Apprentissage supervisé et non supervisé
TP1 : Modèles linéaires pénalisés
M2 SISE - Université Lyon 2 - 2017/2018

Responsable : Julien Ah-Pine

1 Présentation de l'étude de cas

Les données que vous devez modéliser correspondent à des biscuits non encore cuits. Il s'agit de prédire la teneur en sucre (variable à expliquer Y) à partir des données recueillies par la mesure d'un spectre d'absorbance dans le domaine proche infrarouge. En effet, il est couteux de mettre en place à dans une chaîne de production des méthodes classiques de chimie analytique afin de mesurer la composition des aliments et l'on souhaite, à la place, de prédire cette composition (la teneur en sucre notamment) à partir d'un signal (donnant 700 variables explicatives $X^1, \ldots, X^{700}$) obtenu par un appareil de mesure de spectrométrie infrarouge.

Nous avons à notre disposition 40 observations pour lesquelles nous avons la teneur en sucre et également les données du signal.

2 Lecture et decription des données

1. Charger les données contenues dans le fichier `cookie.RData` à l'aide de la commande `load`.
2. Les observations sont représentées par les lignes de la variable `D`. La première colonne de `D` contient la variable représentant la teneur en sucre tandis que les autres colonnes représentent le signal mesuré sur plusieurs dimensions. Stockez dans une variable `Y` la teneur en sucre et dans une variable `X` la matrice contenant en colonne les différentes mesures du signal.
3. Afin de connaître le rang de la matrice `X`, vous pouvez utiliser la commande `qr`[1]. Appliquez cette fonction à la variable `X` et déterminez le rang de cette matrice en accédant à la sous-liste `rank` du résultat de cette fonction.
4. Estimez par MCO le modèle linéaire qui explique Y en fonction des variables explicatives $X^1, \ldots, X^{700}$. Que constatez-vous ?

3 Problèmes de grande dimension et modèles pénalisés

Dans ce cas d'étude le nombre de variables $p = 700$ est très grand. On parle de problème de grande dimension. Les méthodes classiques sont souvent insuffisantes pour traiter ce type de données. Ici, comme $p >> n$, la matrice $\mathbf{X}$ des variables explicatives est de dimension n et non p. Donc $\mathbf{X}^\top\mathbf{X}$ n'est pas de plein rang et elle n'est donc pas inversible. Nous sommes donc dans l'impossibilité d'estimer les coefficients du modèle linéaire par les MCO.

Les modèles pénalisés permettent de surmonter cette difficulté. Il existe plusieurs librairies `R` qui mettent en oeuvre les modèles ridge, lasso et elastic-net mais nous allons nous intéresser à une librairie très générale et très performante : `glmnet`. Dans cette perspective, vous pouvez consulter en parallèle du TP la page suivante : `https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html`.

5. Installez puis chargez la librairie `glmnet`.

1. Décomposition de type QR d'une matrice que l'on peut obtenir avec le procédé d'orthogonalisation de Gram-Schmidt par exemple.

Remarque : la fonction de pénalité d'elastic-net implémentée dans cette librairie est comme suit,

$$R(\mathbf{a}) = \alpha\|\mathbf{a}\|_{\ell_1} + \frac{1}{2}(1-\alpha)\|\mathbf{a}\|_{\ell_2}^2$$

4 Chemin de solutions

Rappelons qu'un modèle pénalisé dépend d'un paramètre λ qui pondère la fonction de pénalité par rapport à la fonction objectf. Dans le cas d'un problème de régression nous avons :

$$\arg\min_{\mathbf{a}\in\mathbb{R}^p} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{p} a_j x_{ij} \right)^2 + \lambda R(\mathbf{a})$$

Nous avons discuté en cours de plusieurs façon d'estimer le paramètre λ. En fait, il existe des méthodes permettant d'estimer efficacement les coefficients pour toutes les valeurs λ ! On parle de chemin de solutions ("regularization path").

6. Entrez, exécutez et commentez les commandes suivantes :

```
reg_mult_ridge=glmnet(x=X,y=Y,family = "gaussian", alpha=0)
plot(reg_mult_ridge)
reg_mult_ridge
```

7. La variable `reg_mult_ridge` est une liste contenant les informations pertinentes pour le chemin de solutions. Parcourez les différentes sous-listes et identifiez les différentes informations importantes.

5 Validation croisée

Pour une meilleure estimation de l'erreur en généralisation, il est indispensable de faire de la validation croisée.

8. Entrez, exécutez et commentez les commandes suivantes :

```
cv_ridge_fit=cv.glmnet(x=X,y=Y,family = "gaussian", alpha=0, nfolds = 5)
plot(cv_ridge_fit)
cv_ridge_fit$lambda.min
coef(cv_ridge_fit,s = "lambda.min")
cv_ridge_fit$lambda.1se
coef(cv_ridge_fit,s = "lambda.1se")
```

6 Régression lasso et elastic-net

9. Sur le même jeu de données, utilisez les commande précédentes pour estimer les coefficients et analyser les résultats du modèle lasso. Comparez les résultats avec ceux du modèle ridge.
10. Mêmes questions avec le modèle elasticnet en prenant $\alpha = 1/2$.