Mini projet à remettre sur e-campus avant le 12 mai 2021 minuit

Consignes

Le mini-projet donne lieu à un compte-rendu *rédigé* à effectuer en *binôme*.

- Ne pas oublier de définir un titre, une introduction pour préciser la problématique étudiée et le plan du travail, et une conclusion.
- Commenter les résultats obtenus, inclure les graphiques pertinents dans le corps du texte.
- Les résultats doivent être justifiés. La notation prendra en compte la clarté et le soin de la rédaction.
- Déposer sous e-campus un compte rendu `pdf` (qui peut être manuscrit puis photographié ou scanné) et un fichier texte `.R` contenant les commandes. Les fichiers seront nommés avec les noms du binôme: `NOM1-NOM2.pdf` et `NOM1-NOM2.R`.
- Aucun retard ne sera admis.

Introduction

Le jeu de données `cookie` est un jeu de données classique en *machine learning*. Il est disponible sur le site du cours sous la forme de deux fichiers, l'un contenant les données d'apprentissage `cookie.app.RData` (40 observations), l'autre des données de test `cookie.val.RData` (32 observations). Il a été originellement fourni par Osbone et al. (1984)[1], et de nombreuses publications le prennent en référence, par exemple wikistat[2] qui décrit précisément la problématique. L'objectif est d'utiliser des mesures de spectrométrie infrarouge (une observation est un spectre mesuré sur 700 fréquences) pour déterminer la teneur en sucre sans avoir à procéder à des analyses chimiques coûteuses.

1 Un peu de théorie

Le nombre de spectres n étant très largement inférieur au nombre de fréquences p, cette étude rentre dans le cadre des données dites de *grande dimension*.

1. Que se passe-t-il lorsqu'on modélise une régression linéaire: $Y = \theta_0 \mathbb{1}_n + X\theta + \varepsilon$, $Y \in \mathbb{R}^n$, $\varepsilon \in \mathbb{R}^n$, $\theta \in \mathbb{R}^p$, $\theta_0 \in \mathbb{R}$?

   Rappeler le cadre de la régression ridge et indiquer son atout dans cette situation.

2. On décide de ne pas pénaliser l'intercept. Calculer l'estimateur de coefficients.

   Ecrire aussi la relation entre la paramétrisation $\tilde{\theta}$ quand les variables explicatives ont été préalablement centrées et θ quand elles ne l'ont pas été.

[1] B. G. Osborne, T. Fearn, A. R. Miller et S. Douglas, *Application of Near Infrared Reflectance spectroscopy to the compositional analysis of biscuits and biscuit doughs*, J. Sci. Food Agric. 35 (1984), 99 - 105

[2] `https://www.math.univ-toulouse.fr/~besse/Wikistat/pdf/st-scenar-app-cookie.pdf`

3. Soit X une matrice de dimension $n \times p$. L'objectif de cette question est de trouver la limite de la matrice $A_\lambda = (X'X + \lambda Id_p)^{-1}X'$ quand λ tend vers 0 dans le cas où X n'est pas injective.

   Soit $\sum_{j=1}^{r} \sigma_j u_j v_j'$ une décomposition en valeurs singulières de la matrice X, où $r = rang(X)$, σ_j^2 sont les valeurs propres non nulles de la matrice $X'X$, $\{u_j\}$ et $\{v_j\}$ sont deux familles orthonormales de $\mathbb{R}^n$ et $\mathbb{R}^p$ telles que: $XX'u_j = \sigma_j^2 u_j$ et $X'Xv_j = \sigma_j^2 v_j$ [pour les curieux.ses, voir la question *bonus* en fin d'énoncé].

   Montrer que $X'X = \sum_{j=1}^{r} \sigma_j^2 v_j v_j'$; en déduire que

$$(X'X + \lambda Id)^{-1} = \sum_{j=1}^{r} \frac{1}{\sigma_j^2 + \lambda} v_j v_j'$$

   puis la limite A_0 de A_λ quand λ tend vers 0.

2 Analyse exploratoire

Afin de prendre contact avec le jeu de données, on commence par une analyse exploratoire.

1. Lire les données: créer les matrices `xtrain` et `xtest` contenant respectivement les variables explicatives du jeu d'apprentissage et du jeu de test, puis les vecteurs `ytrain` et `ytest` pour les réponses (teneur en sucre).

   Tracer les boxplots des variables explicatives, puis les "courbes" (`matplot` avec `type='l'`) des spectres pour chaque observation du jeu d'apprentissage. Etudier la corrélation entre les mesures aux différentes fréquences. Commenter.

2. A l'aide du package `FactoMineR`, effectuer une ACP, tracer le graphe des valeurs propres, commenter leur nombre. Représenter les nuages dans les six premiers axes principaux et commenter.

3. Coder la fonction `reconstruct(res,nr,Xm,Xsd)` qui reconstruit le nuage suivant les `nr` premiers axes de l'ACP `res`, avec le vecteur des moyennes `Xm` et des écarts-types `Xsd` des variables explicatives.

   Vérifier votre code en comparant la reconstruction totale du nuage avec `xtrain`, puis représenter sur une feuille partagée en six la reconstruction pour $nr = 1, \ldots, 5, 39$, en faisant afficher dans le titre l'erreur quadratique moyenne (RMSE) et l'erreur en valeur absolue (MAE).

   Représenter sur un autre graphe les courbes à ces six niveaux pour la variable `X24`.

   Commenter.

3 Régression pénalisée

1. Estimer le modèle de régression *ridge* avec la fonction `glmnet` et la grille de paramètres `grid=10^seq(6,-10,length=100)`. Commenter la variation de la valeur estimée du paramètre d'`intercept`.

   Le recalculer en fonction des estimées des autres paramètres suivant la formule de la section 1.2.

   Quelles modifications quand on centre `ytrain`, ou les variables de `xtrain`, ou les deux?

   Dans le cas où `ytrain` et les variables de `xtrain` sont centrées et réduites, utiliser la question 1-3 pour retrouver l'estimation de θ_λ quand λ tend vers 0.

2. Utiliser maintenant la fonction `lm.ridge` qui est mentionnée dans la fiche wikistat sur `xtrain`, `ytrain` avec la gille `grid`. Retrouvez-vous les mêmes coefficients? Commenter:

- en écrivant directement le code de calcul de l'estimateur ridge et en le comparant avec les sorties des fonctions R dédiées
- en vous aidant de la vignette écrite par les auteurs de `glmnet`[3].

3. Affiner l'étendue de la grille pour éviter les calculs inutiles, puis optimiser la valeur du paramètre de régularisation par validation en quatre plis sur l'échantillon d'apprentissage: définir le germe du générateur aléatoire, utiliser la fonction `cvsegments`, calculer l'erreur sur chaque pli et pour chaque valeur du paramètre.

   Représenter pour chaque valeur du paramètre l'erreur moyenne et un intervalle de confiance de cette erreur représenté sous forme d'un segment.

   Comparer avec la représentation des résultats de la fonction `cv.glmnet`.

   Choisir le paramètre qui vous semble optimum, réajuster sur la totalité du jeu d'apprentissage, puis calculer l'erreur de généralisation.

Bonus La fiche wikistat propose d'utiliser le package `caret`. Montrer que l'on peut aussi appliquer ce cadre à la fonction `glmnet`. Comparer les modèles de régression ridge et lasso, en vous posant la question des hypothèses faites par ces fonctions dans ces deux cas.

4 Régression logistique pénalisée

Ce n'est pas tant la teneur en sucre, que le fait qu'elle dépasse le seuil de 18 qui doit être étudié.

1. Rappeler les hypothèses de la régression logistique, puis créer les variables `z` et `ztest` à prédire. Les jeux d'apprentissage et de test sont-ils équilibrés?
2. Utiliser la fonction `cv.glmnet` pour estimer la régression logistique pénalisée en ridge et en lasso.
3. Comparer les résultats, et tracer les courbes ROC calculées en apprentissage et en test pour les modèles retenus en ridge et en lasso. Peut-on tester leur adéquation?

Bonus pour les curieux.ses

Montrer la formule de la décomposition en valeurs singulières de X, une matrice $n \times p$ de rang r.

1. Montrer qu'il existe une famille orthonormée de vecteurs $\{u_j\}$ et des scalaires $\lambda_j > 0$ tels que $XX' = \sum_{j=1}^{r} \lambda_j u_j u_j'$ et préciser la valeur de r.
2. Montrer que $\sum_{j=1}^{r} u_j u_j'$ est une matrice de projection de $Im(XX')$, puis que les projections sur $Im(X)$ et sur $Im(XX')$ sont identiques.
3. On définit v_j par $v_j = \lambda_j^{-1/2} X' u_j$ pour $j = 1, \ldots, r$. Montrer que les v_j sont normés, puis que la famille $\{v_j\}$ est une famille orthonormée de vecteurs propres de $X'X$.

   En déduire qu'il existe r scalaires strictement positifs σ_j tels que $X = \sum_{i=1}^{r} \sigma_j u_j v_j'$.

[3] `https://web.stanford.edu/~hastie/glmnet/glmnet_alpha.html`