

# TP 1: Régression Linéaire et Régularisation

February 13, 2025

## 1 Résumé

Le but de ce TP est de vous faire découvrir l'apprentissage de fonctions de prédiction linéaires par la méthode des moindres carrés et des moindres carrés régularisés. Vous travaillerez également sur des données réelles ICM et prédirez un mouvement à partir de mesures d'électrocorticographie (ECoG).

## 2 Description des données

Vous utiliserez des données issues de la compétition BCI IV <sup>1</sup>. Vous travaillerez sur le jeu de données 4 de cette compétition, où l'objectif est de prédire la flexion des doigts du sujet à partir de mesures ECoG.

Le fichier `ECoG_Finger.npz` contient :

- `Xall` : Matrice  $\mathbb{R}^{n \times d}$  contenant  $n$  exemples de  $d$  variables.
- `Yall` : Vecteur  $\mathbb{R}^n$  contenant les valeurs à prédire.
- `Fe` : Fréquence d'échantillonnage des signaux.

### 2.1 Évaluation des performances

Les performances sont évaluées via :

- L'erreur quadratique moyenne (MSE) :

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (1)$$

- Le coefficient de corrélation :

$$r = \frac{\text{Cov}(Y, \hat{Y})}{\sqrt{\text{Cov}(Y, Y) \text{Cov}(\hat{Y}, \hat{Y})}} \quad (2)$$

---

<sup>1</sup>Site de la compétition : <http://www.bbci.de/competition/iv/>

## 3 Régression des moindres carrés

### 3.1 Préparation des données

- Charger les données et visualiser les signaux ECoG ainsi que les mouvements des doigts sur la même figure (`numpy.load`, `matplotlib.pyplot.plot`).
- Afficher en 2D la valeur `Yall` en fonction de deux variables de `Xall` (ex. indices 42 et 48) en utilisant `matplotlib.pyplot.scatter`.
- Séparer les données en ensembles d'apprentissage et de test (`train_test_split` de `sklearn.model_selection`).

### 3.2 Régression linéaire

- Estimer les paramètres des moindres carrés en utilisant `LinearRegression` de `sklearn.linear_model`.
- Prédire le mouvement des doigts et mesurer les performances (MSE et corrélation).
- Visualiser les résultats sur les ensembles d'apprentissage et de test.

## 4 Régression Ridge et régularisation

- Utiliser `Ridge` de `sklearn.linear_model` pour estimer les paramètres avec régularisation  $L_2$ .
- Tester différentes valeurs du paramètre  $\lambda$  (`alpha` dans `Ridge`).
- Sélectionner le meilleur  $\lambda$  selon les performances.

## 5 Régression Lasso et sélection de variables

- Utiliser `Lasso` de `sklearn.linear_model` pour estimer les paramètres avec régularisation  $L_1$ .
- Tester différentes valeurs du paramètre  $\lambda$  (`alpha` dans `Lasso`).
- Comparer les performances avec celles de `Ridge`.
- Visualiser les coefficients du modèle et observer quelles variables ont un coefficient nul.
- Discuter des différences entre `Ridge` et `Lasso` en termes de sélection de variables et de performances.