Project Epsilon progress report
The Neural Basis of Loss Aversion in Decision-Making Under Risk

Jo, Min Gu	Kaam, Soazig	Li, Zhuangdi	Yu, Timothy	Zhi, Ye
mingujo	soazig	lizhua	timothy1191xa	ye-zhi

December 1, 2015

1 Introduction

The study *Neural Basis of Loss Aversion in Decision-Making Under Risk* [1] focuses on decision-making process, especially on the correlation between the neural activity and the reluctance to lose. 16 people were presented 255 gambling situations with a 50% of success. Each situation was associated with a potential gain and loss that were randomly selected. The gains were ranging from $10 to $40 while the losses from $5 to $20. The participants were asked to assess their level of willingness to accept or reject the gamble using a 4-point likert scale [1: strongly accept, 2: weakly accept, 3: weakly reject, 4:strongly reject]. The response time was also recorded for each case. The imaging data were collected using the fMRI method. They were processed and analyzed in order to identify the regions of the brain activated by the decision making process. This study also investigated the relationship between the brain activity and the behavior of the subjects towards the gambling situations using a whole-brain robust regression analysis.

2 The data

The data we are using can be found on the OpenfMRI website at the following address: `https://www.openfmri.org/dataset/ds000005`, the dsnum is ds005. For our project, we are specifically using the behavior data and the BOLD data that are organized.

For each of runs per subject (3), the behavior data contains the timestamp of each survey question (onset), the gain/loss combinations (gain and loss), the response for the particular trial (respnum) from the 4-point likert scale. The researcher created a response category (respcat) to be used in their binary choice model that combines the "reject" answers together on one hand and the "accept" answers together on the other hand. BOLD data contains compressed 4-dimensional brain images for each subject's run. The folder also comport Quality Assurance (QA) files and a report.

3 Exploration on Behavior Data

3.1 Initial work

For behavior data, we wrote a function to merge three runs for each subject since we would like to look at the total observations within one subject. Also we noticed that in the data set of each run, there existed observations with "-1" in "respcat", which was meaningless and might be an error during the experiment so we took them out. For bold data, we also wrote a function to unzip all nii.gz files for further analysis.

3.2 Behavior Data

We did some explanatory data analysis and regression analysis using behavior data. For explanatory data analysis, we generated some summary statistics, including correlation among variables and simple plots to better understand the behavior. And then we used regression analysis to mainly answer two scientific questions. The scientific questions that we have are:

- If gain/loss would be significant for individuals who choose to participate and how much time it would take for them to respond.
- If gain/loss would be significant for whether individuals would like to participate in the gamble.

3.2.1 Linear regression

To answer the first question, we built three linear regression models.

1. Response Time $\sim$ gain + loss
2. Response Time $\sim$ diff(gain-loss)
3. Response Time $\sim$ ratio(gain/loss)

Result
We use the result for one subject to explain our findings. In the first model, p value for the predictor loss is 0.002644 while p value for the predictor gain is 0.547262. Basically, we can conclude that people would actually care more about loss than gain. In the second model, p value for the predictor difference is 0.413985. It turns out that we can conclude that the difference between gain and loss wouldn't be a factor for how much time it would take for a individual to respond. In the third model, p value for the predictor ratio is extremely small, showing that ratio has a huge impact on individuals' response time.

3.2.2 Logistic regression

To answer the second question, we fit logistic regression between Accept/Reject Gamble and gain/loss. According to our analysis, the decision to whether take the gamble of most of subjects, in general, is more affected by loss amount rather than by gain amount. For example, below is the analysis on the subject 3. The regression line shows that it well follows the border between the two decisions: 1 (gamble) and 0 (not gamble). Right side of the line illustrates the decision to not gamble and it takes up more area relative to the opposite decision.

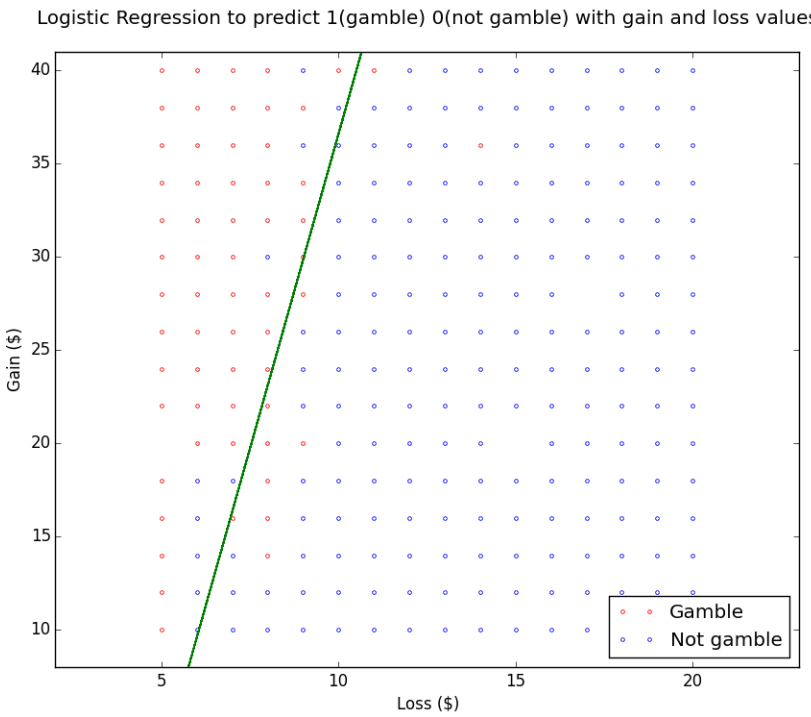

Figure 1: Logistic Regression for Subject 3

3.3 BOLD data

For image data, we did several explanatory data analysis to better under the BOLD data.

- Reproduce Quality Assurance(QA) plots
  We reproduced some of the QA Plots provided in the BOLD folder, including mean signals, Framewise Displacement and DVARS(root mean squared signal derivative over brain mask). The shapes were almost the same, except the y axis. We assumed there was some transformation or preprocessing in the original QA plots.

- Correlation
  We calculated the correlation between task-on/task-off vectors and voxel time courses to identify the active region of the brain.

- Associated the brain image data with behavior data
  We started to associate brain image data(mean/standard deviation across time courses) with gain/loss/ratio, choices(accept/reject) and respcat(strongly/weekly accept/reject).

4 Convolution

4.1 Constructing hemodynamic response function

We can use the gamma function to construct a continuous function that is close to the hemodynamic response we observe for different events in the brain. In our case, we have 4 different events (conditions): task = {0,1}, parametric gain, parametric loss, distance from indifference. However, we cannot really convolve the canonical hemodynamic response function with the neural prediction from text files with condition information because:

1. The events are given at the different onsets in the experiment.
2. The hrf lasts for 30 seconds when the duration for each event is 3 seconds.

Therefore, we tried the normal way to convolve the canonical hrf and the other way to convolve with hrf at the higher time resolution, and compared the resulting MRSS from the general linear regression analysis using the predictor from the normal convolution, and it from the other convolution.

- Convolve with the canonical hemodynamic response function:
  There are 4 condition files.
  use the sum of two gamma distribution probability density functions
  sample from the function, to get the estimates at the times of our TRs (2)
  use this to convolve our neural (on-off) prediction
  We applied this method to all 4 condition files samely and built the design matrix for the linear regression analysis.

- Convolve with the high resolution neural time course:
  There are 4 condition files.
  make a neural and hemodynamic regressor at a finer time resolution than the TRs
  create a new neural prediction time-course where one element corresponds to 1 / 100 of a TR
  sample at the TR onset times. convolve the sampled HRF with the high resolution neural time course

  –> If we compare the MRSS from two linear regressions for three subjects (1,2,3) using convolution predictors from two different methods in the below table, we see the MRSS from linear regression using the latter method has slightly lower values compared to the former method. This makes sense because, using the latter method, we could more elaborately preprocess the data.

5 Linear Regression

5.1 GLM and MRSS

The convolution matrix we get from last section has five columns, which correspond to a column of 1's and 4 cond.txt files from our dataset, respectively. After we have created the convolution matrix, we use it as our design matrix and run the generalized linear regression on our image data. The dimension of our data is (64, 64, 34, 240). First, we reshape our data into 2 dimensional array, which has the shape of (64*64*34, 240). Then we pass our design matrix into the glm function to calculate the related beta hats. These 139624 beta hats we get from the regression correspond to the first three dimensions of our image data. For example, the first beta hat contains the information about the voxel (0,0,0). Then we turn the beta hats back into 4-dimensional shape and run the diagnostic functions on the 4-d beta hats. Based on the predictors, we can calculate the fitted values and then the residuals. We use the MRSS of the first three dimensions as a measurement of our regression.

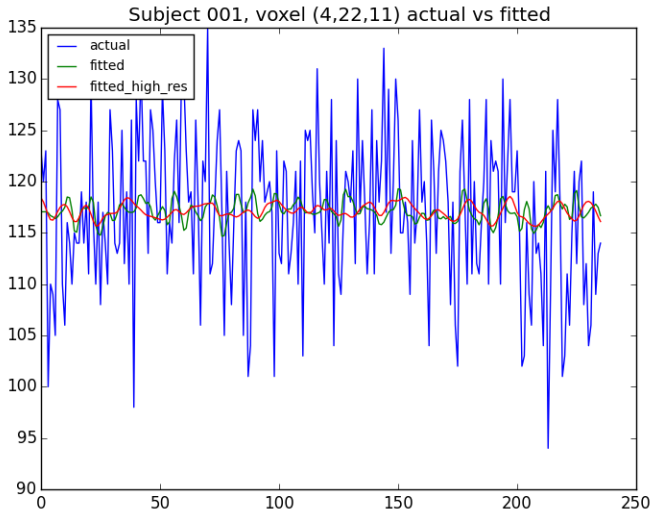

Figure 2: fitted values for the linear regression

After we tried the normal convolution, we also tried the high resolution convolution matrix. It turned out that the MRSS just reduced a little bit. Then we writing the smoothing function to implement the multidimensional Gaussian filter on our data. We repeat the same steps as what we have done in normal convolution on the smoothed data and the MRSS are reduced sharply. Therefore, we concluded that the smoothing method is a good pre-processing when we do the linear regression.

Below are the MRSS values for our subject 1, run 1:

Subject	Run	MRSS_normal	MRSS_high_res	MRSS_smoothed
1	1	40.82952276	40.62297392	2.966249345
1	2	75.42945535	75.1390682	3.914076412
1	3	75.65312045	74.92416373	4.92729471

Figure 3: MRSS for subject 1 run 1

5.2 t-test and p-values

After getting the beta hats, we write the function for t-test, which will give us the corresponding t statistics and p values for our beta hats. Then we set our significant level to be $\alpha = 0.05$; thus, if the beta has a p-value greater than or equal to 0.05, we will say that it is not significant; if the beta has a p-value smaller than 0.05, we will say that this voxel is activated.
Therefore, we write a function called find_activated_voxel to change the voxel positions in the long list back to the 3-d voxel indices. We utilize the functions that we write for our homework to make transitions between them. After we get the positions of these significant voxels for our four predictors, we write them out into txt files.

5.3 Next Step

The affine attribute of the image contains the mapping from voxel coordinates to millimeters in the space of the image. Therefore, in our next step, we plan to get the mm coordinate for a particular voxel coordinate

by applying this affine to our particular voxel coordinate, because the millmeter coordinate makes more sense if we want to do further analysis on the brain image.
Besides, we haven't check the assumptions of regression. Therefore, we might check the normality and constant variance assumptions, but now we assume that these assumptions are satisfied since we have enough data. We also have the function to detect the outliers, so next time, we will get rid of the outliers before regression.

6 Hypothesis testing

6.1 Correlation

Before linear regression and hypothesis testing, we can look at the correlation between different conditions(task on/off, gain, loss, distance) vectors and voxel time courses. After calculating the 3-D correlation matrix, we can visualize the correlation heat map. For visualization, we took the third dimension, height as slices. Thus we have 34 slices in total. Take subject 1 run 1 as an example. We plotted the correlation between gain/loss and voxel time courses. Below are the correlation maps.

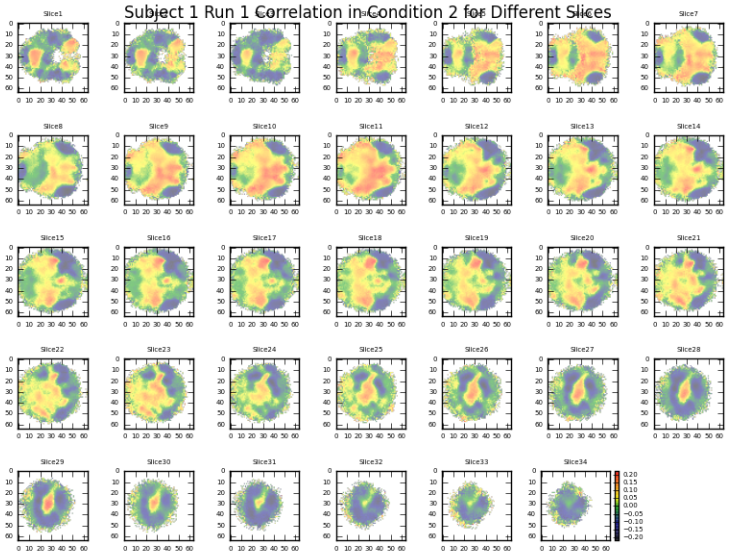

Figure 4: Correlation between Gain and Voxel Time Course

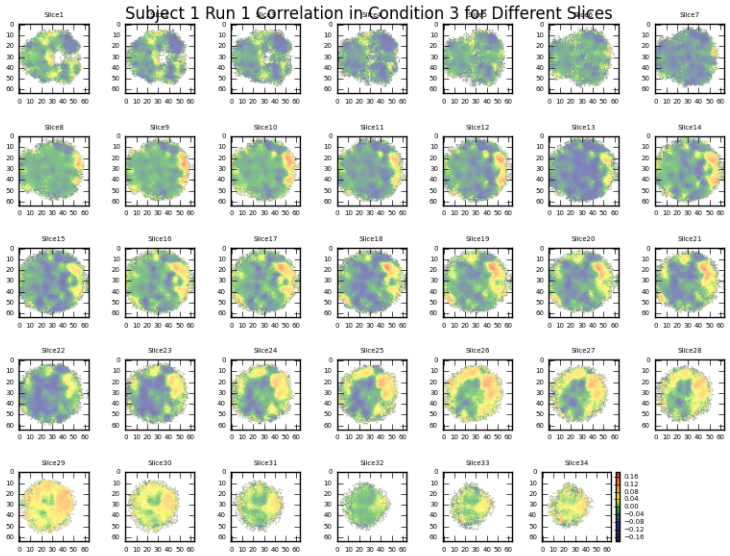

Figure 5: Correlation between Loss and Voxel Time Course

From the correlation figures, the redder, the more correlated to the gain/loss convolved vector. Thus the red spots represents the activated voxels for gain/loss. Gain has more red spots than loss. Paper mention that subject 1 is special that he/she focused on more gain when gambling, which is consistent with the result we got. We can compare the correlation map to t statistics map later.

6.2 Hypothesis Testing

From linear regression, we can get t-statistics for different conditions(task on/off, gain, loss, distance).For each condition, we will have a very similar 3D t-statistics matrix to the correlation matrix. For visualization, we first added mask based the mean voxel and the histogram. We set a boolean mask which takes larger than 400. Also we used smooth function and better color txt to generate a better image. Then we plotted the t statistics map for gain/loss.

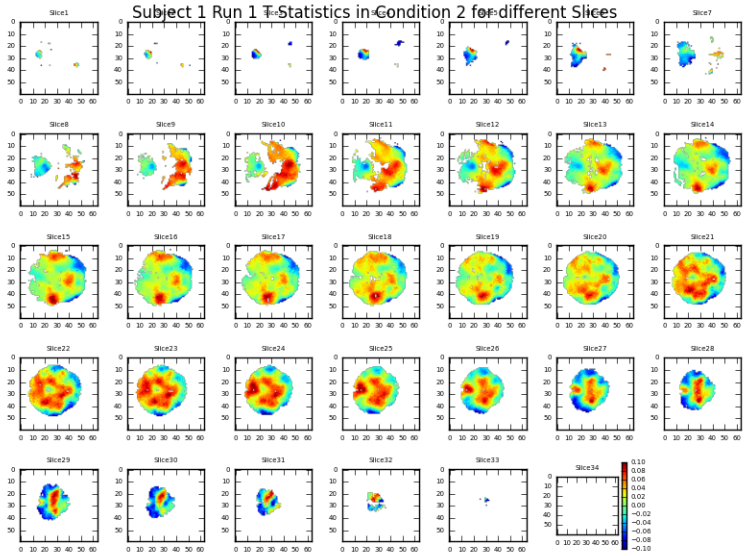

Figure 6: T Statistics for Gain

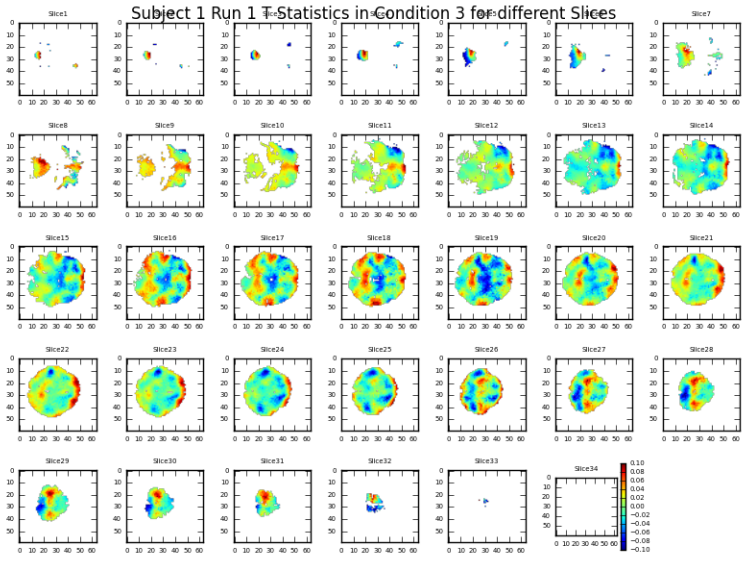

Figure 7: T Statistics for Loss

The larger the t-statistics, the more significant. Thus the red spots represents the activated voxels for gain and loss. Similar to the correlation, gain has more activated voxels. T-statistics figures are similar to correlation, but more clear.

6.3 Findings and Next Steps

From the correlation and t statistics, we can locate and visualize the activated voxels responsible for gain or loss(other condition vectors). For subject 1 run 1, he/she has more activated voxels based on gain than loss. However, it will change across subject. Thus using affine, we can compare activated voxels across subject and run.

7 Noise Modeling

In our quality control analysis, we were interesting to remove the noise from the BOLD images. We first detect the voxels which are inside of the brain. We did not know if the threshold would be the same accross all subjects and runs. After plotting a few histograms of the mean value of the voxels accross time for different subject and runs, we settle for a common threshold value of 300.

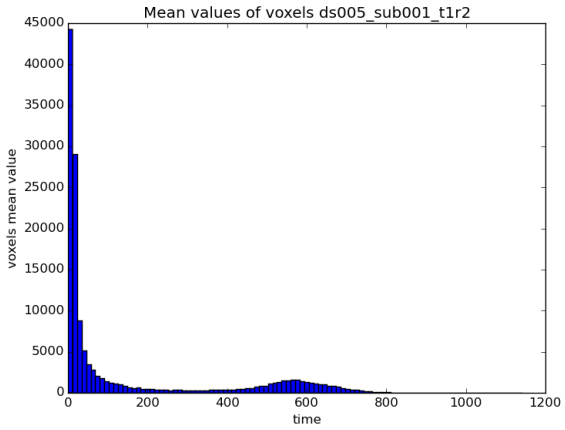

Figure 8: Histogram of mean voxels accross time - subject1 - run1

Applying the mask on the image data allow us to select only the voxels that are in the brain. We then model our signal with some additional drift terms to account for the subject head motion in the scanner. We deceide to add a linear and a quadratic drift terms as regressors. The following brain images justify our design decision.

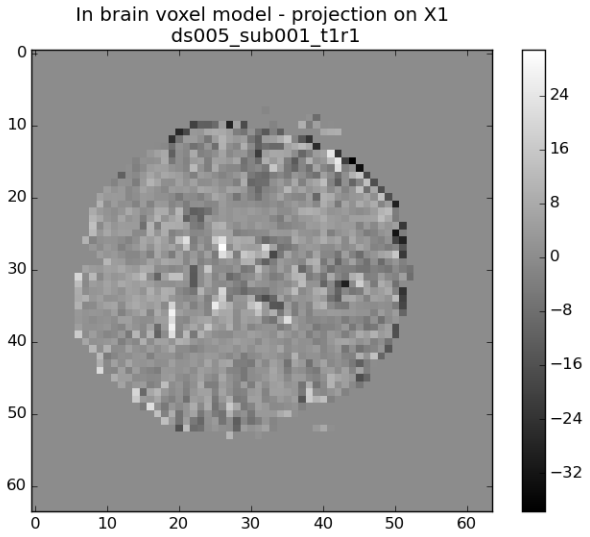

Figure 9: Projection of the brain image on the linear drift related regressor - subject1 - run1

The above image illustrates the effect of the linear drift regressor on the BOLD image. Lighter pixels indicates a higher position of the zone compare to darker zones. We can clearly see the edges of the top right of the brain to be dark (black) and the edges on the bottom left of the brain to be light (white). This suggests a clear linear movement of the head in the direction of the axis.

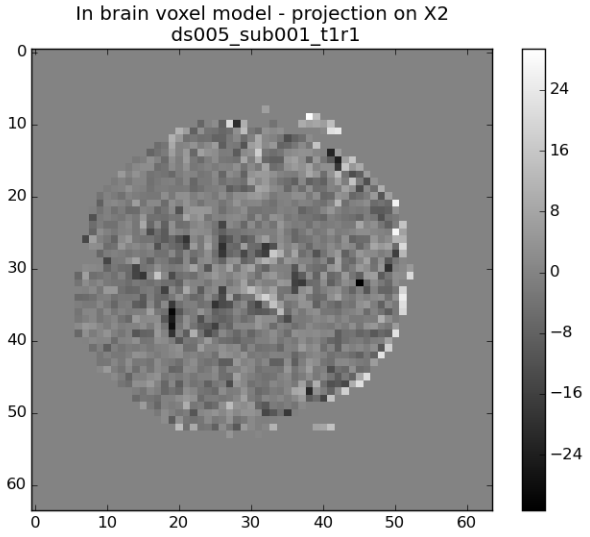

Figure 10: Projection of the brain image on the linear drift related regressor - subject1 - run1

The aboce image clearly illustrates a strong influence on the quadratic regressor in the design matrix. We can clearly distinguish light and dark areas on the above images.

Our design matrix includes the above drift regressors in order to get a more accurate estimation of the task which effects will not be confounded with the movement of the brain.

8 PCA analysis

8.1 PCA and projections

We decided to perform a PCA analysis in order to improve the performance of our linear model. After performing a Single Value Decomposition, we want to detect which which regressor mstly related to noise we have to include in our design matrix. Below the projection of the brain data onto the new basis U.

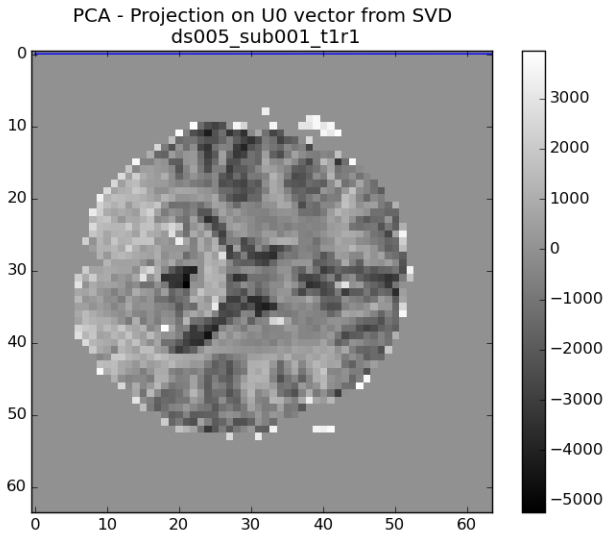

Figure 11: Brain data projected onto the new basis U - projection on U0 - subject 1 - run 1

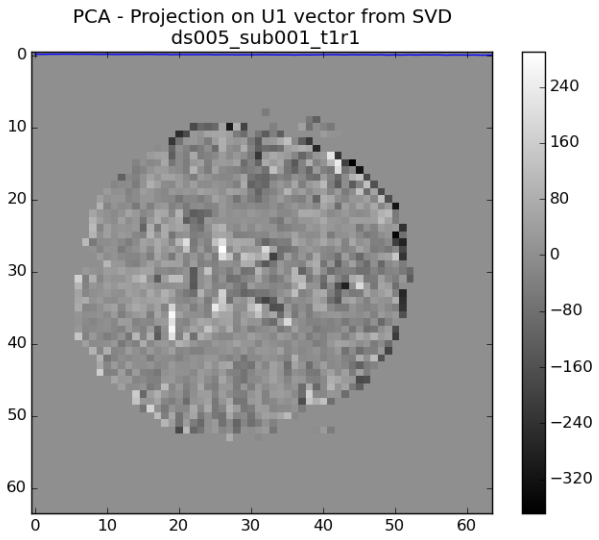

Figure 12: Brain data projected onto the new basis U - projection on U1 - subject 1 - run 1

The two images above, show that strong localization of the voxels at the edges of the brain, which suggests that they reflect brain anatomy rather than activation. We decided to remove these components by regression because they are likely to be related to noise from the scanner or the subject.

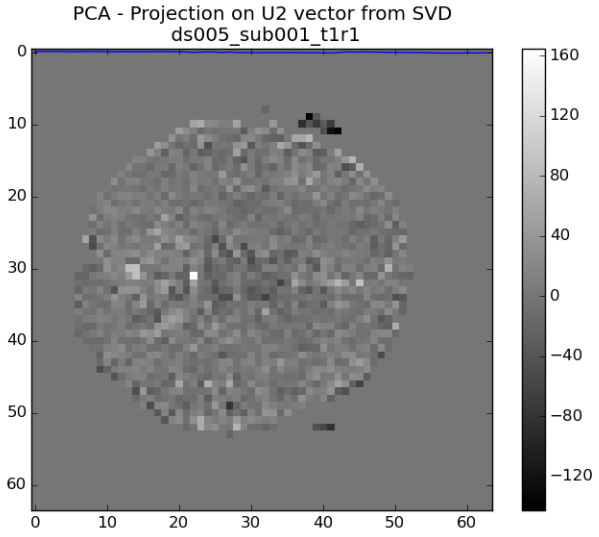

Figure 13: Brain data projected onto the new basis U - projection on U2 - subject1 - run 1

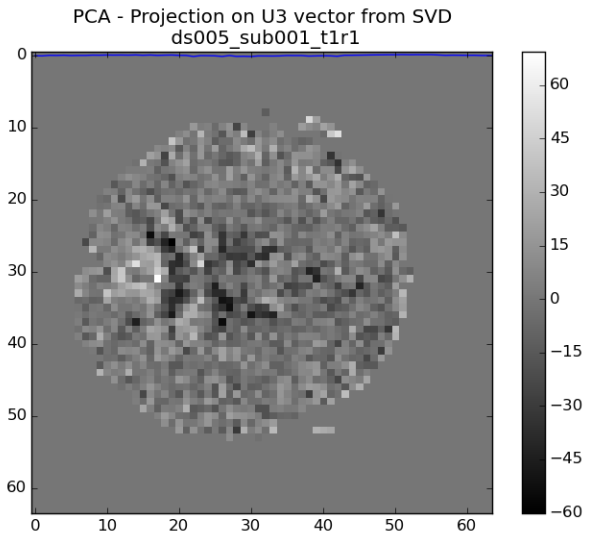

Figure 14: Brain data projected onto the new basis U - projection on U3 - subject 1 - run 1

Contrary to the precedent cases, the above two images of the projection of th 3rd and 4th component don't seem to reveal any random pattern. The projection on the 4th component show a darker area localized on the startium. In order to determine how many more components to remove from the analysis, we plotted the following Variance Explained plot. In the Single Value Decomposition, the vector S contains the square roots of the singular values ordered from greatest to least along its diagonal. Each value indicates the variance of the component vector (time-course) along each dimension of U. In the figure below, each elements of the matrix S has been normalized. We decided to remove the first component from the graph for better clarity. The first principal component accounts for 98 percent of the overall variability, the second explains for 0.03percent. In this particular case, removing the first principan component from the regression makes the more sense, which is a bit different from our previous intuition that considered the first two.

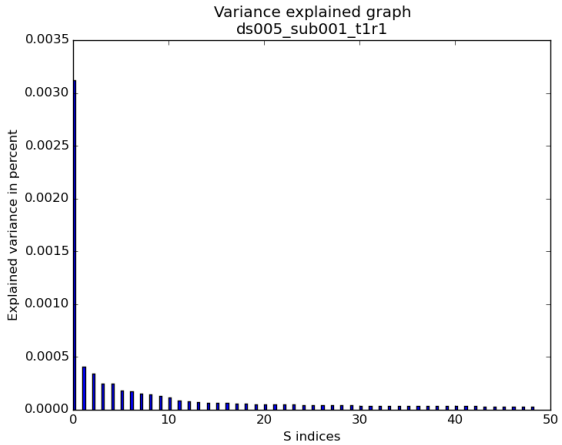

Figure 15: Variance explained for the subject 1 - run 1 (first term remove for clarity of the graph)

After including the first principal component in our design matrix, we perform our linear regression again and a new design matrix. The mean residual sum of square did not different much before and after the PCA analysis. Before the PCA analysis, the mean of the MRSS is 95.9, after it decreases to 94.6 for this particular subject and this particular run.

8.2 Next steps

We need to extend the study accross all subjects of the experiments and we want to validate our model accross runs for the same subject. We find difficulties in validating our model for the same subject with a different run. The challenge comes from the fact that the mask applied to determine which voxels are inside the brain change the dimension of our data.

References

[1] S. M. Tom, C. R. Fox, C. Trepel, and R. A. Poldrack, *The neural basis of loss aversion in decision-making under risk*, Science, 315 (2007), pp. 515–518.