Université d'Angers
Département de Mathématiques

Année 2019-2020
M2 Data Science

Statistique en Grande Dimension et Apprentissage - TP Chapitre 3

Ce TP a pour but d'illustrer le chapitre 3 principalement sur la régression pénalisée.

Exercice 1 (Exemple simulé)**.**

1. Ecrire une fonction de n, p, θ et σ permettant de générer le vecteur $\mathbf{Y}$ de taille n donné par

$$\mathbf{Y} = \mathbf{X}\theta^\star + \varepsilon$$

où $\mathbf{X}$ est une matrice $n \times p$ dont les coordonnées sont i.i.d de loi $\mathcal{N}(0,1)$ et ε est un vecteur de taille n *i.i.d.* de loi $\mathcal{N}(0,\sigma^2)$.

2. On fixe $n = 100$,

$$\theta^\star = \left(\underbrace{1,\dots,1}_{s/2 \text{ fois}}, \underbrace{-1,\dots,-1}_{s/2 \text{ fois}}, \underbrace{0,\dots,0}_{p-s \text{ fois}} \right).$$

$p = 200$ et $\sigma = 0.5$. Créez un objet LASSO_ex1 issu de l'application de la fonction `glmnet` à ce cadre avec une pénalisation LASSO.

3. Affichez les résultats et commentez-les dans les deux cas.
4. Tracez le chemin de régularisation (apparition des coefficients) via la commande `plot(LASSO_ex1)`.
5. Utilisez la fonction `coef` pour une ou plusieurs valeurs pour évaluer l'estimation de θ.
6. Testez la fonction `predict` : exemple. `predict(LASSO_t,newx=nx,s=c(0.1,0.05))`.
7. Choix du λ : à l'aide de la fonction `cv.glmnet`, on peut effectuer une validation croisée pour différentes de valeurs de λ. Créez un objet `cv_Ex1` issue de l'application de cette fonction. La commande `plot(cv_Ex1)` permet de représenter l'évolution de l'erreur de prédiction de validation croisée en fonction de $\log(\lambda)$. Que représentent $\lambda.min$ et $\lambda.1se$?
8. On souhaite évaluer la qualité de l'erreur sur un échantillon test.
   (a) Générez un échantillon test.
   (b) A l'aide de la fonction `predict`, calculez la prédiction pour chaque noyau X_i simulé : `predict(cv_Ex1glmnet.fit, ValSetX, s=bestLambdaLasso))`.
   (c) Que vaut la MSE empirique sur cet échantillon ?
   (d) Comparez le R^2 obtenu via le LASSO avec le λ sélectionné par validation croisée et celui obtenu via le modèle linéaire classique en conservant uniquement les variables sélectionnées (dans le cas $p = 200$).
9. Reproduire certains des tests précédents en changeant la valeur de θ et la valeur de σ ?
10. Faire le même type de tests avec le Ridge.

Exercice 2 (Classification et Leucémie)**.** Cet exercice est tiré d'une école d'été bioinformatique qui se déroule sur Angers/Nantes tous les ans.

1. Téléchargez les jeux de données `leukemia_big.csv` et `leukemia_small.csv`. Chargez la base `leukemia_small.csv` sur `R`.
2. Les colonnes indiquent les patients ALL (acute lymphoplastic) ou AML (acute myeloid). La matrice regroupe les *log-expressions* de $p = 3571$ gènes (resp. $p = 7128$) gènes par individu pour $n = 72$ individus.
3. Créez un vecteur réponse Y constitué de 0 si AML et 1 si ALL. Transposez la matrice de log-expressions pour obtenir une matrice X de taille $n \times p$.
4. Stat. descriptive.
   (a) Tracez un histogramme de la matrice X. Que constatez-vous ?
   (b) Refaites-le pour un gène pris au hasard.
   (c) Calculez le vecteur des moyennes par colonne de X : `apply(X, 2, mean)`. Refaites-le même test par classe.
   (d) Normalité : prendre une colonne au hasard et tracer la droite de Henry associée (`qqnorm(X[,j])` ,`qqline(X[,j])`). Conclusions ? Testez à nouveau en conditionnant aux classes.
5. ACP (... pour voir !) :
   (a)
   ```
   PCALeuk = prcomp(X, center = TRUE, scale = TRUE)
   summary(PCALeuk)
   plot(PCALeuk$x[, 1], PCALeuk$x[, 2], pch = 19, xlab = "Pr Comp 1", ylab
   = "Pr Comp 2", col = 2+Y)
   text(PCALeuk$x[, 1], PCALeuk$x[, 2], labels=rownames(PCALeuk$x), cex=0.7,
   pos = 3, col = 2+Y)
   ```
   Conclusions ?
   (b) Testez à nouveau avec les première et troisième directions de l'ACP puis avec les deuxième et troisième directions de l'ACP.
   (c) Si vous souhaitiez créer une règle de décision à partir de l'ACP ci-dessus, comment procéderiez-vous (voir suite du cours sur la classification linéaire et non linéaire) ?
   (d) Comment testeriez-vous sa pertinence ?

   Du point de vue biologique, l'ACP n'est pas complètement satisfaisante car elle fait potentiellement intervenir toutes les composantes sur une seule direction (Par ailleurs, l'ACP n'est en général pas robuste à la grande dimension...). Une alternative est d'utiliser une ACP sparse. Néanmoins, ici, on regarde un problème supervisé donc il est plus naturel de faire appel à des méthodes supervisées. On peut penser à la LDA ou la SVM (cf suite du cours). Ici, on va tester la régression logistique pénalisée.

   Rappel : La régression logistique fait partie des généralisations du modèles linéaire où "pour faire simple", on fait l'hypothèse que $\mathcal{L}(Y|X)$ appartient à une famille paramétrique de lois ayant une forme "ressemblant" à celle du modèle linéaire. Dans le cas de la

régression logistique on fait l'hypothèse que $\mathcal{L}(Y|X = x)$ suit une loi de Bernoulli de paramètre

$$\pi(x, \theta) = \frac{e^{\langle \theta, x \rangle}}{1 + e^{\langle \theta, x \rangle}}.$$

N.B. Comme d'habitude, on fait ici abstraction du terme constant pour simplifier (quitte à supposer que la première coordonnée de x est 1). Le vrai modèle fait intervenir $\theta_0 + \langle \theta, x \rangle$.

Comme dans le cas du modèle linéaire classique, on choisit alors θ par maximisation de la log-vraisemblance du modèle (qui dans le cas du modèle linéaire classique, s'apparente à la minimisation des moindres carrés). Ici,

$$L(x_1, Y_1, \ldots, x_n, \theta) = \prod_{i=1}^{n} \pi(x_i, \theta)^{Y_i} (1 - \pi(x_i, \theta))^{1-Y_i}$$

et

$$\mathcal{L}(\mathbf{x}, \mathbf{Y}, \theta) = \log L(x_1, Y_1, \ldots, x_n, \theta) = \sum_{i=1}^{n} Y_i \log(\pi(x_i, \theta) + (1 - Y_i) \log(1 - \pi(x_i, \theta)).$$

On maximise ensuite cette fonction de θ par une méthode d'optimisation numérique (déterministe ou stochastique). Comme dans le cas du modèle linéaire, on peut/doit envisager de pénaliser ce problème lorsque l'on est en grande dimension. En général, le problème prend la forme suivante : on cherche

$$\hat{\theta}_\lambda := \text{Argmin}_\theta \{ -\frac{1}{n} \mathcal{L}(\mathbf{x}, \mathbf{Y}, \theta) + \lambda \|\theta\|_r \}$$

avec $r = 1$ ou 2 (LASSO ou Ridge) ou encore

$$\text{Argmin}_{\{\theta, \|\theta\|_r \leq s\}} \{ -\mathcal{L}(\mathbf{x}, \mathbf{Y}, \theta) \}.$$

Là encore, on peut envisager des extensions en modifiant la fonction de pénalité (type Elastic Net). On admet ici que ce type de méthode peut être mis en place en pratique dans ce cadre, en particulier, que des méthodes numériques efficaces permettent de calculer ces optimiseurs. On souhaite ici mettre la méthode à exécution sur `R`.

(a) Testez le modèle non pénalisé :

```
classic_glm=glm(Y ~ X-1, family = binomial(link = "logit"))
```

`# X-1 signifie ‘‘sans terme constant’’`. Qu'en pensez-vous ?

(b) Partagez la base de données en une base d'entraînement de taille $\frac{2n}{3} = 48$ et une base de validation de taille $\frac{n}{3} = 24$ pour tester la qualité du modèle. Pour rappel, on génère ces deux-sous échantillons de manière aléatoire.

`IndTrain=sample`$(1:n)[1:(2*n/3)]$`; IndVal=setdiff(1:n,IndTrain).`

(c) A l'aide de la fonction `glmnet` (option “binomial”), créez un objet `TrainLasso`, résultat de la régression logistique avec pénalisation L_1 (LASSO).

(d) Même question avec Ridge et Elastic Net (pour alpha=$0.25, 0.5, 0.75$).

(e) Faites un `summary(TrainLasso)`. Tentez d'analyser les résultats obtenus. Vous pourrez vous aider du site de T. Hastie dédié à `glmnet`.

(`https://web.stanford.edu/~hastie/glmnet/`)

(f) Affichez le graphe des coefficients ("chemin de régularisation").

(g) Etudiez l'évolution de l'erreur de validation croisée en fonction de λ :

```
LambdaLasso = cv.glmnet(TrainSetX, TrainSetY, family="binomial",
                        type.measure="class", alpha=1).
```

(h) Affichez $\hat{\theta}_\lambda$ pour la meilleure valeur de λ estimée par validation croisée. Relancez le programme. Le résultat est-il similaire ?

(i) Prédire l'échantillon test avec le meilleur choix de λ. Quel résultat obtenez-vous ?

(j) Comparez avec Ridge/Elastic Net.

(k) Considérez la base de données `leukemia_big` et procédez à des tests similaires. Comparez vos résultats.

Exercice 3. Pour compléter, testez le modèle de régression multinômiale sur les données `handdigits` accessibles sur Moodle (Cela signifie de refaire toute la procédure).