

TD 5 — Régression ridge : cas orthogonal, SVD, choix de λ

Séance 5

Christophe Ambroise

2026-10-12

Objectifs

Établir que le coût quadratique est minimisé par $f^*(x) = \mathbb{E}[Y | X = x]$ et que son erreur se décompose en bruit, biais et variance. Démontrer ensuite, sur le cas orthogonal, qu'il existe toujours une pénalité strictement positive meilleure que les moindres carrés, puis mettre ridge au travail sur le jeu diabète : chemin de régularisation calculé par une seule SVD, choix de λ par validation croisée, et mesure de ce que la pénalité apporte — ou n'apporte pas.

Les exercices 1 et 2 se traitent **sans ordinateur**. Les exercices 3 à 5 portent sur le jeu **diabète** : 442 patients, 10 variables déjà centrées, une réponse continue.

Partie théorique

Exercice 1 — Le coût quadratique et le compromis biais-variance

La séance 1 a annoncé la règle optimale $f^*(x) = \mathbb{E}[Y | X = x]$. On la démontre ici, puis on isole la part de l'erreur qui revient au modèle.

1. **Règle optimale.** On cherche la fonction f qui minimise le risque quadratique $R(f) = \mathbb{E}[(Y - f(X))^2]$. À x fixé, montrer que $c \mapsto \mathbb{E}[(Y - c)^2 | X = x]$ est minimale en $c = \mathbb{E}[Y | X = x]$. En déduire que $f^*(x) = \mathbb{E}[Y | X = x]$ minimise R parmi toutes les fonctions f .
2. Pour une fonction f quelconque, écrire $Y - f(X) = (Y - \mathbb{E}[Y | X]) + (\mathbb{E}[Y | X] - f(X))$. Montrer que le double produit est d'espérance nulle, puis en déduire

$$\mathbb{E}[(Y - f(X))^2] = \mathbb{E}[(Y - \mathbb{E}[Y | X])^2] + \mathbb{E}[(\mathbb{E}[Y | X] - f(X))^2].$$

3. Le premier terme ne dépend pas de f . Retrouver le résultat de la question 1, et montrer que le risque minimal vaut $\mathbb{E}[\text{Var}(Y | X)]$ — le **bruit irréductible**.
4. f^* est inconnue : on l'estime par $\hat{f}$ apprise sur D_n . À x fixé, l'espérance portant sur le tirage de D_n , montrer

$$\mathbb{E}_{D_n}[(\hat{f}(x) - f(x))^2] = \underbrace{(\mathbb{E}_{D_n}[\hat{f}(x)] - f(x))^2}_{\text{biais}^2} + \underbrace{\text{Var}_{D_n}(\hat{f}(x))}_{\text{variance}}.$$

5. Assembler les deux décompositions :

$$\mathbb{E}[(Y - \hat{f}(X))^2] = \underbrace{\mathbb{E}[\text{Var}(Y | X)]}_{\text{bruit}} + \underbrace{\mathbb{E}[\text{biais}^2]}_{\text{modèle}} + \underbrace{\mathbb{E}[\text{variance}]}_{\text{modèle}}.$$

Le premier terme échappe au modèle. Sous un modèle bien spécifié, lequel des deux autres les moindres carrés annulent-ils ? Lequel laissent-ils libre ?

Exercice 2 — Le cas orthogonal et le λ optimal

On se place dans le modèle $\mathbf{y} = \mathbf{X}\beta + \varepsilon$, $\varepsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_n)$, avec un plan **orthogonal** :

$$\mathbf{X}^\top \mathbf{X} = n \mathbf{I}_p.$$

L'erreur quadratique de **prédiction** d'un estimateur $\hat{\beta}$ est

$$\text{MSE}(\hat{\beta}) = \frac{1}{n} \mathbb{E} \|\mathbf{X}\hat{\beta} - \mathbf{X}\beta\|^2.$$

1. Montrer que $\hat{\beta}_\lambda = \frac{1}{n + \lambda} \mathbf{X}^\top \mathbf{y}$, puis que $\hat{\beta}_\lambda = \frac{n}{n + \lambda} \hat{\beta}_{\text{MCO}}$.
2. Décomposer $\hat{\beta}_\lambda - \beta$ en une partie déterministe et une partie aléatoire. Identifier biais et variance.
3. En déduire

$$\text{MSE}(\hat{\beta}_\lambda) = \sum_{j=1}^p \left\{ \frac{\lambda^2}{(n + \lambda)^2} \beta_j^2 + \frac{n\sigma^2}{(n + \lambda)^2} \right\}.$$

Indications : $\|\mathbf{X}\mathbf{u}\|^2 = n\|\mathbf{u}\|^2$, et $\mathbb{E}\|\mathbf{X}\mathbf{X}^\top \varepsilon\|^2 = \sigma^2 \text{tr}(\mathbf{X}\mathbf{X}^\top \mathbf{X}\mathbf{X}^\top) = \sigma^2 n^2 p$.

4. Dériver par rapport à λ et montrer que le minimum est atteint en

$$\lambda^* = \frac{p\sigma^2}{\|\beta\|^2},$$

strictement positif dès que $\beta \neq \mathbf{0}$.

5. Interpréter : que devient λ^* quand $\sigma \rightarrow 0$? quand $\|\beta\| \rightarrow \infty$? quand p croît ? Chiffrer le gain pour $n = 25$, $p = 10$, $\sigma = 1$, $\beta_j = 1$.

Partie empirique

Exercice 3 — Colinéarité et chemin ridge par la SVD

```
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler

d = pd.read_csv("../data/diabetes.csv")
X, y = d.drop(columns="target"), d["target"]
Xs = StandardScaler().fit_transform(X)

def ridge_svd(X, y, lambdas):
    """Chemin ridge complet à partir d'UNE seule décomposition."""
    y_centre = np.asarray(y) - np.mean(y)
    U, d, Vt = np.linalg.svd(np.asarray(X), full_matrices=False)
    coord = U.T @ y_centre
    facteurs = d[None, :] / (d[None, :] ** 2 + np.asarray(lambdas)[:, None])
    return (facteurs * coord[None, :]) @ Vt, d
```

```

d <- read.csv("../data/diabetes.csv")
X <- scale(as.matrix(d[, setdiff(names(d), "target")]))
y <- d$target

ridge_svd <- function(X, y, lambdas) {
  y_centre <- y - mean(y)
  s <- svd(X)
  coord <- t(s$u) %*% y_centre
  t(sapply(lambdas, function(lam)
    as.vector(s$v %*% (s$d / (s$d^2 + lam) * coord))))
}

```

1. Ajuster les moindres carrés sur les variables standardisées. Donner les coefficients, leurs erreurs-types et les statistiques t . Quelles variables ont les plus grandes erreurs-types, et à quelle corrélation cela correspond-il ?
2. Calculer les valeurs singulières de $\mathbf{X}$. Quel est le rapport d_1/d_p ? Le conditionnement de $\mathbf{X}^\top \mathbf{X}$?
3. Tracer le chemin de régularisation sur $\lambda \in [10^{-2}, 10^4]$. Quels coefficients bougent le plus vite ? Est-ce cohérent avec la question 1 ?
4. Vérifier votre implémentation contre `sklearn.linear_model.Ridge` (respectivement `glmnet(alpha = 0)`). Quel écart obtenez-vous ?
5. Combien de décompositions la fonction fait-elle pour 120 valeurs de λ ? Combien en ferait un calcul naïf ?

Exercice 4 — Choisir λ

```

import numpy as np
from sklearn.linear_model import Ridge, LinearRegression
from sklearn.model_selection import KFold, cross_val_score
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler

plis = KFold(n_splits=10, shuffle=True, random_state=0)
lambdas = np.logspace(-2, 4, 120)

moyennes, erreurs = [], []
for a in lambdas:
    modele = make_pipeline(StandardScaler(), Ridge(alpha=a))
    scores = -cross_val_score(modele, X, y, cv=plis,
                              scoring="neg_mean_squared_error")
    moyennes.append(scores.mean())
    erreurs.append(scores.std() / np.sqrt(10))
moyennes, erreurs = np.array(moyennes), np.array(erreurs)

mco = -cross_val_score(make_pipeline(StandardScaler(), LinearRegression()),
                       X, y, cv=plis, scoring="neg_mean_squared_error").mean()

i = int(moyennes.argmin()) # lambda retenu
seuil = moyennes[i] + erreurs[i] # règle du 1 erreur-type
j = max(k for k in range(len(lambdas)) if moyennes[k] <= seuil)
print(lambdas[i], lambdas[j], mco)

```

1. Tracer l'erreur de validation croisée en fonction de λ , avec l'erreur des moindres carrés en référence. Quel est $\hat{\lambda}$? Quel gain?
2. Pourquoi la standardisation est-elle **dans** le pipeline et non appliquée une fois pour toutes avant la validation croisée?
3. Appliquer la **règle du 1 erreur-type** (séance 4) : retenir le plus grand λ dont l'erreur reste sous $\min + \text{ET}$. Quel λ ? Quels degrés de liberté? Que coûte-t-il?
4. Calculer $\text{df}(\lambda) = \sum_i d_i^2 / (d_i^2 + \lambda)$ en $\hat{\lambda}$. Combien de paramètres effectifs le modèle retenu utilise-t-il, sur $p = 10$?
5. Implémenter la **validation croisée généralisée** et comparer le λ qu'elle retient à celui des dix plis :

$$\text{GCV}(\lambda) = \frac{1}{n} \sum_i \left(\frac{y_i - \hat{y}_i}{1 - \text{tr}(\mathbf{H}_\lambda)/n} \right)^2.$$

Exercice 5 — Intérêts et limites de ridge

```
def ajoute_bruit(X, m, graine=0, reserve=200):
    """m colonnes gaussiennes prélevées dans une réserve tirée une seule fois."""
    bruit = np.random.default_rng(graine).normal(size=(len(X), reserve))
    return np.hstack([np.asarray(X), bruit[:, :m]])
```

1. Rejouer la validation croisée sur des **sous-échantillons** de taille $n \in \{20, 30, 50, 100, 200, 442\}$. Comparer moindres carrés et meilleur ridge. Tracer le gain en fonction de n .
2. Que se passe-t-il pour $n = 20$ avec $p = 10$? Expliquer à partir de la formule $\text{Var} = \sigma^2/d_i^2$.
3. Reprendre $n = 442$ et ajouter $m \in \{0, 10, 50, 200\}$ variables de bruit. Comparer moindres carrés et ridge. Que se passe-t-il à $m = 200$, et pourquoi les moindres carrés tiennent-ils encore?
4. Pour $m = 50$ et le λ retenu, examiner les coefficients des 50 variables de bruit. Combien valent **exactement** zéro? Quel est le plus grand en valeur absolue?
5. Conclure : à quoi sert ridge, et quelle méthode faudrait-il pour obtenir des zéros?

Remarque finale

! À retenir

Il existe toujours un $\lambda > 0$ qui bat les moindres carrés : c'est un théorème, et il ne dit rien de la taille du gain. Celle-ci se lit sur le rapport p/n — nulle à 44 observations par paramètre, décisive à 2. Ridge contracte tous les coefficients sans jamais en annuler aucun.

Les deux scripts exécutables correspondants sont `python/td05_ridge.py` et `r/td05_ridge.R`. La dérivation complète de $\text{Var}(\hat{\beta}_{\text{MCO}}) = \sigma^2(\mathbf{X}^\top \mathbf{X})^{-1}$, sa lecture direction par direction et la vérification numérique sur le diabète sont dans `proofs/variance-mco.md`.