 0$ si $i \neq j$.
Donner l'expression de l'estimateur des moindres carrés généralisés de β_2.

Correction

L'estimateur des moindres carrés généralisés de β_2 est solution de la minimisation de la fonctionnelle ci-dessous

$$RSS(\beta_1, \beta_2) = \sum_{i=1}^{k} (y_i - \beta_1 - \beta_2 x_i)^2 + \left(\frac{1}{2}\right) \sum_{i=k+1}^{n} (y_i - \beta_1 - \beta_2 x_i)^2 \ .$$

$RSS(\beta_1, \beta_2)$ est une fonction strictement convexe. Pour trouver le minimum, il faut donc résoudre le système de 2 équations à 2 inconnues :

$$\frac{\delta RSS}{\delta \beta_1}(\hat{\beta}_1, \hat{\beta}_2) = -2 \sum_{i=1}^{k} \left(y_i - \hat{\beta}_1 - \hat{\beta}_2 x_i\right)^2 - \sum_{i=k+1}^{n} \left(y_i - \hat{\beta}_1 - \hat{\beta}_2 x_i\right)^2 = 0 \quad \text{et}$$

$$\frac{\delta RSS}{\delta \beta_1}(\hat{\beta}_1, \hat{\beta}_2) = -2 \sum_{i=1}^{k} x_i \left(y_i - \hat{\beta}_1 - \hat{\beta}_2 x_i\right)^2 - \sum_{i=k+1}^{n} x_i \left(y_i - \hat{\beta}_1 - \hat{\beta}_2 x_i\right)^2 = 0 \, .$$

$$\frac{3n}{4} \hat{\beta}_1 + s_x \hat{\beta}_2 = s_y \quad \text{et} \quad s_x \hat{\beta}_1 + s_{x^2} \hat{\beta}_2 = s_{xy} \quad \text{avec}$$

$$s_y = \sum_{i=1}^{k} y_i + \left(\frac{1}{2}\right) \sum_{i=k+1}^{n} y_i, \ s_x = \sum_{i=1}^{k} x_i + \left(\frac{1}{2}\right) \sum_{i=k+1}^{n} x_i, \ s_{xy} = \sum_{i=1}^{k} x_i y_i + \left(\frac{1}{2}\right) \sum_{i=k+1}^{n} x_i y_i,$$

$$s_{x^2} = \sum_{i=1}^{k} x_i^2 + \left(\frac{1}{2}\right) \sum_{i=k+1}^{n} x_i^2.$$

Nous obtenons

$$\hat{\beta}_1 = \frac{4s_y}{3n} - \frac{4s_x}{3n}\hat{\beta}_2$$

et

$$s_x\left(\frac{4s_y}{3n} - \frac{4s_x}{3n}\hat{\beta}_2\right) + s_{x^2}\hat{\beta}_2 = s_{xy}$$

$$\iff \left(s_x - 3n/4\frac{s_{x^2}}{s_x}\right)\hat{\beta}_1 = s_{xy} - \frac{s_{x^2}s_y}{s_x}$$

$$\iff \hat{\beta}_2 = \left[\frac{s_{xy} - \frac{4s_x s_y}{3n}}{s_{x^2} - \frac{4s_x^2}{3n}}\right].$$

Exercice 3 (5 pts)

Expliquer en détails les résultats produits par le code R ci-dessous

```
n <- 100
x1 <- rnorm(n)
x2 <- rnorm(n)
x3 <- rnorm(n)
u <- rnorm(n)
y <- 1+1.5*x1+2*x2-0.7*x3+u
d <- 80
z <- matrix(runif(n*d),n)

app <- data.frame(y=y,x1=x1,x2=x2,x3=x3,z=z)
names(app)[-(1:4)] <- paste("z",1:80,sep="")

model1 <- lm(y~1,data=app)
model2 <- lm(y~.,data=app)
model3 <- lm(y~x1+x2+x3,data=app)

library(glmnet)
library(caret)
x <- as.matrix(app[,-1])
model4 <- glmnet(x,y,family="gaussian",nlambda=50,alpha=1)
model5 <- train(x,y,method="glmnet",metric="RMSE",
                trControl=trainControl(method="repeatedcv",
                                       number=5,repeats=100),
                tuneGrid=data.frame(alpha=1,lambda=model4$lambda))

x1test <- rnorm(n)
x2test <- rnorm(n)
x3test <- rnorm(n)
utest <- rnorm(n)
ytest <- 1+1.5*x1test+2*x2test-0.7*x3test+utest
ztest <- matrix(runif(n*d),n)

test <- data.frame(y=ytest,x1=x1test,x2=x2test,x3=x3test,z=ztest)
names(test)[-(1:4)] <- paste("z",1:80,sep="")

mean((predict(model3,test)-ytest)^2)
# 1.000278
mean((predict(model1,test)-ytest)^2)
# 10.37723
```

```
mean((predict(model2,test)-ytest)^2)
# 6.840056
mean((predict(model5,test)-ytest)^2)
# 1.217017
```

Correction

Cet exercice vaut un total de 5 points, chacun des points suivants vaut un point :

- Le modèle est $y = 1 + 1.5x_1 + 2x_2 + 0.7x_3 + u$. Il comprend 3 prédicteurs auxquels nous rajoutons 80 prédicteurs de bruit. Nous disposons de $n = 100$ observations.
- Dans le modèle 1, seule l'ordonnée à l'origine est incluse. En revanche, le modèle 2 intègre toutes les variables, y compris les variables de bruit. Le modèle 3 est le modèle Oracle qui inclut seulement les variables pertinentes.
- Le modèle 5 incorpore toutes les variables et utilise la régularisation Lasso.
- Le paramètre de régularisation Lasso est sélectionné par validation croisée à 5 ensembles, répétée 100 fois, en utilisant le critère RMSE.
- Les résultats du modèle Oracle et du modèle Lasso sont très similaires sur un ensemble de données de test.