TP : Régression linéaire

Rémi Flamary, Alain Rakotomamonjy

Résumé

Le but de ce TP est de vous faire découvrir l'apprentissage de fonction de prédiction linéaires par la méthode des moindres carrés et des moindres carrés régularisés. Vous devrez aussi travailler sur des données réelles ICM et prédire un mouvement à partir de mesures ECoG.

1 Description des données

Vous utiliserez dans ce TP des données proposées lors de la compétition BCI IV. Une description détaillée des données ainsi que les données brutes elles même sont disponibles sur le site web de la compétition [1].

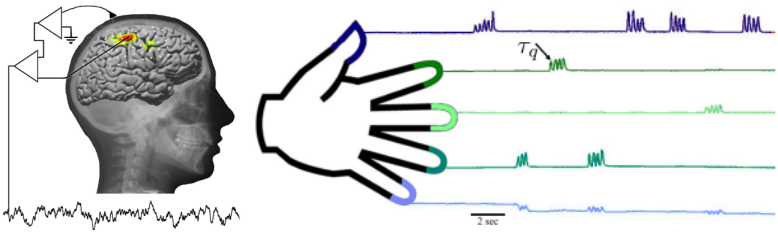

Fig. 1: Données de la Compétition ICM IV. Elles contiennent des mesures ECoG ainsi que des mesures de mouvement de doigts

Données Vous allez travailler sur le jeu de données 4 de cette compétition dont le but est de prédire la flexion des doigts du sujet à partir de mesures ECoG. Vous travaillerez dans ce TP sur les signaux d'apprentissage du sujet 3 et vous devrez prédire la flexion de son pouce au cours du temps.

Protocole expérimental Les trois sujets de la base de données sont des patients épileptiques ayant des capteurs ECoG en place pour des raisons médicales. Les sujets sont équipés de gants permettant de mesurer la flexion de chacun de leurs doigts. Il est demandé aux sujet de bouger un de ses doigts avec un ordre visuel. Le signal mesuré pour le sujet 3 consiste en 10 minutes d'enregistrement à 1000Hz sur 64 électrodes.

Pré-traitement des données Le signal ECoG a été filtré avec un filtre passe bas de Savitsky-Golay. Ce filtre permet d'atténuer le bruit et a déjà montré de bonnes performances pour des tâches de prédiction de mouvement. Ensuite le signal est sous-échantillonné de manière à obtenir

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

un signal temporel échantillonné à $Fe = 50$Hz. Le signal cible (mouvement des doigts) est également sous-échantillonné et uniquement le canal contenant le mouvement du doigt est conservé. Finalement une détection grossière de mouvement est effectuée et seuls les instants temporels correspondant à un mouvement sont conservés. Le fichier `ECoG_Finger.npz` contient les variables suivantes :

Xall Matrice de $\mathbb{R}^{n\times d}$ contenant n exemples de d variables.

Yall Vecteur de $\mathbb{R}^n$ contenant les valeurs à prédire.

Fe fréquence d'échantillonnage des deux signaux.

Évaluation des performances en prédiction Les performances en prédiction seront évaluées de deux manières. Tout d'abord en calculant l'erreur au carré moyenne :

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

que devra bien évidemment être la plus petite possible et le coefficient de corrélation :

$$r = \frac{Cov(\mathbf{Y}, \hat{\mathbf{Y}})}{\sqrt{Cov(\mathbf{Y}, \mathbf{Y})Cov(\hat{\mathbf{Y}}, \hat{\mathbf{Y}})}} \tag{2}$$

où $Cov(.,.)$ est la covariance entre deux vecteurs. Cette mesure est celle qui a été utilisée pour départager les candidats de la compétition. Elle sera égale à 1 si les prédictions sont parfaitement alignées et égale à zéro si la fonction prédit au hasard.

Ces mesures de performances peuvent être calculées simplement en python, pour la MSE, avec la fonction `np.mean` et l'opérateur puissance `**`, pour le coefficient de corrélation directement avec la fonction `np.corrcoef` qui retourne une matrice de corrélation.

2 Régression des moindres carrés

Lors du TP vous aurez besoin des bibliothèque Python `numpy` `pylab` et `scipy`. Il vous est conseillé de les importer dès le début avec le code suivant :

```
import numpy as np
import pylab as pl
import scipy as sp
```

Vous pourrez ensuite accéder aux fonctions de ces modules à l'aide du point (par exemple `np.zeros(10)` pour la fonction `zeros` de numpy). Dans la suite du TP, pour chaque question la liste des fonctions `numpy/pylab` nécessaires est donné entre parenthèses.

2.1 Visualisation des données

— Charger les données en mémoire, visualiser les signaux ECoG et les mouvements des doigts sur la même figure (`np.load,pl.plot,pl.subplot`).
— Visualiser les données en 2D + la valeur à prédire y en fonction de deux variables de $Xall$ 42 et 48. (`pl.scatter` avec `Yall` qui contrôle la couleur).
— Découper les données en un ensemble d'apprentissage et un ensemble de test ($n = 1000$ et `x=Xall[:n,:]` pour conserver les n premières lignes).

2.2 Régression des moindres carrés

— Créer la matrice d'apprentissage `X` telle que nous l'avons vu dans le cours en ajoutant une colonne de 1 aux exemples d'apprentissage (`np.concatenate,np.ones`).
— Estimer les paramètres des moindres carrés sur les données d'apprentissage. stocker ces paramètres dans un vecteur $\mathbf{w}$ et un biais b (`np.dot,np.linalg.solve`).
— Prédire le mouvement des doigts du sujet pour les signaux ECoG d'apprentissage et de test. Tracer les vraie trajectoire et votre prédiction pour les données d'apprentissage et de test (`pl.plot`). Mesurer la performance (MSE et coefficient de corrélation) dans les deux cas. Que remarquez vous ?

3 Régression Ridge et régularisation

— Estimer les paramètres de la fonction linéaire dans le cadre de la régression ridge (`np.eye ,np.linalg.solve`).
— Prédire les mouvements des doigts sur les exemples d'apprentissage et de test, pour différentes valeurs du paramètre de régularisation λ (`for i,reg in enumerate(lst_reg):`).
— Sélectionner la paramètre permettant d'obtenir les meilleurs performances en prédiction.
— Visualiser sur la même figure les vrai Yall et les prédictions en 2D (`pl.subplot,pl. scatter`)

4 Interprétation et sélection de variables

Une fonction linéaire a l'avantage d'être interprétable. En effet le vecteur $\mathbf{w}$ contient des coefficient correspondant à chaque variable dans $\mathbf{X}$. L'amplitude de ces coefficient donnera donc l'impact de chaque variable dans la fonction de décision.

— Visualiser la valeur absolue des coefficients du vecteur $\mathbf{w}$. Repérer les capteur les plus importants dans la fonction de prédiction (`np.abs,pl.stem`).
— Sélectionner uniquement les variables les plus importantes et refaire l'apprentissage. Évaluer les performances en utilisant uniquement ce sous-ensemble de variables.