# Apprentissage en grande dimension (Partie #3)

Thomas Verdebout

Université de Lille

# Contenu du cours

1. Introduction.
2. Apprentissage supervisé: régression.
3. Apprentissage supervisé: classification.
4. Apprentissage non-supervisé

# 2: Apprentissage supervisé: régression (Ridge et Lasso).

De nos jours, les données à haute dimension sont très présentes en économie, en génomique, en imagerie biomédicale et en finance.

Nous avons ici à l'esprit des situations où la matrice de covariables $\mathbf{X}$ est une matrice $n \times p$ avec $p$ aussi grand que $n$ ou plus grand que $n$.

Dans le modèle de régression linéaire, tout le mécanisme des moindres carrés vu précédemment ne fonctionne pas bien dans ces situations.

On se souvient que le modèle linéaire considéré jusqu'ici est de la forme

$$\mathbf{Y} = \mathbf{X}\beta + \varepsilon,$$

où $\varepsilon \sim (\mathbf{0}, \sigma^2 \mathbf{I}_n)$. L'estimateur des moindres carrés pour $\beta = (\beta_1, \ldots, \beta_p)$ est donné par

$$\begin{aligned} \hat{\beta} &= \operatorname{argim}_{\beta \in \mathbb{R}^p} \|\mathbf{Y} - \mathbf{X}\beta\|^2 \\ &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}, \end{aligned}$$

c.f. les leçons précédentes.

Quand $p > n$ la matrice $\mathbf{X}'\mathbf{X}$ est non-inversible.

Il y a plusieurs propositions dans ces situations pour estimer $\beta$.

Une alternative est la régression "ridge" (Hoerl and Kennard, 1970), qui remplace la somme des carrés des résidus par une version pénalisée

$$\|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|^2,$$

où $\lambda > 0$ est une pénalité qui contrôle la taille (en norme $L_2$) de $\beta$. On l'appelle parfois paramètre de rétrécissement (shrinkage penalty or parameter).

Le second terme ci-dessus est clairement très petit si $\beta$ est proche de zéro.

La solution ridge est donnée par

$$\hat{\beta}_{\text{ridge}}(\lambda) := (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}_p)^{-1}\mathbf{X}'\mathbf{Y},$$

qui donc ajoute $\lambda$ aux éléments diagonaux de $\mathbf{X}'\mathbf{X}$ pour ensuite calculer l'inverse.

On a que

$$\begin{aligned} \mathrm{E}[\hat{\boldsymbol{\beta}}_{\mathrm{ridge}}(\lambda)] &= \mathrm{E}[(\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}_p)^{-1}\mathbf{X}'(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon})] \\ &= (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}_p)^{-1}\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \end{aligned}$$

et donc que le biais tend vers zéro quand $\lambda \to 0$.

En posant $\mathbf{W}_\lambda := (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}_p)^{-1}\mathbf{X}'\mathbf{X}$, on a que

$$\begin{aligned} \mathbf{W}_\lambda \hat{\boldsymbol{\beta}} &= \mathbf{W}_\lambda (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y} \\ &= (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}_p)^{-1}\mathbf{X}'\mathbf{Y} = \hat{\boldsymbol{\beta}}_{\mathrm{ridge}}(\lambda), \end{aligned}$$

de sorte que

$$
\begin{aligned}
\mathrm{Var}[\hat{\beta}_{\mathrm{ridge}}(\lambda)] &= \mathbf{W}_\lambda \mathrm{Var}[\hat{\beta}]\mathbf{W}'_\lambda \\
&= \sigma^2 \mathbf{W}_\lambda (\mathbf{X}'\mathbf{X})^{-1}\mathbf{W}'_\lambda \\
&= \sigma^2 (\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}_p)^{-1}\mathbf{X}'\mathbf{X}(\mathbf{X}'\mathbf{X} + \lambda \mathbf{I}_p)^{-1},
\end{aligned}
$$

Donc lorsque $\lambda \to \infty$, la variance de $\hat{\beta}_{\mathrm{ridge}}(\lambda)$ converge vers 0. Il y a un donc un "trade-off" entre biais et variance pour le choix de $\lambda$.

On peut comparer les variances/MSE's de $\hat{\beta}$ et $\hat{\beta}_{\text{ridge}}(\lambda)$. Pour simplifier un peu les choses, on suppose ici que $\mathbf{X}$ est semi-orthogonale (design orthogonal); ce qui signifie que $\mathbf{X}'\mathbf{X} = \mathbf{I}_p$. Dans cette situation, nous avons que

$$\text{Var}[\hat{\beta}] = \sigma^2 \mathbf{I}_p$$

et

$$\text{Var}[\hat{\beta}_{\text{ridge}}(\lambda)] = \frac{\sigma^2}{(1+\lambda)^2} \mathbf{I}_p,$$

ce qui signifie que lorsque $\lambda > 0$, $\hat{\beta}_{\text{ridge}}(\lambda)$ a une plus petite variance. Attention, $\hat{\beta}_{\text{ridge}}(\lambda)$ est biaisé!

La question naturelle est alors, que donne la MSE? De façon générale, la MSE d'un estimateur $\hat{\boldsymbol{\theta}}$ de $\boldsymbol{\theta}$ est donnée par $\mathrm{E}_{\boldsymbol{\theta}}[\|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|^2]$. Dans le design orthogonal,

$$\mathrm{E}[\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2] \quad = \quad \mathrm{E}[\|\mathbf{X}'\boldsymbol{\varepsilon}\|^2] = \sigma^2 p$$

et

$$\begin{aligned}
\mathrm{E}[\|\hat{\boldsymbol{\beta}}_{\mathrm{ridge}}(\lambda) - \boldsymbol{\beta}\|^2] &= \mathrm{E}[\|(1+\lambda)^{-1}(\boldsymbol{\beta} + \mathbf{X}'\boldsymbol{\varepsilon}) - \boldsymbol{\beta}\|^2] \\
&= (1+\lambda)^{-2}\mathrm{E}[(-\lambda\boldsymbol{\beta} + \mathbf{X}'\boldsymbol{\varepsilon})'(-\lambda\boldsymbol{\beta} + \mathbf{X}'\boldsymbol{\varepsilon})] \\
&= \frac{\lambda^2}{(1+\lambda)^2}\boldsymbol{\beta}'\boldsymbol{\beta} + \frac{\sigma^2 p}{(1+\lambda)^2}.
\end{aligned}$$

Exercice: montrer qu'un minimum en $\lambda$ est obtenu pour $\lambda = \sigma^2 p / \beta' \beta$.

Dans le cas général, on peut montrer qu'il existe un $\lambda$ tel que $MSE(\hat{\beta}_{\text{ridge}}(\lambda)) < MSE(\hat{\beta})$.

La ridge fonctionne plutôt bien en présence de multicollinéarité et lorsque $p$ n'est pas trop grand.

En régression et plus particulièrement en grande dimension, l'intérêt est la sélection des variables importantes et la prédiction.

En grande dimension, la régression ridge a progressivement perdu en popularité en faveur de la méthode Lasso dont le terme de pénalité est de la form $\sum_{j=1}^{p} |\beta_j|$ et force donc l'estimation des petits $\beta_j$ à être zéro.

Par rapport à la pénalité ridge, elle minimise la somme des carrés des résidus sous une contrainte sur la somme des valeurs absolues des coefficients de régression

$$\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \sum_{j=1}^{p} |\beta_j|,$$

qui est la forme Lagrangienne de

$$\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2$$

sujet à $\sum_{j=1}^{p} |\beta_j| < c$. Pour chaque valeur de $\lambda$, il y a un "$c$" qui lui correspond. Un gros $c$ correspond à un $\lambda$ très petit. Dans la suite on utilise la notation $\sum_{j=1}^{p} |\beta_j| =: \|\boldsymbol{\beta}\|_1$.

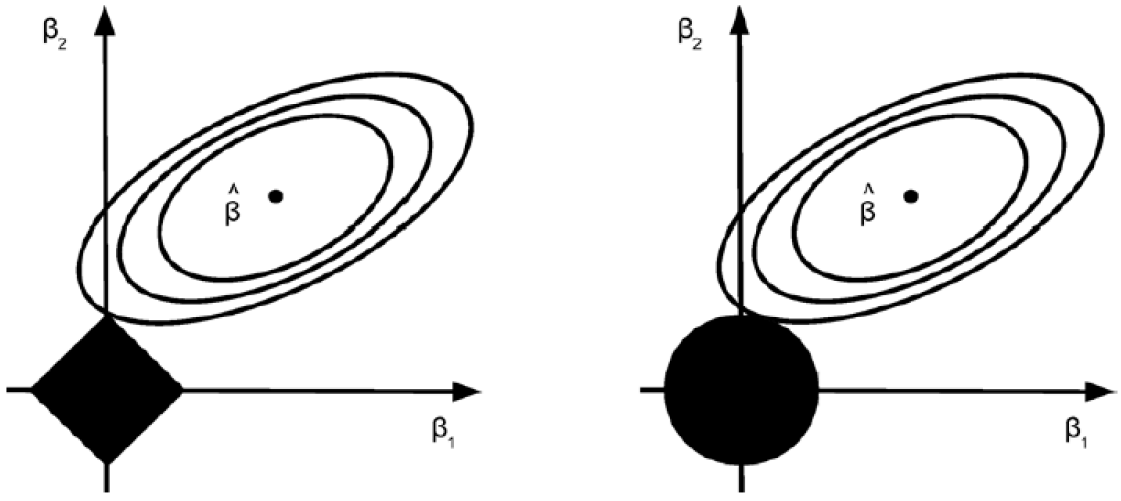


Figure: Contour lines of the residual sum of squares $\|\mathbf{X}(\hat{\beta} - \beta)\|^2$ with $l_1$-balls on the left and $l_2$-balls on the right.

Prenons une situation très simple avec $n = p = 1$, une observation $y$ et $X = 1$. On obtient dans ce cas, respectivement le régresseur ridge avec

$$(y - \beta)^2 + \lambda\beta^2$$

et Lasso avec

$$(y - \beta)^2 + \lambda|\beta|.$$

Ridge est donc solution en $\beta$ de

$$-(y - \beta) + \lambda\beta = 0,$$

et donc $\hat{\beta}_{\text{Ridge}}(\lambda) = \frac{y}{1+\lambda}$.

Pour Lasso, nous devons donc résoudre

$$-(y-\beta)+(\lambda/2)\text{sign}(\beta)=0.$$

ou de façon équivalente

$$\beta = y-(\lambda/2)\text{sign}(\beta).$$

- si $y > (\lambda/2)$, $\beta > 0$ et donc $\hat{\beta}_{\text{Lasso}}(\lambda) = y-(\lambda/2)$.
- si $y < -(\lambda/2)$, $\beta < 0$ et donc $\hat{\beta}_{\text{Lasso}}(\lambda) = y+(\lambda/2)$
- si $|y| < (\lambda/2)$, on a clairement que $\hat{\beta}_{\text{Lasso}}(\lambda) = 0$.

Considérons un modèle de régression sans "intercept" et $p$ covariables. L'ensemble des entiers

$$S_0 := \{j : \beta_j \neq 0,\ j = 1, \ldots, p\}$$

représente les indices des coefficients de régression non nuls. On appelle généralement cet ensemble l'"active set". Le nombre $s_0 = \mathrm{card}(S_0)$ est appelé l'indice de sparsité (*sparsity index*) de $\beta$.

Pour un estimateur $\hat{\beta}$ de $\beta$ on peut définir la version empirique de $S_0$ par

$$\hat{S}_{\hat{\beta}} := \{j : \hat{\beta}_j \neq 0,\ j = 1, \ldots, p\}.$$

Meinshausen and Buhlmann (2006) montrent que si (ici $p = p_n$)

$$\inf_{j \in S_0^c} |\beta_j| > \sqrt{s_0 \log(p_n)/n}$$

et si $\lambda > \sqrt{\log(p_n)/n}$,

$$\mathrm{P}[\hat{S}_{\hat{\beta}_{\mathrm{Lasso}}} = S_0] \to 1$$

quand $n \to \infty$.

La mise en oeuvre de la regression ridge et du lasso nécessite un procédé de sélection d'une valeur du paramètre $\lambda$.

La validation croisée est une méthode très simple pour sélectionner $\lambda$.

Algorithme:

step 1: Partitioner l'échantillon en deux parties: l'échantillon d'entrainement d'un côté, l'échantillon test (ou de validation) de l'autre.

step 2: Sélectionner une grille (assez fine) de valeurs de $\lambda$

step 3: Pour chaque valeur de $\lambda$, calculer $\hat{\beta}(\lambda)$ avec l'échantillon d'entrainement uniquement

step 4: Calculer pour chaque estimateur (chaque valeur de $\lambda$) l'erreur

$$\mathrm{err}(\lambda) := \sum_{i \in \mathrm{test}} (Y_i - \mathbf{X}_i' \hat{\beta}(\lambda))^2$$

step 5: Sélectionner le $\lambda$ avec $\mathrm{err}(\lambda)$ minimal.

La validation croisée est une méthode simple et facile à mettre en œuvre. Mais elle présente deux inconvénients potentiels:

- Les erreurs $\mathrm{err}(\lambda)$ sont clairement fonction des observations incluses dans l'ensemble d'apprentissage et des observations incluses dans l'ensemble de validation.
- Seul un sous-ensemble d'observations - celles qui sont incluses dans l'ensemble d'apprentissage plutôt que dans l'ensemble de validation - est utilisé pour ajuster le modèle. Puisque les méthodes statistiques ont tendance à être moins performantes lorsqu'elles sont entraînées sur un nombre réduit d'observations, cela suggère que le taux d'erreur de l'ensemble de validation peut avoir tendance à surestimer le taux d'erreur du test pour le modèle ajusté sur l'ensemble des données.

Il existe d'autres méthodes:

- Méthode "leave one out": on entraîne avec toutes les observations sauf une et on calcule l'erreur de prédiction avec l'observation laissée de côté. On répète l'opération pour chaque observation et on agrège.
- Validation croisée $k$-fold: l'échantillon est séparé en $k$ groupes. $k-1$ groupes sont utilisés pour l'entrainement et 1 groupe pour le test. On répète l'opération $k$ fois (chaque groupe joue à son tour le rôle de groupe de validation)

Elastic Net: c'est une combinaison de Ridge et Lasso, elle améliore ces méthodes en particulier dans le cas d'une forte corrélation des variables explicatives: chercher le $\beta$ qui minimise

$$\|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2, \quad \lambda_1, \lambda_2 > 0.$$

## Ridge avec R:

```
library(glmnet)
grid <- 10^seq(10, -2, length = 100)
ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid)
dim(coef(ridge.mod))
ridge.mod$lambda[50]
coef(ridge.mod)[, 50]
sqrt(sum(coef(ridge.mod)[-1, 50]^2))
ridge.mod$lambda[60]
coef(ridge.mod)[, 60]
sqrt(sum(coef(ridge.mod)[-1, 60]^2))
predict(ridge.mod, s = 50, type = "coefficients")[1:2
set.seed(1)
train <- sample(1:nrow(x), nrow(x) / 2)
test <- (-train)
y.test <- y[test]
```

```
ridge.mod <- glmnet(x[train, ], y[train], alpha = 0,
lambda = grid, thresh = 1e-12)
ridge.pred <- predict(ridge.mod, s = 4, newx = x[test
mean((ridge.pred - y.test)^2)
mean((mean(y[train]) - y.test)^2)
ridge.pred <- predict(ridge.mod, s = 1e10, newx = x[t
mean((ridge.pred - y.test)^2)
ridge.pred <- predict(ridge.mod, s = 0, newx = x[test
exact = T, x = x[train, ], y = y[train])
mean((ridge.pred - y.test)^2)
lm(y ~ x, subset = train)
predict(ridge.mod, s = 0, exact = T, type = "coeffici
x = x[train, ], y = y[train])[1:20, ]
set.seed(1)
cv.out <- cv.glmnet(x[train, ], y[train], alpha = 0)
bestlam <- cv.out$lambda.min
ridge.pred <- predict(ridge.mod, s = bestlam,
newx = x[test, ])
```

Lasso avec R:

```
library(glmnet)
grid <- 10^seq(10, -2, length = 100)
### The Lasso
lasso.mod <- glmnet(x[train, ], y[train], alpha = 1,
 lambda = grid)
plot(lasso.mod)
set.seed(1)
cv.out <- cv.glmnet(x[train, ], y[train], alpha = 1)
plot(cv.out)
bestlam <- cv.out$lambda.min
lasso.pred <- predict(lasso.mod, s = bestlam,
 newx = x[test, ])
mean((lasso.pred - y.test)^2)
out <- glmnet(x, y, alpha = 1, lambda = grid)
lasso.coef <- predict(out, type = "coefficients",
    s = bestlam)[1:20, ]
lasso.coef
lasso.coef[lasso.coef != 0]
```