The Neural Basis of Loss Aversion in Decision-Making Under Risk

Kong, Victor VictorKong94 | Li, Ce karenceli | Liu, Anna liuanna | Xia, Yunfei yfxia | Qin, Weidong j170382276

November 30, 2015

Abstract

It is a well-known fact that certain areas of the brain play a larger role in some tasks than others. What is not as well-known at the moment is which parts of the brain correspond to what roles, as well as how much they contribute to these roles. This study follows up on another investigation that found an area of the brain that plays a substantial role in decision-making, employing additional algorithms to corroborate and add to the findings reported. The goal of this project is to demonstrate the importance as well as the proper process of reproducible computational science.

1 Introduction

This study was inspired by "The Neural Basis of Loss Aversion in Decision Making Under Risk," a 2007 article by Sabrina M. Tom and her associates at the University of California, Los Angeles [1]. The original study it describes made several major findings that will be corroborated and further investigated using methods of computational research and machine learning.

One finding of interest to this study is that certain areas of the human brain responded with increasing activity given higher potential gains, but that no area responded similarly given higher potential losses. Furthermore, many of the same areas responded with decreasing activity given higher potential losses, but no area responded similarly given higher potential gain. This would suggest that the same areas of the brain are responsible for processing possible gains and losses, an idea that will be investigated in detail.

This report will describe briefly attempts to reproduce particular methods used in the original study, but will focus more on novel approaches not employed by the previous researchers. The hope is that some of these methods differ drastically from those that have already been used, and henceforth have the potential to shed light on discoveries that were not possible before.

Lastly, this study will place heavy emphasis on the theme of reproducibility, which is defined to be how easily a line of analysis can be replicated in its entirety, either by the original analyst or an independent party. To achieve this, any code that is used to interpret the data, be the outcome conclusive or not, will be well-written, documented, and reviewed prior to publication.

2 Data

The original study recruited sixteen right-handed, English-speaking individuals between the ages of 19 and 28. There is confusion about the sexes of the individuals as the supplement to the published article reports seven males and nine females, while the data made publicly available reports eight males and eight females. It is known that all subjects were physically healthy and free of neurological and psychiatric history, and that their informed consent was acquired prior to their participation in the study.

These subjects participated in three runs of 85-86 trials of a single task. This task involved offering the subject a wager with an equal chance of winning or losing known amounts of money. Potential gains were in even denominations between \$10 and \$40, while potential losses were in integer denominations between \$5 and \$20. The participants were prompted to either strongly accept, weakly accept, weakly decline, or strongly decline the wager in each trial. At the same time that investigators collected these behavioral responses, data pertaining to neural activity was recorded via fMRI. This data set is available for download from OpenFMRI.org under the name “Mixed-gambles task” (accession number ds005), and offers both the neural activity data as well as the behavioral data.

The neural activity data for each run is saved under the filename `bold.nii.gz`, each file containing 240 volumes obtained over the course of eight minutes. Each volume further contains exactly 139264 voxels, being 64 points in length, 64 points in width, and 34 points in depth. As per usual with blood-oxygen-level dependent (BOLD) analysis, each voxel also corresponds to a signal at a given time. A larger signal indicates increased oxygenated blood flow to the corresponding are of the brain, which this paper will refer to as “activation.”

The behavior data for each run is stored separately under the alias `behavdata.txt`. Each file contains a table for the run, in which each trial of the task is stored as a row. Each row then contains seven elements, referred to in the file as `onset`, `gain`, `loss`, `PTval`, `respnum`, `respcat`, and `RT`. The first three elements respectively denote the time at which the trial was presented, the proposed monetary reward given in the event the subject wins the trial, and the proposed monetary loss suffered in the event the subject lost the trial. The fourth element is a standardized value indicating the relative expected gain from the wager. The fifth element represents the subject’s response: `1` for strong acceptance, `2` for weak acceptance, `3` for weak rejection, `4` for strong rejection, and `0` for nonresponse. The sixth element contains the patient response converted to a binary variable: `0` for rejection, `1` for acceptance, and `-1` in the outstanding case of nonresponse. The final element is the time it took the subject to submit a response, measured in seconds and with precision to within a millisecond.

3 Methods

3.1 Smoothing

Prior to statistical analysis, we perform a series of preprocessing steps to minimize the influence of data acquisition and physiological artifacts.The principle justification for using the Gaussian as a smoothing filter is due to its frequency response. Most convolution-based smoothing filters act as lowpass frequency filters and are designed to remove high spatial frequency frequency components from an image. If the spatial extent of a region of is larger than the spatial resolution, smoothing may reduce random noise in individual voxels and increase the signal-to-noise ratio within the region. Here, our 4-dimensional image data were spatially smoothed using a 5 mm FWHM Gaussian kernel as suggested in the reference material.

In addition, additional covariates may be included to account for periodic noise present in the signal, such as heart-rate and respiration. Since more often than not, the effects of physiologicla noise are left unmodeled, the aliased physiologicla artifacts may give rise to the autocorrelated noise in fMRI data. We thus perform exploratory data analysis on residuals, i.e., difference between reponse vector (data in the bold image) and fitted values calculated from linear regression, to fit it with AR(1), AR(2) and model this seasonal component.

3.2 Convolution

Since our study is constructed in event-related design, the onsets are not evenly spaced. This causes problems when we use the traditional way to convolve the canonical hemodynamic response function with the neural prediction that we get from condition files. Therefore, we manually calculate the convolution. Another problem we encountered is that the duration of each second is too short. Usually, the hemodynamic response lasts 30 seconds when an individual is facing a task. However, in our case, the duration for each task is only three seconds. That is, all hemodynamic responses are overlapped, and we need to be cautious when we convolve. So, at the same time point, we add all the hemodynamic response and convolve with hemodynamics response regarding to different onset, duration and amplitude. Since the one run lasts 480 seconds and time to repetition is 2, we remove the all the convolved BOLD signals after 240 TR.

3.3 Linear Regression

Since we have four condition files for one subjects, we get four neural stimulus from these four files and convolve with hemodynamic response respectively. We construct a design matrix where the first column is all 1’s and the rest four columns are BOLD signals we obtain from convolution. The response for linear regression is the BOLD images. Then by performing linear regression, we will be able to get regression coefficients for intercepts and four BOLD signals which corresponds to four conditions. We also calculate the residual errors. In addition, we also identify outliers, perform linear regression for two situations: with or without outliers, and compare the difference.

3.4 Logistic Regression

We select the feature gain, loss, response (0 or 1) behavioral data and calculate the euclidean distance from each gain/loss combination to the diagonal of the gamble matrix to perform the logistic regression. We construct logistic model according to the definition of logistic regression. We calculate the log likelihood and find the coefficients that result in the maximum log likelihood. In order to make the model more accurate, we add a penalty term to our model. Therefore, we have to estimate the tunning parameter for this penalty term using cross validation.

3.5 Hypothesis Testing

Logistic Regression is performed to test and verify a point stated in the paper, "This latter variable was included because of behavioral evidence suggesting greater difficulaty making a decision for trials in which participants had the weakest preference." The null hypothesis states that the coefficient in front of the third beta is 0. The alternative hypothesis is tested against that. The positive weight means that the variable increases the probability of the outcome, while a negative weight means that the variable decreases the probability of the outcome. A large weight means that the variable strongly influences the probability of the ourcome while a near-zero weight means that the variable has litle influence on the probability of the outcome.

The linear regression is performed on the neural data to test the hypothesis that which stimulus (out of four conditions) has a influence/ correlation for each participant (out of sixteen).

4 Results

4.1 Neural Results

By convolution, we get four simulated BOLD signals for four conditions, as shown in [Figure 1]. Before we perform linear regression, we try to find outliers first. So we plot the SD for each volume, as shown [Figure 2]. However, we did not find outliers. Then we perform linear regression and get MRSS about 40.38 which is relatively low. So this model fit the data pretty well.

4.2 Behavioral Results

In logistic model, we try to get the coefficients for each features and perform hypothesis testing, which we are currently working on it. Basically, we want to know which feature influence an individual's behavior most.

4.3 Smoothing Results

In order to apply a multi-dimensional Gaussian filter to 'smooth' the bold image, we use `gaussian_filter` function from scipy module. We make this decision because convolution does not truncate the kernel. Using zero padding, the points towards the edge get pulled down towards zero because they are part-made of the results of taking the product of zero sith the kernel values. This way, we prefer some other method of dealing with the data off the edge of the image. [Figure 3]

5 Discussion

5.1 Discussion of Results

Though we haven't finished yet, the project has been in progress as we planned. For one thing, prior to statistical analysis, we have smoothed the raw data and manually convolved the canonical hemodynamic

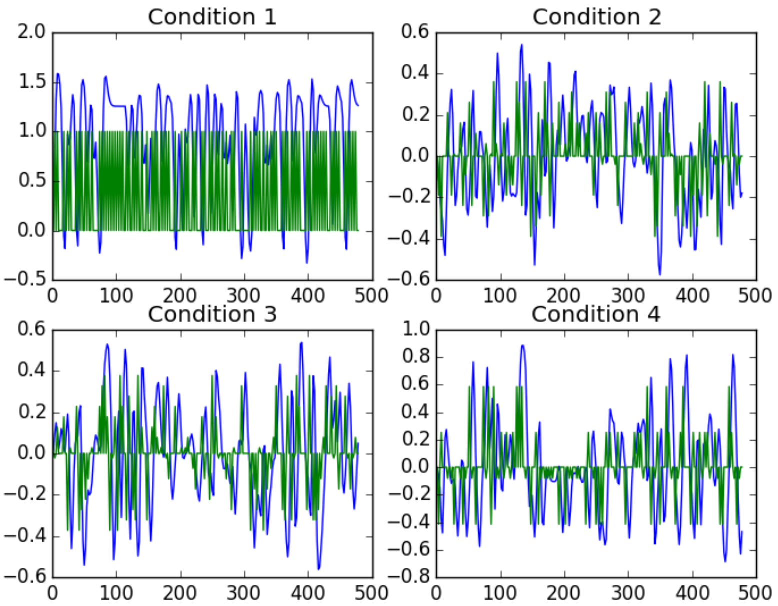

Figure 1: BOLD signals for four conditions

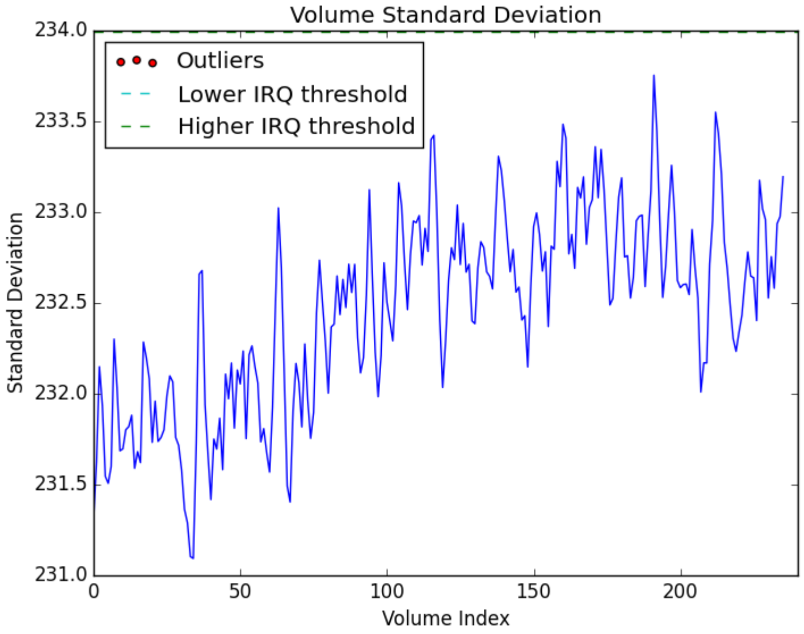

Figure 2: SD at each timepoint

response function. For another, we have applied several methods to do both neural decision analysis and behavioral decision analysis.

Firstly, as for neural decision analysis, we performed linear regression here and found no outliers, which implied this model fits the data very well. Secondly, when doing the behavioral decision analysis, we have constructed logistic models as well as performed hypothesis tests, which have not been finished yet. Nevertheless, we aim at finding out which feature has the biggest influence on an individual's behavior in this part. Last but not least, we also applied a multi-dimensional Gaussian filter to smooth

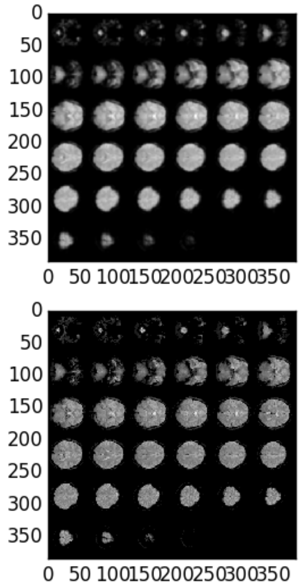

Figure 3: Compare two bold images before and after smoothing

the bold image, based on our thought that convolution does not truncate the kernel.

5.2 Discussion of Future Work

One idea that arose during the preliminary stages of the project that has now become a work currently in progress is the implementation of support vector machines. A support vector machine is a powerful algorithm used for prediction in machine learning, often for classification purposes. In theory, it can allow the user to draw flexible yet robust boundaries between data points given some criterion. If properly implemented, models generated by this algorithm could shed light on an entire class of inquiries, like what parts of the brain would activate given increasing parametric gains and losses, and vice versa. The mechanism by which support vector machines work make such topics natural to work with, while also introducing an additional degree of resolution and confidence in findings that pertain to these topics.

Another scheme that has the potential to be even more difficult to implement is decision trees. What decision trees lack in predictive power, they make up many fold in interpretability. Models generated via a decision tree can clearly and simultaneously show the principle variables responsible as well as how multiple variables interact with each other in the prediction of an outcome. This would greatly facilitate the investigation of such questions as how much does a brain area's activity have to change before the subject changes his/her answer, or how great of an increase in potential gain is necessary for the brain to recognize it over a given potential loss.

References

[1] S. M. Tom et al., *The neural basis of loss aversion in decision-making under risk*, Science, 315 (2007), pp. 515–518.