FMRI Classifier to Decode Cognitive State
Final Project Report

Abhishek Maheshwari, Nitish Joshi, Rohit Singh Kharanghar
Satyandra Guthula, Susmit Wagle
161110{01,16,19,20,52}@iitk.ac.in
Indian Institute of Technology, Kanpur

June 3, 2017

Introduction

In our brain, various neural activity happens at all time. In each activity, neurons fluctuate as we perform a simple task like walking, talking, to complex tasks like cognitive processing, perception, comparison, sensitization etc. These various activities generates different pattern of neural BOLD signal in our brain. fMRI is the technology to capture these pattern. It can detect tiny metabolic(like blood oxygen concentration) changes which are coupled with neural activity.

In our project we train a classifier that processes fMRI generated images and predicts the cognitive state of subject. To be more precise, we give an fMRI image(in form of a 21K dimensional vector) as input and predict what object the subject was thinking about while the image was being taken.

Previous Work

Wang et al [1] have applied a number of classifier training methods like Gaussian naive bayes(GNB), Support vector machine(SVM), KNN to train a classifier for a similar problem. Also instead of individual human subjects, they train the classifier so that it can be applied on multiple human subjects.
Norman et al[2] have done multi-voxel pattern analysis of functional fMRI data.This analysis is beneficial in ways like 1) It makes more sensitive detection of cognitive states. 2) It helps in characterizing how cognitive states are represented in the brain.
Viviani R[3] has described a principal component analysis (PCA) method for functional magnetic resonance imaging (fMRI) data based on functional data analysis, an advanced non parametric approach.

Data-set Description

Our data set consist of 300 fMRI images (each of which is represented using a 21764 dimensional feature vector of voxel intensities) captured while the subject was being shown a word along with a

"line drawing" representing the word. Each word comes from a small vocabulary of 60 nouns and also has an associated 218 dimensional feature vector where the features describe some human-defined attributes of each word.

The test data consists of 60 fMRI images and, for each test image, the goal is to predict which word (from two candidate words) the subject was thinking about.

2-vs-2 prediction : 2-vs-2 prediction prediction accuracy is the percentage of tests where the correct assignment is chosen among given two choices corresponding to each test fMRI brain activation voxel intensity.

Implementation

1 Regression

1.1 Ridge Regression

Linear regression with l_2 regularizer is a basic squared error loss function with l_2 constrain on weight vector W.
we define loss function for this as

$$L(\lambda, W_i) = \arg\min_W \sum_{n=1}^{N} (y_n - W^T x_n)^2 + \lambda ||W||_2^2$$

Here x_n is an fMRI image, while y_n is a word related to it. For each 218 feature we learned a separate weight vector . We got an accuracy of 70%(42 out of 60 correct predictions).

1.2 LASSO

LASSO is a shrinkage and selection method for linear regression. It is defined as squared error loss function with bound on sum of absolute value of each weight vector entry. In simpler terms it is linear regression with l_1 used as regularizer. We mathematically define it as

$$L(\lambda, W) = \arg\min_W \sum_{n=1}^{N} (y_n - W^T x_n)^2 + \lambda ||W||_1$$

For each 218 feature we learned weight vector W. With this we got an accuracy of 68%(41 out of 60 correct predictions).

1.3 Elastic Net

Elastic net is the hybrid of ridge regression and LASSO method. We define Elastic Net

$$L(\lambda_1, \lambda_2, W) = \arg\min_W \sum_{n=1}^{N} (y_n - W^T x_n)^2 + \lambda_1 ||W||_2^2 + \lambda_2 ||W||_1$$

. We can write it in term of optimization problem as

$$L(\lambda_1, \lambda_2, W) = \arg\min_W (Y - W^T X)^2 \;\; S.t \;\; (1-\alpha)|W|_1 + \alpha |W|^2 \leq t \;\; t > 0 \;\; , \alpha = \lambda_1/(\lambda_1 + \lambda_2)$$

For each 218 feature we learned weight vector W_i. With this we got an accuracy 65%(39 out of 60 correct predictions).

2 Multi-class SVM

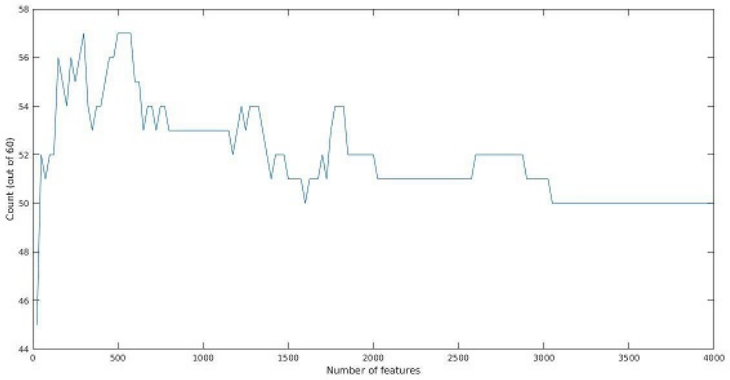
In multiclass SVM we train a classifier per class(in our case 60 classes). In training classifier, let say for class A, we consider all example of class A as positive and rest all as negative. Now at testing time,for each test example we calculate score for each class. The score is calculated by following rule,

$$f(X) = \sum_{j=1}^{N} \alpha_j y_j G(X_j, X) + b$$

where $(\alpha_1, \alpha_2, ..., \alpha_N, b)$ are estimated SVM parameter $G(x_j, x)$ is the dot product in the predictor space between x and the support vectors, and the sum includes the training set observations. We got an accuracy of 81% (48 out of 60 correct predictions) by this method.
Note: We used **predict** function of matlab to implement this method.We also applied the whole algorithm on raw data(without any feature selection/extraction).

3 Feature selection using NCA for classification

We apply feature selection using Neighborhood Component Analysis to select most relevant features. After selecting features we train our model by those features only. Finally we used only those features of test data to make prediction. In our implementation we select top k(from 25 to 4000 in interval of 25) features and plot a graph between no of features and Count value(out of 60).



We get a highest accuracy of 95% by using NCA for feature selection.
Note: We used **fscnca** function to perform feature selection.

4 Joint Matrix Factorization

The idea behind joint matrix factorization is basically to combine two different kinds of information about same object into one and finally use this combined information to predict over test data. So in this paradigm we will try to find a combined information matrix A ($300 \times$ l, l is latent dimension),that is derived from both the fMRI data and word feature data. We preprocess the given data in step 1 and apply algorithm consequently as follows:
1) Arrange word feature matrix data such that each row of this matrix corresponds to the same word (noun) for which brain (fMRI) voxel intensities recorded. (This step gives us Y_feature(300×218) matrix.
2) We used alternating gradient descent over three matrices to evaluate their value such that recon-

struction error is minimize w.r.t following function.

$$\min_{A,D^{(c)},D^{(b)}} \sum_{i=1}^{w} |X_i - A_{i,;} \times D^{(c)}||^2 + \sum_{i=1}^{w'} |Y_i - A_{i,;} \times D^{(b)}||^2 + \lambda||A||_2 + \lambda_a||D^{(a)}||_2 + \lambda_b||D^{(b)}||_2$$

Subject to following constraints :-

$$D_{i,;}^{(a)} D_{i,;}^{(a)^T} \leq 1, \forall 1 \leq i \leq l$$
$$D_{i,;}^{(b)} D_{i,;}^{(b)^T} \leq 1, \forall 1 \leq i \leq l$$

Note: These constraints are needed to avoid solutions where A matrix is made arbitrarily small by making D matrices large.
3) Updates for D^a and D^b matrices are done column wise. They simply reduce to sqaured error linear regression penalised with l_2 regularizer.
4) Updates for rows of matrix A are hard to come up with a closed form solution so we have approximated the calulations by following concatenations over X_train and Y_features (column wise) to get X_new (300$\times$ 21K+218) and from concatenation of D^a and D^b to get D_new (1 $\times$ 21K+218) matrix. Now we use gradient descent to optimize for A with following loss function.

$$\hat{A} = \arg\min_{A} ||X\_new - A * D\_new|| + \lambda||A||_2$$

Prediction accuracy with above procedure for original data is very low and never exceeding 55So we used X_feature_selected instead of X_train, created with 4000 top ranking dimensions of X_train. Following Results was observed for above settings with maximum 60% prediction accuracy of 2-vs-2 test.

Number of iterations	Latent Dimensions	Count value(Out of 60)
30	200	29
10	400	33
40	400	36
10	800	36

5 Ensemble Methods

5.1 Bagging ensamble of regression tree

The basic outline of this method is as follows
First Generate a regression model on fMRI data and one of the word feature.Then using these 218 ensambles to predict the word feature of a given test data. Now classify using these predicted word features.
One of the approaches of using ensemble is bagging. It is a fairly straightforward approach where each training data is given the same importance and an average of the prediction generated by each regression tree is taken. Following the results are obtained by this approach.

Number of Trees	Sampling with Replacement	Count value(Out of 60)
2	yes	42
3	yes	43
7	yes	44
9	yes	46
10	yes	42

5.2 Adaboost on regression tree

Basic outline of this methods is same as above. The only difference is that we are boosting the ensemble of regression trees using a variant Adaboost method. This variant uses least square boosting[3].Since this is a boosting method for decision trees, it may tend to overfit. So we started using this method on decision stumps(trees with depth one). We obtained following results by this approach.

Number of iterations	Count value(Out of 60)
2	38
3	47
7	48
9	49
10	47
12	48
15	50

References

[1] Alona Fyshe, Partha P Talukdar, Brian Murphy, Tom M Mitchell1 "Interpretable Semantic Vectors from a Joint Model of Brain- and Text- Based Meaning" (2014)

[2] Julien Mairal, Francis Bach, J Ponce, and Guillermo Sapiro. 2010. "Online learning for matrix factor- ization and sparse coding". The Journal of Machine Learning Research, 11:1960.

[3] Jerome Friedman Trevor Hastie Saharon Rosset Robert Tibshirani and Ji Zhu "Discussion of Boosting Papers" (2003)

[4] Jacob Goldberger, Sam Roweis, Geoff Hinton, Ruslan Salakhutdinov "Neighbourhood Components Analysis" (2005)

[5] Hui Zou and Trevor Hastie "Regularization and Variable Selection via the Elastic Net" (2004)