EE 511 Final Report

Read the Mind

Yeyun Lu

Abstract: The task of this project is predicting which word was being read using 3D fMRI images data. After preprocessing data, principal component analysis is for feature reduction, Lasso regression is used to map voxel features X into semantic features Y and L1/L2 distance is used for binary classification. The results for each person is 74%, 68%, 73%, 72%, 66%, 74%, 72%, 72% and 74% (the mean is 72% and the baseline is 62%). In addition, result for all the data set (9 persons together) is 65% since the activation pattern for each person is different.

1. Introduction

1.1 Problem Description

Given an image and two candidate words, predict which of those words was being read by the subject. Brain scans were taken of a subject in the process of a word reading task. In this task, I need to predict what word the participant is reading based on the activation patterns learned from fMRI images in their brain.

The data is collected from nine right-handed adult participants over time (There are 360 trials per person) [1]. At trial i, word i is shown to the subject. Take subject 1 as an example. The fMRI is recorded as a 21764 dimension vector X_i . Each dimension corresponds to the activity in a voxel (a cube in the 3-d coordinates of the brain). There are 360 trials, for which the results are put into $X \in R^{360\times 21764}$ and $Y \in R^{360\times 1}$. $X_{i,j}$ is the signal at j^{th} voxel at trial i, and $Y_{i,j}$ is the j^{th} feature of the word displayed at trial i.

1.2 Overview of Approach

- Step 1: PCA

$$X \in R^{360\times 21764} \longrightarrow X \in R^{360\times 300}$$

- Step 2: Lasso Regression (sparse data)

$$X \in R^{360\times 300} \longrightarrow Y \in R^{360\times 218}$$

- Step 3: Binary classification

$$Y \in R^{360\times 218} \longrightarrow Y \in R^{360\times 1}$$

2. Approach

2.1 Data Preprocessing

Step 1: Load input data and standardize X to have mean 0 and unit norm for each column and center Y to have mean 0 for each column

Step 2: From original to word id: load words from 'info' class, search specific word in the word-id dictionary and encode output Y with the word id.

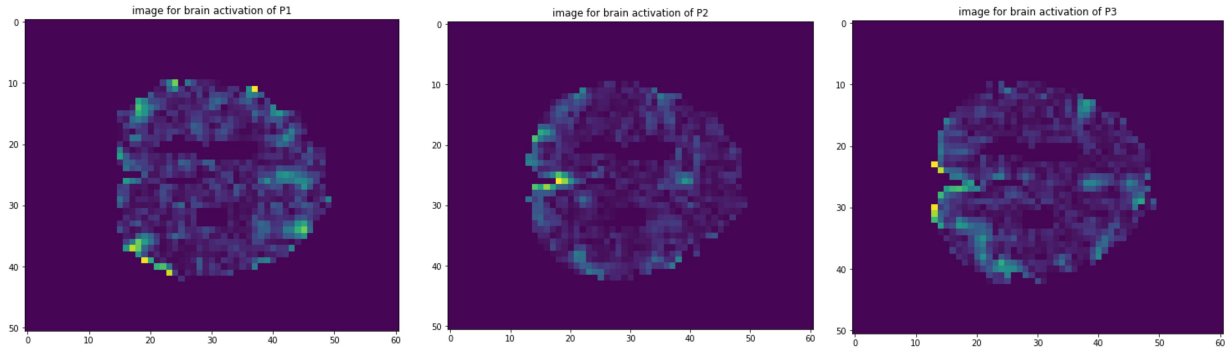
Step 3: From word id to word feature: The vocabulary set contains 60 nouns covering different categories. For each word, 218 semantic features are extracted. Each feature corresponds to a question about the semantic meaning of this word. The value is a score ranging from 1 to 5 provided by a human labeler as his response to the question. For example, feature 1 corresponds to the question: "IS IT AN ANIMAL". The word "ant" has value 4 for feature 1, whereas the word "airplane" has value 1. This can help us project our word id into word features.

Step 4: Actually, we preprocess Y_train and Y_test differently: there is one single $Y \in R^{360\times218}$ matrix for Y_train, but two matrices for Y_test (one for true word index and one for random word index so that we can use them for binary classification.)

2.2 Visualization for the data

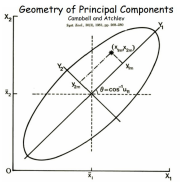
Get the x,y,z coordinate and arrange corresponding data into a 3D array 'visual', then transfer it into pixel.

Get the brain activation images for several persons at trial 0 as follows (z from 0 to 20):



Visualized Image for brain activation for 9 persons

2.3: Feature Reduction



Use principal component analysis to project 2000+ features into 300 features.

2.4 Lasso Regression:

Lasso regression is good at dealing with sparse problem, which is suitable for our data set.

Loss function for Lasso Regression:

$$F(w) = \sum_{i=1}^{n}(x^{(i)} * w + w0 - y^{(i)})^2 + \lambda \sum_{j=1}^{d} |w_j|$$

Coordinate descent for lasso with normalized features, update one single feature at each time:

Initialize $\hat{w} = 0$ (or smartly…)
while not converged
for j=0,1,…,D

compute: $\rho_j = \sum_{i=1}^{N} h_j(x_i)(y_i - \hat{y}_i(\hat{w}_{-j}))$

set: $\hat{w}_j = \begin{cases} \rho_j + \lambda/2 & \text{if } \rho_j < -\lambda/2 \\ 0 & \text{if } \rho_j \text{ in } [-\lambda/2, \lambda/2] \\ \rho_j - \lambda/2 & \text{if } \rho_j > \lambda/2 \end{cases}$

$$\partial_{w_j}[\text{lasso cost}] = \begin{cases} 2z_j w_j - 2\rho_j - \lambda & \text{when } w_j < 0 \\ [-2\rho_j - \lambda, -2\rho_j + \lambda] & \text{when } w_j = 0 \\ 2z_j w_j - 2\rho_j + \lambda & \text{when } w_j > 0 \end{cases}$$

$$= 0$$

$$\hat{w}_j = \begin{cases} (\rho_j + \lambda/2)/z_j & \text{if } \rho_j < -\lambda/2 \\ 0 & \text{if } \rho_j \text{ in } [-\lambda/2, \lambda/2] \\ (\rho_j - \lambda/2)/z_j & \text{if } \rho_j > \lambda/2 \end{cases}$$

2.5 Binary Classifier:

The classifier is defined as follows. Since the output Y is a n*218 matrix, I use both L1 distance and L2 distance to calculate the similarity between prediction and true Y, and the wrong Y is produced from random words. After calculate the distance, we make a decision according to similarity.

3. Results and Analysis

3.1 Tune hyperparameter

Use validation data set to tune hyperparameter lambda:

```
lambda: 0.05
Accuracy for L1: 0.75
Accuracy for L2: 0.7333333333333333
lambda: 0.1
Accuracy for L1: 0.7666666666666667
Accuracy for L2: 0.75
lambda: 0.15
Accuracy for L1: 0.7833333333333333
Accuracy for L2: 0.7833333333333333
lambda: 0.2
Accuracy for L1: 0.7333333333333333
Accuracy for L2: 0.75
```

From the result above, we choose lambda = 0.15 for training and testing of Person 1. Similarly, we need to tune hyperparameter for other persons. In fact, for each person, the best lambda is different.

3.2 Test result for 9 persons separately:

It tooks long time to train the model for even one single person if I use all the feature.
There are 2 methods:

1. Use all the matrix as input data, but use randint(0,self.p-1) to choose 200 feature idx randomly to update, which will produce a sparse matrix.
2. Use PCA to reduce the dimension into that less than 300 (number of training samples).

Using method 1, since the input matrix is 300*20000+, it takes long to train the mode (more than one hour for one single person). And I have got some results as follows:

```
---------The 1st Person--------
Accuracy for L1: 0.7333333333333333
Accuracy for L2: 0.7166666666666667

---------The 2nd Person--------
Accuracy for L1: 0.675
Accuracy for L2: 0.6583333333333333

---------The 4th Person--------
Accuracy for L1: 0.7166666666666667
Accuracy for L2: 0.7166666666666667
```

In this method, L1 is usually better.Although the result is acceptable, it is quite time-consuming (there would be about 10 hours for all the subjects).
So I switch to method 2. Using method 2, it takes shorter time (about 5 minutes for each person with 20 iteration). And the result is as follows:

```
---------The 1st Person--------
Accuracy for L1: 0.6916666666666667
Accuracy for L2: 0.7416666666666667

---------The 2nd Person--------
Accuracy for L1: 0.65
Accuracy for L2: 0.6833333333333333

---------The 3rd Person--------
Accuracy for L1: 0.7333333333333333
Accuracy for L2: 0.7333333333333333

---------The 4th Person--------
Accuracy for L1: 0.7
Accuracy for L2: 0.7166666666666667

---------The 5th Person--------
Accuracy for L1: 0.6416666666666667
Accuracy for L2: 0.6583333333333333

---------The 6th Person--------
Accuracy for L1: 0.7466666666666667
Accuracy for L2: 0.7466666666666667

---------The 7th Person--------
Accuracy for L1: 0.7
Accuracy for L2: 0.7166666666666667

---------The 8th Person--------
Accuracy for L1: 0.7166666666666667
Accuracy for L2: 0.7166666666666667

---------The 9th Person--------
Accuracy for L1: 0.7416666666666667
Accuracy for L2: 0.725
```

Although they are similar in accuracy, PCA method is more time-saving. And in the second method, L2 is better.
Finally, we choose the better result among these two methods and the result for each person separately is as follows:

	Baseline	P1	P2	P3	P4	P5	P6	P7	P8	P9
Accuracy	62%	74%	68%	73%	72%	66%	74%	72%	72%	74%

The result is acceptable, but the accuracy is not so high. The mean of the accuracy is 72%, one of the reason may be that the iteration is not enough.

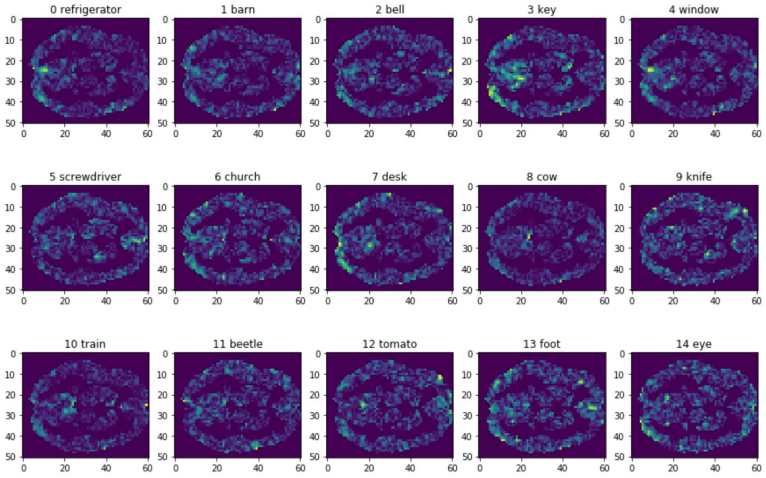
3.3 Result for the total data set (9 persons):

When combine the data of all the 9 persons together, the input would be 2700 * 20000+. It is a large matrix, so we need to project the data into lower dimension. Using PCA method, the result is as follows:

	Baseline	P1+P2+P3+P4+P5+P6+P7+P8+P9
Accuracy	62%	65%

It takes several hours to run the whole data set. The accuracy is higher than baseline but not good enough. The reason may be that the activation pattern for differents persons would be different.

3.4 Dimensionality Reduction: project to x-y plane
When looking into the projection images, it seems that we can project all the data in (xi,yi) to a 2D plane and get different image for different words. We choose z from 3 to 10 in the following image:



Visualized Image for brain activation for different words

Since dimension for x,y,z is the same for all the 9 subjects (i.e. (51,61,23)), I can projected the 3D image into x-y plane and got the data set in $R^{3240\times3111}$ $(\text{where } 3111=51*61)$, which is another way for feature reduction.

4. Summary

4.1 Conclusions:

(1) When using PCA method before Lasso regression, L2 distance for binary classification is better. When only using Lasso regression, L1 distance is better.
(2) The result for each subject separately is better than using the whole data set, because the fMRI image different from person to person.
(3) Choosing meaningful features based on expert knowledge is import to the final result of this problem.

4.2 Feature work:

Actually, in the original data set, different words belong to different category. For example, 'hammer' belongs to 'tool', so it may be a meaningful work it do such a multiple classification.

Reference:

[1] http://www.cs.cmu.edu/afs/cs/project/theo-73/www/science2008/data.html
[2] https://www.coursera.org/lecture/ml-regression/deriving-the-lasso-coordinate-descent-update-6OLyn
[3] http://dai.fmph.uniba.sk/courses/CSCTR/materials/CSCTR_07sup_Mitchell2008Science.pdf