Curds and Whey Regression Applied to fMRI

Chris Gagne and Alan Cowen

April 26, 2015

1 Introduction

When making inferences on multiple responses (Y), performance on each can be improved by pooling information across problems. Moreover, the amount of improvement depends in part on how similar the problems are. For fMRI data, inference is typically done for each individual voxel, but we anticipate that gains can be made by combining inference across multiple voxels. fMRI data is characterized by spatial correlations across voxels, some of which is due to a functionally-modular topographic organization of cortex. Therefore, this spatially correlated signal may make fMRI particularly conducive to a technique that joins inference across voxels. The particular technique we chose was the "Curds and Whey Regression" (CW) proposed by (Breiman and Friedman 1997)(BF).

Our project has three major parts. First, we wanted to replicate the simulation results of Breiman and Friedman to verify that our implementation was correct, and assess CW's performance across a range of situations. Second, we wanted to apply CW to simulated fMRI data to assess its performance in a more realistic simulation. Finally, we applied CW to a real fMRI data set, in which we predict held out data in order to choose scientific models of cortical organization.

2 Background

2.1 Setup

Our setup is one of predicting multiple voxel response time courses given multiple experimental predictors (stimuli presented to the participant). We are interested in both predicting new voxel responses, and estimating the 'true' parameters underlying the voxel responses.

The voxel responses at a given time point is a row vector $\mathbf{y'} = (y_1, \ldots, y_q)$. We've assumed that each voxel's response can be described as a linear combination β of responses to our experimental manipulation $\mathbf{x} = (x_1, \ldots, x_p)$ and additional zero-mean error.

$$y_i = \mathbf{x}\beta_i + \epsilon_i$$
$$y_q = \mathbf{x}\beta_q + \epsilon_q$$

2.1.1 OLS

In the typical approach to fMRI analysis, we stack the y's and x's across the n time points and solve using OLS, in which the estimation for each y is solved independently of all others.

$$\hat{\beta}_{ols} = (\mathbf{X'X})^{-1}\mathbf{X'Y}$$
$$\hat{Y} = \mathbf{X}\hat{\beta}_{ols}$$

Although OLS is the minimum variance unbiased estimator, improvements in prediction can be made by shrinking the estimates towards zero. Copas (1983) showed that the expected slope of a regression of future y's on our OLS estimates $\hat{y}$ will be too large. Therefore, his method and many others shrink the $\hat{y}$ prior to prediction.

2.1.2 Ridge

One such method is Ridge Regression which estimates the amount of shrinkage λ using cross-validation. We currently use Ridge on our data.

$$\hat{\beta}_{rr} = (\mathbf{X'X} + I_p\lambda)^{-1}\mathbf{X'Y}$$

2.1.3 OLS-Curds-Whey

Both Ridge Regression and Copas' method shrink the OLS estimates based just on x. BF proposed that further improvements can be made in situations with multiple responses. Their method is based multiplying the OLS estimates by a shrinkage matrix S that incorporates the relationship between multiple y's and x's.

$$\hat{Y}hat = \mathbf{X}\hat{\beta}_{ols}S^{cw}$$

BF derive the optimal shrinkage matrix (in an idealized setting), and highlight its relation to a canonical correlations analysis (CCA) of x and y. To see this, we start with the following combination of covariances for y and x,

$$Q = E(yy^T)^{-1}E(yx^T)E(xx^T)^{-1}E(xy^T)$$

which BF show relates to optimal shrinkage in the following way (where r is the ratio of p/n):

$$S^{cw} = ((1-r)I + rQ^{-T})^{-1}$$

In CCA, Q is usually decomposed into its eigenvectors and eigenvalues.

$$Q = TC^2T^{-1}$$

where T represents a new basis for Y in which the covariance between Y and X is a diagonal matrix C, and diagonals are maximized across all possible bases (for X and Y)(which are known the squared canonical correlations). Doing this, we can re-write the optimal shrinkage matrix as,

$$S^{cw} = TDT^{-1}$$
$$D = \frac{c^2}{c^2 + r(1-c^2)}$$

where c^2 are the diagonal elements of the C^2.

Thus this decomposition uncovers what BF's shrinkage achieves. Multiplying $\hat{Y}$ by S^{cw} effectively transforms y into a new basis where x and y are maximally correlated, shrinks relative to the canonical correlations, and inversely transforms back. Intuitively, CCA has found a new basis for our voxels y and experimental manipulation x in which they have the strongest mutual relationship. For example, an important dimension in Y space may be one in which many voxels respond similarly to a certain experimental manipulation. Shrinkage by D will preserve this dimension while shrinking less important ones (ie. noise).

In practice, we estimate Q with our observed data:

$$Q = E(Y^TY)^{-1}E(Y^TX)E(X^TX)^{-1}E(X^TY)$$

Then because our sample estimates for the values of c^2 tend to be too high, we adjust those as well. Using generalized cross validation, we can approximate how much we should decrease the canonical correlations.

$$D = \frac{(1-r)(c^2-r)}{(1-r)^2c^2 + r^2(1-c^2)}$$

This is the D we use in our implementation. It is important to note that when q>p, there are only p canonical correlations. Therefore, like BF, we set those elements in D=0.

2.1.4 Ridge-Curds-Whey

In situations where p is close to n, it may be beneficial to combine ridge and CW shrinkage. To do this, you multiply the estimates found via ridge regression by a similar shrinkage matrix.

$$\hat{Y}hat = \mathbf{X}\hat{\beta}_{rr}S^{cw}$$

Two differences to the above procedure are that the CCA was done on $(Y, \hat{Y}\_\{ridge\})$ rather than on (Y, X), and r is calculated using

$$\frac{1}{N}trace((\mathbf{X}^T\mathbf{X} + I_p\lambda)^{-1}\mathbf{X}^T)$$

which can be considered the effective' degrees of freedom of the model.

2.1.5 Double Ridge

As an alternative approach to using CW and CCA, we tried to estimate the optimal shrinkage directly by predicting predicting held out Y from $\hat{Y}$ using cross validation with added L2-regularization:

$$\hat{S} = argmin_S(||Y - S\hat{Y}|| + \gamma||S||$$

This is essentially a second step of ridge regression (i.e. ridge was first used to predict Y from X on a voxel-by-voxel basis, then for each voxel used to predict Y from $\hat{Y}$ of voxels in a surrounding neighborhood).

To determine the regularization parameter, γ (to be distinguished λ from the first ridge regression of Y onto X), we used 9-fold cross-validation in the estimation set. For each fold, $ S_k $ was chosen as

$$S_k = argmin_S(||Y_{estimation,trainingfold} - S\hat{Y}_{estimation,trainingfold}|| + \lambda||S||$$

And used to produce

$$\hat{Y}_{new,estimation,testingfold}$$

After cross-validation, we selected the γ that resulted in the highest correlation between $(Y, S\hat{Y})$. This was done separately for each voxel, i.

3 Part I: Simulations (BF)

3.1 Summary

The two primary goals of these simulations were to verify our implementation of the CW shrinkage applied to OLS and Ridge Regression, and to identify situations in which it is most beneficial. We ran 50 simulations on each of the 4 models (excluding Double Ridge) with each of the following values of parameters:

- n observations (200,400)
- p parameters in our design matrix (11,21,51)
- q responses y (5,10,20,50)
- SNR (signal-to-noise ratio) (.01,.1,.5,1,2)
- bsig: variance in the distribution of true B (.01,.1,.5,1,2)

3.2 Data Generation

The data for each simulation was generated in a similar fashion to BF.

1. X: Each x was chosen as a random draw from multivariate Guassian with a covariance $\mathbf{V}$ that was itself randomly generated for each simulation. Each entry in the covariance was random number from -1 to 1.

2. B: The true parameters were generated independently across x, but dependently across y. For each x, a random value was chosen from -1 to 1 to be the mean of the true parameter B. The B were then sampled from that Guassian with a width determined by our free parameter (bsig). With a small variance, the y's would have similar B, a situation that should favor CW.

3. F: The signal matrix was calculated as $\beta X X^T \beta^T$. This was then used to adjust the error variance σ so that different values of bsig would not adjust the overall SNR. σ was related to the overall signal so that SNR = mean(F)/sigma.

4. ϵ: Were independently drawn from a gaussian with mean zero and covariance $I_q \sigma$.

5. We then divided into training and test sets (centering within each), fit the model on the training set and predicted the test set. We assessed performance on both the training set (estimation of true B) and the test set (prediction accuracy).

3.3 Performance Metrics

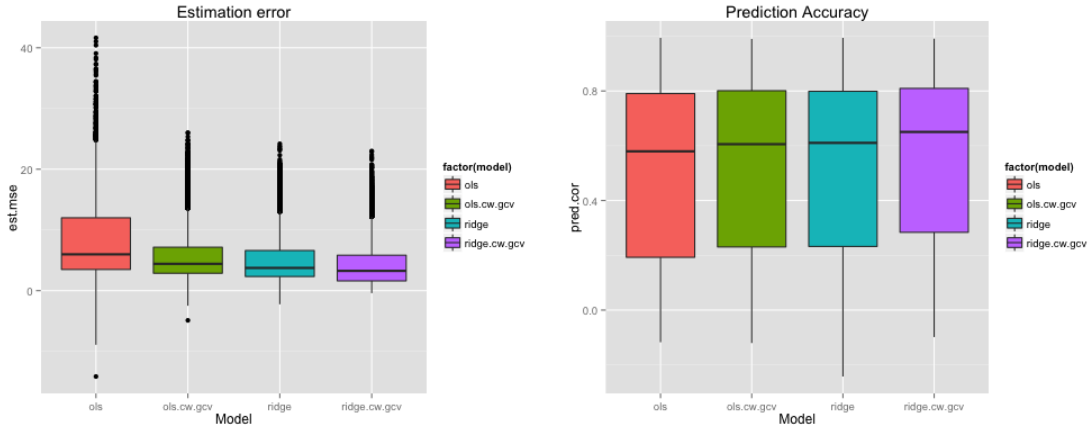
- **Estimation error**: As fMRI analyses are often aimed at estimating the effects of some experimental manipulation, we wanted to assess how well the models estimated the true parameters in the training set. Following BF we used the following loss function, where V is the true covariance used to generate the X.

$$est.mse = |\beta - \hat{\beta}|^T V |\beta - \hat{\beta}|$$

- **Prediction accuracy**: For predictive performance, we measure the correlation of each response $ y_i$ in the test set with its estimate $\hat{y}_i$

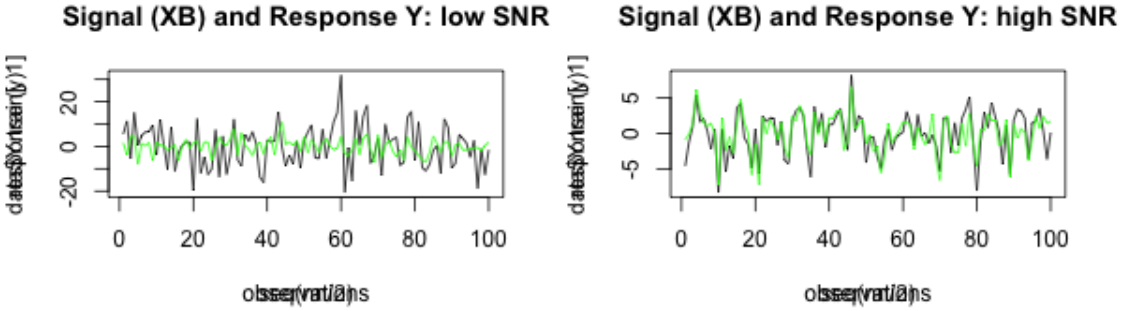
3.4 Results: Overall

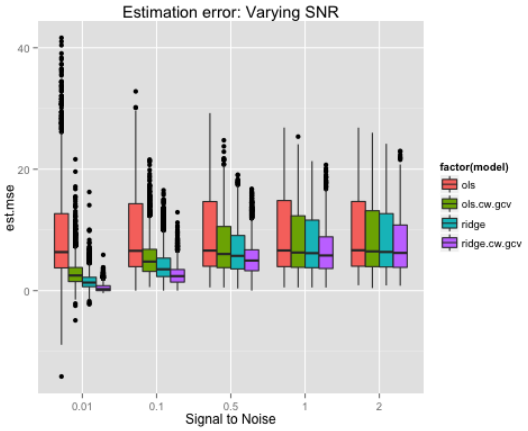
Looking at the average performance across all simulations, CW performed better than OLS, ridge and CW performed similarly, and combining with ridge with CW performed the best. However, depending on the situation these relative performances shifted.

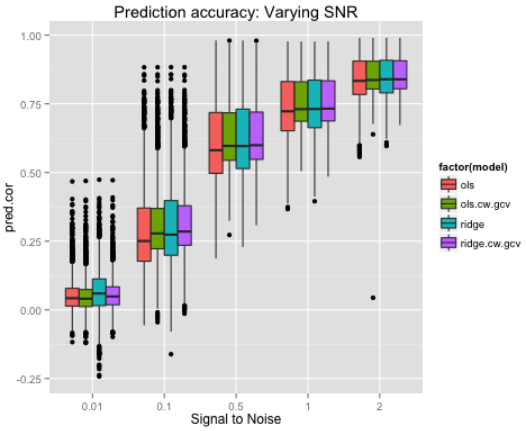


3.5 Results: Signal-to-Noise

Averaging across all other parameters, we looked how the signal-to-noise ratio effected model performance. In general, prediction accuracy increased with SNR, however estimation MSE did not. Importantly, the effect of shrinking the estimates (either with ridge or CW) had improvements over OLS in cases of low SNR (0.01, 0.1) for both estimation and prediction.



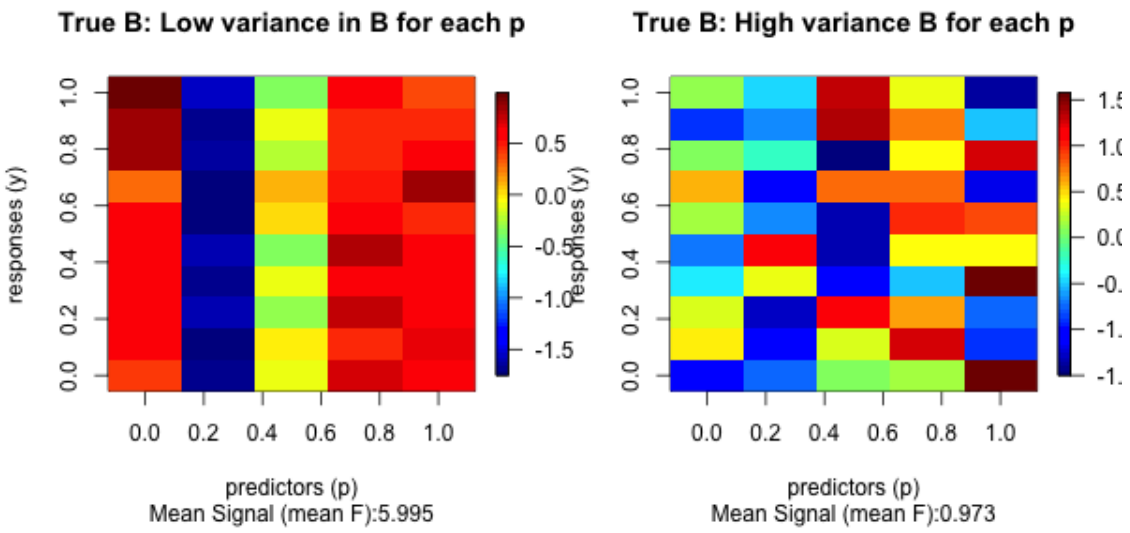


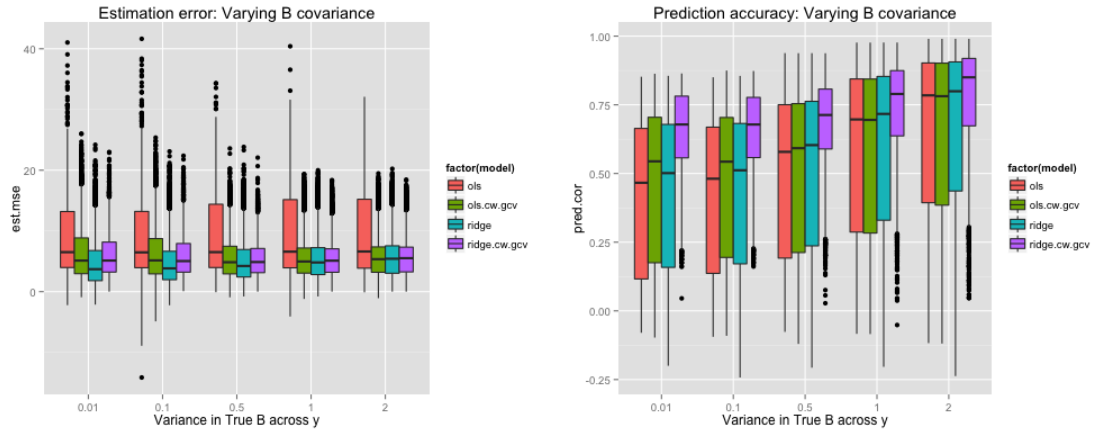


3.6 Results: B variance

Next, we averaged across all parameters except bsig, which determined the variance of B for each predictor. Low variance meant that B's were similar across voxels for a given predictor (left), whereas high variance meant that each voxel could have different B's (right). At each level of B variance, the SNR was kept constant by adjusting σ so the effects for this parameter on performance could be isolated.

The clearest benefit from CW is seen in prediction. For low variance in B's, CW applied to both OLS and Ridge improves performance (right green/purple). This benefit decreases as the variance in the B's increases. For estimation MSE, Ridge actually seems to do better than CW, but more simulations need to be run, and the results might need to be subset by other factors

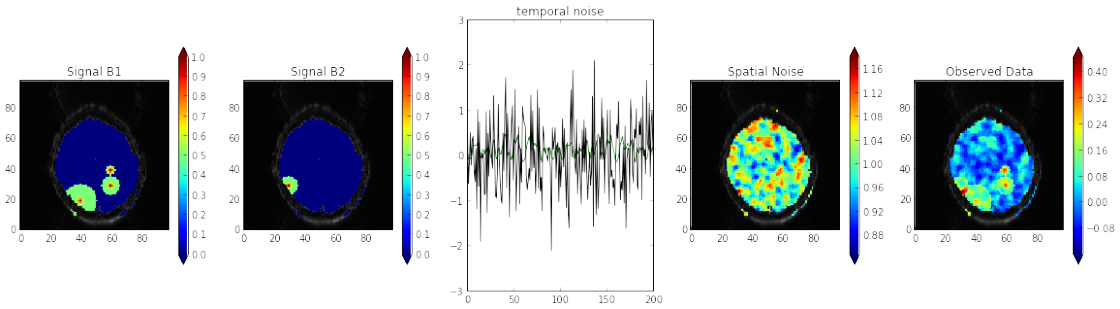




4 Part II: Simulations (fMRI)

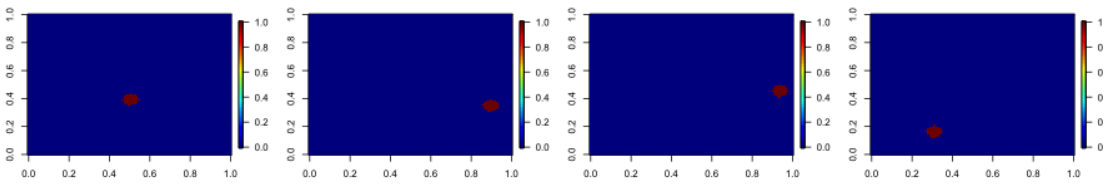
4.1 Summary

In this section, we wanted to extend the simulations to data more similar actual fMRI data. We simulated a simple experiment in which two conditions occurred in a random sequence for a total of 400 seconds. Each condition lasted 1 second (e.g. like an image presentation) and there was a delay between conditions. Several regions were simulated which each responded to one condition or the other. These are shown in the left two panels below. These spatial effects 'B1' and 'B2' were multiplied by the design matrix to create a temporal response for each voxel. Several types of noise were then added. For this particular simulation, we added zero-mean Guassian and AR1 noise to each voxel's time course, and then spatial noise in the form of a gaussian random field with 5 voxel FWHM. The spatial average of the final simulated data is shown on the far right panel. The data simulation was done using a combination of in-house software, and some basic functions (e.g. gaussian random field) from the R package neuRosim.



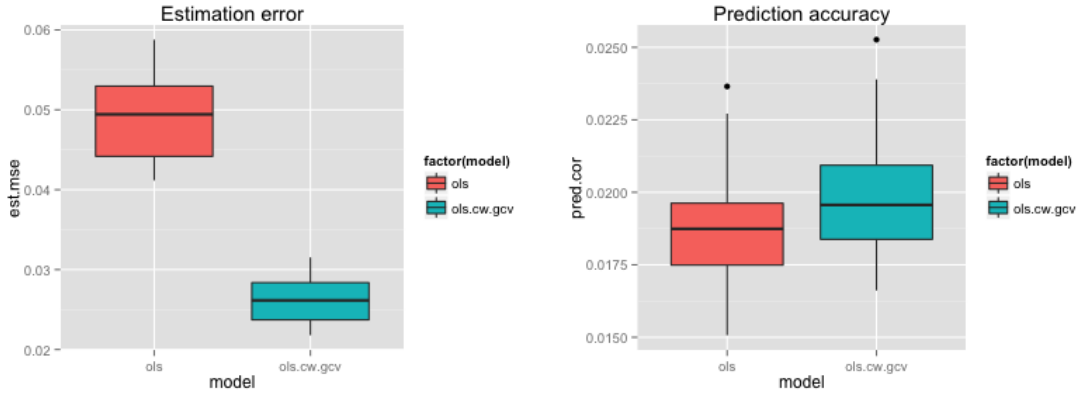
4.2 Search-Light Curds-Whey

Instead of applying CCA to all voxels and our two predictors, we embedded CCA in a searchlight procedure. For each voxel, we took the 25 nearest voxels by euclidean distance. CCA was then calculated on these voxels and the predictors to get a shrinkage matrix. This matrix was applied to the predictions of voxel in the center of the searchlight, and the procedure was repeated for all voxels. Below is an example of the searchlight applied to the simulation.

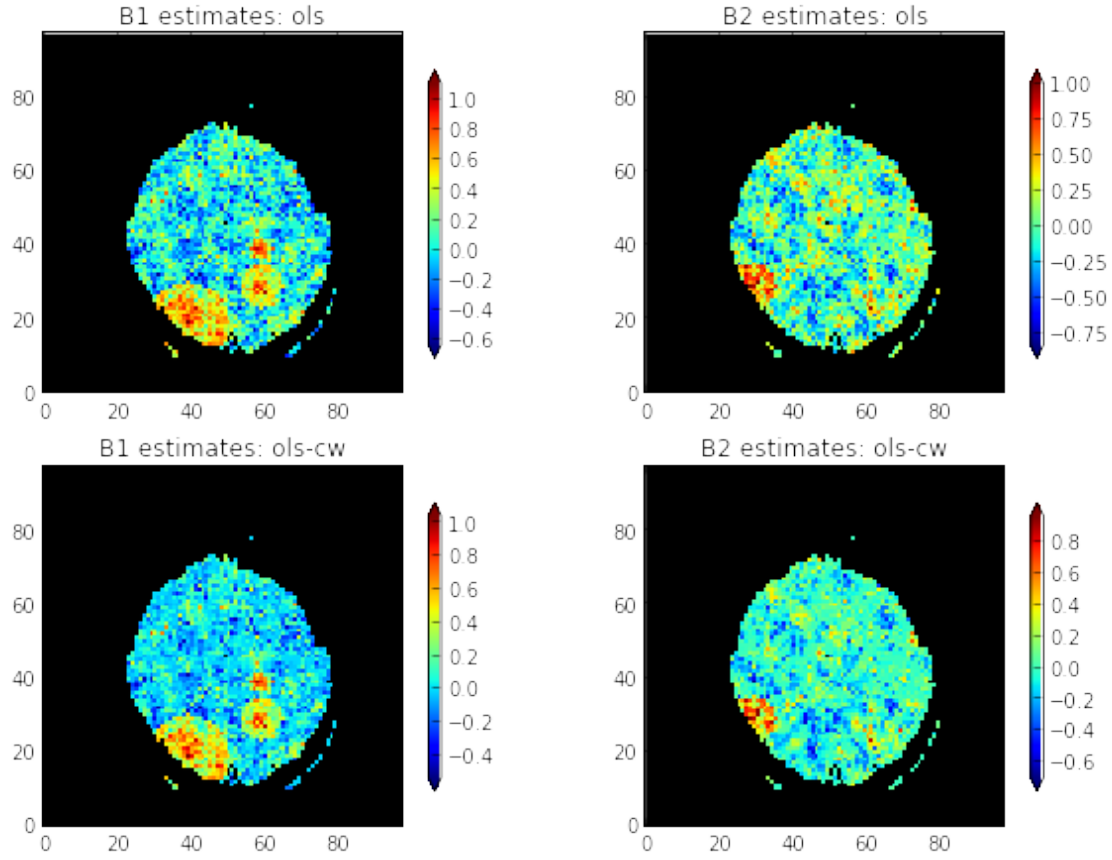


4.3 Results

In this simulation, we only looked at OLS and OLS-CW. Assessed by the same metrics as above, CW showed improvements over OLS.



Looking at maps of the B estimates (below), it is apparent that OLS-CW gave lower B weights to noise voxels that had B=0 (upper upper sections of the brain maps).



5 Part III: fMRI Dataset

5.1 Summary

Face perception is essential to human social interactions on a daily basis. It enables us to identify a virtually unlimited number of people, interpret a wide range of facial emotional cues, and draw rapid trait inferences from facial appearance. These abilities are made possible by a network of brain regions that seem to be largely dedicated to face perception, including the fusiform face area (FFA), occipital face area (OFA), and posterior superior temporal sulcus (pSTS).

While it is well understood that these brain regions activate in response to face images, it is less clear what aspects of face processing they each perform. Our approach is to measure brain activity in response to a large set of naturalistic faces, then use L2-regularized regression to see if we can predict brain activity as a function of different features of the images. Figure 1 shows how this works.

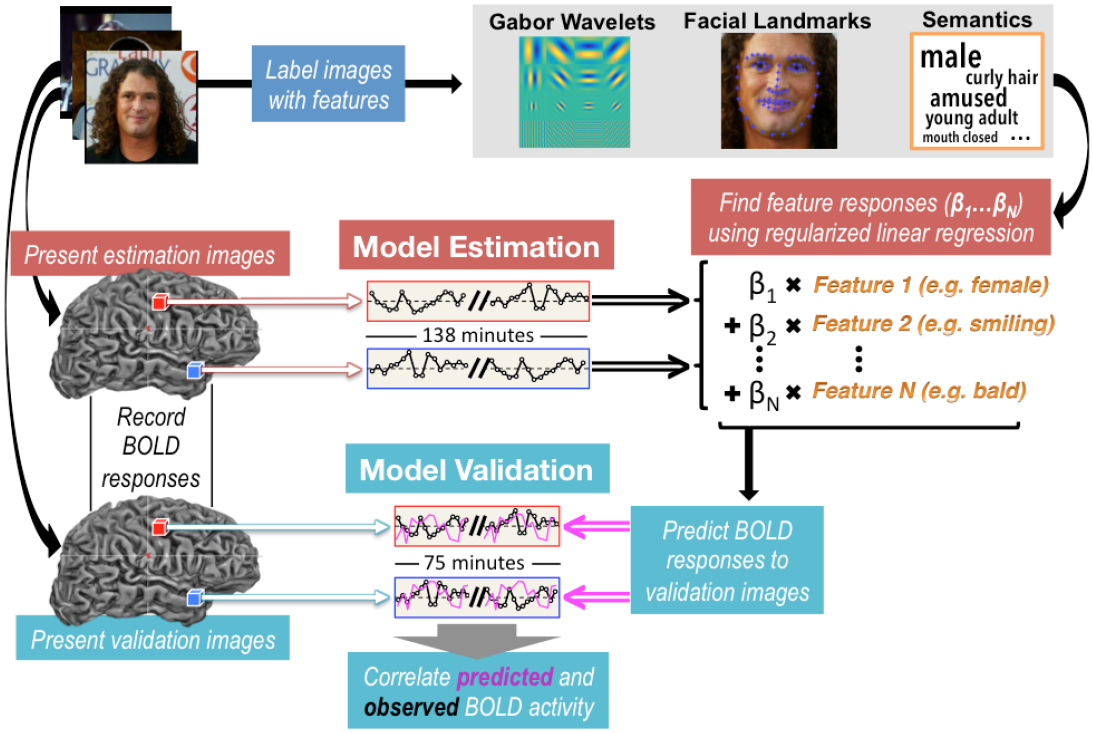
In general, we can compare the ability of different feature models to predict voxels in different regions across cortex. Here, for simplicity, we use only a semantic model. Membership of each face image in 102 semantic categories was independently determined by five naive raters and then averaged. 53 terms described variant facial attributes including 10 aspects of facial posture (e.g. mouth open, squinting) and 43 emotional expressions. The other 49 terms described invariant facial attributes such as gender and race.

The model was fit separately to the 1.92 hours of estimation data collected within each individual voxel. To model the slow hemodynamic response, variables within each model were assigned distinct time-

inseparable finite impulse response filters with four bins at delays 2-4 s, 4-6 s, 6-8 s, and 8-10 s after stimulus onset. All model parameters were simultaneously fit using L2-regularized linear regression. The regularization parameter (λ) for regression was selected with nine-fold cross-validation. Each voxel's prediction scores were taken as the correlation coefficient (Pearson's r) between the actual and predicted BOLD responses for that voxel. The optimal λ for all voxels was determined by testing ten values and selecting the one for which the maximum number of voxels had prediction scores exceeding a threshold r value (0.06). The model-fitting procedures were performed with in-house software written in Matlab (MathWorks). After fitting each voxel-wise encoding model, the proportion of variance explained in each voxel was estimated by predicting the BOLD responses to the validation set and correlating these predictions with the data.

5.2 Methods

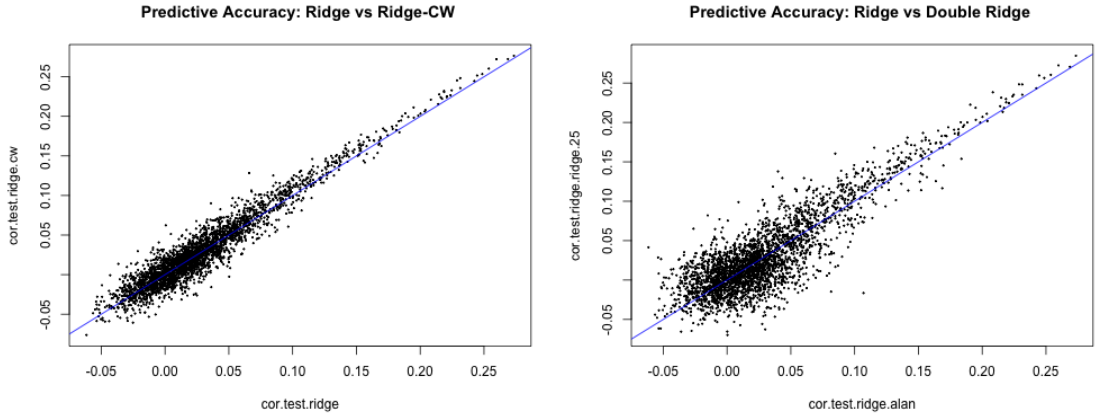
Our goal was to use multi-output regression to improve the predictions of the semantic model. To that end, we tested two different formulations of multi-output regression as described above: searchlight Ridge-Curds-and-Whey and searchlight Double-Ridge. For each voxel, we only used the 25 nearest neighbors in either Ridge-CW or the second step of Double-Ridge. For both methods, the out-of-sample response estimates obtained during cross-validation to select λ in the estimation set were taken as $\hat{Y}$. For double-ridge, we used an additional 9-fold cross-validation within the estimation set to choose γ and then retrained the second ridge step across all of the estimation data to estimate $\hat{S}$ for this γ. Finally, we adjusted our predictions of Y in the validation set with $S * \hat{Y}$ and correlated these adjusted predictions with our measured BOLD responses (Y). These correlations were compared to the correlations between our original $\hat{Y}$ and Y.



5.3 Results

These figures show a scatter plot of the voxel-wise prediction correlations when using searchlight Ridge-CW compared to Ridge (left), and searchlight Double-Ridge compared to Ridge (right). Points lying above the line correspond to voxels where there was an improvement using the new multi-response methods. In both

cases, multi-response methods clearly improved correlations for the majority of relevant voxels, albeit by a small margin.



Below, we show the each voxel's prediction correlation on an inflated representation of the subject's cortex for Ridge (left) vs Double-Ridge (right). Outlined in green, is a region where noticeable improvement occurred.