Predicting the Word from Brain Activity

Aman Garg 14073 | Apurv Gupta 14124 | Ayushya Agarwal 14168 | Gopichand Kotana 14249 | Kushal Kumar 14346

Abstract

In this project, our objective was to predict the word a person is thinking of, based on the fMRI scan data of his brain. For this we had a set of 300 fMRI scans of 5 different persons thinking of one of 60 words one at a time, on which we trained various machine learning models and applied various different approaches to best predict the words in the test set of 60 fMRI images; and compare the results for each model. By establishing relationships, testing different models and selecting important features in the data, we were able to achieve an accuracy of up to **95%** on the test data based on the pair wise prediction rule as specified.

1. Introduction

One of the motivations to study brain imaging is to answer if it is possible to tell what someone is currently thinking based only on measurements of their brain activity. In this project we deal with a very simplified version of this. Recent advances in human neuroimaging have shown that it is possible to accurately decode a persons conscious experience based only on non-invasive measurements of their brain activity.[1]

Another question cognitive and linguistic scientists want to answer is, does the inclusion of brain data improve semantic representations learned from corpus data ? And, if brain activity could replace corpus data as input to a Vector Space Models representing lexical meaning by assigning each word a point in high dimensional space, and contemporary imaging techniques allow us to attempt this.[2]

Variety of experimental results have led to competing theories of how the brain encodes meanings of words and knowledge of objects, including theories that meanings are encoded in sensory-motor cortical areas and theories that they are instead organized by semantic categories such as living and nonliving objects.[3]

Conventional approaches seek to determine how a particular cognitive state is encoded in brain activity, by determining which regions of the brain are involved in a task. This is achieved by measuring activity from many thousands of locations in the brain repeatedly, but then analyzing each location separately. The sensitivity of human neuroimaging can be dramatically increased by taking into account the full spatial pattern of brain activity, measured simultaneously at many locations. Such pattern-based or multivariate analyses have several advantages over conventional univariate approaches that analyze only one location at a time. First, the weak information available at each location can be accumulated in an efficient way across many spatial locations. Second, even if two single brain regions do not individually carry information about a cognitive state, they might nonetheless do so when jointly analyzed.[1]

2. The Problem statement and the data-set

We have a data set X_{train}, which is a $(300 * 21764)$ dimensional matrix representing voxel intensities of parts of brain from 300 samples using fMRI imaging. Each voxel represents a specific brain areas response out of several such responses, which occur during interpreting a word. Our task is to understand which of them are important to determine the words at hand and then predict a new word, based on our learning. We need to predict the word for X_{test} which is `(60x21764)` test data, after training a model. We have also been provided with a word semantic features matrix which contains 218 semantic features for 60 words. The (i,j)th value of this matrix represents how well the i'th word is correlated to the j'th cue(or, simply speaking the j'th question) and these correlation values are derived from a text corpus.
The immediate challenge with the data set is that the number of training examples is quite small and the number of features for each training example is very high. Naively using classification algorithms would give us very low accuracy, as we will see in some of the methods described ahead. Feature selection/extraction thereby becomes very important. The prediction rule doesnt ask us to do an absolute prediction, instead we have to check whether our algorithm predicts the first word with more probability than the second word.

3. Feature Selection and Extraction

The training data X_{train}, is a $300 * 21764$ matrix. The number of examples is low and using all the features might lead to over-fitting. Also, fmri data generally has noise. Hence, It is very essential to choose features. Each feature selection algorithm has its own advantages as well as limitations so it is important to apply the right algorithm on the data.

PCA- We tried to find principal components in the training data of voxel intensities (300×21764) but because the number of data points was small as compared to the dimension of each data point, our pca algorithm gave principal dimensions as 299 which was the maximum it could. We also used the adjoined matrix of training (300) and test (60) data that out pca algorithm suggested 359 dimensions. We found that using these many principal components the accuracy of prediction according to the given rule shooted up from 21 to 36 in one of our algorithm. From this it was clear that for such data naively using pca was not entirely useful as number of features selected by PCA is bounded by min(N,D) and, we lost on important features in the data.

Lasso - It was an immediate choice as the data is high dimensional. It regularizes as well as selects the features. This makes the model more computationally efficient and also increases the interpret-ability. The accuracy which we obtained by selecting features using this method is close to the PCA. It has some limitations, if there is a group of variables among which the pairwise correlations are very high, then the lasso tends to select only one variable from the group and does not care which one is selected. If there are high correlations between predictors, it has been empirically observed that the prediction performance of the lasso is dominated by ridge regression. [4] Even though classifiers perform well on the data obtained after applying Lasso, we believed there could be a better feature selection algorithm because the prediction rule is 'easy' and a classifier should perform exceptionally well for such a rule. A tailor made feature selection algorithm is necessary.

Stability selection - This feature selection based on sub-sampling using randomized Lasso. The idea is to apply Lasso on 5000 subsets with each subset consisting of 75% of training examples which are randomly chosen. After repeating the process 500 times, the selection results are aggregated. Each of the selected feature is given a score, each time it is selected. We can expect strong features to have high scores, since they are always selected when possible. Weaker, but still relevant features will also have non-zero scores, since they would be selected when stronger features are not present in the currently selected subset, while irrelevant features would have scores zero (almost), since they would never be among selected features. This selected data has been tested using softmax which gave identical results to the data obtained from other feature selection algorithms. But on other classification algorithms this data has performed poorly than the data obtained from other feature selection algorithms hence, this was not used for the final predictions.[5]

Sparse PCA - PCA seeks the linear combinations of the original variables such that the derived variables capture maximal variance. It does feature extraction by forming linear combination of input variables to form new variables which correspond to the direction of maximum variance. Since the number of examples available is low, pca could only reduce the dimensions to the order of the number of examples. Sparse pca reduces the dataset to a smaller number of dimensions while still capturing most of the variance. In the fMRI dataset, the dimension of the data (21764) was much higher than the number of training example (300), and hence should have performed better than our standard PCA.Sparse PCA was modelled as a nonlinear eigenproblem and efficiently be solved by a nonlinear Inverse Power Method, as mentioned in one of the research papers. This Sparse PCA reached its local maximum at about 50 features & performed worse than our standard pca which had reduced the data to 299 dimensions.[6]

Voxel Stability Criterion - The voxel stability criterion was given in [3]. The data consisted of fMRI scans of brains of 5 individuals as they were shown 60 words. The feature selection method began by arranging the value of every voxel in the i_{th} column of the 300*21764 data matrix in a 5*60 matrix, This matrix corresponds to the activation of that particular region of the brain of different individuals as they were shown the different words. The voxels from one individual were stored in one row and those from one word were stored in one column. Now the correlation between the pairs of rows of this constructed matrix was calculated and the average value of this correlation stored. This process was repeated for each voxel column. The order of importance of the dimensions was based on the average correlation values. Those dimensions that showed a high correlation in the rows of the constructed matrix were considered to be more important than those with lower values of correlation. This was done because the correlated dimensions showed consistency, which was indicative of information stored in them. The uncorrelated dimensions just corresponded to the voxels that were affected by noise.

From this ordered dimension space we could choose the best k features that we wanted to choose. This can be considered as a form of Filter feature selection method.

4. Algorithms implemented:

Multiclass Logistic Regression- This algorithm is most intuitive to the given problem statement. The prediction rule given can be directly applied, as we can get log odds of all the classes for every test example once we have trained the model. This can be seen in the performance while using One vs All scheme (the class is fitted against all the other classes) using liblinear solver and L1 penalty. When all the features are considered (No regularization) (300, 21764) it only classifies only 7 out of 60 correctly. This shows that the data overfits and feature selection is required. Lasso is an immediate choice. As we start doing feature selection (regularization using lasso) the accuracy increases, 21642 features: 15 out of 60; 15465 features: 37 out of 60; 12543 features: **52 out of 60**; 7399 features: 45 out of 60; 4215 features: 45 out of 60; 2996 features: 44 out of 60; 1409 features: 46 out of 60; 759 features: 45 out of 60; 225 features: 39 out of 60; 44features: 40 out of 60; 16 features: 36 out of 60. This shows that training a classifier on the entire feature set is a bad idea as the relevant features seem to be very less compared to the given number of features. Using multinomial scheme (the loss minimised is the multinomial loss fit across the entire probability distribution), Newton conjugate gradient solver and L2 penalty predicts, with 11040 features: 51 out of 60, and with very high regularization (9689 features) 44 out of 60 correctly. Feature selection is done on the training data using PCA which reduces the 21764 dimensions to 299 on which applying One vs All scheme using liblinear solver and L2 penalty gives 51 out of 60 correct predictions. Using Newton conjugate gradient solver and a multinomial scheme and L2 penalty gives 48 out of 60 correct predictions.

Multinomial solver learns a true multinomial logistic regression model , which means that its probability estimates should be better calibrated than the one-vs-rest setting. The optimization problem is decomposed in a one-vs-rest fashion so separate binary classifiers are trained for all classes.

Multiclass Support Vector Machines- An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap using hyperplanes such that the gaps are as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall. Theoretically, an SVM would work effectively on datasets in which the number of dimensions is higher than the number of examples, making it easier to find separating hyperplanes. Since we have a small training dataset which might be imbalanced as well, we applied a one vs all svm approach which used K(K 1)/2 binary support vector machine (SVM) models, where K is the number of unique class labels. After doing PCA pre-processing, a hard label was predicted for each of the test data. Using the word_feature_centered matrix, a score was calculated, which represented the L1 distance of the two candidate labels from the hard label, based on their features in the word_feature_centered matrix. Our prediction was correct if we got a lower score for the first word. An accuracy of 36 words was obtained using this method. However, when we used multiple output linear regression as the feature selection step instead of the usual PCA feature extraction, an accuracy of **42 out of 60** was obtained.

Naive Bayes- It is a supervised learning algorithm based on applying the Bayes Theorem with the naive assumption of independence amongst the predictors. The class-conditional independence assumption is a naive assumption for practical reasons, as it is

almost impossible to get a set of predictors which are completely independent to each other. In our dataset, this was clearly violated, since the words like ant - bee , church - barn in the dataset are related to each other but still it was expected to perform well as seen in one of the research papers[8]. Still, naive Bayes is able to obtain 76.66% accuracy. This algorithm was first directly applied to the dataset with no pre-processing on the data, which gave an accuracy of 33 out of 60 words, but it increases to 46 out of 60 words after applying Dimension Reductionality through PCA, which reduced the dimension of the input vector to 299. Feature selection through Voxel Stabilization slightly improved the accuracy to **49 out of 60** words when 1247 features were selected. Since the dataset has less training data and relatively large predictor classes, assumption of class-conditional independence of the predictors allows the Naive Bayes classifier to give relatively accurate classification even with less training data than other classifiers.

Linear Discriminant Analysis- It is a classic classifier which learns linear decision boundaries, assuming that different labels generate data based on different Gaussian distributions. The model assumes that each class generates data using a multivariate normal distribution, that is data has a Gaussian mixture distribution. Each class has it's own mean but the same covariance matrix. To predict the classes of new data, the trained classifier finds the class with the smallest misclassification cost. We trained a classifier, which estimates the parameters of a Gaussian distribution for each label. A prediction is incorrect or correct based on the same L1 norm score rule described in SVM above. An accuracy of **43 out of 60** is obtained using this method after using PCA as a pre-processing method.

5. Learning a semantic map:

This algorithm gave remarkably good result, and the idea is based on a research paper.[3] This computational modeling framework is based on two key theoretical assumptions. first, it assumes the semantic features that distinguish the meanings of words are reflected in the statistics of their use within a very large text corpus. This assumption is drawn from the field of computational linguistics, where statistical word distributions are frequently used to approximate the meaning of documents and words. Second, it assumes that the brain activity observed when thinking about any word can be derived as a weighted linear sum of contributions from each of its semantic features. The linearity assumption is consistent with the widespread use of linear models in fMRI analysis.

In this algorithm we tried to learn a map from voxel intensities to the corresponding semantic features of the word that these voxel intensities represented. First, we created a $(300 * 218)$ matrix F by picking up semantic values corresponding to the output word of each of the 300 training examples. Then we learned a map, M from X_{train} to the matrix F by a simple rule: $X_{\text{train}} \times M = F$, which implied,

$$M = (X_{\text{train}} X_{\text{train}}^T)^{-1} (X_{\text{train}}^T)^{-1} F$$

But the problem we encountered with this method was that since this method required finding inverse of $X_{\text{train}} X_{\text{train}}^T$ which was (21764×21764) this method would be computationally inefficient. Also to avoid any invertibility problem we used *pinv* function to find the inverse. Hence feature selection was a key to working of this method. We first implemented this using the 299 principal components learned from X_{train} using PCA algorithm which

gave the prediction accuracy of 21/60. Then on evaluating this on 259 dimensions from PCA on the adjoined matrix of X_{train} and X_{test}, we got an accuracy of 36/60.

Voxel stability feature extraction algorithm helped us choose the number of voxel intensities we consider sufficient for the data set. Since the data set was small enough we used the test data as our cross validation data to learn the best number of such features to give a good accuracy. It turned out that on selecting 1443 voxels we could get an accuracy of **57/60** according to the prediction rule, when we use L1 norm to find closeness of test word in the semantic feature space representation from the two test words that were given . The actual prediction made by the algorithm using 1-NN approach in the semantic feature space after obtaining the semantic feature space mapping of the test image through regression as done previously turns out to be 17/60 which is reasonable for the simplicity of the algorithm and the still significant dimension size. The absolute accuracy may be made better by better prediction rules than 1NN. The above results though obtained without regularizing the loss function proved to be correct even when the L2 regularizer was used.

6. Zero Shot Learning

The method of classification that dealt with learning a map from the voxel space to the semantic features space also implemented a zero shot approach to classification implicitly.

For classifying words on which the model had not been trained before the only requirement was that the semantic features of the word that is encountered should be known. This is a requirement that is easy to meet as there exist a lot of computational methods to develop semantic features of different words from a huge text corpus. Even though the use of these computationally generated semantic features leads to lesser accuracy than when manually generated semantic features are used, these computational feature spaces win out on the sheer number of words that they map.

With this assumption of knowledge of the semantic feature space value of the word that we are classifying, but have not trained the model on, we can see why the above mentioned classifying method will work. The fMRI image data shall be mapped into the semantic feature space using the mapping that the model has learned through the training data. The mapping has been assumed to remain constant which seems intuitively to be a valid assumption. Now these generated semantic space features can be used in classification using methods like the *1-NN*. The problem is transformed from one which we have not encoutered (*the fMRI image*) to one in which data is present for classification (*the semantic features of the word are already known*)

We implemented this by removing 2 words ,at random, completely from the training data set i.e.converting the data set into a 290*21764 matrix and then selecting relevant features as well as obtaining the mapping from the voxel space to semantic feature space. This obtained mapping was then used on the original test data set. The results of the classification showed that the model in its solution to 'predicting the correct out of the given two words' problem, correctly predicted the two words that had been left out while training the model. Thus 'zero shot training' was effectively implemented.

7. Conclusion

The defining feature of this project was the large dimensionality of the data. It was a huge impedance in the classification of the fMRI data as it led to huge computational costs as well as inaccuracies due to noise. After dealing with this through various methods we focused on predicting a word solely based on the brain-wide fMRI activations of the given word. Considering the poor accuracies that we were getting in the implemented classification methods we tried to incorporate the semantic features of words for prediction. This inclusion improved the prediction significantly. we can thus conclude that there is a huge similarity in the way a brain distinguishes meanings of words and semantic features of words obtained from large text corpus and that different individuals visualise different words in the same manner in terms of pure physiology in the brain.While we were implementing the different classification models we cross verified the different feature selection/extraction methods and found that the voxel selection based on voxel stability criterion method gave the best results.The best classification method implemented gave us an accuracy of 57/60 in the selecting the correct one out of the two given problem and an accuraccy of 17/50 in terms of hard prediction.
This project has helped us explore an area of research which we were oblivious to. It is concerned with a very important and interesting objective and, further research will help us understand how perceptual and cognitive states are encoded in the human brain.

Technologies Used:
Python : scikit-learn, Ipython, scipy, numpy, theano
Matlab

References

[1] John-Dylan Haynes and Geraint Rees *Decoding mental states from brain activity in humans* 2006: Nature

[2] Alona Fyshe, Partha P. Talukdar, Brian Murphy and Tom Mitchell *Interpretable Semantic Vectors from a Joint Model of Brain- and Text-Based Meaning* Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014, long paper)

[3] T. M. Mitchell, S. V. Shinkareva, A. Carlson, K.M. Chang, V. L. Malave, R. A. Mason, and M. A. Just *Predicting Human Brain Activity Associated with the meanings of Nouns* Science, 320, 1191, May 30, 2008. DOI: 10.1126/science.1152876

[4] Hui Zou and Trevor Hastie *Regularization and variable selection via the elastic net* J. R. Statist. Soc. B (2005)67, Part 2, pp. 301320

[5] Nicolai Meinshausen, Peter Buhlmann *Stability selection* Journal of the Royal Statistical Society: Series B Volume 72, Issue 4, pages 417-473, September 2010

[6] M. Hein and T. Bhler *An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA.*

[7] M. Palatucci, D. Pomerleau, G. Hinton, T. Mitchell *Zero-Shot Learning with Semantic Output Codes* Neural Information Processing Systems (NIPS),2009

[8] Shinkareva SV, Mason RA, Malave VL, Wang W, Mitchell TM, et al (2008) *Using fMRI Brain Activation to Identify Cognitive States Associated with Perception of Tools and Dwellings.* PLoS ONE 3(1): e1394. doi:10.1371/journal.pone.0001394

[9] Abdelhak Mahmoudi, Sylvain Takerkart, Fakhita Regragui, Driss Boussaoud, and Andrea Brovelli, Multivoxel Pattern Analysis for fMRI Data: A Review, Computational and Mathematical Methods in Medicine, vol. 2012, Article ID 961257, 14 pages, 2012. doi:10.1155/2012/961257