

TP no 6 - modèles linéaires pénalisés et glmnet - solution exercice 1

Master parcours SSD - UE Apprentissage Statistique I

1 Exercice 1

Dans cet exercice nous allons illustrer l'utilisation du package `glmnet` sur un problème de classification.

Pour cela nous travaillerons sur le jeu de données **South Africa Heart Disease**, utilisé à fins illustratives dans le livre *Elements of Statistical Learning* que l'on peut télécharger sur cette page.

Nous nous limiterons ici à un problème de classification binaire, mettant donc en jeu un modèle de régression logistique, mais cette vignette illustre l'utilisation du package de manière bien plus complète.

```
## Warning: package 'knitr' was built under R version 3.6.3
```

1.1 Question 1 : charger le jeu de données

Le jeu de données est contenu dans le fichier **SAheart.data**. Notons que la première colonne du fichier contient le nom des lignes et qu'il convient de les interpréter en tant que telle.

```
##### STARTING EXERCICE 1 ####
tab = read.csv("datasets/SAheart.data", row.names = 1)
```

1.2 Question 2 : mettre en forme le jeu de données

1. extraire la variable réponse, qui est contenue dans le champ **chd**
2. transformer le descripteur qualitatif **famhist** en descripteur(s) quantitatifs
3. standardiser les descripteurs

```
# extract outcome #
#-----#
y = tab$chd
# convert to a factor
y = factor(y)

# convert famhist to numeric #
#-----#
# (NB: only two levels, so can use a single variable)
tab$famhist = as.numeric(tab$famhist) - 1

# extract matrix #
#-----#
# discard outcome
X = tab[, -which(colnames(tab) == "chd")]
# convert to matrix
X = as.matrix(X)
```

```
# scale columns
X = scale(X)
```

1.3 Question 3 : construire un modèle lasso et représenter le chemin de régularisation obtenu

On construit le modèle avec la fonction `glmnet` et on représente le chemin de régularisation obtenu avec la fonction `plot.glmnet`.

Notons que par défaut la fonction `glmnet` considère 100 valeurs du paramètre de régularisation définies automatiquement (se référer à la documentation pour davantage de précisions).

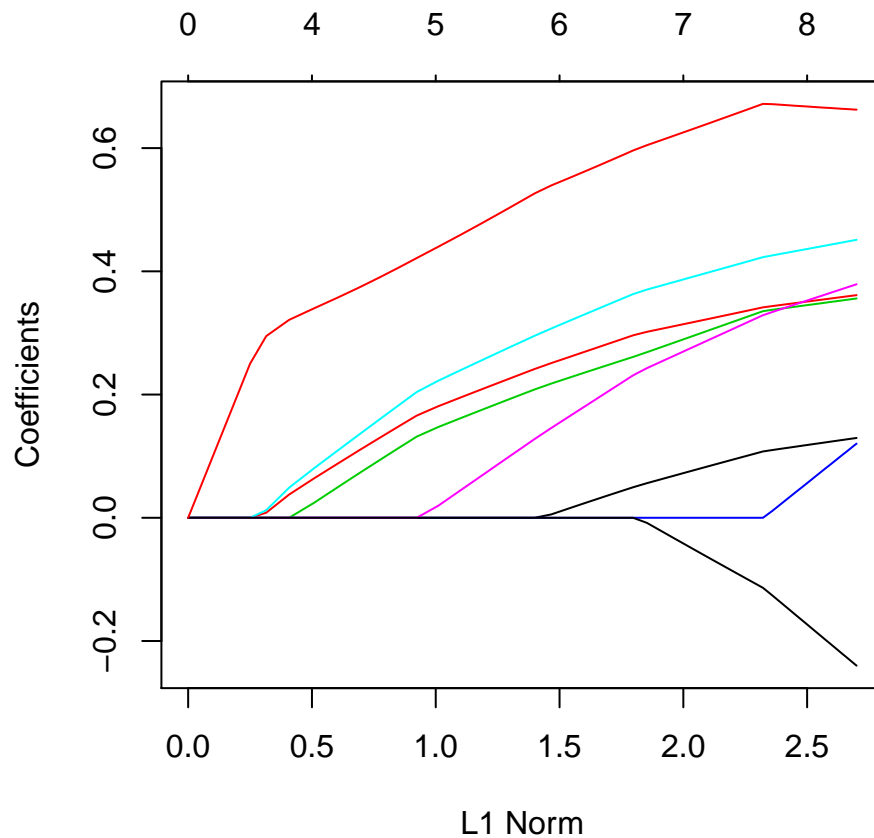
```
# load package
library(glmnet)

## Warning: package 'glmnet' was built under R version 3.6.3

## Loading required package: Matrix

## Loaded glmnet 4.0-2

# fit model
fit.lasso = glmnet(x = X, y = y, family = "binomial")
# plot
plot(fit.lasso)
```



1.4 Question 4 : faire de même pour une pénalité “ridge”.

Il suffit pour cela de modifier le paramètre α qui définit la pénalité **elastic-net**:

$$\Omega(w) = \alpha \|w\|_1 + \frac{1 - \alpha}{2} \|w\|_2^2.$$

Le paramètre α vaut par défaut 1, ce qui correspond à un modèle lasso. Se référer à la documentation pour davantage de précisions.

```
# fit model
fit.ridge = glmnet(x = X, y = y, family = "binomial", alpha = 0)
# plot
plot(fit.ridge)
```

