Group 9

Kelly Huang
Jeffrey Lin
Swupnil Sahai

The Kaggle Contest

The main goal of this Kaggle contest is to predict the brain's responses to visual images. The Gallant Lab at UC Berkeley provided a training set of 1,500 observations with 10,921 predictors, response values to the training set, and a test set of 370 observations without response values. From this, we created a model to predict responses for the test set observations. A submission to Kaggle takes in our predictions and calculates the difference between our predictions and the actual responses in the test set. Kaggle randomly splits our predictions in half, with half of the predictions being used to give a public leader board ranking and the other half for the final scores. Root mean squared errors (RMSEs) are calculated for each of these halves. A smaller root mean square error corresponds to a better prediction, and therefore a higher ranking on the leader board or final score.

Our Contest

The Gallant Lab at UC Berkeley conducted research on the relationship between visual stimuli and fMRI activity in the visual areas of the brain. This data includes 10,921 transformed variables, which represent features of the 1,500 images shown after being reduced down to a vector of grey-scaled pixels. These variables are obtained from fMRI scans. fMRI brain responses are recorded while subjects viewed novel, natural images. Conventional methods can identify position, orientation, and object category of images if the person has had prior exposure to the same image. This is possible because we already have a baseline comparison of fMRI responses recorded from the first time the subject saw the image. Therefore, instead, researchers at the Gallant Lab focused on the use of novel images to try and identify the position, orientation, and object category of images with no baseline. Natural, novel images are used because they have more complex statistical structures with no baseline comparison.

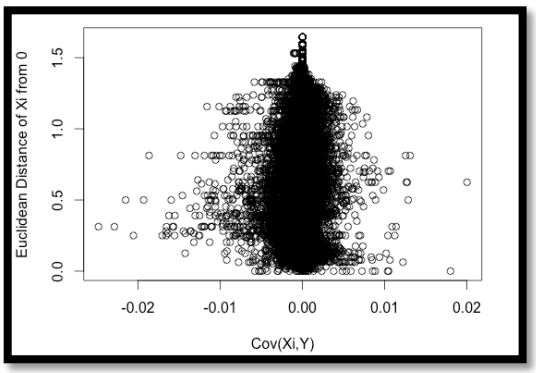
The given predictor variables are 10,921 features of the images shown in the training set and the images shown in the test set. There were originally 16,384 pixels in each image, which were then transformed and reduced down to 10,921 features. The response vector is a single given voxel's response to each image. The training set is comprised of 1,500 images and the corresponding brain activity to the 10,921 features of the images. We're given the voxel's responses to each of the 1,500 images, simplified down to a single number. We used a model built off this training set data to predict the response in a particular voxel in the visual area of the brain when shown a particular image for the 370 images shown in the test set.

Exploratory Data Analysis

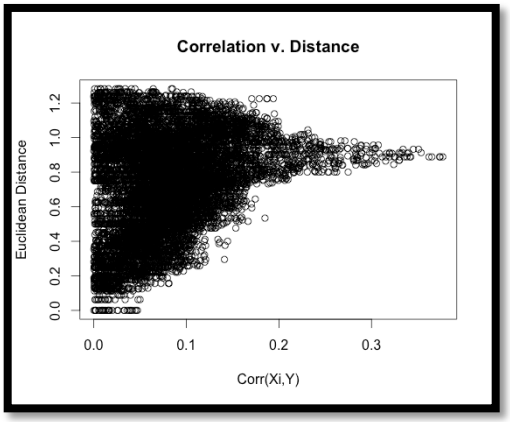
We initially partitioned our training data so that we would have a sample size of 1000 for a training set, 250 for a validation set, and 250 for a sample test set. Unfortunately, since there were so many variables, we could not get an easy grasp of

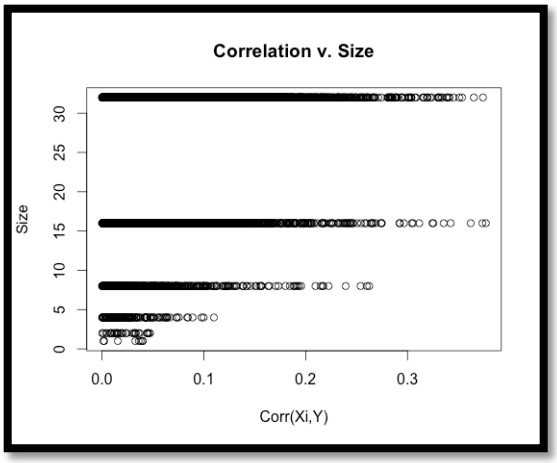
Kelly Huang
Jeffrey Lin
Swupnil Sahai

how the variables may be interrelated. Furthermore, it was difficult to plot the relationships between particular variables and a response, so we could not see if there was any type of specific parametric relationship. Eventually, we decided to look at the correlation between the variables and response, and plot them against the various column descriptions of those variables. Initially, we found that the spread of covariance decreased as Euclidean distance increased (see figure on the right).

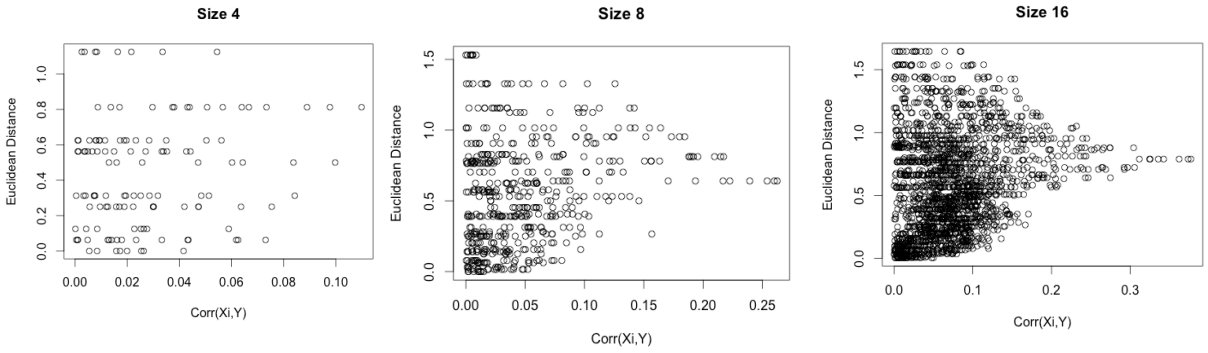


However, plotting the absolute value of correlation versus Euclidean distance (square root of the sum of the X value squared and the Y value squared) gave a very different picture, which confused our analysis somewhat (figure on the left).





Plotting absolute value of correlation versus the "size" description showed somewhat of a trend in that the maximum magnitude was higher as size increased. This gave us impetus to start separating the variables by size and then redo the plots. In doing so, however, we found there was almost no relationship between orientation and correlation within each of the sizes. Furthermore, the plots of absolute value of correlation against Euclidean distance, within each size, also showed a strange relationship:



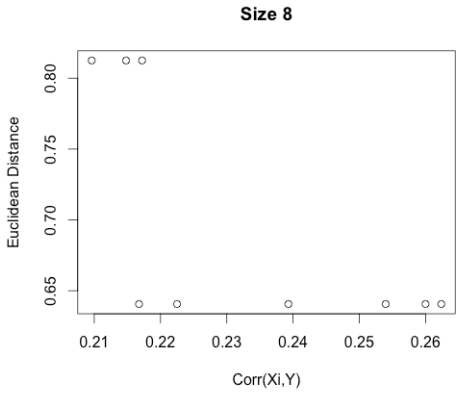
Kelly Huang
Jeffrey Lin
Swupnil Sahai

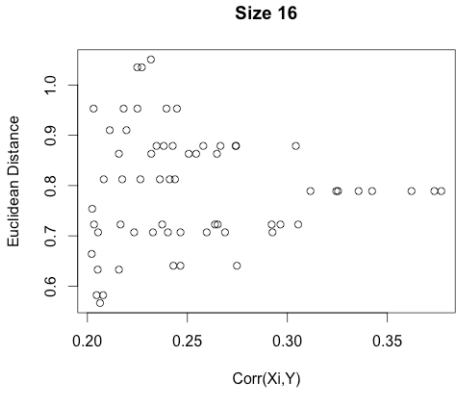
We also found almost no relationship between absolute value of correlation and orientation. As such those plots have not been including in this report.

Variable Selection

Our first step was to clean the columns of the data by removing all columns of the training partition that were all zero. Although this did not imply independence, for our training partition these columns gave no extra information about the response, so it was not useful to keep them when creating the new models. Our first plan of attack was then to install the "leaps" package to run the function "regsubsets" in an effort to find the best subsets of size 1 to 500. As a test, we then ran GBM on all these subset sizes and calculated RMSE on validation set for all subsets. Unfortunately, most did not improve RMSE from those created by the models generated by the clean training partitions.

As such, we decided to go back to our exploratory analysis and remove those variables whose correlation with the response was less than 0.2. This particular number was initially arbitrary and we determined an optimal cutoff value when tuning our GBM model. This cutoff resulted in the removal of all features with size 1, 2, or 4, as well as several other features within sizes 8 and 16. We found the following modified plots for absolute value of correlation versus Euclidean distance.





These plots showed us that for "strong features" (i.e. those with absolute value of correlation greater than 0.2) of size 8, correlation decreased with distance, but the same phenomenon was not as pronounced for strong features of size 16. At this point it may have nevertheless made sense to remove features located past a certain threshold Euclidean distance, but we elected to instead restrict our feature selection to merely a correlation cutoff, as the relationship with distance wasn't as strong as we would have liked.

Group 9

Kelly Huang
Jeffrey Lin
Swupnil Sahai

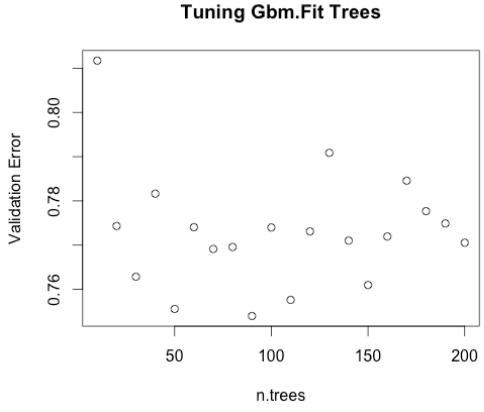
Models

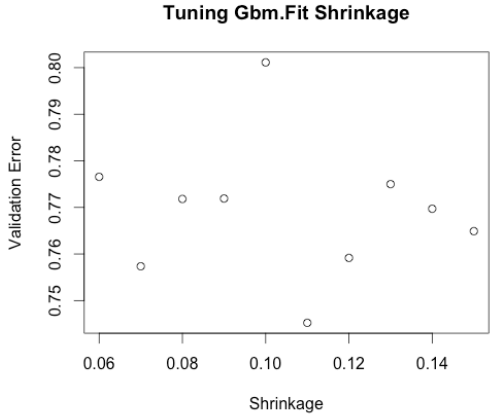
OLS:

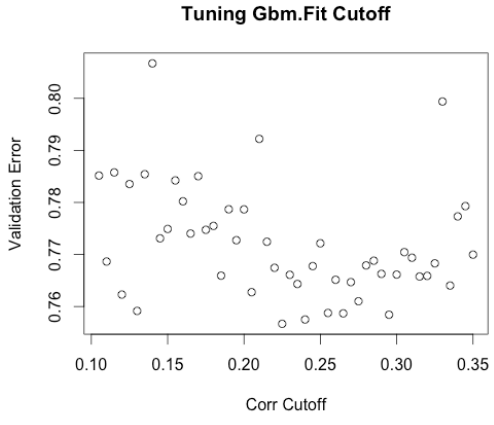
Initially, we had many more variables than observations so there was no unique solution to OLS. Once we narrowed down the variables to approximately the best 300 using the correlation cutoff, then OLS was more feasible. However, the validation error was always significantly higher than for the other models, and even when blending with the other models, our validation error was not being optimized. As such, we did not use OLS in the final model.

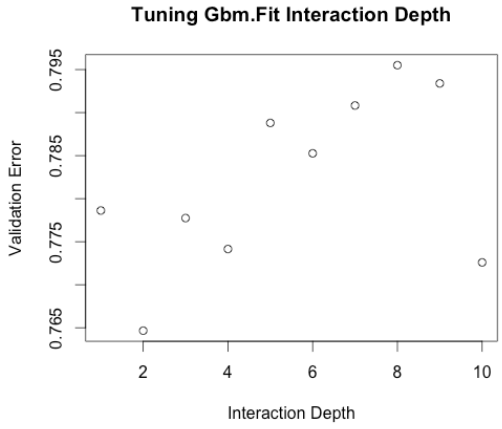
GBM:

Initially, we ran GBM because it was the model that ran quickest for the untouched training partition. Once we narrowed down the variables to around 300 (by removing those with zero covariance with the response, as well as with the correlation cutoff), we again ran GBM and it outperformed all other models that we tried. We then proceeded to tune GBM based on the number of variables we kept, the maximum number of trees to use in the initial run of the model, the shrinkage factor, and the interaction depth. We ran loops to tune the GBM parameters several times and generated several plots of how these parameters affected validation error. Included below are single plots for each parameter generated after just one run of the tuning loop.





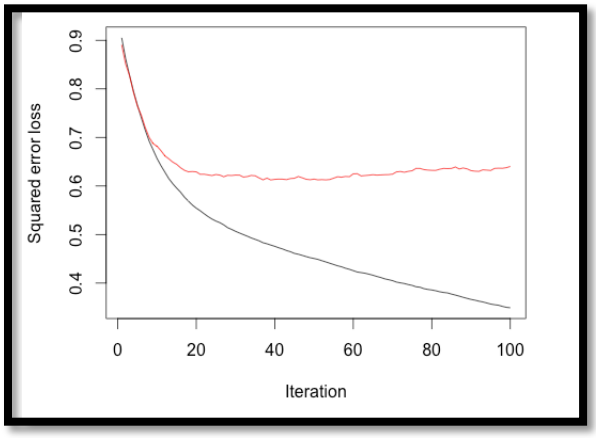




Kelly Huang
Jeffrey Lin
Swupnil Sahai

The plots were repeatedly similar to these after several iterations of the loops. On average, validation partition error was minimized when n.trees=90, shrinkage=0.11, and interaction depth=2 for the initial run of any particular gbm model. After tuning those three parameters, we tuned the correlation cutoff itself and found that, on average, the validation partition error was minimized when the absolute value correlation cutoff was set to 0.225.

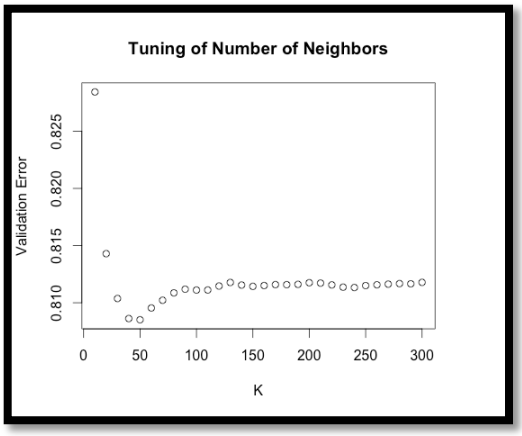
We then ran the gbm function several times using the above parameters until we found a model that most minimized the validation error, resulting in the final plot of our best possible gbm model. The best iteration ended up being 39.



GLMNet-Based LASSO with CV:

As we were unable to get the predict function to work properly using the LASSO model in the lars package, we found a new model under the glmnet package, written by the Stanford authors of our textbook. Using this model, we actually found that inputting the original training partition, rather than inputting the cleaned up training partition with only ~300 variables, gave the minimum validation error. This is likely because the LASSO regression was able to select the best possible subsets more efficiently and systematically then the way we manually selected our best subsets. Nevertheless, even the best possible tuned LASSO model could not outperform our best GBM model on our test partition. Despite this, however, when including this LASSO model in our final blending, we were able to optimize our test partition error.

K-Nearest Neighbors:



Once we narrowed down the variables to around 300 (by tuning the validation error of GBM), we decided to tune k-NN to minimize validation error with respect to the number of neighbors. After several iterations of the tuning loop for k-NN, on average the validation error was minimized when the number of neighbors was 50. We then proceeded to tune the model with respect to the kernel. We received the following validation errors for

different kernel types: 0.8497664 (recatangular), 0.8453364 (triangular), 0.8444132 (epanechnikov), 0.8442451 (biweight), 0.8458203 (triweight), 0.8447197 (cos), 0.8512349 (inv), 0.8515829 (Gaussian). Hence, we decided to use "biweight."

Ridge-Regression, GAM, and Logistic Regression:

These methods were not pursued because we felt more comfortable using the other models based on the experience we had with using them on homework assignments as well as in class during the discussion section. We did, however, attempt to run the GAM model but found several errors and decided it was not worth our time to pursue that model.

Blending:

Using the four tuned models above, we ran OLS to blend all possible combinations of their responses on the test partition and found the test error was minimized when using the k-NN, GBM, and the GLMNet-derived LASSO models' predictions. Fortunately, we actually overestimated the test error (.8034) and were somewhat delighted upon submission to Kaggle (.78971).

Conclusion and Notes

Our initial attempts were poor because we attempted to submit a working model to Kaggle as soon as possible without doing a thorough exploratory analysis. However, in doing so, we completely avoided removing variables from the training data, which made our regressions extremely slow to calculate. Even when we began attempting to winnow out "weaker" variables, we accidentally performed the variable selection on the entire training set when we should have been doing so only using the training partition we initially created, as this analysis was likely biasing our models towards the partitioned validation and test sets, which should have ideally remained "untouched."

In terms of model selection, we were hoping to submit a model using the lars package but ran into constant issues with the predict and predict.lars functions as these functions outputted lists, which we were unable to successfully convert into matrix form. Furthermore, we did not ideal tune the models using smaller increments of the parameter values, as that would have taken too much time – ideally, we would have done this as well.

With respect to exploratory analysis, we would have hoped to look at variance-covariance plots to determine the extent of interdependence among the variables, which may have allowed us to remove possibly redundant variables. Furthermore, we still did not feel like we understood 100% of what the variables

really meant, and this lack of a strong grasp of the data made it more difficult for us to efficiently conduct a productive exploratory analysis.

In conclusion, our best model showed us that the vast majority of the 10921 features were not absolutely necessary to make accurate predictions of the voxel response. In reducing the number of features to less than 300, we were able to not only generate predictions more quickly than with the original data, but we also had more accurate predictions. This was expected, however, as when an individual is shown an image, one would not expect every part of the image to generate a strong response for certain voxels of the brain. Only the most important features of an image are truly captured by the human brain, and it is these few features that truly stimulate the brain. Indeed, just as our code quickly made predictions without requiring or comprehensively understanding all of the test data, so does the brain recognize images without truly understanding them down to each and every pixel.

Division of Work

Swupnil worked on variable selection based on correlation and Euclidian distance as well as on the individually tuning and comprehensive blending of GBM, k-NN, and GLM. He wrote the exploratory data analysis and variable selection portion of the report.

Kelly worked on the lasso and k-NN models and wrote the list and methodology of models used.

Jeffrey worked on variable selection through backward stepwise selection after removing zero columns and the GBM model. He wrote the contest and data introduction and helped write the variable selection portion of the report.