Examining Self-Control Through the BART Study

Chen, Kent `kentschen` Lee, Rachel `reychil` LeRoy, Benjamin `benjaminleroy` Liang, Jane `janewliang` Udagawa, Hiroto `hiroto-udagawa`

December 14, 2015

Abstract

Self-control is an interesting field of behavioral research with broad implications for our day-to-day experiences. Being able to appropriately regulate and check our impulses and reactions to various everyday stimuli is necessary for maintaining health and high-functionality in society. Thus, relating a subjects ability to control risky behavior to an area of the brain has been the focus of many studies. One approach to capturing these neurological facets is to use functional magnetic resonance imaging (fMRI). Our goal is to identify active regions of the brain using fMRI data from a Balloon Analogue Risk Task (BART) study described in Cohen's *The Development and Generality of Self-Control* [2]. When possible and logically sound, we will attempt to reproduce the data preprocessing and analysis outlined in the paper. However, many of our approaches deviate considerably from the methods used by Cohen, either out of necessity from our lack of pre-packaged software or when we were inspired explore other directions. Ultimately, we will compare the active regions identified by our analysis with those detected by the Cohen's paper.

1 Introduction

The Development and Generality of Self-Control [2] and its associated fMRI studies are concerned with the similarities and differences across the brain relating to different forms of self-control. The paper in its entirety explores multiple studies (of multiple study types), but we will only focus on the third study. The original study compares four different types of self control among healthy adults to see if the four are related to each other. Very little relationship was found between these different behavioral tasks — in contrast to the vast majority of existing literature, which argues for a unified notion of self-control. We have decided to narrow our focus and data analysis to just the Balloon Analogue Risk Task (BART) study, which purportedly measures control over risky behavior, for feasibility reasons. fMRI scans from the study show blood flow to the brain, which may be relatable to control of risk-taking behavior during participation.

We initially strove to reproduce the original analysis as faithfully as possible. Our lack of access to the packaged software used by Cohen to clean the data was one early limitation that led us to develop our own pre-processing pipeline. However, the more we progressed in our analysis, the more we found it difficult to justify pursuing all of the same analytical decisions made by the paper, since many of them involved unfamiliar or "black-box" procedures. Furthermore, we identified several steps and assumptions in Cohen's analysis that we found difficult to theoretically justify. Thus, our final analysis actually deviates considerably from Cohen's approach. Our ultimate goal will be to compare the active regions identified by our analysis with those detected by the original paper.

We experimented with different procedures to spatially smooth the voxels and to convolve and time-correct the time courses for each subject. After pre-processing the data, we built linear regression models for each subject's voxel time courses. Several design matrices for the models were explored, and model selection was used to identify optimal combinations of predictors, such as linear drift as well as Fourier series and principal components of the voxel time courses. The resulting estimated coefficients from the linear regression models can be used to perform t-tests to examine the significance of activity in different voxels. However, the validity of these tests is highly dependent on the validity of our model assumptions, such as the normality of our errors. Since we are performing a large quantity of tests, the Benjamini-Hochberg correction was used to control the false positive rate. Finally, we clustered the raw coefficients, t-statistics, and corresponding “p-values” to identify regions of the brain with high activity. We then compared our results with those of the paper.

2 Data

The Balloon Analogue Risk Task (BART) measures risk-taking behavior by presenting participants with a computerized balloon. The participant can earn money incrementally by pumping up the balloon, but after an unknown threshold, the balloon will explode. At any time, the participant may elect to cash out his or her earnings, but doing so eliminates the potential to gain additional money through pumps. If the balloon explodes, the participant loses all of the money for the trial. For this study, BART and fMRI data for 24 subjects were collected and deemed to be of good quality. The mean age of the subjects was 20.8, and ten of the subjects were female. Four behavioral variables were recorded for each subject: the average number of pumps for each balloon, the average amount of money earned across runs, the number of exploded balloons, and the number of trials. There were also three model conditions: events for inflating the balloon (excluding the very last inflation of each trial), the last inflation before an explosion, and the event of cashing out (the balloon explosion was not included as an event). We are interested in the blood-oxygen-level dependent (BOLD) imaging data recorded for each subject during the course of task. Each subject’s BOLD data was recorded in 64 by 64 image matrices in 34 slices, with a variable number of time points.

Much of our analysis focuses on developing our own procedure for cleaning and preprocessing the raw BOLD data so that the signal (about 5%) can be more readily separated from the noise (95%). However, a cleaned version of the data was later made available by Ross Poldrack and the OpenfMRI project. The cleaned scans had received motion correction, high-pass filtering in time, and registration to the standard MNI anatomical template. Although we did not have the time to heavily investigate the pre-cleaned version of the data, it would have been interesting to compare the results from using our cleaning procedure versus the provided procedures.

It should also be noted that neither set of preprocessed data uses the exact same cleaning procedure as that used in *The Development and Generality of Self-Control* [2], which describes using various software and black-box methods that were not be available for our use.

3 Methods

3.1 Smoothing

Due to the inherently random nature of human subjects and their movements, smoothing must be performed on the spatial dataset. That way, the “noisy” data can be cast off from the data that actually represents significant changes in blood flow in the brain. Each voxel of the brain is represented by a measure of blood flow intensity, so a series of steps must be taken so that the data is correctly convolved to most closely and accurately depict what was happening at a certain point in the brain at a certain time. We decided to use smoothing via a Gaussian kernel in order to reduce noise in the three dimensional data. Originally, we were going to try and write a smoothing function from scratch, by implementing a rudimentary average-over-neighbors method. However, discussion with mentors lead us to the scipy module `ndimage.filters` that has a function to perform a Gaussian filter on n-dimensional data. A more detailed discussion about our approach and the theory behind the smoothing of the hemodynamic response can be found in Appendix B.

3.2 Convolution and Time Correction of Hemodyamic Response

3.2.1 Convolution

Our study is structured around event-related neurological stimuli, rather than block stimuli as was the case of the data used in class examples. So, we could not repurpose the class approach for representing the hemodynamic response to our analysis.

It is assumed that there is a relationship between the hemodynamic response to the neurological stimuli. Further, there is the assumption that a single stimulus generates a delayed hemodynamic response that mirrors a double-gamma function, and that multiple stimuli have an additive nature, as defined below:

$$r(t) = \sum_{i=1}^{n} \psi_i \phi_i (t - t_i) \tag{1}$$

where ψ_i is the amplitude of the response stimulus (assumed to be always 1 in our case), and ϕ_i is the hemodynamic response started at the ith stimulation (t_i).

We attempted five approaches that can each be grouped into one of three subcategories: **(1)** a strict replication of equation 1; **(2)** a matrix multiplication equivalent to **(1)**; and **(3)** a complex function that takes advantage of the speed of `np.convolve`. This complicated function **(3)** first splits the (two-second) intervals between each scan into a given number of even slices, then puts the stimulus into the closest slice with respect to time, and finally calls `np.convolve` on this much longer time series and a detailed hrf function, before reducing it back down to the dimensions of the original scan time series at two-second intervals. Detailed exploration of this matter can be found in Appendix C.

We compared these methods for on accuracy and speed. Figure 1 displays an accuracy comparison, and Table 1 shows the speed per loop based off of `iPython`'s `%timeit` magic command.

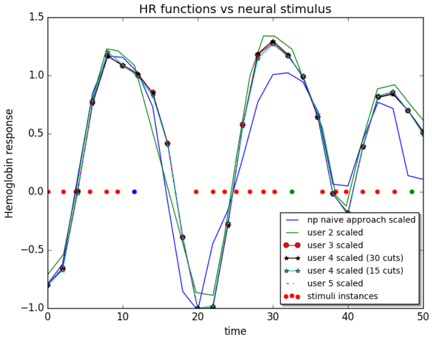

Figure 1: Different convolution functions vs. the Neural stimulus

name in graph	Speed per loop
np naive approach	14.4 μs
user 2	972 ms
user 3	1.15 s
user 4 (15 cuts)	98.3 ms
user 4 (30 cuts)	185 ms
user 5	110 ms

Table 1: Speed to create HRF predictions for Subject 001, all conditions

The first method in the table, "np naive approach", blindly plugs our data into the `np.convolve` function. This approach is ill-advised, because our data fails the `np.convolve` assumption of equidistant spacing for stimuli and scans. It is only demonstrated to showcase potential speed. The failure of the "np naive approach" was the motivating factor behind the rest of the hemodynamic response convolution analysis. The "user 2" and "user 3" functions fall under subcategory **(1)**. "user 2" was the first approach designed to approximate the theoretical background, but it matches the stimulation times and not the scan times. "user 3" is the most theoretically sound model (and is our standard for accuracy). "user 5" falls under subcategory **(2)** and is the matrix algebra version "user 3", with identical accuracy and gains in speed. "user 4" falls under subcategory **(3)**, the methods that use the grid cut usage of `np.convolve` with notations for the number of slices between each scan. We concluded that "user 4 (15 cuts)" was the best approach since it gives us much more speed and very close accuracy to the golden standard — "user 3".

3.2.2 Time Correction

The fMRI scanner scans each voxel at a slightly different time. In our case, the lowest horizontal slice was scanned first, with the later scans obtained progressively in order toward the top of the brain. The signs of this linear change in time of scan were observed when we ran simple linear regression on the data and found that the hemodynamic response $\hat{\beta}$ values from all conditions were grouped together. We corrected for the time differences by shifting the times of stimuli "backwards" for voxels scanned later to directly correct for the delay of the scan (assuming that each layer of the scan took 2/34 of a second).

3.2.3 Multiple Conditions

Originally, we tried using multiple linear regression to account for the three different types of stimuli (pump, explode, cash-out) and examined if the separation of these stimuli can better describe the response. We did this by creating separate predicted hemodynamic responses for each condition to allow for different amplitudes associated with each type of condition. As will be noted in Section 3.4 later, we did not observe a large difference in the results values we obtained from predicting the hemodynamic responses together for all conditions, so we did not continue with this exploration. In Figure 2, we can see the different conditions separated the responses for each condition.

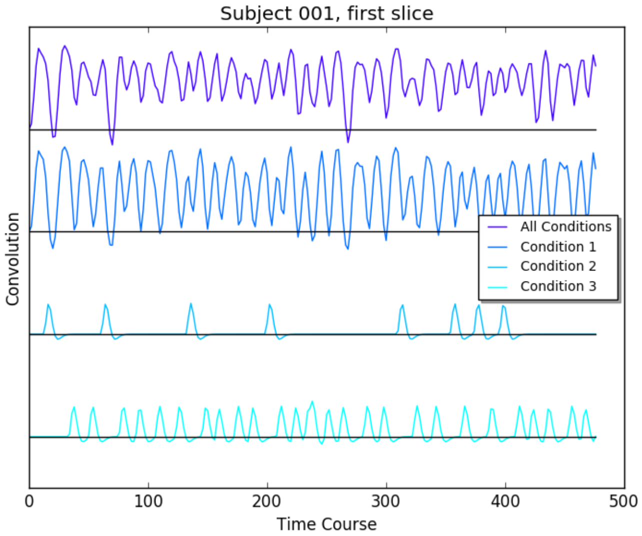

Figure 2: Plotting all predicted HR for conditions.

A more detailed discussion about our approach and the theory behind the convolution of the hemodynamic response with the neurological response can be found in Appendix C.

3.3 Linear Regression

A simple and straightforward way to model the voxel time courses for each subject is to perform linear regression. Initially, we just used the convolved predicted hemodynamic response (HR) and either all of the conditions together or each of conditions individually. After realizing that the HRs themselves did not explain enough of the BOLD ratio, we attempted to add features in order to reduce or explain the noise we observed. Additional features that we examined included a linear drift and some of the time courses' Fourier series and principal components.

Linear regression assumes a linear relationship between a response vector y and a design matrix of predictors X. Each element of y represents a single observed response, and each row of X represents a corresponding vector of predictor values. If one including the intercept as a term (as we have elected to do), the first column of the X design matrix should be a vector of 1s. The linear model can then be expressed as:

$$y = X\beta + \epsilon \tag{2}$$

It is further assumed that the errors ϵ_i for each observation i are independent and identically distributed with $N(0, \sigma^2)$, and that the errors are independent of X. The vector of coefficients β with length equal to the number of predictors in X can be estimated with the closed-form solution:

$$\hat{\beta} = (X^T X)^{-1} X^T y \tag{3}$$

Even when $(X^T X)$ is not invertible, $\hat{\beta}$ can be estimated using the pseudo-inverse of $(X^T X)$, represented as $(X^T X)^-$ to get a non-unique value for $\hat{\beta}$.

To consider the strength of the effects of these predictors, we will use t-tests of the corresponding estimated coefficients for each voxel and subject, as discussed under Section 3.6. The validity of the model and of the "p-values" produced by performing these t-tests is dependent on whether or not the many assumptions of the linear model are actually met. In particular, we will discuss the assumption of normal errors by analyzing the residuals in Section 3.5.

3.3.1 More about potential features:

Other than the basic HR feature(s) and a column of 1s (to account for an "intercept" term or non-zero average value), we experimented with additional predictors for our design matrix X. Among these were the first few principal components of the voxel × time matrix of voxel time courses and the first few functions of the Fourier series for the time courses. As noted above these, additional features helped account for the noise in the observed BOLD ratio fluctuation.

Principal Components

One approach for reducing the noise in the linear model is to include principal components of the voxel × time voxel time course matrix. Instead of using the entire matrix, it may be possible to just include the first few principal components as features that explain a great deal of the variance in the entire matrix. To get the principal components, we obtained the singular value decomposition (SVD) of the time × time covariance matrix. We tried this with and without first masking the voxels. To standardize the voxels, we subtracted the column means (mean across voxels) from the voxel by time matrix. There is also a very strong effect of mean over time in the data that dominates other effects, so we subtracted the row means (mean over time) as well.

As you can see in Figure 3, which compares the variance explained by including up to ten components, with and without masking the voxels, masking explains more variance at each component. This trend was observed across all subjects. So, between the better performance and the logical rationality of using the masked data (we are not actually interested in the behvavior of voxels outside the brain), we decided to only work with the masked data's principal components.

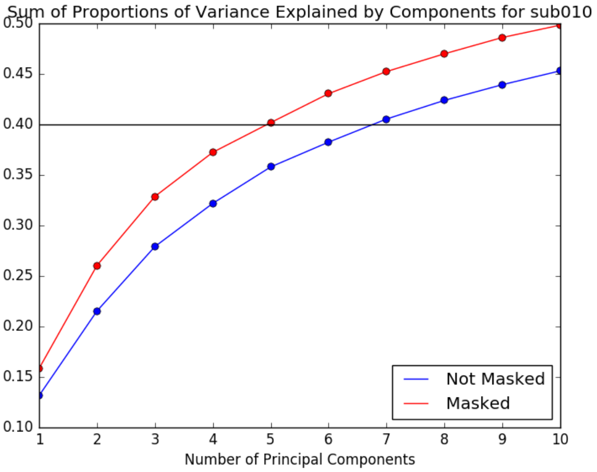

Figure 3: Comparing proportion of variance explained by Subject 10's principal components, with and without masking the data.

An important issue to consider is how many principal components to include in the design matrix. Figure 4 compares the the amount of variance explained by including successively more principal components across subjects. A few observations should be noted. First, there is considerable variation between subjects in how much variance the early principal components capture. Second, by including only the first six components, it is possible for most subjects' voxel time course matrix to capture at least 40% of the variance. Moving forward, we chose to include six principal components as additional features when considering models that reduce noise. This cutoff of six components or 40% of the variance explained was somewhat arbitrary, with the idea being we wanted to only include a few components without sacrificing too much of the variance explained and that we wanted to use the same number of components for each subject. It will be seen later that the variation between subjects in how much variance is captured by the first six components has strong ramifications on the results.

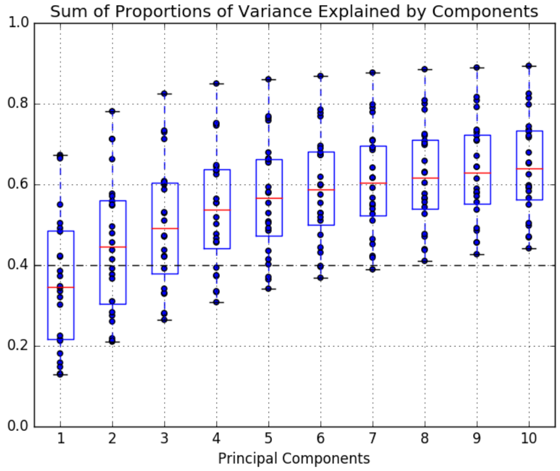

Figure 4: Boxplots comparing the amount of variance captured by principal components for each subject. The data was masked beforehand. All subjects are able to capture at least 40% of the variance when using just six principal components.

Fourier Series

We also considered including 6 features related to the first few functions of the Fourier series. A full Fourier series is represented as the following:

$$f(x) = \frac{1}{2} \cdot a_0 + \sum_{n=1}^{\infty} a_n \cdot cos(nx) + \sum_{n=1}^{\infty} b_n \cdot sin(nx) \tag{4}$$

We wanted to represent low level sinesoidal fluctuations, which requires a few periods over the full range of the time course $(0, \text{num of TR})$. As such, we changed the Fourier series to:

$$f(x) = \frac{1}{2} \cdot a_0 + \sum_{n=1}^{\infty} a_n \cdot cos(\frac{n}{\text{num of TR}}x) + \sum_{n=1}^{\infty} b_n \cdot sin(\frac{n}{\text{num of TR}}x) \tag{5}$$

We used $\sum_{n=1}^{3} a_n \cdot cos(\frac{n}{\text{num of TR}}x) + \sum_{n=1}^{3} b_n \cdot sin(\frac{n}{\text{num of TR}}x)$ to be 6 features to try to get a low order sinusoidal fluctuations.

3.4 Model Selection

In order to select the best set of features for our X matrix, and also to compare the use of a single condition feature vs. each of the three different types of conditions as three separate features, we decided to utilize model comparison metrics; specifically, the AIC, BIC, and adjusted R^2. Using a small but expressive subset of the subjects (002, 003, and 014), we averaged the metrics across all voxels and people, an admittedly not theoretically sound approach. We visualized values in Figures 5, 6, and 7.

From these plots we can observe that **(1)** separating the conditions into individual features to convolve the hemodynamic response did not provide much gain in these metrics and **(2)** the inclusion of the 6 principal components comparably tends to create better models than the inclusion of the 6 Fourier series features. We initially interpreted this as a vote to include the first 6 principal components and not the Fourier features. Unfortunately, including the 6 principal components lead to overfitting and collinearity with the HRF features for some subjects. These problems tended to arise when the proportion of variance explained by the first 6 principal components was much greater that 40 %. Overall, we went with the 6 Fourier features instead and observed similar t-statistics for the HRF feature in the models with those when using the first 6 principal components when the variance explained was around 40 %.

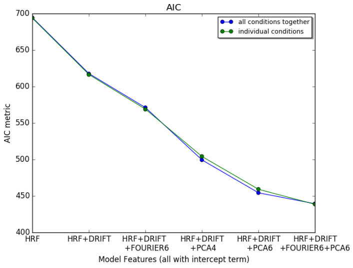

Figure 5: AIC

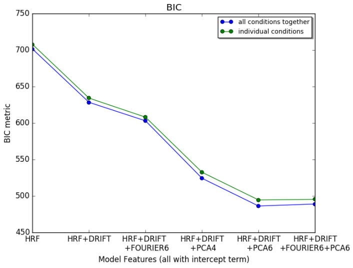

Figure 6: BIC

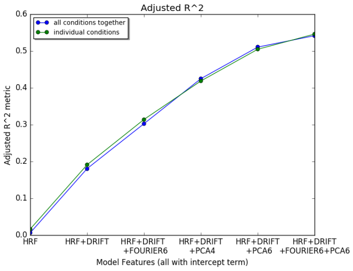

Figure 7: Adjusted R^2

3.5 Normality Assumptions

The validity of our hypothesis tests for the estimated $\hat{\beta}$ values from the chosen linear regression model are largely dependent on whether we can reasonably assume that the errors in our model are independent and identically distributed from some normal distribution with mean zero and constant variance. We focus here on checking the normality assumption. It is generally wise to use visualizations, such as residual vs. fitted values plots and quantile-quantile plots to inspect residuals for patterns and abnormalities. However, considering the sheer quantity of data we are working with — each of the 24 subjects has 64 × 64 × 34 voxels that can each in turn be fitted to a model — visual inspection is not practical.

For this reason, we used the Shapiro-Wilk test for normality, which tests the null hypothesis that the data in question is normally distributed. A Shapiro-Wilk test was performed for each set of residuals corresponding to a single voxel's time course. That is, each test used around 200 observations, or the number of time points for that particular subject. 200 observations is not an especially large sample size, and for this reason, we express some concern because normality tests have low power for small sample sizes. Shapiro-Wilk may incorrectly fail to reject the null hypothesis due to this bias [3].

The average proportion of Shaprio-Wilk test "p-values" above 0.05 was 0.742 for the unmasked residuals across both subjects and voxels and noticeably lower at 0.630 for the masked residuals. However, since using the masked data is more theoretically justifiable (the unmasked data contains many voxels outside of the brain), we use the masked data for our analysis despite the lower proportion of voxels whose residuals meet our normality check. Note that these proportions suggest that only about two-thirds of the masked voxels have residuals that are approximately normal, and that the others deviate significantly from the normal distribution (especially when considering our concerns about the power of Shaprio-Wilk tests for small sample sizes). So when discussing our models and especially when looking at the conclusions made by our hypothesis tests, one should exercise caution about the validity of those conjectures.

Figures 8 and 9 compare the spatial distribution of the Shapiro-Wilk "p-values" for Subject 10's masked and unmasked data. The unmasked figure is very difficult to interpret, but the masked figure does suggest that while the spatial distribution of voxels with approximately normal residuals is reasonably uniform in many regions, there are a few areas that consistently have very low "p-values" at around or below the 0.05 threshold. Examples of these regions for Subject 10 include a small area between the front and center of the brain and a few spots near the sides. However, these observations are not consistent across all subjects, so much of it may simply be due to noise.

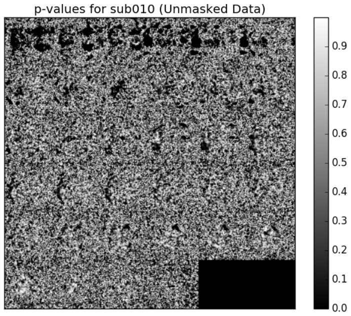

Figure 8: Subject 10's brain slices, with voxels colored by the magnitude of the "p-value" in the corresponding Shapiro-Wilk test for normality.Using unmasked residuals.

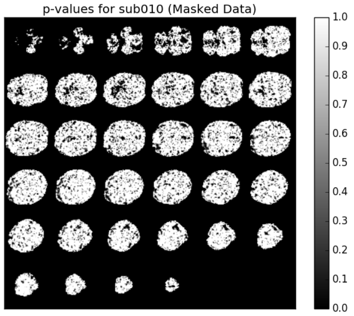

Figure 9: Subject 10's brain slices, with voxels colored by the magnitude of the "p-value" in the corresponding Shapiro-Wilk test for normality. Using masked residuals.

3.6 Hypothesis Testing

We now have chosen our linear model and tested its normality assumptions by analyzing the residuals from our model. Next, in order to measure the strength of the association between the voxel time courses and the HRF, we run a hypothesis test on the coefficients of the linear regression model for each subject.

There is an individual linear model associated with each voxel in a subjects image (and a total of $64 \times 64 \times 34$ voxels per subject). Thus we ran a t-test on each voxel's $\hat{\beta}_1$ coefficient, which is associated with the HRF response. The null hypothesis for each test is that $\hat{\beta}_1 = 0$, with the (two-sided) alternative hypothesis being $\beta_1 \neq 0$. Once we obtained each t-statistic, we compared this value across voxels in two ways. First, we simply compared the t-values with voxels within a subject. In this case, we took into account the sign of the t-statistic in our analysis. Second, we converted this t-statistic into a "p-value" (assuming that the null distribution of the t-statistics is student's t), in which case the sign of the t-statistic becomes irrelevant. Here, we compared across voxels without taking into account this sign. We later performed a multiple comparisons correction using Benjamini-Hochberg method in order to identify the voxels with the most significant activation. We also implemented clustering methods to analyze the $\hat{\beta}$ and t-statistics, taking into account the problem of multiple comparisons.

3.7 Clustering

The heart of our analysis consists of different implementations of procedures for multiple comparisons between the 24 subjects. More specifically, we focused on an implementation of the Benjamini-Hochberg procedure, the grouping of t-statistics, and the grouping of $\hat{\beta}$ values. For each of the three cases, we found appropriate parameters to be used for all subjects by observing the images from a few subjects at different points of input. The parameters that we played around with include the false discovery rate, the cutoff value for significant t-values and $\hat{\beta}$- values, and the number of neighbors to use when smoothing data. We also considered hierarchical clustering. For hierarchical clustering using the ward metric, we compared different parameter selections for the number of clusters.

3.7.1 Benjamini-Hochberg Correction

When conducting multiple comparisons, it is important to have an idea of the quantity of Type I errors that may be prevalent in the analysis. In our analysis of the voxel time course data, we decided that limiting and controlling the number of Type I errors is important to the process of identifying active regions. The processes of limiting the number of Type I errors are called FDR-controlling procedures. In the grand scheme of things, FDR-controlling procedures give greater statistical power at the cost of producing more Type I errors.

Once we coded the hypothesis function that will return t-test values and "p-values", we implemented the Benjamini-Hochberg procedure to control the proportion of rejected null hypotheses in the data. The Benjamini- Hochberg procedure works by multiplying each of the "p-values" to a ratio of the number of tests times the rank of the “p-value” in the ordered set, and the chosen false discovery rate — from these adjusted “p-values”, only the values that are less than the chosen false discovery rate will be chosen to be returned. This way, we are able to adjust the proportion of null hypotheses that will be rejected and the proportion that will return the desired proportion of significant tests. This will reduce the number of false positives returned in the data and extend greater statistical power in later analysis performed on the voxel dataset.

The Benjamini-Hochberg procedure was not the only analysis we performed over multiple subjects. After completing grouping of the t-statistics and $\hat{\beta}$ values. we noticed that the variability between the subjects of the study is different between the Benjamini-Hochberg procedure outputs, but that the t-statistics and $\hat{\beta}$ value grouping procedure outputs. Moreover, when performing the Benjamini-Hochberg procedure, along with the t-statistics grouping procedures, we did play around with smoothing over neighboring voxels. Ultimately, however, we decided to forgo the neighbor smoothing to prevent loss of information in the plots that we generated in the end, and neighbor smoothing didn't visually appear very needed.

3.7.2 t-Statistics Grouping

We use the t-statistics that we found earlier for another type of cutoff analysis. The t-statistics measure the size of the difference relative to the variation in the data. Thus, a small “p-value” corresponds directly to large absolute values of t-statistics. In fact, the “p-value” and the t-statistics are related by the following statement: More extreme t-statistics will return lower “p-values”, which increases the chances of the null hypothesis being indicated as false. Although the t-statistics go hand in hand with the Benjamini-Hochberg analysis on the “p-values”, we ultimately decided that an additional process of selecting t-statistics based on a threshold was necessary because the “p-values” assume a normal distribution while the t-statistics do not assume this from the data. It should be noted that the t-statistics grouping analysis performed as well, if not better, than the “p-value” analysis using the Benjamini-Hochberg procedure.

Since the magnitude from zero represents how significant the test is for t-statistics, we collected the absolute value t-statistics that were above a certain threshold. This was the opposite of the procedure for finding the significant tests in the Benjamini-Hochberg process. The threshold, also called the cutoff in the scripts that were written when developing this process, is calculated in a very similar fashion to the Benjamini-Hochberg false discovery rate. This is because the t-values and the “p-values” are strongly linked, in fact there is a 1-to-1 relation between the absolute value of t - values to p-values.

Implementing the t-value grouping function was almost trivial because it is the flip side of the Benjamini-Hochberg function that was implemented previously. However, since we do not know if there's a function that relates the absolute values of t-statistics to “p-values”, there was a lot of experimenting with different values to see which values of cutoff points versus false discovery rates (the Q value in the Benjamini-Hochberg function) would deliver similarly focused results.

3.7.3 $\hat{\beta}$ Grouping

Using the same function that we created to group a certain subset of the t-statistics, we also looked over the $\hat{\beta}$ values as a part of our analysis. Much of what we know in statistics tells us that observing $\hat{\beta}$ values on top of looking at t-statistics would not tell us much since the two variables are not explicitly connected; however, the main reason why we decided it was of utmost importance to look at both the t-statistics and the $\hat{\beta}$ values is that there might be a relationship between the trends in each respective variables' output to the cutoff point, that is lost when going from $\hat{beta}$ to t-statistic (dividing by variance).

Granted, it is pretty clear that the t-statistics grouping is very theoretically similar to the Benjamini-Hochberg procedure. This is why the $\hat{\beta}$ grouping portion of the analysis was added on. Since $\hat{\beta}$ values do not directly use normality assumptions, we thought it would offer either a different angle on how to approach analysis, or confirm the results found in the analysis that depended on the assumptions of normality.

3.7.4 Hierarchical Clustering

Using the across-subject average t-statistics for every voxel in the brain, we are left with a 3-d array of t-statistics that contain both negative and positive values. Instead of manually observing patterns in these images, we instead implemented a clustering algorithm to split the entire 3-d images into clusters based on the voxels' relative location to each other as well as the values of their t-statistics.

In order to find a proper clustering algorithm, we decided to treat this problem like a grayscale image segmentation problem and implemented a agglomerative hierarchical cluster using Ward's method. Agglomerative means that the clusters are built bottom up with each observation starting as its own cluster and pairs being moved up the hierarchy. Ward's method creates clusters based on a minimum variance criterion that minimizes the total within-cluster variances. An example of this implementation for a 2-d image is seen here: `http://scikit-learn.org/stable/auto_examples/cluster/plot_lena_ward_segmentation.html`.

In our implementation, we defined a structure to our data using a connectivity graph in order to ensure that each cluster is spatially constrained. Also, since our scenario uses a 3-d image, the connectivity graph will also have to take into account this extra dimension.

Ultimately, we did not use this clustering algorithm in our main paper because it did not add any useful information on top of our other clustering methods we used. See figure 10 to see similarities. Furthermore, we faced a few problems when trying to implement the algorithm. Most significantly, creating the 3-d connectivity graph was a problem, and we were not able to properly implement this in z-direction. Furthermore, runtime was an issue when compared to our other options. Ultimately, the idea around using a hierarchical clustering method was intriguing; however, the same results could be achieved using simpler and faster methods.

3.7.5 Comparison of Clustering Techniques

Generally the clustering techniques for the multiple comparison and the hierarchical clustering produces similar results, which can be seen if a side by side comparison of the hierarchical clustering method using Ward against the quantile-based clustering algorithm for t-statistics (we've included the actual t-values as well in the center for comparisons) [Figure 10]. Depending upon the subject, the clustering from Benjamini Hochberg, t grouping and $\hat{\beta}$ grouping can be very different or very similar.

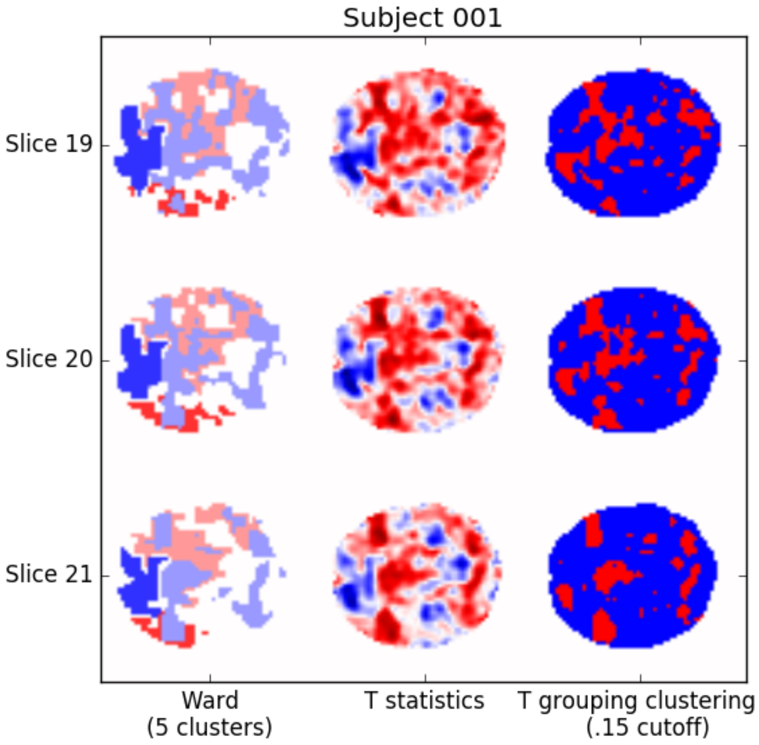

Figure 10: Clustering Comparison between WARD, t-statistics, and t-grouping

4 Results

Pre-Processing

Smoothing performed well in reducing outliers and extreme values in the time-series per voxel. We used a value of $\sigma = 1$ for the Gaussian smoothing. We performed HRF convolution and time correction as well with success.

Model Selection

For our linear regression model, we ended up choosing the design matrix model with a single HRF feature with all conditions, a linear drift feature, and three pairs of Fourier features. The dimension of our final feature matrix is time × 9. As noted previously, this selection of the model was also due to reduce problems with overfitting and collinearity with the HRF features.

Testing Normality Assumptions

Based on the Shapiro-Wilk test for normality on the residuals with a p-value threshold of 0.05, we found that the normality assumptions of the linear models were violated roughly 35% of the time. This suggests that we should be cautious of the validity of hypothesis testing and possibly explore alternatives for analyzing the coefficients.

Clustering We used two perspectives to cluster, one based of multiple correction and another based on hierarchical clustering using Ward's method. Hierarchical clustering was computationally costly against the large number of voxels and was not feasible. We used 3 different approaches to multiple comparison correction.

The most canonical approach to multiple correction we used was Benjamini- Hochberg method, which utilizes p-statistics (with the underlying assumption of normality). Based on the fact that the normality assumptions did not hold for a large proportion of the voxels, we also utilized analyses that did not rely on such assumptions. Our other methods that mirrored our clustering approach to Benjamini-Hochberg, where we utilized the $\hat{\beta}$ and t-statistics corresponding to the HRF coefficient of the chosen linear model design matrix. We created active region clusters when the $\hat{\beta}$ and t-statistics for just absolute values of these statistics that were higher than 85% of the rest.

Identifying Active Regions

We considered the results from each of the approaches for clustering with multiple comparison corrections. We tuned the false discovery rate cutoff for Benjamini-Hochberg and the quantile-based cutoffs for the t-statistic and $\hat{\beta}$ methods using subjects 1, 2, 3, 4, and 14. As alluded to previously, the BH approach was less interpretable and required per-subject rather than agglomerative tuning of parameters to obtain logical results. Thus, the BH results were largely dependent on the optimal choice of false discovery rate cutoff, which varied between subjects, and so the results from using the tuned parameter were not consistent. Thus, it was generally a less favorable approach than the t-statistic and $\hat{\beta}$ quantile-based clustering.

These observations are visualized in Figures 13, 14, and 15 (at the end of the paper, before the appendix), which compare the t-statistic, $\hat{\beta}$, and Benjamini-Hochberg clustering results, respectively, for a single subject (Subject 6). In the former two approaches, distinct active regions can be readily identified, whereas the Benjamini-Hochberg seems largely unable to pull out spatially grouped significant voxels. While some subjects had BH results that were reasonably interpretable, others subjects also experienced similarly scattered significant voxels. However, the other two methods were highly consistent with each other within the same subject. Small differences in the t-statistic and $\hat{\beta}$ results can largely be accounted by the presence of coefficient variances in the former.

A region frequently identified with high HRF activity across subjects for the t-statistic and $\hat{\beta}$ approaches was the occipital area of the brain. Other areas with potentially high activity include a small region toward the center and front and some areas on the left edge. In Figures 16 and 17, which show the quantile-based clustering results for the t-statistics, one can see the strong similarities in active regions for Subjects 3 and 11, particularly in the occipital area of the brain. There are some noticeable differences. The slices in which the occipital active regions are most prominent are shifted down for Subject 11 compared to Subject 3.

However, not all subjects behaved in this similar fasion, with the main active region being the occipital area of the brain. Among them were subjects with unusually-shaped brains, as well as other subjects like Subject 7 Figure 18. Curiously, Subject 7's active regions seem almost backwards, with the highly active regions in the front of the brain and very little in the back. So the identified active regions are not always consistent or fully conclusive.

5 Discussion

5.1 Discussion of Results

Cohen's paper identifies sixteen regions of brain activation during the course of the BART study. In particular, we able to detect high activity in the dorsal lateral prefrontal cortex and the occipital lobe. The prefrontal cortex encompasses the frontal lobe and stores the brain's short-term memory as well as quick decision-making processes. The occipital lobe can be found in the back part of the brain and this region is responsible for processing visual inputs. We were able to consistently detect the occipital lobe in our analysis. Both active regions are showcased for Subject 23 in Figure 11.

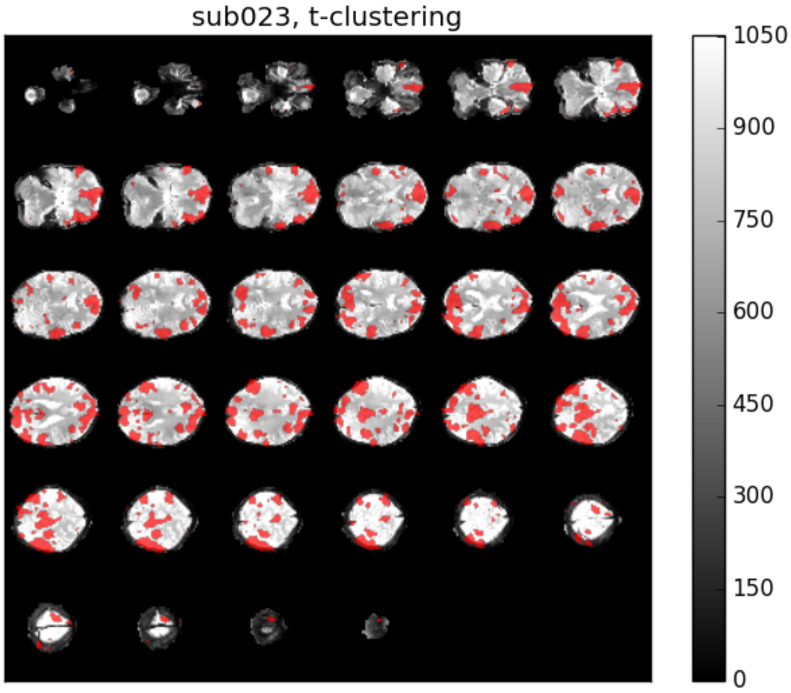

Figure 11: Quantile-based clustering for Subject 23's t-statistics. (Red areas denote significant regions)

BART studies measure risk-taking behavior, which is a subset of self-control. There is activity in both the prefrontal cortex and the occipital lobe. This is expected since subjects use visual processing centers (the occipital lobe) during the study. While visual stimulus is not something that the BART study is interested in assessing, it is reasonable that the occipital lobe was nevertheless an site of high activity, since there was a strong visual aspect to the study.

Additionally, subjects utilize their short term memory and their decision making during the experiment while choosing to continue pumping the balloon or cash out. Thus, it makes sense that both regions of the brain will be activated during the experiment.

5.2 Discussion of Future Work

The implementation of permutation tests, which have few assumptions and are easy to interpret (but computationally intensive), would have been useful for testing the significance of the estimated coefficients from linear modeling. This can serve as an alternative to t-tests, which make several assumptsion about the data structure. Somewhat relatedly, additional tests and checks for model assumptions would also be valuable for assessing the appropriateness of our existing hypothesis testing.

One major issue that we were unable to fully address is how to appropriately aggregate the results from individual subjects to make more general conclusions about activation regions, as opposed to the activation region of a particular subject. Much of this is due to the considerable variety in brain positioning and shape observed between different subjects; there is really no such thing as an "average" subject. One approach that we tried was to simply average across subjects the binary t-statistic masks from the quantile-based clustering (Figure 12). The result was a single "average" brain, but since there is considerable variance in brain size, shape, and so on, the justification for this method was not theoretically sound. Nevertheless, having that single average brain makes interpretation easier. For our situation, this method was able to tease out the high activity in the back of the brain, but not elsewhere.

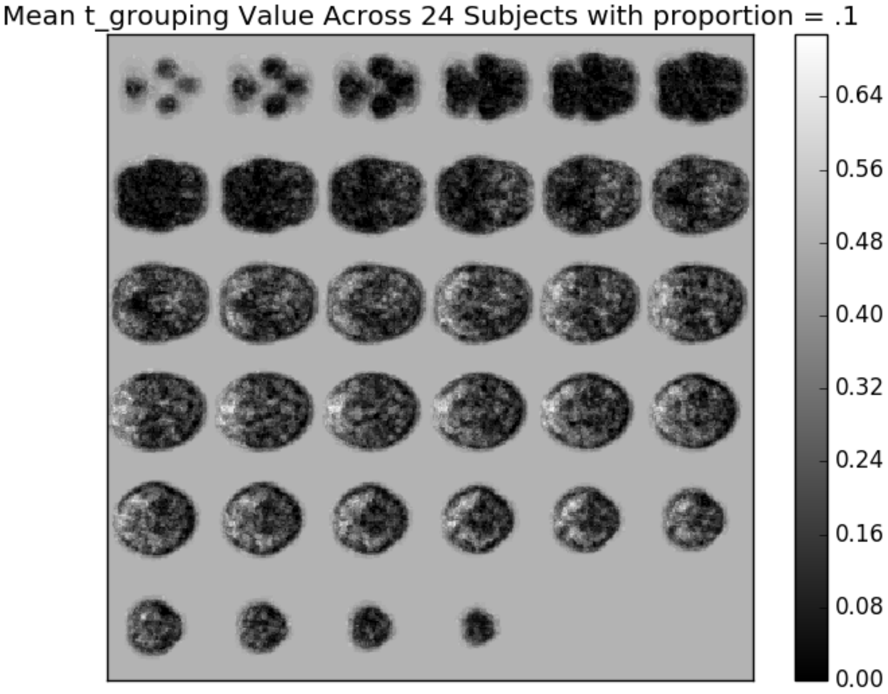
Mean t_grouping Value Across 24 Subjects with proportion = .1
0.64
0.56
0.48
0.40
0.32
0.24
0.16
0.08
0.00

Figure 12: Slices from averaging the binary t-statistic masks across subjects. Colors indicate the proportion of subjects that identified a voxel as significant.

Additional future work could also be concerned with reproducing our approach for identifying actived regions on the pre-cleaned version of the study's data provided by the organizers of the OpenfMRI project. Those results could then be compared with the results from using our own pre-processing techniques. Since fMRI data is notoriously noisy, the different decisions made when cleaning the data to seperate the signal from the noise can greatly alter the results. So while ideally, both our pre-processed data and the alternatively pre-processed data would identify the same active regions, we acknowledge that there is a reasonably high chance that this would not be the case.

Relating back to the issue of aggregating subjects, one great advantage to using the cleaned data from Ross Poldrack and the OpenfMRI project is that the subjects were registrated to a standard MNI anatomical template. Averaging clustering results across subjects would make more sense here, since the standard template should account for many of the differences in brain shape and positioning in the fMRI scanner. Near the end of our project we attempted to incorporate the new data, but with 8 × as much data per subject and the nonlinear order of growths caused problems in the short term.

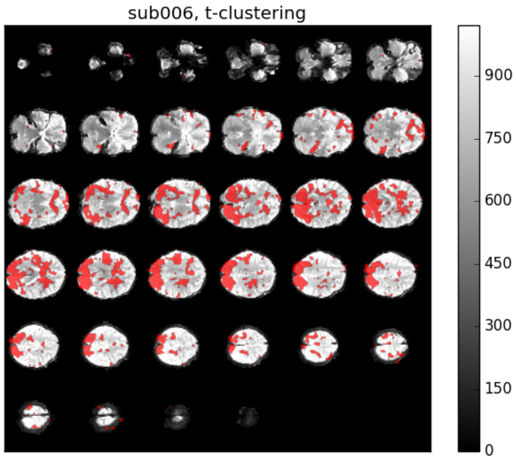

Figure 13: Quantile-based clustering for Subject 6's t-statistics. (Red areas denote significant regions)

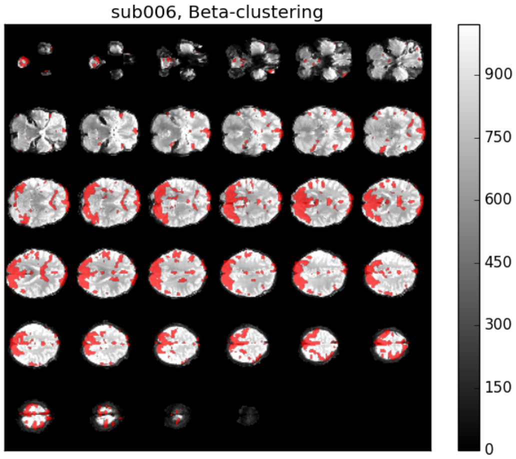

Figure 14: Quantile-based clustering for Subject 6's $\hat{\beta}$ coefficients. (Red areas denote significant regions)

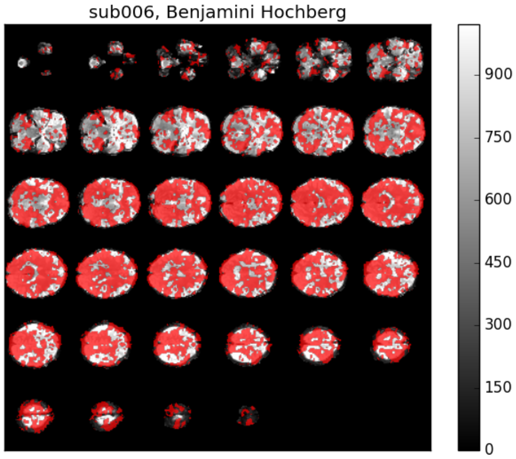

Figure 15: Benjamini-Hochberg clustering for Subject 6. (Red areas denote significant regions)

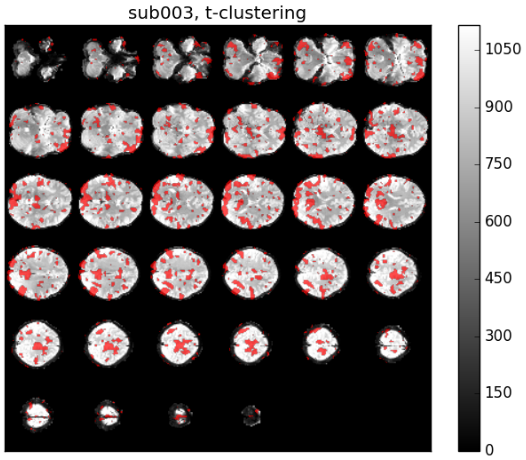

Figure 16: Quantile-based clustering for Subject 3's t-statistics. (Red areas denote significant regions)

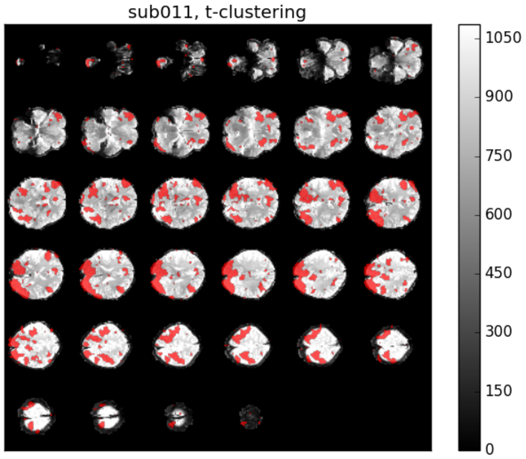

Figure 17: Quantile-based clustering for Subject 11's t-statistics. (Red areas denote significant regions)

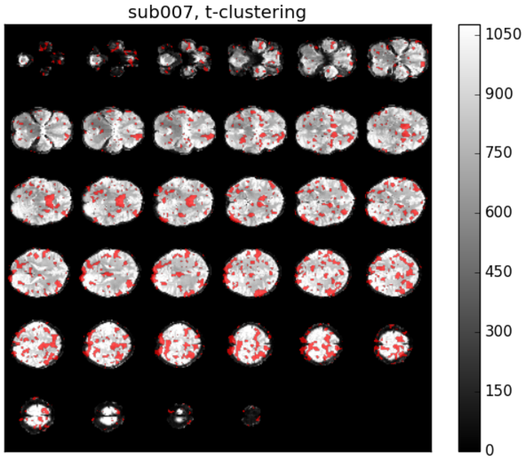

Figure 18: Quantile-based clustering for Subject 7's t-statistics. (Red areas denote significant regions)

Appendix

A Outlier Removal

We considered following the procedure implemented as part of Homework 2 to detect and remove outlier 3-D volumes from the 4-D image scans of each subject. The process involves finding the root mean squares (RMS) of each 3-D volume across time and then getting the difference values. When a given volume is very different from the preceding volume, this may indicate a potential outlier or the sign of an artifact. We used thresholds based on 1.5 × IQR added to the 75th percentile and subtracted from the 25th percentile to create the cutoffs for the RMS difference outliers. We then extended the RMS difference outliers by labeling the volumes on either side of the outlier RMS difference as being outliers.

Each subject was considered independently, as considerable variation in measurement is expected between different subjects. However, we found that reductions in mean residual sum of squares from running simple linear regression before and after removing the extended RMS difference outliers were minimal. Visually speaking, we also did not observe the presence of egregiously different points. So, we opted to refrain from removing outliers, at least through the extended RMS difference method.

B Smoothing

We used a Gaussian filter to spatially smooth noise in the brain images. At first, we played around with implementing a function similar to a mean filter, where each voxel would be the average of a certain radius of its neighbors. Ultimately, we decided to use a kernel with a Gaussian (bell-like) shape. While it is a linear type of smoothing, like the mean-smoothing approach, the Gaussian filter is a stronger choice than the mean-smoothing because it is not as affected by sharp spikes in the data.

The property that makes the Gaussian filter a stronger and more reliable candidate for smoothing the voxel data is that it outputs a weighted average of the voxel and its neighbors. The more heavily weighted values are at the center of each of the neighborhood of voxels we examine. On the other hand, a regular mean filter would use a uniformly weighted average, which means that there is the risk of oversmoothing, along with additional complications of handling the voxels along the edges of the brain image data. Furthermore, a Gaussian filter is ideal for use in noisy voxel data because of its steady frequency response. By choosing an appropriate filter, we can gain more control of the range of spatial frequencies left after smoothing the data. Additionally, Gaussian filters are non-negative for all voxel data. Thus, the output of smoothing the voxel data with a Gaussian filter will still be a valid image.

We decided to go with the module for a Gaussian filter for several reasons, the main one being that Gaussian filters can remove noise yet preserve the high frequency edges in the brain image data. Rather than using a self-implemented mean filter function that would cause issues with high-frequency edge cases as well as conglomerating data into thoughtless averages, we used a Gaussian filter which handles these situations better because the smooth, bell-shaped curve of the convolution does not have a sharp cutoff at edges. The Gaussian filter also distributes weighted averages across the voxels such that the smoothing keeps high-frequency data points into consideration for the end product.

C Convolution Analysis

C.1 Introduction

fMRI data presents a distinct challenge for relating neural stimuli to BOLD (blood-oxygen-level dependent) response. fMRI scans record changes in oxygenation levels of hemoglobin in the brain. However, there is a delay between the neural stimulus and the change in blood oxygen levels in a given area. In our case, the neural stimulation comes from a event-related experimental design. A commonly assumed hemodynamic response to a neurological stimulus is the double gamma function that can be seen in Figure 22. The complete hemodynamic response function needs to be modeled in order to better relate stimulation and the BOLD response from the fMRI scan. It should be noted that the BOLD response is highly noisy and we are really trying to capture the blood oxygenation level change to the stimulation.

C.2 Mathematics

C.2.1 Convolution Theory and Mathematics

To relate stimuli to BOLD response, we convolved the time courses of discrete stimuli with the assumed response to a single stimulus. At a basic level, convolution is a distinct combination of two functions (say f and g). This combination is just the "integral that expresses the amount of overlap of f as it is shifted over another function g" [4]. There are many examples of this, but the following is basic idea that we will expand off of later.

Let us define the function f as a sum of two gamma functions and g as a "continuous" specialized step function (we will examine why these functions are valuable later). Graphically, we can see their plots in Figure 19 and 20, and mathematically we will define them as in the following equations 6 and 7, respectively.

$$f(t) = \frac{.6}{.17} \cdot \left[G_1(6,t) - .35 \cdot G_1(12,t)\right] \tag{6}$$

where $G_1(k,t) = \frac{1}{\Gamma(k)} t^{k-1} e^{-t}$ (the gamma pdf with $\theta = 1$)

$$g(t) = \begin{cases} 0 & \text{if } 5.85 \leq \text{t} \leq 6.15 \\ .6 & \text{otherwise} \end{cases} \tag{7}$$

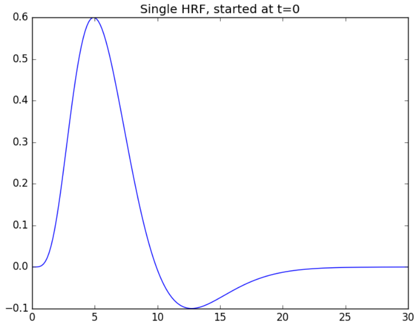

Figure 19: f ("Stabilized Function").

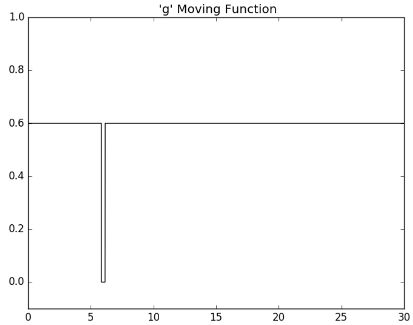

Figure 20: g ("Moving function").

As mentioned in the earlier definition, if we move g across f from left to right, we will see something similar to Figure 21 for discrete time intervals. If we plot these values (the integration of the differences), we will get a plot very similar to that of f when f is starting at a certain point. (The plot actually "cheats" when f is negative, and we would have to alter definitions a little bit). If we had multiple peaks in our g function (i.e. multiple distinct"zero" places), we would expect to get multiple non-zero differences between the functions at each time capture.

C.2.2 Convolution Applied to Stimulus

The "continuous" nature of the step function "g" does not extend well into the discrete time series that we have. However, one approach for fMRI analysis is to approach the convolution as something slightly different: mathematical sums. For example, in the previous section, we can treat f as the same, and g as g' defined in equation 8.

$$g'(t) = \begin{cases} 1 & \text{if t=6} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

We could then find the value of the convolution of g' and f for discrete integers as in equation 9.

$$r(t) = f(t-6) \tag{9}$$

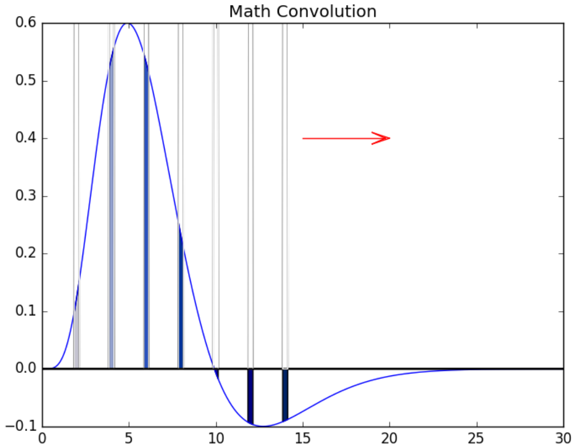

Figure 21: Convolution of f and g.

If we allow for multiple non-zero periods in g', we can get a more general model in equation 10, where each t_i is a value when $g'(t_i) \neq 0$:

$$r(t) = \sum_{i=1}^{n} f(t - t_i) \tag{10}$$

This equation gives a good glimpse into what the hemodynamic response would be after stimulus at time t_i for $i \in 1, ..., n$. Moreover, one could extend the idea to include a "strength" value of the stimulus by changing the $g'(t_i)$ to values other than 1. If that was the case, we would change the response equation to Equation 11 to allow us to include all discrete t into the equation where $g'(t_i)$ is now expected to be zero (so n becomes much larger).

$$r(t) = \sum_{i=1}^{n} g'(t_i) f(t - t_i) \tag{11}$$

With this new equation, we can consider function f and g' displayed graphically [Figure 22, 23, respectively] and their "convolved" output [Figure 24].

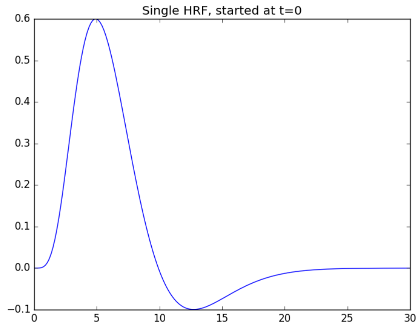

Figure 22: f ("Stabilized Function").

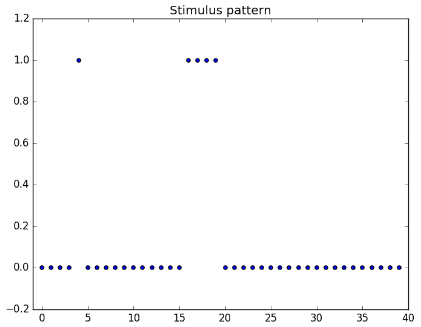

Figure 23: g' ("Moving function").

C.3 Approach to our Specific Problem

C.3.1 Returning to Our fMRI Data

We can now apply this discrete approach to convolution between f and g' to our data. The f is actually a common representation of the hemodynamic response, and the g' is a good representation of

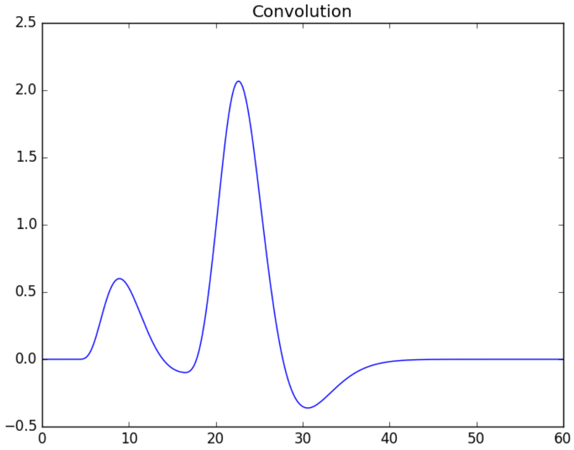

Figure 24: Convolution of f and g'.

the stimuli from an event-related trial [1].

C.3.2 Naive Approach (using `np.convolve`)

A `numpy` function `np.convolve`, which takes advantage of fast Fourier transforms for efficiency (it boils down to fewer computations using roots of unity) is commonly used to do discrete convolution. A naive approach for convolving two functions would use this function directly, but `np.convolve` assumes that the intervals between stimuli mirror the desired intervals between prediction intervals. As such, we should not naively apply `np.convolve` to our data (though we do for a base model). Even so, exploring this naive approach for convolving the hemodynamic response gives an idea about what not to do, and sets a higher bar for accuracy.

C.3.3 Needed Improvements: Moving beyond naive `np.convolve`

Our data fails to meet the assumption that the intervals are equidistant, which is required to naively apply `np.convolve`. Especially in our case, there was not an simple fix, such as performing some basic rounding in order to then correctly utilize `np.convolve`. All the following approaches improve on the basic `np.convolve` approach's accuracy, but ultimately circle back to incorporating `np.convolve` to improve the speed of the convolution.

Our condition file (`cond1`) lists stimulus times for when the individual pumped the balloon but did not pop it. For subject 001, the first 10 data points are as follows [Figure 25]:

0.0671	2.1251	3.7681	5.6601	7.8673	9.3443	19.7831	22.0402	23.5837	25.1434

Figure 25: First 10 values for Sub 001, condition 1.

Clearly, this short time series does not align with idealized scans that start at $t = 0$ and occur every two seconds apart. As such, we had to go back to the drawing board to try to reproduce our expected hemodynamic response for the entire time course.

C.4 Summary of Approaches

Our first approach attempts to correctly match the theory underlying our data. Our second approach tries to utilize `np.convolve` by expanding the grid of desired results (thanks to advice from Matthew Brett, Jean-Baptiste Poline, and Jarrod Millman).

C.4.1 Initial Correction to Represent Theoretical Idea

To account for our data's lack of any easily identifiable grid structure between when a stimulus was recorded and when our scans occurred (on the order of every 2 seconds), we went back to the theory of convolution and implemented code to recreate equation 11 directly. To do so, we also had to create a function that works with all discrete points of f, the stimulus response as potential starts of the hemodynamic response, multiplied by the actual value of f, as seen in equation 12:

$$r(t) = \sum_{i=1}^{n} g'_i f(t - t_i) \tag{12}$$

where g'_i is the value of g' at t_i (allowing for zeros and varying non-zero values of g').

C.4.2 Matrix Multiplication

Equation 12, reproduced below

$$r(t) = \sum_{i=1}^{n} g'_i f(t - t_i)$$

can be rewritten as a matrix multiplication problem and can be seen below:

$$r(t) = g^*(t)^T f^*(t) \tag{13}$$

where g^* is a vectorized function of g' of t as a scalar output and f^* is the vector of f values (irrespective of location, as the t^* takes that into account). This is a useful representation, since matrix multiplication is faster for Python's numpy arrays.

C.4.3 Using FFT with `np.convolve`

The "theoretical" solution lacked computation efficiency (despite considerable speed improvements from matrix multiplication), so we also approached the problem by creating a denser grid between each scan (two seconds apart). Then we rounded the actual times of the stimulus to meet this more finely scaled grid. This allowed us to utilize `np.convolve` with its faster, FFT-based algorithms, before reducing back down to our two-second grid.

C.5 Example

Now that we have discussed the theory and possible implementations for convolving event-related stimulation, we will look at a basic example from our data. In doing so, we examine the trade-offs between theoretical accuracy and computational efficiency. We will consider just subject 001's condition files.

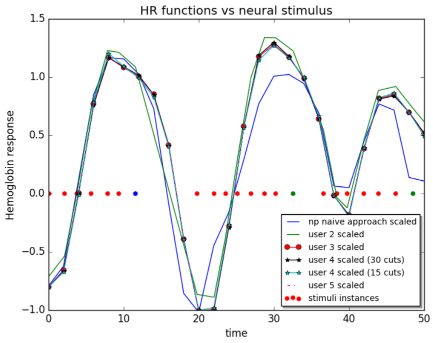

Figure 26: Different convolution functions vs. the Neural stimulus

name in graph	Speed per loop
np naive approach	14.4 μs
user 2	972 ms
user 3	1.15 s
user 4 (15 cuts)	98.3 ms
user 4 (30 cuts)	185 ms
user 5	110 ms

Figure 27: Speed to create HRF predictions for Subject 001, all conditions

The first method in the table “np naive approach” method blindly plugs our data into the `np.convolve` function and is provided to showcase potential speed. The “user 2” method was the first approach to match the theory, though it matches the stimulation times instead of the scan times. The “user 3” method is the most theoretically sound model (and is our standard for accuracy). The “user 5” model is our matrix version of “user 3” and has the same accuracy, but is observably faster. The “user 4” methods involve the grid cut usage of `np.convolve` with notations for the number of slices between each scan. We concluded that ”user 4 (15 cuts)” was the best approach since it gives us speed and very close accuracy to the golden standard — ”user 3”.

D Time Series Analysis

Cohen’s paper [2] discusses analyzing the data with time series using FILM (FMRIBs Improved Linear Model). While we are not familiar with the FILM method, we did try modeling individual voxels in the framework of an autoregressive integrated moving average (ARIMA) process. We focused only on a single voxel from the first subject, but the method could easily be extended to additional or aggregate voxels. Let $\{Y_t\}$ be a single volume’s value at time t and assume that the dth difference $W_t = \nabla^d Y_t$ is weakly stationary, defined to be when W_t has a constant mean function and autocovariance dependent only on lag k and not time t. Then we can try to model W_t as a linear combination of p autoregressive terms (or the number of most recent values to include) and q moving average terms (the number of lags to include for the white noise error terms):

$$W_t = \phi_1 W_{t-1} + \phi_2 W_{t-2} + ... + \phi_p W_{t-p} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - ... - \theta_q e_{t-q}.$$

White noise is defined as a sequence of independent, identically distributed random variables. In order to fit an ARIMA process, the three orders p, d, and q must be first be specified, and the the associated coefficients estimated. We used a combination of visual inspection and quantitative methods to specify the ARIMA orders, and then used the maximum likelihood method to estimate parameters.

Having specified the order for d, we turned to the problem of specifying p and q. We used a combination of visually inspecting the autocorrelation and partial autocorrelation plots of the first difference, and looking at the Akaike information criteria (AIC) and Bayesian information criteria (BIC) computed from a grid of possible models. The latter method suggested specifying $p = 1$ and $q = 1$ (based on either the AIC or the BIC), which was also supported by the visual inspections.

We estimated the parameters for an ARIMA(1,1,1) model using the exact maximum likelihood estimator via Kalman filter. The residuals appear to be normally distributed, and its autocorrelation and partial autocorrelation plots also do not raise any red flags. Furthermore, when visually comparing the fitted time series to the true observed data, the ARIMA process seems to approximate the observed data much better than any of the linear regression models. However, since specifying the correct ARIMA process orders and estimating the associated parameters must be done separately by hand for each individual voxel of interest, we decided to eliminate this direction of analysis from our main pipeline.

Had we decided to continue pursuing time series analysis, we may have tried to forecast future observations based on previous ones. As an example, we modeled an ARIMA(1,1,1) process based on the first half of the observations for a single voxel. This process was then used to forecast the second half of the observations. A comparison between the true observations and the forecasted predictions is shown in [Figure 28]. While the forecasted observations look reasonable for approximating the true values, more quantitative and robust metrics for assessing performance need to be implemented.

One such procedure for assessing performance would be to design a permutation test. A null voxel time course could be simulated from by performing a Fourier transform on the observed time course, permuting the phases, and then transforming back to the original space. That way, the simulated time course has the same autocovariance as the observed time course, but random signal (as under the null case). We can then fit the same ARIMA model to the permuted process and examine how much, if at all, the ARIMA process fitted to the observed data makes improvements over the null case. Generating confidence intervals for the parameter estimates and forecasting future observations may also be of interest. Other considerations for modeling voxels as time series include exploring efficient and reasonable techniques for comparing multiple voxels both within and across subjects.

Other approaches for modeling time series could also be considered, such as spectral density or modeling the conditional variance instead of the conditional mean.

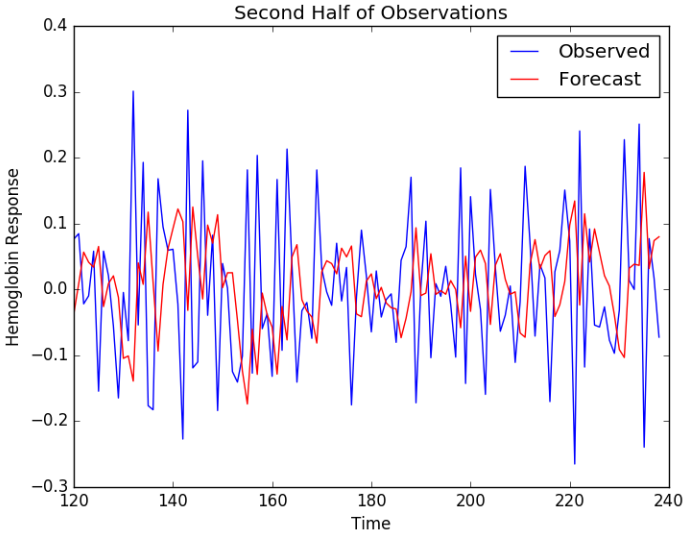

Figure 28: Forecasting the second half of observations based on the first half.

References

[1] M. Brett and J.-B. Poline, *Course on practical neuroimaging in python — practical neuroimaging analysis*. "http://practical-neuroimaging.github.io/on_convolution.html".

[2] J. R. Cohen, *The development and generality of self-control*, ProQuest, (2009), p. 164. "http://gradworks.umi.com/34/01/3401764.html".

[3] A. Ghasemi and S. Zahediasl, *Normality tests for statistical analysis: a guide for non-statisticians*, International journal of endocrinology and metabolism, 10 (2012), p. 486. "http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3693611/".

[4] E. W. Weissten, *Convolution, mathworld - a wolfram web resource*. "http://mathworld.wolfram.com/Convolution.html".