Predicting fMRI-based task-related activation with Machine Learning

Cognitive Science and AI: Assignment 1

January 23, 2023

1 Instructions for submission

Maximum marks - 40

- You may do the assignment in Jupyter or Colab notebook.
- You need to submit a notebook specified by Roll Number roll_no.ipynb in Moodle before deadline.
- Include the assignment number, your name and roll number in the notebook as well for better identity.
- No late submissions are accepted.
- IMPORTANT: Make sure that the assignment that you submit is your own work. Do not copy any part from any source including your friends, seniors. Any breach of this rule could result in serious actions including an F grade in the course.
- Your marks will depend on the correctness of answers and output. In addition, due consideration will be given to the clarity and details of your answers and the legibility and structure of your code.
- Do not copy or plagiarise, if you're caught for plagiarism or copying, penalties are much higher (including an F grade in the course) than simply omitting that question.

2 Objective

Multivariate pattern analysis of fMRI data There are two inputs that you can use to predict experimental conditions like face or house stimuli from fMRI data [1].

1. Raw timeseries signals data (4D images) – no statistical analysis applied
2. Activation maps (3D) or β-maps – statistical analysis applied

The objective of this assignment is to address the following two questions:

[1]The data that is pre-processed and have quality checked for any errors if at all

1. What are the differences between 1 and 2 in terms of building classification model? Discuss the trends in classification patterns, for example which of the following input i.e., raw timeseries or statistically derived maps makes more sense for predicting experimental stimuli and why?
2. Which classification model match with the conclusions of the original paper (Haxby et al. 2001).

3 Dataset

The Haxby dataset to be used for the assignment can be downloaded with Nilearn `https://nilearn.github.io/stable/modules/generated/nilearn.datasets.fetch_haxby.html#nilearn.datasets.fetch_haxby`

4 Tasks

Build classification model with Support Vector Machines [2] or model of your choice with and without feature selection [3]. While building classification model, use cross-validation[4]. For instance, the strategies for cross-validating model should be something interesting to avoid information leakage.

We already know how to transform Nifti images (4D) to 2D matrix and how to extract conditions/classification labels for the objectives mentioned above.

What we don't know is about the statistical maps or β-maps and how to estimate them using General Linear Model (K. J. Friston et al. 1994). See some notes on GLM modeling is on Appendix A for your technical understanding.

To estimate β-maps, we follow GLM model implemented in Nilearn. At first level, β-maps or statistical maps per condition is estimated. At second level, we use the first level statistical maps and suppress the baseline activity ("rest") – a way of contrast specification and use these statistical outputs (3D maps) for classifying two conditions "face" versus "house". The classification outputs should show the face versus house accuracy scores for three masks: ventral temporal, face responsive and house responsive masks.

A notebook is prepared to estimate such β-maps.
`https://github.com/KamalakerDadi/Tutorial/blob/master/IIITH/Teaching/Notebooks/CSAI_Assignment_1_helper_GLM_estimation.ipynb`

5 Deliverables

Replicating the appearance (box plots visualization) as shown as example on Figure 1, for each of the mask shipped with Haxby dataset: ventral temporal, face responsive brain regions mask, house responsive brain regions mask depicted in each column. This multi-class classification seen on Figure 1 is more relevant to objective 1. For objective 2, we use only face versus house conditions aka binary classification.

[2] `https://scikit-learn.org/stable/modules/svm.html#svm`
[3] `https://scikit-learn.org/stable/modules/feature_selection.html`
[4] `https://scikit-learn.org/stable/modules/cross_validation.html`

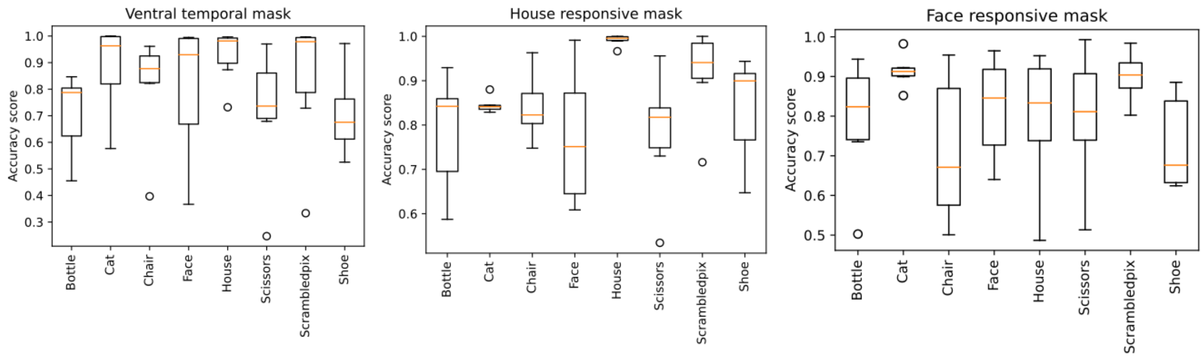

Figure 1: Distribution of classification scores predicting conditions/classification labels without "rest" condition with Linear SVC. These accuracies are obtained while predicting classification labels on raw timeseries data.

References

Abraham, Alexandre et al. (2014). “Machine learning for neuroimaging with scikit-learn.” In: *Frontiers in Neuroinformatics* 8.

Friston, K. J. et al. (1994). “Statistical parametric maps in functional imaging: A general linear approach.” In: *Human Brain Mapping* 2.4, pp. 189–210.

Friston, K.J. et al. (1998). “Event-Related fMRI: Characterizing Differential Responses.” In: *NeuroImage* 7.1, pp. 30–40.

Haxby, James V. et al. (2001). “Distributed and Overlapping Representations of Faces and Objects in Ventral Temporal Cortex.” In: *Science* 293.5539, pp. 2425–2430.

A Encoding statistical β maps

Standard analysis in task fMRI relates psychological manipulations to brain activity separately for each voxel also known as mass univariate analysis. It models the BOLD signal as a linear combination of experimental conditions – the General Linear Model (GLM, (ibid.)) that produces the brain responses in reply to the experimental conditions, Figure **??**. The BOLD signal forms a matrix

$$Y = X\beta + \varepsilon$$

where Y is the acquired fMRI data of shape $\mathbb{R}^{n \times p}$, where p is the number of voxels and n is the number of scans/timepoints; X is the design matrix formed by k temporal regressors of interest $X \in \mathbb{R}^{n \times k}$. Each regressor is an indicator of occurrence of stimuli in the experimental design The design matrix also includes some nuisance confounds such as subject motion, as well as other noisy signals like low-frequency drifts present in the data. $\varepsilon \in \mathbb{R}^{n \times p}$ denotes noise (K. Friston et al. 1998). The GLM is presented for one signal on Figure 2.

What we need to estimate are the $\beta \in \mathbb{R}^{k \times p}$ denotes the coefficients (the weight of the regressors), and $\varepsilon \in \mathbb{R}^{n \times p}$ denotes a noise component. For each voxel j, this noise can be modeled as a Gaussian white noise with zero mean and variance σ_j^2. It is often modeled with some process as the BOLD signal is auto-correlated in time domain. Then estimator becomes, $\hat{\beta} = X^{\dagger}Y$.

In our assignment, we use the Nilearn library (Abraham et al. 2014) to estimate such β maps or statistical maps.

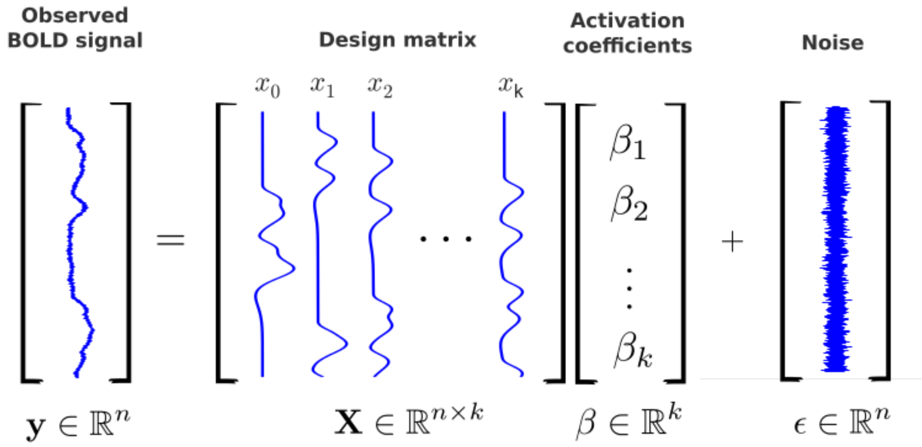

Figure 2: The GLM model for one voxel. The model expresses the acquired BOLD signal as a linear combination of regressors plus a noise term. Each regressor of the design matrix is the convolution of a reference HRF and the stimulus function. Each element of the (needed to be estimated) activation coefficients $\beta_1, \beta_2 ... \beta_k$ represent the relative amplitude of a given condition.