Final Project
Stat 215A, Fall 2017

Hector Roux de Bézieux

December 11, 2017

Abstract

Being able to understand how the brain encode and decode visual input has always been a challenge and a goal in neurology. A first aspect is being able to predict how different regions of the brain react when presented with different images. In this paper, we relied on fMRI of subjects looking at grey-scale pictures to build a predictive model of the brain response in the visual region. Using various model selection and regression tools, we chose the most appropriate model and studied how the spatial location of voxels are linked to their response to images. Our model performed especially well on voxels near the origin (center of the brain?). However, our model selection step was unstable, which limited the quality of our prediction methods. Further work would therefore require a more efficient way to select the best model.

1 Introduction

Deciphering how the brain respond to visual inputs, and in particular images, has long been a key interest in neuroscience. An image is viewed by the eye and becomes an electric signal which is then transmitted to the brain. Such a signal activates or repress neurons in certain regions of the human brain, depending on its content, light intensity and many other such features. Linking those image features to specific brain regions will therefore go a long way towards understanding how the human brain reacts to images. It is also a first step towards a more ambitious but fascinating goal: estimating the image seen by the person just by looking at its neurological response.

The focus of this lab is to develop a model that would properly estimate the response of several brain regions given an image. This paper approaches the problem through several steps: model selection, regression, diagnostic, interpretation and discussion of the model.

2 Data description

The data consists of a set of 1750 gray-scale images that have been shown to a subject. Each image consists of 16384 pixels (128×128). After transformation using Gabor wavelet pyramid, images are represented as vectors of length 10921. As images are shown to the subject, the brain activity is recorded using functional magnetic resonance imaging or fMRI. This techniques discretizes the brain into cubes and measure the magnetic (and hence electric) activity of each of those regions. For our study, we consider 20 voxels that have been previously linked to visual functions. A more detailed description of the location of those regions will follow in part 5.

The data is therefore a 1750×10921 X matrix of image features and a 1750×20 Y matrix of voxel responses. A first separation done on the data is between a training/validation set and a test set. To do so, we randomly select 20% of the data (i.e 20% of the images and their associated response) to set out for a test set. This leaves us with 1400 images in the training/validation set and 350 in the test set.

3 Methods

3.1 Setting the model with Lasso

Because we have more covariates than observations, we are in a case of over-specification. Any modeling done using the full range of the predictors will result in obvious over-fitting. A method for selecting a reduced number of covariates needs to be used. In this paper, we use the Lasso regularization technique. While Ordinary Least Squares (OLS) find the $\hat{\beta}$ that minimizes $\sum_{i=1}^{n} (Y_i - X_i\beta)^2$, Lasso aims to minimize $\sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda|\beta|_1$, with λ a parameter to choose. This will shrink $\hat{\beta}$ by setting more and more coefficients to zero as λ increases. In practice, the algorithm selects an initial λ big enough such as all coefficients are zero and then decreases λ until we get close to over-specifying.

3.2 Choosing λ

To choose the value of λ, we used an objective function that we aimed to minimize. We considered 5 objective functions in total (see 3.4 for details). To obtain more robust results, we used 4-fold cross validation. The training/validation set is divided into 4 blocks (images are randomly assigned to each fold but the folds are the same for all voxels). One after the other, each block is left-out and used as validation set. The three others are used as training. On each training set, we run Lasso with a wide range of λ, starting with high values where all coefficients of $\hat{\beta}$ are zero and stopping when we have nearly as many non-zero coefficients as there are images in the three folds (1050). Then, we find the value of λ such as the associated $\hat{\beta}$ minimizes each of the five objective function on the training set. We then only keep the covariates that have non-zero coefficients.

3.3 Regression

Then using only those covariates, we can build a regression model on the training set constituted by the three folds. We use 2 regression models: usual OLS and ridge (where we minimize $\sum_{i=1}^{n} (Y_i - X_i\beta)^2 + \lambda(|\beta|_2)^2$). For Ridge, the value of λ is chosen by 10-fold cross validation (picking the one that minimizes Mean Standard Error or MSE on the left-out fold). Then, we validate both those models on the left-out fold that acts as a validation set. We then measure 2 criterion for selecting the best model: MSE and the correlation between the fitted and real values.

3.4 Objective functions

As said above, we consider 5 objective functions:

- Akaike Information Criterion (AIC). $AIC = -2\ell\ell(\hat{Y}) + 2k$, with k the number of parameters in the model (i.e the numbers of covariates selected by the Lasso regularization). Here, by noting n the number of observations, $\ell\ell(\hat{Y}) = -\frac{n}{2}log(\sigma^2) + \frac{RSS}{\sigma^2} + C_1$, where C_1 is independent of the model. Since we do not know σ^2, we estimate it with $\hat{\sigma^2} = RSS$ so that $\ell\ell(\hat{Y}) = -\frac{n}{2}log(RSS) + C_2$. Since we use the AIC to compare between models, we can drop the constant and we have : $AIC = n \times log(RSS) + 2k$.

- Bayesian Information Criterion (BIC) is similar to the AIC but with a different penalty for the dimension of the model. $BIC = n \times log(RSS) + log(n) \times k$.

- AICc is the corrected AIC for finite samples. It adds an extra penalty for model complexity : $AICc = n \times log(RSS) + 2k + \frac{2k(k+1)}{n-k-1}$

- Cross-validation (CV): we break the training set of three folds into 10 smaller folds. We leave out one fold at a time. On the nine other folds, we fit a Lasso regression and then compute the MSE on the tenth fold. We pick the λ which has minimal average MSE.

- Estimation Stability with Cross Validation (ESCV) also relies on a 10-fold cross-validation but here we pick the λ greater than the one picked by CV which also minimizes $\frac{Var(\hat{Y}(\lambda))}{||(mean)(\hat{Y}(\lambda))||_2^2}$. Sometimes, the λ for CV and ESCV may well be the same (as is the case with other objective functions).

3.5 Global model selection method

In total, we have:

$$\boxed{(20 \text{ Voxels}) \times (4 \text{ Folds}) \times (5 \text{ Objective Functions}) \times (2 \text{ Models}) = 800 \text{ Measures for 2 criteria}}$$

3.6 Remarks

- Computing all this takes $\sim$ 50 minutes with 10 cores on the clusters. The code used to produce such a result can be found in **R/parallel.R**. The output is the file **data/output.RData**

- The choice of 4 global folds was also done for practical computational purposes (the more folds, the longer it takes to run).

- We always impose that at least one coefficient be non-zero so that the regression can always happens, even though a model with zero coefficient (fitting Y_i with $\bar{Y}$) sometimes minimizes the objective function even more.

4 Results of model selection

4.1 Models performances

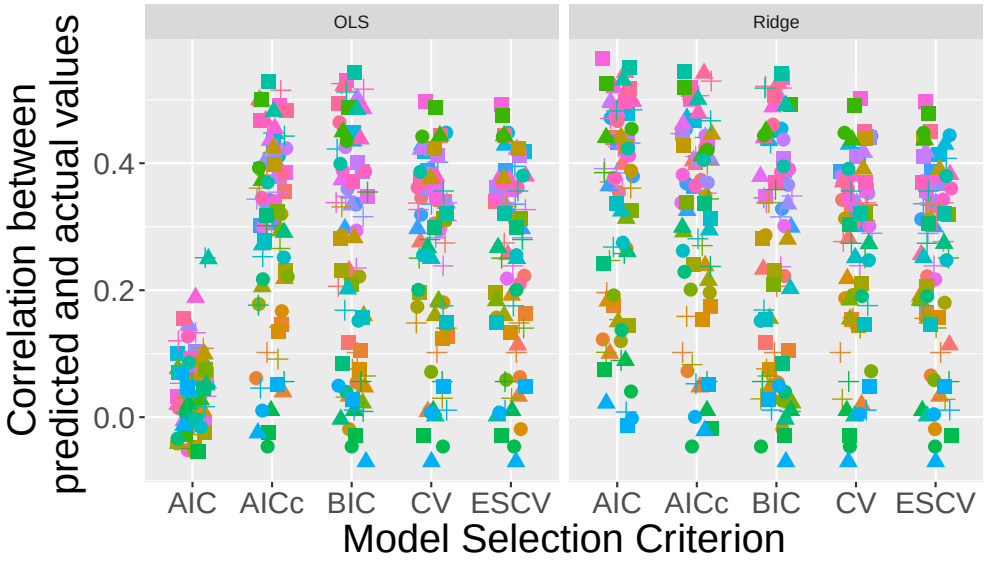


Figure 1: Correlation between the fitted and real values for all models, voxels (colors) and CV fold (shape)

We plot the results in Fig.1 . For each model and each objective function, we plot the correlation between fitted and real values. Each point corresponds to one voxel (colors) and one fold (shape). We can make several observations from that. The first is that, apart from AIC + OLS, all the models are fairly similar in their prediction power. We can compute the mean correlation and we get table 1.

models	mean_corr	max_corr	mean_mse
AIC + Ridge	0.34	0.50	0.30
AICc + Ridge	0.33	0.50	0.29
AICc + OLS	0.30	0.47	0.31
CV + Ridge	0.27	0.42	0.30
CV + OLS	0.26	0.42	0.31
BIC + Ridge	0.26	0.50	0.30
BIC + OLS	0.26	0.50	0.30
ESCV + Ridge	0.25	0.42	0.31
ESCV + OLS	0.25	0.42	0.31
AIC + OLS	0.05	0.15	71.66

Table 1: Mean Correlation, maximum correlation and mean MSE for each model

As could be seen in Fig 1, the worst model by far is the AIC + OLS. A little surprisingly, the best model is AIC + Ridge. Overall, the extra layer of regularization added by Ridge always improve performances. Also, our predictive power is still not optimal: the best correlation that we have is 0.5.

When we look more closely at the models, it can be seen that AIC tends to over-fit the training set by keeping too many parameters while BIC, CV and ESCV tend to under-specify the models: in some cases, only one or two covariates are kept. One possible explanation as to why those models are not optimal is that we choose the λ that minimizes the MSE, plus a penalty for model complexity or some cross-validation, while our parameter of interest is the correlation. And, as can be seen from table 1, the ranking changes quite a lot between the two, even though the first two and the last models are shared.

4.2 Selecting the best model

As we can see the two best models are AIC + Ridge and AICc + Ridge but it is hard to break the tie based on the metrics in table 1. If we look at other metrics to select the best model, we can see that they are pretty consistent. The rankings in maximum correlation and minimum MSE are the same. The two best models are Lasso with AIC + Ridge and Lasso with AICc + Ridge. It is hard to select one model over the other with those metrics. One way to distinguish between them is to look at their consistency across voxels. For each of those models, we compute the standard deviation of the correlation across the 4 folds, for each voxel. The standard deviation for correlation is the same but not for the MSE (it is 22.18 for AIC + Ridge and 26.70 for AICc + Ridge). AIC + Ridge does seem the best model but since the differences are still tenuous, we will keep both of the models for further analysis.

5 Results for each individual voxel

5.1 Quality of prediction

Another result of interest is that there is much more difference between voxels than within voxels (for the 4 folds). The mean standard deviation of the correlation for each model and CV fold is 0.15 while the mean standard deviation for correlation for each voxel and model is 0.05. This stability in our results is reassuring: the biological differences are stronger than the variations in our statistical models.

We can show the results for each voxel in our two best models in table 2. Depending on the voxel one considers, the best model might be either AIC + Ridge or AICc + Ridge. Moreover, as could be seen in Fig.1, the quality of our prediction varies wildly

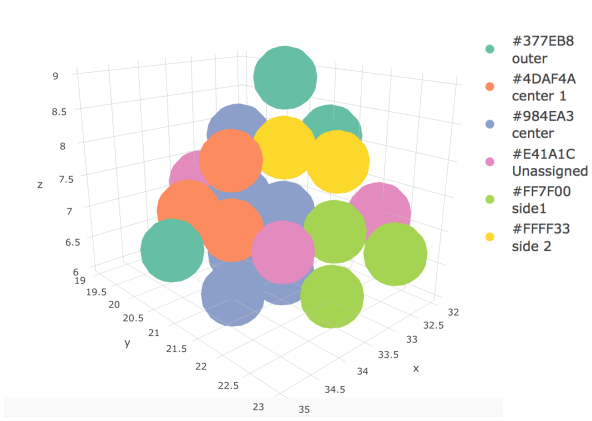
Voxel	AIC + Ridge	AICc + Ridge	Voxel	AIC + Ridge	AICc + Ridge
1	0.39	0.40	11	0.30	0.29
2	0.15	0.10	12	0.48	0.47
3	0.20	0.18	13	0.00	0.02
4	0.40	0.41	14	0.40	0.37
5	0.13	0.18	15	0.43	0.40
6	0.27	0.28	16	0.44	0.44
7	0.45	0.42	17	0.43	0.40
8	0.07	0.00	18	0.49	0.48
9	0.22	0.28	19	0.45	0.45
10	0.50	0.48	20	0.50	0.50

Table 2: Mean Correlation for each voxel and each model

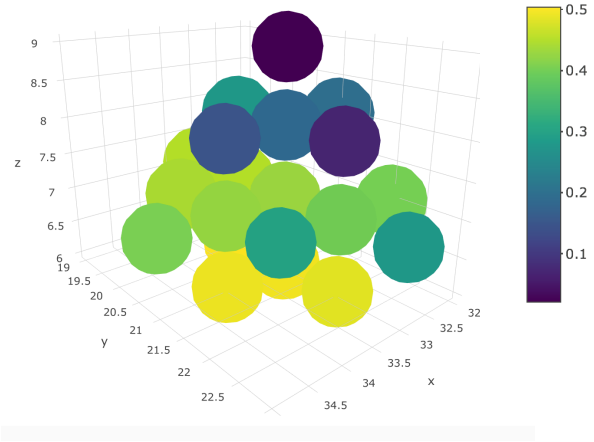
5.2 Link between quality of prediction and voxel characteristics

To be able to explain why we can better predict some voxels than others, we need to look a bit deeper at the voxel first. Based on the responses to all images, we can use hierarchical clustering to make a dendogram on the voxels and establish several clusters. If we then plot the 3D-structure of the voxels and color each voxel by its cluster, we can clearly see, in Figure 2a, that statistical clusters based on responses to images are linked to spatial regions of the brain. The responses of the brain are clearly linked to the specific position of the voxels. We define descriptive names for the regions based on their position. The code to obtain the dendogram ans this plot can be found in **R/brain3D.R**.

We can also look at the quality of our predictors based on their spatial location on Fig 2b. We color the voxels based on the correlation of the fitted values with the real values, for the best model of each voxel (between the top two). There is a clear link between the spatial location and the precision of our models. The z-axis is the most linked to the precision but lower x and y are also associated with better fits. Overall, the closer the voxel is to $(0, 0, 0)$, the better the fit.



(a) Voxel identified based on their responses to the images



(b) Voxel identified based on the quality of the fitted values

Figure 2: Spatial location of the voxels

6 Diagnostics

6.1 Stability

A first way to ensure stability has been discussed. Using 4-fold cross-validation: for each voxel and model, we have four measures for the correlation between the fitted and real values. As we have seen before, we can see that this correlation is quite stable across various folds, and that the differences between voxels are much more important.

Another measure of stability can be done on variable selection: we run a 5-fold cross-validation on the training/validation set. Such an analysis was run with **R/stability.R** and the output is **data/output.RData**. Using AIC and AICc, we find all selected covariates and we can compare between folds. We then count how many times each covariates is selected among the 5 folds and compute the frequencies of occurrences (from 0 to 1). We only show the results for 4 voxels with the highest fit on the validation set and not the zeros (they can be deduced quite easily from the table). As we can see in table 3, not many parameters are picked more than once or twice. Therefore, the selection of parameters is not very stable. The ridge regularization that happens after probably shrink all the parameters that are not common between the folds, which is why the final result is more stable.

Occurences	voxel10	voxel12	voxel18	voxel20
1	1524	1425	1468	1494
2	680	691	665	664
3	387	407	383	424
4	221	223	254	232
5	132	134	126	114

Table 3: Frequencies at which a parameter is picked

6.2 Validity of the fit

Then we test out models on the test set. To so so, we use the full training/validation set to select the restricted covariates based on Lasso and choosing the appropriate λ_{Lasso} with AIC and AICc. Then, we fit a Ridge regression to the training/validation set, choosing λ_{ridge} with a 10-fold cross validation. Finally, we predict the values of the test set. We do that on the 4 voxels from before. Te code is in textbfR/best_models.R and the results where too heavy to push to Github. In table 4, we can see that, for 3 out of 4 voxels, the results are very good and close to what we have on the validation set in 4.1. However, for voxel 20, the fit is now very poor.

	voxel20	voxel10	voxel18	voxel12
AIC	-0.04	0.15	0.53	0.43
AICc	0.08	0.24	0.51	0.45

Table 4: Fit of our model on the test data

We can look at the number of covariates selected for explanations in table 5. As mentioned before, AIC overfits at first but Ridge regularization corrects that. However, for voxel 20, AICc only select 2 covariates. So there seems to be only a few powerful predictive covariates for voxel 20. Given how unstable the parameter selection is, it is therefore not surprising that the predictions are poor on voxel 2.

	voxel20	voxel10	voxel18	voxel12
Dimension with AIC	1399	1396	1399	1399
Dimension with AICc	2	38	102	84
Common dimensions	2	26	71	51

Table 5: Dimensions of the model, per voxel

7 interpretation

We have already see which regions of the brains are more easily predicted than others. Now, we will look at which covariates are the most selected. We choose the predictors that are picked at least 8 times among the 5 folds from 6.1, and with AIC or AICc. The number of covariates that we select range from 0 for voxel 16 to 23 for voxel 18 so we have a lot of heterogeneity. There are 64 different covariates thus selected, and 11 covariates appear in at least 3 voxels. When looking at the list of covariates selected, we can notice that only one of them comes from the columns of X above 5000 and 15 from above 1000. However, this is just a coincidence. Randomly permuting the columns of X before re-doing the analysis deleted that phenomenon.

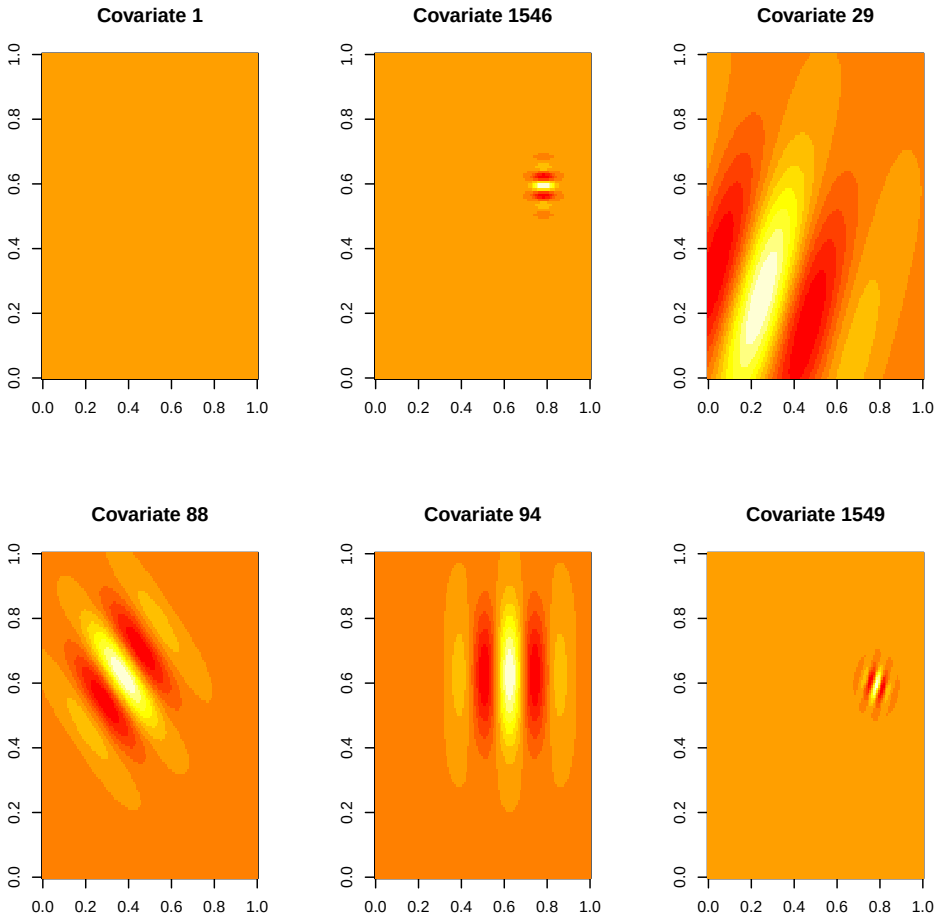


Figure 3: Wave functions of the covariates

We can still, however, look at those covariates. For example, we can plot the top 6 covariates (in term of

how many times we find them selected in the voxels), as in Fig. 3. We can notice some characteristics. Covariates 1546 and 1549 look similar, as do covariates 29, 88 and 94. We can also look at whether some results are coherent with the brain structure. The voxels where covariate 1549 matters are all the clusters from side 2 and center 1, from Fig 2a so we find once again a link with the neurology. However, the other covariates cannot be linked so easily to apparent clusters.

8 Predicting the new data

The response of the 20 voxels to the 120 new images was done as for the test set. The code used is in **R/prediction.R** and the results in **output/predv1_hectorrouxdebezieux.txt**.

9 Discussion

We have establish several results in this paper: using Lasso for model selection and Ridge regularization for regression, we build a predictive model for the brain activity in 20 voxels of the visual region. We can see that voxel react similarly to spatially close voxels and that our model performs much better for some voxels than others (ranging from a correlation of 0 to 0.5). The correlation is relatively stable.

However, as we have seen, the model selection method is quite imperfect, even though Ridge regularization partly make up for that. In particular, the choice of covariates is not very stable with cross-validation. To mitigate this, the model-selection step could be stabilized with cross-validation. Parameters selected in k of the folds would be kept. k can be a bit too low because related over-fitting will be dealt with with Ridge regularization afterwards.