TD4 : Sélection de variables par pénalisation

M2 MMA - Université de Paris

2020-2021

Exercice 1 : Découverte

Importation des données

Pour cet exercice, nous travaillons sur le jeu de données `Ozone` disponible sur Moodle :

```
ozone <- read.table("Ozone.txt")
```

Ce jeu de données contient des informations concernant la pollution de l'air, notamment :

- `maxO3` : maximum de concentration d'ozone observé sur la journée en μgr/m3
- `T9`, `T12`, `T15` : Température observée à 9, 12 et 15h,
- `Ne9`, `Ne12`, `Ne15` : Nébulosité observée à 9, 12 et 15h,
- `Vx9`, `Vx12`, `Vx15` : Composante E-O du vent à 9, 12 et 15h,
- `maxO3v` : Teneur maximum en ozone observée la veille,
- `vent` : orientation du vent à 12h,
- `pluie` : occurrence ou non de précipitations.

```
dim(ozone)
```

```
## [1] 112  13
```

Questions

1. Après avoir enlevé les variables qualitatives `vent` et `pluie`, ajuster un modèle de régression linéaire multiple non pénalisée pour expliquer la concentration d'ozone `maxO3v` en fonction des autres variables. Identifier les variables significatives.
2. Centre et réduire les données puis ajuster une régression ridge en faisant varier la pénalité λ sur une grille préalablement définie. Tracer les chemins de régularistation et commenter.
3. Ajuster cette fois une régression Lasso en faisant varier λ sur une grille puis tracer le chemin de régularisation de chacune des variables.
4. Ajuster finalement une régression Elastic Net puis tracer le chemin de régularisation de chacune des variables.
5. Pour chacune des trois régressions linéaires pénalisées, calibrer la pénalité par validation croisée puis comparer les variables du modèle retenu.

Exercice 2 : Comparaison des méthodes

Pour cet exercice, on simulera trois types de jeu de données, qui nous permettront d'évaluer les avantages et les inconvénients de chacune des méthodes.

Jeu de données 1 : petit signal et beaucoup de bruit

On simulera des données vérifiant les conditions suivantes :

- $p = 5000$ et $n = 1000$,
- variables décoréleées : $\forall 1 \leq i \leq n, X_i \sim_{i.i.d} \mathcal{N}(0, I)$,
- modèle sparse : $\beta = (1, ..., 1, 0, ..., 0)^T$ (seules les 15 premières variables expliquent la réponse Y).

Jeu de données 2 : gros signal et beaucoup de bruit

On simulera des données vérifiant les conditions suivantes :

- $p = 5000$ et $n = 1000$,
- variables décoréleées : $\forall 1 \leq i \leq n, X_i \sim_{i.i.d} \mathcal{N}(0, I)$,
- modèle non sparse : $\beta = (1, ..., 1, 0, ..., 0)^T$ (les 1000 premières variables expliquent la réponse Y).

Jeu de données 3 : signal varié et variables corrélées

On simulera des données vérifiant les conditions suivantes :

- $p = 50$ et $n = 100$,
- variables coréleées : $\forall 1 \leq i, j \leq n, Cov(X_i, X_j) = (0.7)^{|i-j|}$,
- modèle varié : $\beta = (10, 10, 5, 5, rep(1, 10), rep(0, 36))^T$.

Pour chacun des 3 jeux de données, créer un échantillon d'apprentissage et un échantillon test puis répondre aux questions suivantes :

1. Mettre en place un modèle de régression Lasso, Ridge et Elastic Net défini sur l'ensemble d'apprentissage.
2. Tracer les chemins de régularisation.
3. Calculer l'erreur de prédiction sur l'ensemble test.
4. Comparer les méthodes.

Exercice 3 : Comparaison `pls` et `Lasso`

Importation des données

Pour cet exercice, nous travaillerons sur le jeu de données `Colon` disponible dans le package `plsgenomics`.

```
library(plsgenomics)
data(Colon)
```

Si l'installation du package échoue, télécharger le jeu de données depuis Moodle.

```
load("Colon.rda")
length(Colon)
```

```
## [1] 3
```

L'objet `R Colon` contient trois éléments :

- `X` des données d'expression des gènes ($n = 62$ et $p = 2000$),
- `Y` une variable binaire (deux valeurs `1` et `2`) indiquant le type de tissus d'origine (tumoral ou normal),
- `gene.names` un vecteur contenant les noms des 2000 gènes du jeu de données.

Transformer les données de la manière suivante :

```
X <- Colon$X
Y <- Colon$Y
Y <- Y-1
gene <- Colon$gene.names
Colon <- data.frame(X=I(X),Y=Y)
```

L'objectif de cet exercice est de comparer la prédiction faite à l'aide d'un modèle linéaire pénalisé avec celle faite en petite dimension suite à une PLS.

Questions

1. Créer un ensemble d'apprentissage et test permettant d'apprendre le modèle et d'en tester les capacités prédictives.
2. Appliquer la fonction `glmnet()` par défaut (avec 100 valeurs différentes valeurs de λ) pour `alpha` qui vaut 1 puis 0.5. Indiquer à quoi correspondent ces deux cas. Pour ces deux valeurs de `alpha`, tracer les trajectoires des coefficients en appliquant `plot()` au résultat. Commenter les figures.
3. En utilisant la fonction `glmnet()`, proposer un modèle de prédiction.
4. Proposer un modèle de prédiction basé sur la fonction `plsr()`.
5. Appliquer ces deux modèles et les comparer en termes de qualité de prédiction et de nombre de gènes sélectionnées sur la base d'un jeu d'apprentissage et d'un jeu test.