Practical Session : Linear regression

Rémi Flamary, Alain Rakotomamonjy

1 Data description

In this session you will use a subset of the dataset from the BCI (for Brain Computer Interface) competition IV. A detailed description and the raw data is available on the competition website [1].

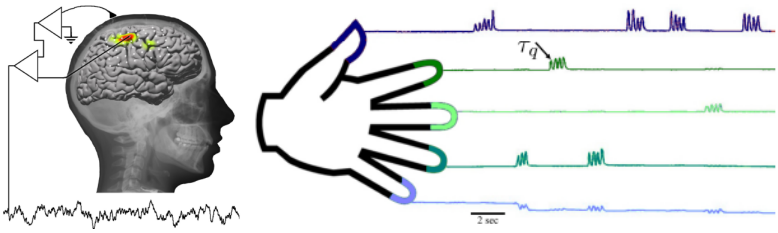

Fig. 1: Dataset 4 from BCI competition IV. It contains measured ECoG signals and the corresponding finger flexion on the subjects.

Dataset You will use the dataset 4 from the competition. The objective on this data is to predict the finger flexion of a subject using only ElectroCorticoGram (ECoG) signals recorded simultaneously on the brain of the subject. In this session you will use the data from subject 3 and predict the flexion of its thumb along time.

Experimental protocol The three subjects from the dataset were epileptic patients with ECoG sensors in place for medical reasons. The subjects were equipped with gloves measuring the flexion of each of there fingers simultaneously with the ECoG signals. The subjects were asked to move their fingers with visual cue. The signal measured consists in 10 minutes recording at 1000Hz on 64 electrodes.

Data pre-processing The ECoG signal was filtered with a Savitsky-Golay band-passed filter that has been shown to work well on similar applications. The signal is then sub-sampled in order to obtain a sampling of $Fe = 50$Hz. The target signal (finger flexion) is also sub-sampled to achieve the same sampling. Finally we only keep temporal samples where the finger is detected to be in movement.

The file `ECoG_Finger.npz` contains the following variables:

Xall Matrix of $\mathbb{R}^{n \times d}$ ceontaining n examples of d variables.

Yall Vector of $\mathbb{R}^n$ containing values to predict.

Fe Sampling frequency for the two signals.

Prediction performance evaluation The performance of a given model can be computed either using the average squared error :

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

[1] Competition website:http://www.bbci.de/competition/iv/

that should obviously be the smallest possible, and the correlation coefficient :

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

where $Cov(.,.)$ is the covariance between the two vectors. This last measure of performance was used in the competition for the final ranking of the candidates methods. It is a normalized performance measure that will be equal to 1 for a perfect prediction of 0 for a random prediction.

2 Least Squares regression

During the practical session you will need to use the Python `numpy pylab` and `scipy` toolboxes. It is recommended that you import them at the beginning of all your scripts with the following code :

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

You will then be able to access most of their functions using their short name followed by a dot (for instance `np.zeros(10)` for the function `zeros` of numpy). In the following, for each question, the list of necessary `numpy/pylab` functions is given between parenthesis .

2.1 Data visualization

- Load the dataset in memory and plot the EcoG signals and the finger movement on the same figure (`np.load,pl.plot,pl.subplot`).
- Visualize the data as a scatter plot where the color of the samples is the value to predict y and their position are variables 42 and 48 of $Xall$ (`pl.scatter` with `Yall` controling the color).
- Cut the data in a trining and a testing set ($n = 1000$ with `x=Xall[:n,:]` to select the n fisrt lines in a matrix).

2.2 Least Squares regression

- Create the training matrix `X` as defined in the course by concatenating a columns of 1s to the training samples (`np.concatenate,np.ones`).
- Estimate the LS parameters on the training data. Store those parameters as a vector $\mathbf{w}$ and a bias b (`np.dot,np.linalg.solve`).
- Predict the finger flexion of the subject on the training and test sets. Plot the predictions along with the true y (`pl.plot`). Compute the performance as MSE and correlation coefficient on training and test data. Conclusions?

3 Ridge regression

- Estimate the parameters on the training data for the ridge regression (`np.eye,np.linalg.solve`).
- Predict the finger flexion of the subject on the training and test sets for different values of the regularization parameter λ (`for i,reg in enumerate(lst_reg):`).
- Select the value of λ having the best performance on test data.
- Visualize with a scatter (as in section 2.1) plot the true values and the predictions (`pl.subplot, pl.scatter`)

4 Interpretability and variable selection

A linear function is quite simple but can be easily interpreted. Indeed $\mathbf{w}$ contains the weights of each variables of $\mathbf{X}$. The magnitude of these coefficients is then a good if simple proxy on the impact that each variable has in the prediction function.

- Visualize the absolute value of the coefficients in $\mathbf{w}$. Find the sensors (variables) having the most impact. (`np.abs,pl.stem`).
- Select only those sensors and train on the data with only those sensors. Evaluate the performance of the model trained on partial data.
- Use `sklearn` to estimate a lasso estimator from the data and compare the selected variable with the coarse magnitude selection above (`sklearn.linear_model.Lasso` where `clf.coef_` for a trained classifier returns vector $\mathbf{w}$).
- Plot the evolution of the parameters as a function of the regularization parameter for the Lasso estimator and the ridge estimator.