Applying Learning Methods to Brain Imaging Data

Samuel Chan, Farshad Darvish, Henry Zou
Tufts University
419 Boston Ave, Medford, MA 02155
`Github:` https://github.com/sauch168/CS135-Project-A2

Abstract

In our study, we applied machine learning algorithms to classify the cognitive states on a data set of 6 human subjects, using observations of their brain activity during functional Magnetic Resonance (fMRI). Classifying such cognitive states is a fundamental topic in the fields of cognitive science and clinical applications as doing so constitute virtual sensors of the hidden cognitive states. Previous studies on this data set have reached around 80% classification accuracy. In this study, we will try to extend that line of research by employing the design of various machine learning algorithms, including Gaussian Naive Bayes, Support Vector Machine, and k Nearest Neighbors, to train multiple subject classifiers. After applying these learning approaches, we reported the experimental results, and then analyzed and compared the success of these algorithms for classification of the fMRI data sets.

1 Introduction

The development of fMRI has allowed it to be possibly to observe neural activity of the entire human brain with high spatial resolution. An fMRI is an experiment that produces time-series data to represent the activity of the brain as a collection of 2D slices of the brain. Multiple 2D slices are captured in order to form the 3D image. Altogether, the fMRI time series represent a high-resolution 3D movie of the activation across the brain. fMRI is used as it one of the best methods to view activity across human brain.

In particular, researchers have used fMRI to perform studies to investigate cognitive tasks, such as reading and puzzle solving. These researchers use fMRI to identify which regions of the brains are being activate on averages whenever the human subject performs a cognitive task. By averaging the fMRI observations over multiple time intervals, researchers are able to tell when a subject performs a cognitive task. In light of this, we tackling the problem of training classifiers to accurately classify the subject's cognitive state at a single instance or interval in time. Doing so, will allow use to use these classifiers as virtual sensors of hidden cognitive states.

2 functional Magnetic Resonance Imaging

fMRI is a technique for constructing three dimensional images of the brain actively through a time period. It does this by measuring the ratio of oxygenated hemoglobin in the blood to the deoxygenated hemoglobin, which conveys a blood oxygen level dependent (BOLD) response. This response is then used as an indicator of neural activity as neural activity influences blood oxygen levels.

3 Data Set / Experiment Background

The data set that contains a time series of images of brained activation, which are measured using fMRI (one image every 500 msec). During that time, 6 human subjects performed 54 trials of a sentence-picture comparison task, which the task of includes reading a sentence, observing a picture, and determining if the sentence correctly describes the picture. Each trials takes about 30 seconds and each image is around 5,000 voxels. To analyze the data we use various machine learning algorithms to makes predictions for the two case studies: sentence versus picture, and syntactic ambiguity. Note that we restrict the classifier input to the 7 regions of interest (ROIs) in order to be most relevant.

3.1 Case Study 1: Sentence versus Picture

This Case Study includes when during the trail, the subjects were shown a sentence and a picture, and then asked if the sentences described the corresponding picture. This was done in order to determine when the subject is considering a sentence vs a picture during a time interval. Half of the trials had the picture presented first, then the sentence, which we will refer as the SP data set. The latter half of the trials had the sentence presented first, then the picture, which we will refer as the PS data set.

In our report, given a sequence of n images collected during a contiguous time interval, we trained various classifiers to determine whether the subject was viewing a picture or sentence during the time interval. Form of classifier:

$$f : \langle I_1, \ldots, I_n \rangle \rightarrow \{Picture, Sentence\}$$

where I_i is the ith image captured during the interval, and maps to either a picture or sentence. Note that I_1 is the first image captured were the picture or sentence was presented.

3.2 Case Study 2: Syntactic Ambiguity

This Case Study includes when the subjects were presented with an ambiguous and an unambiguous sentence, and then either answered yes or no to the corresponding sentence. An example ambiguous sentence is "The experienced soldiers warned about the dangers conducted the midnight raid." An example of an unambiguous sentence is "The experienced soldiers spoke about the dangers before the midnight raid. This data set is referred as SA. Form of classifier:

$$f : \langle I_1, \ldots, I_n \rangle \rightarrow \{Ambiguous, Unambiguous\}$$

where I_i is the ith image captured during the interval, and maps to either ambiguous or unambiguous. Note that I_1 is the first image captured were the sentence is first presented.

4 Data Structure

In this study, we used the data set from StarPlus fMRI data. The data set contains six fMRI .mat files that can be loaded directly to Matlab, which each of the file represents the fMRI results for a human subject. The data set is split into three main variables: meta, info, and data.

4.1 meta

The meta variable provides information about the overall data set. Primarily, it includes information on the study, trials, and dimensions. Some relevant features:

- Subject: identifier for the human subject
- Number of trials (ntrials) : 54
- Snapshots (nsnapshots): 2800
- Number of voxels (nvoxels): 4634
- Dimensions (x,y,z): 64 x 64 x 8

4.2 info

The info variable provides information about the experiment in terms of the trials. An info is represented by a 1 x 54 struct array, describing the 54 trials. These time intervals described by the info, corresponds to the trials where the subject views a picture and a sentence, and press a button to indicate whether the sentence correctly described the picture. Some relevant features of the info variable were

- Cond: Describes the condition of the brain activity (takes values of (0,1,2,3))
  - Cond = 0 indicates the data in the segment should be ignored
  - Cond = 1 indicates the segment is a rest interval
  - Cond = 2 indicates the interval is a sentence/picture trial where the sentence is not negated
  - Cond = 3 indicates the interval is a sentence/picture trials in which the sentence is negated
- firstStimulus: Describes whether a picture or sentence was shown first in the trial (takes values of ('P', 'S')). This is primarily used for the Case Study 1: Sentence versus Picture
  - firstStimulus = 'P' indicates a picture was presented before the sentence. Note that this were done for the first half (27) trials
  - firstStimulus = 'S' indicates a sentence was presented before the picture. Note that this were done for the latter half (27) trials
- actionAnswer: Describes if the subject should answer the question for the trial (takes values of (-1, 0)). This is primarily used for the Case Study 2: Syntactic Ambiguity.
  - actionAnswer = 0 indicates the subject is expected to answer either yes or no
  - actionAnswer = -1 indicates the subject should not answer, which is a rest trial.

4.3 data

The data variable provides information about the raw observed data from the fMRI. data is a [54x1] cell array, where each cell represents a trial in the experiment. Notably, the element data$\{x\}(t, v)$ gives the fMRI observation at voxel v at time t for trial x, where the full image at t can be given by data$\{x\}(t, :)$.

5 Learning Methods

To classify cognitive state of activity of the human brain, we will use machine learning algorithms in the form of:

$$f : \langle I_1, \ldots, I_n \rangle \to \{Cognitive\ State\}$$

where I_i represents the ith fMRI image collected. The classifier training methods performed during the study are Gaussian naive Bayes Classifier, Support Vector Machine, k Nearest Neighbor.

5.1 Gaussian Naive Bayes (GNB)

In this study, we will use the Gaussian naive Bayes classifier to estimate the probability distribution of fMRI observations givens the cognitive state of the subject. Naive Bayes Classifier is a classifier that assumes the features in the training data are independent. In order words, the presence of a particular feature in one class is unrelated to the presence of any any feature. Bayes theorem gives the way of calculating the posterior probability $P(c|x)$ from $P(c)$, $P(x)$, and $P(x|c)$

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

where

- $P(c|x)$ is the posterior probability of class (c, target) given predictor (x, attributes)
- $P(c)$ is the prior probability of class
- $P(x|c)$ is the likelihood which is the probability o predictor given class
- $P(x)$ is the prior probability of predictor

To perform Gaussian Naive Bayes on our fMRI data set, we used matlab's implementation of Gaussian Naive Bayes.

5.1.1 How Naive Bayes works

1. Convert the data set into frequency table
2. Create Likelihood table by finding probabilities
3. Use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of the prediction

5.2 Support Vector Machine (SVM)

Support Vector mechanic is a supervised learning algorithm that is used for classification and regression analysis.

To perform SVM on our fMRI data set, we used matlab's implementation of SVM with the linear kernel and the rbf (gaussian) kernel.

5.2.1 How SVM works

Given a set of training data, each marked to a set of two categories, the SVM algorithm develop a model that assigns new data to one of the two categories (binary classification case). The SVM maps the training data to points in space in order to maximize the width of the gap between the two categories. SVM can perform linear classification, as well as non-linear classification by using the kernel trick, which maps the inputs to high dimensional feature spaces. Additionally, SVM can also be extended beyond binary classification and into multi-class classification.

5.3 k Nearest Neighbor (kNN)

kNN classification is a classifier algorithm where the object is classified by its neighbors. The object will be assigned to the most common label on among its k nearest neighbors. In the case that, $k = 1$, then the object is assigned to the class of its single closest neighbor. To perform kNN on our fMRI data set, we used matlab's implementation with three different cases of $k = 5, 7, 9$

5.3.1 How kNN works

1. Initialize k to a value of the chosen number of neighbors
2. For each data point, calculate the distance between the query data point and the current data point
3. Store the distance and index of the data point to a list
4. Sort the list of distances and indices by distances in ascending order
5. Pick first k entries from the sorted list
6. Return mode of the k labels

6 Feature Selection

When analyzing fMRI data, a challenge that arises is how the data are very high dimensional as an fMRI image contains over thousands voxels, and the training data is relatively sparse (there is only a few dozen training examples). Thus we need to first reduce the apparent dimensional of the learning task, which can be done by discovering useful abstractions of the fMRI signals. Normally, two basic abstractions are used, where the first one is the mean (averaging the activity of all voxels in ROI to get an average voxel), and the second one is the top n active voxels in an ROI under t-test. In our report, we test both of these abstractions (using *average* and *Active(n)*), then a combination of the two (*ActiveAvg(n)*).

6.1 Abstraction methods

Here are the following methods for feature abstraction:

- *Average*. Calculate the mean activity over voxels in ROI. This is done for each ROI, and the ROI means as the input features.

- *ActiveAvg(n)* Select the n most active voxels, then calculate the means of those values. The most active voxels are determine by those whose activity while performing the task varies the most when their activity is at the rest state. This is done for each ROI, and the ROI means as the input features.
- *Active(n)* Select the n most active voxels. These will be the input features.

6.2 Registering Data for considering multiple subjects

When there are multiple subject, there will be different sizes and shapes for the different brains. Thus mapping the voxels in one brain to another is not possible. We used two different methods to produced representation of the fMRI data for use considering multiple subjects:

- ROI mapping: Obtain the voxel data in each of the subjects' brain by using *Average* or *ActiveAvg(n)* method, and then use resulting average activity for each ROI as the normal representation for the subjects. Note that this can also be done by using *Active(n)*, which will essentially be the same as *ActiveAvg(n)*, but without taking the average.
- Talairach coordinates: Transform the coordinate system of each brain so that it is into the coordinate system of a standard brain (Talairach-Tournoux coordinate system). This results in the brains having the same shape and size, with a drawback that he transformation generally is imperfect.

In our report, we will use ROI mapping as our method for feature selection since it allows us to easily apply all of our different feature abstraction methods of *Average*, *ActiveAvg(n)*, and *Average(n)*. Additionally, although the Talairach coordinates method has the benefit of converting all the subject's brain to the same coordinate system, its drawbacks of having imperfect transformation is not ideal for the complexity of the fMRI data.

7 Experimental Results

In our investigation, we trained classifiers to classify cognitive states across multiple human subjects. For each of the following feature selection methods that were performed, each classifier algorithm were individually tested (GNB, SVM, KNN), where KNN consisted of testing 1NN, 3NN, and 5NN. We used six data sets from StarPlus fMRI data.

For each of the three training methods we used, we tested how each feature selection method compared. In the case of using *ActiveAvg(n)*, where n represents the top n active voxels, we tested all values of multiples of 10 between the range 10 to 200. We then used the best result for n as our choice for n in *ActiveAvg(n)* which was 100. Similarity, for *Active(n)*, to get the best value for n, we tested all values of multiples of 10 between the range 10 to 200, and then used the best result for n as our choice for n in *Active(n)*, which was 50.

7.1 Choosing training data and testing data

In our experiment, in order to have a balance between not over fitting the data set, and having enough training data, we choose to split the data sets using two methods:

- Split the 6 data sets into 3 testing sets and 3 training sets.
- Split the 6 data sets into 1 testing sets and 5 training set.

To account for may bias that may occur, we applied our learning methods to all combinations of the data sets. To instance, for 3 testing sets, and 3 training sets, there will be 6 choose 3 combinations, which is 20, and for 1 testing sets, and 5 training sets, there will be 6 choose 1 combinations which is 6. We then used the average accuracy from all the combinations as the accuracy for the corresponding learning method and the corresponding study case.

7.2 Considering Error with the Hinge Loss Function

To analyze the error of our machine learning algorithms, we needed to implement a cost function that incorporates a margin of error for the classification boundary. For our study, we implemented the hinge loss function to identify the margin of errors in the result. The hinge loss function can be described by the equation

$$l(y) = \max(0, 1 - t \cdot y)$$

- t is the intended output that takes values of $-1, +1$
- y is the classifier score
- $l(y)$ is the hinge loss of the prediction y

We choose to use the hinge loss function as our error measurement for our classification methods because it heavily penalizes classifications, and it leads to better accuracy at the cost of sensitivity. Additionally, the hinge loss function is very connected to the support vector machine model, and since we are implementing support vector machines, we thought it would be interesting to compare the errors results between the classifiers with the hinge loss function.

7.3 Results for Case Study 1: Sentence vs Picture

The following table contains the results for Case Study 1, where it SP represents the case where the sentence was given first, and PS represents the case where the picture was given first. (SP + PS) represents the union of the two cases, which we expect to have lower accuracy then the other two columns.
Note that for the following tables, the values on the left represent the results for the 3 testing sets and 3 training sets case, and the values on the right in the parentheses represent the results 1 testing sets and 5 training set case.

Table 1: Average accuracies in Sentence versus Picture study

Feature Method	Classifier	SP	PS	SP+PS
Average	GNB	82.42% (85.00%)	69.92% (76.67%)	76.17% (80.83%)
Average	SVM (Linear)	91.17% (92.50%)	83.83% (85.83%)	64.38% (68.33%)
Average	SVM (RBF)	92.83% (94.17%)	77.83% (77.50%)	64.33% (68.75%)
Average	5NN	64.25% (60.83%)	85.00% (87.50%)	55.63% (54.58%)
Average	7NN	42.40% (40.71%)	23.01% (20.36%)	46.71% (47.92%)
Average	9NN	65.33% (70.00%)	85.58% (86.67%)	56.42% (57.08%)
ActiveAvg(100)	GNB	82.58% (87.50%)	68.17% (77.50%)	75.37% (82.50%)
ActiveAvg(100)	SVM (Linear)	77.67% (80.00%)	78.17% (78.77%)	77.92% (78.33%)
ActiveAvg(100)	SVM (RBF)	81.50% (84.17%)	70.92% (75.83%)	76.21% (80.00%)
ActiveAvg(100)	5NN	66.67% (61.17%)	83.17% (84.17%)	74.50% (76.67%)
ActiveAvg(100)	7NN	36.99% (34.76%)	26.48% (22.74%)	26.48% (22.74%)
ActiveAvg(100)	9NN	67.83% (69.17%)	81.58% (85.83%)	74.71% (77.50%)
Active(50)	GNB	78.42% (76.67%)	47.00% (55.00%)	62.71% (65.83%)
Active(50)	SVM (Linear)	70.00% (73/33%)	58.75% (63.33%)	64.38% (68.33 %)
Active(50)	SVM (RBF)	73.00% (71.67%)	55.67% (65.83%)	64.33% (68.75%)
Active(50)	5NN	36.76% (46.67%)	73.58% (62.50%)	55.63% (54.58%)
Active(50)	7NN	58.29% (53.57%)	35.14% (42.26%)	46.71% (47.92%)
Active(50)	9NN	38.83% (49.17%)	74.00% (65.00%)	56.42% (57.08%)

Here we see that most of our results are significantly higher than the 50% accuracy expected by randomly classifying the cognitive states, some results from the kNN classifier, which possibility suggest that kNN classifier is not suitable for this data. Nevertheless, ignoring, the kNN classifiers, the overall classifiers accuracy of around 80%, and the highest of 94.17% with rbf SVM using *Average*, indicate that it is feasible to train classifiers to distinguish cognitive states of human subjects beyond just the training set.

Comparing the data set of SP, PS, SP+PS, we see that SP consistently had the best accuracy, but PS had the worst. Originally, we expected SP and PS to both be the best and have similar accuracy, as we thought it would be harder to classify the union of the two sets SP+PS. However, due to the significantly lower accuracy rate of PS compared to SP, PS had the lowest overall accuracy, and the union had a better accuracy as the high accuracy of SP improve its. The higher accuracy of the SP data set possibly indicates that the human brain is undergoing a different process when it is first seen a picture (SP data set) vs when it is first seen a sentence (PS data set). Due to the different reactions, our data set reveals that the visual sensors for the SP data set is easier to classify than the PS data set.

Comparing the feature selection methods, we see that *Average* performed the best, and *Active* performed the worse. This possibly suggest that when classifying for the sentence vs picture case, it is more important to consider an average of the voxels for the regions of interest than considering which voxels are the most active, as both of the average feature selection methods performed better than the method of simply choosing the n most active voxels. This is the opposite of what happened in similar studies. These studies consider the active voxel as they hypothesize that active voxels reveal more information about the cognitive state of the brain. The difference from this study and the other study could be due to difference in the methods for determining which voxels are most active.

Table 2: Hinge Error in Sentence versus Picture study

Feature Method	**Classifier**	**SP**	**PS**	**SP+PS**
Average	GNB	0.1763 (0.1421)	0.3035 (0.2316)	0.2399 (0.1869)
Average	SVM (Linear)	1.1712 (1.1358)	1.2404 (1.1954)	1.4508 (1.4278)
Average	SVM (RBF)	1.2452 (1.2016)	1.3341 (1.2815)	1.4658 (1.4489)
Average	5NN	0.4098 (0.3900)	0.2182 (0.1900)	0.4787 (0.5000)
Average	7NN	0.4240 (0.4071)	0.2301 (0.2036)	0.4671 (0.4792)
Average	9NN	0.4298 (0.4056)	0.2427 (0.2120)	0.4729 (0.4824)
ActiveAvg(100)	GNB	0.1828 (0.1516)	0.3244 (0.2351)	0.2536 (0.1933)
ActiveAvg(100)	SVM (Linear)	1.2629 (1.2467)	1.2648 (1.2658)	1.2704 (1.2557)
ActiveAvg(100)	SVM (RBF)	1.2849 (1.2591)	1.3360 (1.2991)	1.3104 (1.2791)
ActiveAvg(100)	5NN	0.3672 (0.3400)	0.2217 (0.2316)	0.3117 (0.2808)
ActiveAvg(100)	7NN	0.3699 (0.3476)	0.2648 (0.2274)	0.3173 (0.2875)
ActiveAvg(100)	9NN	0.3771 (0.3565)	0.2754 (0.2389)	0.3263 (0.2977)
Active(50)	GNB	0.2179 (0.2273)	0.5297 (0.4516)	0.3738 (0.3395)
Active(50)	SVM (Linear)	1.4327 (1.4118)	1.4690 (1.4438)	1.4508 (1.4278)
Active(50)	SVM (RBF)	1.4428 (1.4302)	1.4887 (1.4676)	1.4658 (1.4489)
Active(50)	5NN	0.6045 (0.517)	0.3528 (0.4383)	0.4787 (0.5000)
Active(50)	7NN	0.5829 (0.5357)	0.3514 (0.4226)	0.4671 (0.492)
Active(50)	9NN	0.5849 (0.5389)	0.3609 (0.4529)	0.4729 (0.4824)

Similar to the results of the accuracies, for the feature methods, *Average* had the overall less error, closely followed by *ActiveAvg(n)*, and then *Active(n)* had significantly worst error than the both of them. Notably, unlike in the tables of accuracies, for the classifier methods, GNB had the clear lowest error, followed by kNN, but then both SVM methods had significantly higher error. This is strange as SVM had the best accuracy along with GNB. The large error in SVM could be due to how we choose to use the Hinge loss function as our measurement for error, which the Hinge loss function is closely tied to the model for SVM.

7.4 Results for Case Study 2: Syntactic Ambiguity

The following table contains the results for Case Study 2. Note that there is only one accuracy column for this study case as this is the SA data set, which consider if the subject was reading an unambiguous or ambiguous sentence.

Table 3: Average Accuracies in Syntactic Ambiguity study

FEATURE METHOD	CLASSIFIER	SA
Average	GNB	60.04% (59.58%)
Average	SVM (Linear)	65.00% (67.08%)
Average	SVM (RBF)	63.50% (67.08%)
Average	5NN	58.58% (58.75%)
Average	7NN	60.92% (61.67%)
Average	9NN	58.67% (54.58%)
ActiveAvg(100)	GNB	60.75% (61.25%)
ActiveAvg(100)	SVM (Linear)	63.75% (65.42%)
ActiveAvg(100)	SVM (RBF)	61.12% (60.42%)
ActiveAvg(100)	5NN	55.58% (54.17%)
ActiveAvg(100)	7NN	55.58% (54.17%)
ActiveAvg(100)	9NN	56.46% (56.25%)
Active(50)	GNB	55.77% (57.14%)
Active(50)	SVM (Linear)	55.55% (56.85%)
Active(50)	SVM (RBF)	56.08% (56.19%)
Active(50)	5NN	50.59% (49.88%)
Active(50)	7NN	51.63% (51.19%)
Active(50)	9NN	51.55% (50.95%)

Like in Case Study 1, most of our results are higher than the 50% accuracy expected by randomly classifying the cognitive states, but they are considerably lower than the accuracies from the Case Study 1. This suggest certain cognitive activity are more difficult to classify than others, indicating that some virtual sensors for the cognitive states are more difficult to identify. Since Case Study 2 had a consistently lower accuracies than Case Study 1, it provides evidence that classifying when the cognitive state is asking yes or no is more difficult than classifying when the cognitive state is presented a sentence versus picture. Notably, Case Study 2 did have more consistent result than Case Study 1 with kNN classifier as Case Study 1 had many outlier accuracies with kNN that reach values of around 20%. Altogether, since the accuracies for Case Study are still consistently higher than 50%, where the highest is 67.08% from using SVM with *Average*. The accuracy provide modest evidence that it is feasible to train classifiers to distinguish cognitive states of human subjects beyond just the training set.

Comparing the feature selection methods, like in Case Study 1, *Average* performed the best, and *Active(n)* performed the worse. However, in Case Study 2, the results of *ActiveAvg(n)* were more similar to the *Average*, which makes sense as answering an ambiguous sentence should directly influence the active of the voxels.

Table 4: Hinge Error in Syntactic Ambiguity study

FEATURE METHOD	CLASSIFIER	SA
Average	GNB	0.3984 (0.4003)
Average	SVM (Linear)	1.3923 (1.3802)
Average	SVM (RBF)	1.4305 (1.4124)
Average	5NN	0.4912 (0.4954)
Average	7NN	0.4532 (0.4464)
Average	9NN	0.5867 (0.5458)
ActiveAvg(100)	GNB	0.4057 (0.3943)
ActiveAvg(100)	SVM (Linear)	1.3978 (1.3796)
ActiveAvg(100)	SVM (RBF)	1.4440 (1.4269)
ActiveAvg(100)	5NN	0.3862 (0.3746)
ActiveAvg(100)	7NN	0.4618 (0.4524)
ActiveAvg(100)	9NN	0.5646 (0.5625)
Active(50)	GNB	0.4439 (0.4319)
Active(50)	SVM (Linear)	1.4698 (1.4644)
Active(50)	SVM (RBF)	1.4754 (1.4675)
Active(50)	5NN	0.4912 (0.4954)
Active(50)	7NN	0.4926 (0.4942)
Active(50)	9NN	0.5155 (0.5095)

The results of these errors are very similar to the results of the error in Case Study 1. However, the error of GNB in this study is significantly higher than the error of the GNB in Case Study 1, which had an average error of around 0.20.

8 Analysis

Overall, the accuracies for the studies cases were mostly all above 50%, which indicates that these machine learning algorithms out perform the method of randomly classiflying the cognitive state, which should have expected an average of 50%. Even through our accuracies does suggest that it is feasible to have learning methods classify cognitive states of the human brain, our accuracies could be better, especially for Case Study 2. Since Case Study 1 had the highest accuracy of 94.17%, and Case Study 2 had a the highest accuracy of of about 67.08%, the difference in accuracies possibility imply there is some other feature, other than active voxel, that may be a indicator for cognitive states. Ideally, we want to have these accuracy be around 95%, which we have closely reached in Case Study 1, but not in Case Study 2. Due to the complexity of fMRI data as well as the cognitive states of the human mind, our overall accuracies are reasonable and do provide evidence that machine learning techniques can be efficiently applied to the field of cognitive brain science.

8.1 Comparing 3 testing and 3 training vs. 1 testing and 5 training

Overall the accuracies for the two cases were the similar, but 1 testing and 5 training had an slight improvement. This improvement was typically around 1-2%. For example in the Case Study 1, of the *Average*, SVM (RBF), SP dataset, the resulting accuracy for the 3 vs 3 was 92.83%, and the accuracy for 1 vs 5 was 94.17%, a 1.34% increase.

The slight improvement in the accuracy for the 1 vs 5 case can be explained by how there is more training data for the model to train on. Since we have such limited number of data sets, as we only have data on 6 human subject, having more training data is likely to improvement the accuracy, and not tend to over fit the model by a significant amount.

8.2 Comparing the learning methods

Considering the learning methods, we found that the classifier that worked best were SVM and GNB, which had similar overall accuracies, and the kNN performed the worst.
For Case Study 1, in terms of the highest accuracies, follows:

- SVM (RBf): 94.17%
- SVM (Linear): 92.50%
- GNB: 87.50%
- 5NN: 87.50%
- 9NN: 86.67%
- 7NN: 58.29%

For Case Study 2, in terms of highest accuracies, follows:

- SVM (Liner): 65.00%
- SVM (RBF): 63.50%
- 7NN: 60.92%
- GNB: 60.04%
- 9NN: 58.67%
- 5NN: 58.58%

The results between linear kernel SVM and rbf kernel SVM were almost the same. This is as expected as When the number of features is large, the linear kernel tends to perform very well. In these cases, nonlinear kernels like RBF are not usually a significant improvement over the linear one. Generally, this corresponds to how a linear kernel is generally used when the number of features is larger than the number of observations, and the Gaussian kernel is the reverse.

In Case Study 1, despite 5NN and 9NN having efficient highest accuracies around 85%, the overall accuracies for kNN varied much more than the other classifiers. Strangely, when $k = 7$, the accuracies significantly dropped. The lack of consistently in the results of the kNN accuracies suggest that kNN may not be the most suitable classifier for our task, as SVM and GNB perform better and had much more consistent results.

In Case Study 2, the accuracies ranges were much less varied than Case Study 1. This suggest that for the syntactic ambiguity case, choice for the learning method was not as important as Case Study 1. However, like in Case Study 1, SVM was the machine learning method that performed the best.

9 Conclusion

Altogether, the goal of this study on applying machine learning to fMRI was to better understand and analyze how machine learning algorithms can decode mental states across multiple human subjects. The overall high accuracy in results demonstrated that it is feasible to use machine learning to classify cognitive states of human subjects.

To perform our study, three feature selection methods were explored to reduce the dimensions of the system in order to train multiple-subject classifiers, using the fMRI data. These feature selection methods consisted of *Average*, *ActiveAvg(n)*, and *Active(n)*. By using these feature selection methods, we are able to apply machine learning algorithms to multiple brains, despite each brain being a different size and shape. The machine learning algorithms that were tested were Gaussian Naive Bayes, support vector machine, and k nearest neighbors. Our results reveal high accuracy in Case Study 1, indicating that our machine learning algorithms can feasibly classify the cognitive states of the human brain, However, even through our accuracies in Case Study 2 were above 50%, since the overall accuracy was around 60%, there is much room for improvement and does not provide reliable evidence that our trained classifies can accurately detect cognitive states performing a ambiguous task.

9.1 Future Directions

If we have more time, improving our classifiers on Case Study 2 would be the primarily focus for improvement. Some possible avenues that we could explore to improve this accuracy for this case is possibility changing our feature selection to the case of Talirach Coordinates transformation. Although this method can lead to imperfect transformation it could possibility be better for this study case. Additionally, we can also use different machine learning methods such as Dynamic Bayesian Networks, or Hidden Markov Models.

9.1.1 Limitations

Although our study has successfully provide support that machine learning algorithms can effectively classify the cognitive states of the human brain, it is important to consider the limitations of our study. One of the main limitation of our study was the lack of data sets. We were only given 6 fMRI data sets, where each one represented one human subject. If we had more data to work our, our model could be trained with more data and possibly reach higher accuracies.

10 References

[1] T. M. Mitchell (2004) "Learning to Decode Cognitive States from Brain Images"

[2] Friedman (1997) "Bayesian Network Classifiers"

[3] X. Wang, R. Hutchinson, and T. M. Mitchell (2003) "Training fMRI Classifiers to Detect Cognitive States across Multiple Human Subjects ," Neural Information Processing Systems.

[4] Ramish, J. (2004) Learning common features from fMRI data of multiple subjects. Summer project report

[5] T.M. Mitchell, R. Hutchinson, R.S. Niculescu, F.Pereira, X. Wang, M. Just, and S. Newman, (2004) "Learning to Decode Cognitive States from Brain Images"