Predictive Shrinkage in High Dimensional Regression

Runjing Liu & Jake A. Soloff

May 10, 2018

Abstract

In ordinary least squares (OLS), the magnitude of predictions on average are too large for test data, and Stein shrinkage can improve prediction mean squared error. We study the relationship between regularized least squares and predictive shrinkage to understand how the latter might be beneficial in high-dimensional data analysis. We then study predicting multiple correlated responses from a large number of explanatory variables.

Contents

1 Background

1.1 Predictive Shrinkage

Suppose we are given an $n \times p$ centered data matrix X and an $n \times 1$ vector of responses Y—the pair (X, Y) constitutes the training data. Assume the linear model $Y_i = \alpha + (\beta^*)^\mathsf{T} X_i + \varepsilon_i$, where $\varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2)$. Define the ordinary least squares (OLS) estimate

$$(\widehat{\alpha}, \widehat{\beta}) := \arg\min_{\alpha,\beta} \left\{ \|Y - \mathbf{1}\alpha - X\beta\|_2^2 \right\}, \tag{1}$$

$$\text{given by } \widehat{\alpha} = \overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i \text{ and } \widehat{\beta} = \left(X^\mathsf{T}X\right)^{-1} X^\mathsf{T}(Y - \overline{Y}\mathbf{1}). \tag{2}$$

If we have test data $x \sim (0, \Sigma)$ and $y = \alpha + x^\mathsf{T}\beta^* + \varepsilon$, the OLS prediction is $\widehat{y} = \widehat{\alpha} + \widehat{\beta}^\mathsf{T} x$. The bias-variance decomposition of prediction mean square error (PMSE) is

$$\mathbb{E}\left[(y - \widehat{y})^2\right] = \left(1 + \frac{1}{n}\right)\sigma^2 + \mathbb{E}\left[(\beta^* - \widehat{\beta})^\mathsf{T}\Sigma(\beta^* - \widehat{\beta})\right] \tag{3}$$

$$= \left(1 + \frac{p+1}{n} + \frac{1}{n}\mathrm{tr}\left(\mathbb{E}\left[(\Sigma - S)\,S^{-1}\right]\right)\right)\sigma^2, \tag{4}$$

where $S = n^{-1}X^\mathsf{T}X$. If the design of the training set is fixed (so S is constant) and we assume the test set follows the distribution of the training set in the sense that $\Sigma = S$, then the last term vanishes and so the PMSE is $\left(1 + \frac{1}{n} + \frac{p}{n}\right)\sigma^2$. Alternatively, if the rows X_i of X are all i.i.d. $\mathcal{N}(0, \Sigma)$, then nS is a Wishart matrix, and so $\mathbb{E}\left[(nS)^{-1}\right] = \frac{\Sigma^{-1}}{\nu}$ where $\nu = n - p - 1$, yielding a larger overall PMSE of $\left(1 + \frac{1}{n} + \frac{p}{\nu}\right)\sigma^2$.

Under the first set of assumptions, where $\Sigma = S$ exactly, we can write the last term in equation (3) as $\mathbb{E}\left[(\beta^* - \widehat{\beta})^\mathsf{T}\Sigma(\beta^* - \widehat{\beta})\right] = \mathbb{E}\left[\|\widehat{\xi} - \xi\|_2^2\right]$, where $\widehat{\xi} = \Sigma^{1/2}\widehat{\beta} \sim \mathcal{N}(\xi^*, (\sigma^2/n)I_p)$ and $\xi^* = \Sigma^{1/2}\beta^*$. This is a normal-means estimation problem, so when $p > 2$ we can achieve lower MSE $\mathbb{E}\left[\|\widetilde{\xi} - \xi^*\|_2^2\right] < \mathbb{E}\left[\|\widehat{\xi} - \xi^*\|_2^2\right]$ with the James-Stein estimate

$$\widetilde{\xi} = \left(1 - \frac{(p-2)(\widehat{\sigma}^2/n)\nu}{(\nu+2)\|\widehat{\xi}\|_2^2}\right)\widehat{\xi}, \tag{5}$$

yielding the shrunk regression coefficients $\widetilde{\beta} = \widehat{K}\widehat{\beta}$, where $\widehat{K} = \left(1 - \frac{(p-2)(\widehat{\sigma}^2/n)\nu}{(\nu+2)\widehat{\beta}^\mathsf{T}S\widehat{\beta}}\right)$. It follows that $\widetilde{y} = \widehat{\alpha} + \widehat{K}\widehat{\beta}$ has strictly better PMSE than OLS $\widehat{y}$. Since $\widehat{K} < 1$ we are left to conclude that the OLS predictions on held-out data were too large in magnitude. Pre-shrunk predictors of this form were first studied by Copas (1983), who also provided the Stein-shrinkage interpretation.

Another form of shrinkage is *ridge regression*

$$(\widehat{\alpha}, \widehat{\beta}(\lambda)) := \arg\min_{\alpha,\beta} \left\{ \|Y - \mathbf{1}\alpha - X\beta\|_2^2 + \lambda\|\beta\|_2^2 \right\}, \tag{6}$$

$$\text{given by } \widehat{\alpha} = \overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i \text{ and } \widehat{\beta}(\lambda) = \left(X^\mathsf{T}X + \lambda I_p\right)^{-1} X^\mathsf{T}(Y - \overline{Y}\mathbf{1}). \tag{7}$$

In particular, $\widehat{\beta}(0) = \widehat{\beta}$ is OLS and $\lim_{\lambda\uparrow\infty}\widehat{\beta}(\lambda) = 0$. Note $\widehat{\beta}(\lambda) = \left(X^\mathsf{T}X + \lambda I_p\right)^{-1}(X^\mathsf{T}X)\widehat{\beta}$ shrinks $\widehat{\beta}$ towards zero in a manner that accounts for the variability in the covariates X. For $\lambda > 0$ it is not the case that $\widehat{\beta}(\lambda) = K\widehat{\beta}$ for some K, which is to say the family of estimators $\widetilde{\beta}(K, \lambda) = K\widehat{\beta}(\lambda)$ is not overparametrized. We allow any $K > 0$, since for λ large, $\widehat{\beta}(\lambda)$ tends to overshrink. The estimator minimizes the following loss functions:

$$\widetilde{\beta}(K, \lambda) := \arg\min_{\beta} \left\{ \|K(Y - \mathbf{1}\overline{Y}) - X\beta\|_2^2 + \lambda\|\beta\|_2^2 \right\} \tag{8}$$

$$= \arg\min_{\beta} \left\{ \|Y - \mathbf{1}\overline{Y} - X\beta\|_2^2 + \frac{\lambda}{K}\|\beta\|_2^2 + (K^{-1} - 1)\|X\beta\|_2^2 \right\}. \tag{9}$$

The first minimization performs ridge regression on the centered design X and shrunk (or inflated) centered responses $K(Y - \mathbf{1}\overline{Y})$. The second is like ridge with an additional penalty. If $K < 1$ then $(K^{-1} - 1)\|X\beta\|_2^2$ penalizes large $X\beta$, and if $K > 1$ it penalizes small $X\beta$.

When $\lambda > 0$, the ridge regression estimate $\widehat{\beta}(\lambda)$ is biased, but has the advantage of being well-defined even in the high dimensional case $n < p$. Whether we can benefit from the additional pre-factor K depends on how reliably we can detect whether we need to inflate or shrink. In the case where β^* is known to be sparse, we can ask the same set of questions about the magnitude of our predictions after LASSO, where the regularization term in (6) is replaced with $\lambda\|\beta\|_1$. We start with simulation studies to get a sense of in what cases we can significantly improve predictions by including K.

1.2 Motivation: fMRI data

We examine fMRI data from Kay et al (2008). In their experiments, a subject is presented with an image, and fMRI signals in the primary visual cortex (V1) are measured. Our focus will be restricted to voxels 1-20, and we wish to predict the responses of these twenty voxels to each image. Features from each image were extracted using a Gabor wavelet pyramid, resulting in $p = 10921$ features for each image, and there were a total of $n = 1705$ images.

Neurons in this region have a localized receptive field, and cortical columns in V1 have been well modeled using Gabor transformations. Hence it is reasonable to assume the fMRI response in a certain voxel is (roughly) a linear function of a small number of Gabor transforms of the image.

Therefore, we chose to fit a LASSO model to predict voxel responses for this particular application. Furthermore, we investigate whether or not choosing a pre-factor K can improve our LASSO predictions. In section 2 we fit independent LASSOs to each voxel, and investigate the benefits of additionally choosing K.

However, we also have knowledge that adjacent voxels are likely to be strongly correlated, so instead of fitting 20 independent prediction models, we would like to take advantage of these correlations to improve predictive accuracy. We explore ways to extend the curds-and-whey method proposed by Breiman & Friedman (1997) to this high-dimensional, sparse case.

2 Single-Response Prediction in High Dimensions

We explore the performance of choosing a pre-factor K in addition to choosing the traditional regularization parameter λ in LASSO or ridge regression. We propose several methods for selecting K, and run a suite of simulation studies to compare their performance. We then apply these methods to predicting voxel responses on the fMRI dataset. In this section, we restrict our attention to the prediction of a single-dimensional response. For the fMRI data, we treat the predictions on each voxel as an independent regression problem; in the sequel, we will consider multi-response approaches to predictive shrinkage.

2.1 Choosing K

Recall the ridge regression objective given in (6), and let $\hat{\beta}_\lambda^{ridge}$ be the solution for some λ. On new test data (x_{new}, y_{new}), we can predict y_{new} with $\hat{y}_{ridge} = x_{new}^T \hat{\beta}_\lambda^{ridge}$ for some chosen λ.

Analogously, the LASSO procedure solves,

$$\hat{\beta}_\lambda^{lasso} = \arg\min_\beta \|y - X\beta\|^2 + \lambda\|\beta\|_1 \tag{10}$$

and we can predict y_{new} with $\hat{y}_{lasso} = x_{new}^T \hat{\beta}_\lambda^{lasso}$.

In both LASSO and ridge regression, λ must be chosen by some method such as minimizing an information criterion (AIC, BIC, AICc) or by cross-validation. Inspired by Copas (1983), we ask whether the predictive accuracy of ridge (or LASSO) can be improved by choosing some factor K in addition to the selection of λ, and predict y_{new} with $Kx_{new}^T \hat{\beta}_\lambda^{ridge}$ (or $Kx_{new}^T \hat{\beta}_\lambda^{lasso}$).

We compare three methods for choosing a factor K:

1. We select λ and K *jointly* using cross-validation. That is, we search through a grid of proposed combinations (λ, K), and predict y with $K\hat{\beta}_\lambda^T x$, choosing the combination that minimizes the cross-validation MSPE.

2. We select λ and K in *two-stages*. We split the data into V folds like in cross-validation. We run ridge (or LASSO) and select lambda, with cross-validation, using only data in $V-1$ folds. Let $\tilde{\beta}_\lambda$ be the estimate from fitting the data on everything but the vth fold. We can predict the y's on the held-out fold using $x^T\tilde{\beta}_\lambda$. We then regress the y in the held out fold on our predictions $x^T\tilde{\beta}_\lambda$ to get an estimate for K, $\tilde{K}$. We do this procedure V times, once each for fold, and average the estimates $\tilde{K}$ to get a final estimate for K.

3. When $n > p$, we also compare against the procedure proposed by Copas (1983). Here, we run OLS, minimizing the squared error without any penalty term (i.e. $\lambda = 0$), and set K to be

$$K_{copas} = 1 - \frac{p-2}{pF} \tag{11}$$

where $F = n\hat{\beta}_{OLS}^T V \hat{\beta}_{OLS}/(p\hat{\sigma}^2)$, $\hat{\sigma}^2 = \frac{1}{n-p}\sum_{i=1}^n (y-\hat{y})^2$. Our prediction is $\hat{y} = \hat{K}_{copas}\hat{\beta}_{OLS}^T x$. When $p > n$, OLS is not well-defined, so we cannot run the procedure proposed by Copas exactly. However, we can first run LASSO, choosing λ using cross-validation, for the purpose of feature selection. We can then run OLS with the Copas procedure, using only the selected features.

2.2 Ridge Regression Simulation Study

We use a simulation study to examine the predictive performance of the three methods above. We compare the benefits (if any) of selecting K, against running ridge regression and OLS. Here, ridge regression refers to optimizing the objective (6) to solve $\hat{\beta}_\lambda^{ridge}$, and choosing λ with ten-fold cross-validation.

2.2.1 Simulation Setup

For the ridge regression simulation study, we remain in the low-dimensional setting, where $p < n$, so we follow the simulation setup of in Copas (1983). We pick a matrix $X_{train} \in \mathbb{R}^{n\times p}$, with centered columns, and a vector $\beta \in \mathbb{R}^p$, to be fixed for the remainder of this analysis. (At the beginning of the analysis, each entry of X_{train} was drawn iid $\mathcal{N}(0, 10)$ and each entry of β was drawn iid $\mathcal{N}(0, 1)$).

We then draw $y_{train} \sim \mathcal{N}(X_{train}\beta, I_{n\times n}\sigma^2)$. New data is then drawn by $x_{new} \sim \mathcal{N}(0, S)$, where $S = \frac{1}{n}\sum_{i=1}^m X_{train}^T X_{train}$, the empirical covariance of the x's in our training sample; y_{new} is drawn from $\mathcal{N}(\beta^T x_{new}, \sigma^2)$.

We fix the number of training samples to be $n = 1000$ and the dimension of our problem $p = 20$. We then draw $N = 10000$ new observations (x_{new}, y_{new}), and examine the MSPE of our predictors $\hat{y}$,

$$\text{MSPE}(y_{new}, \hat{y}) = \frac{1}{N}\sum_{i=1}^N (y_{new,i} - \hat{y}_i)^2 \tag{12}$$

We also compute the slope of y_{new} on $\hat{y}$, given by

$$\text{slope}(y_{new}, \hat{y}) = \frac{(y_{new} - \bar{y}_{new})^T(\hat{y} - \bar{\hat{y}})}{\|\hat{y} - \bar{\hat{y}}\|_2^2} \tag{13}$$

Note that slopes greater than 1 imply that our predictions are too small, and inflating our predictions may be helpful; conversely, a slope less than 1 implies that our predictions are too large, and shrinkage is necessary.

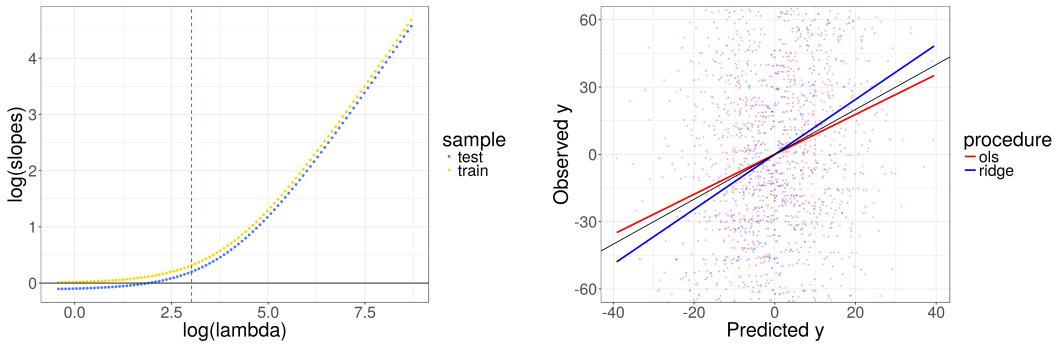
We run this trial 500 times, each time drawing new training responses y_{train} and a subsequent test sample X_{new}, y_{new}. For each trial, we compute $\text{MSPE}(y_{new}, \hat{y})$ and $\text{slope}(y_{new}, \hat{y})$, and we examine the distribution of these quantities for the five prediction methods: OLS, OLS with Copas K, ridge regression, joint (λ, K) selection, and two-step (λ, K) selection.

2.2.2 Simulation Results

Whether or not our ridge regression predictor $\hat{y} = x^T \hat{\beta}_\lambda^{ridge}$ needs to be shrunk or inflated certainly depends on our choice of λ. In the extreme cases, if $\lambda = 0$, then our predictors are equivalent to the OLS predictor, and Copas (1983) showed that the predictions are generally too large so shrinkage (choosing some $K < 1$) improves MSPE; conversely, as $\lambda \uparrow \infty$, we predict identically 0. Hence, if our choice of λ is too large, we conclude that we must rather *inflate* our predictors (choose some $K > 1$).

This is demonstrated in figure 1a, where we show slope$(y_{new}, \hat{y}_{ridge})$ as a function of λ. As discussed, when λ is small, we are in the OLS regime, and our slope is less than 1, i.e. we need to further shrink our predictions; when λ is large, the slope is greater than 1, and hence we need to inflate our predictions. On this particular training set, cross-validation chose a λ that returned a slope greater than 1, and ridge regression over-shrunk in this case.

Figure 1b displays the scatterplot of 10000 observed datapoints y_{new} against the predicted $\hat{y}$, for $\hat{y}$ computed by OLS, and for $\hat{y}$ computed by ridge regression. We see that the slope is less than 1 for OLS; conversely, the slope is greater than 1 for ridge regression, with its particular value of λ chosen by cross-validation.



(a) (b)

Figure 1: (a) Log(slope) of y_{new} on $\hat{y} = x_{new}^T \hat{\beta}_\lambda^{ridge}$ as a function of log(λ). Vertical red line is the λ chosen by cross-validation. Yellow are slopes in the training sample, blue are slopes in the test sample. (b) Scatterplot of $(y_{new}, \hat{y})$ in the validation sample. Black line is the 45 degree line. Red and blue lines are the regression lines of y on $\hat{y}_{OLS}$ and y on $\hat{y}_{ridge}$, respectively. The particular λ used to compute $\hat{y}_{ridge}$ is the same as the λ displayed in (a).

We now repeat this experiment many times, each time drawing a set of 1000 training observations (x_{train}, y_{train}), 10000 subsequent draws of (x_{new}, y_{new}), and examine slope$(y_{new}, \hat{y})$, as we did above. We wonder if this over-shrinkage by running ridge regression is a common occurrence, and if so, whether or not choosing a pre-factor K with choosing λ can fix this problem.

In figure 2, we display the histogram of the slopes over 500 trials. We see that for most trials, when $\hat{y}$ is computed with OLS, slope$(y_{new}, \hat{y})$ is less than 1, as predicted by Copas (1983), and shrinkage is clearly needed. The distribution of slope$(y_{new}, \hat{y})$ when $\hat{y}$ is the ridge regression predictor seem reasonably centered at 1, though there is a noticeable right tail of large slopes.

We then examine the performance of choosing a pre-factor K. In figure 3a, we compare the distribution of slopes when choosing K to the slope distribution of OLS and ridge regression. We see that choosing this factor K in addition to choosing λ indeed lowers the slope, but by a bit too much. For all the methods of choosing K, the median slopes are again below 1, though they are closer to 1 than the OLS slopes. The median slope of the Copas procedure is very close to 1, which is unsurprising given that the simulation setup is the same as in Copas (1983); however, the median slope of our two-step procedure is also not far from one.

We then compare the distribution of test set MSPEs in figure 3b, and we find that K selection does not seem to yield significant improvements. The median MSPE for selecting K jointly was slightly larger than the median MSPE for ridge regression, while the median MSPE was slightly smaller for the Copas and the two-step procedure. The median MSPEs for all the types of shrinkage considered here, ridge, Copas, joint or two-step K selection, was smaller than simple OLS, however.

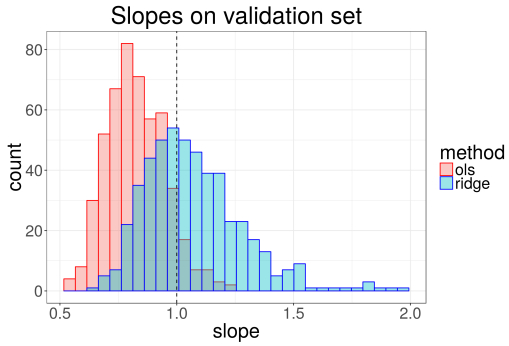


Figure 2: Histogram of slope$(y_{new}, \hat{y})$ over 500 trials. Red are slopes when $\hat{y}$ is obtained by OLS. Blue are slopes when $\hat{y}$ are obtained by ridge regression and cross-validation selected λ.

In sum, both ridge regression and our addition of K improved upon OLS, but the selection of K in the two-step procedure gave only slight improvements over ridge regression.

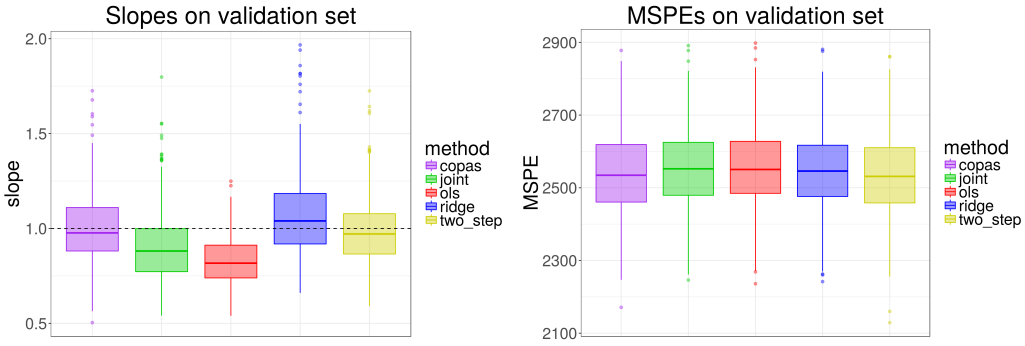


Figure 3: (a) Distribution of slope$(y_{new}, \hat{y})$ for $\hat{y}$ computed with OLS, OLS with Copas K, ridge regression, joint (λ, K) selection, and two-step (λ, K) selection; (b) Distribution of MSPEs in the test set, relative to OLS. In this plot we set $\sigma = 50$

Varying the noise:

Next, we consider what happens as we vary the noise in our simulations. The experiments above had σ set at 50; below, we rerun the experiment above (each experiment again with 500 trials) for several different σs. In figure 4a, we plot the median slope$(y_{new}, \hat{y})$ for the 500 trials against σ. We see that the slope for OLS starts near 1 when σ is small, and decays rapidly. This makes sense: as our data become more noisy, OLS overfits to the noise in the training set, and hence its performance on the test set suffers. Ridge regression, with its regularization term, avoids this overfitting problem, as its median slopes are consistently above 1 for all σ – while it does consistently *over-shrink*, the performance does not seem to become worse with σ, like for OLS. We hoped that choosing K would fix the over-shrinking problem, but choosing K seemed to overcompensate; the slopes become consistently less than 1. However, it is encouraging that again, the slopes do not seem to become as bad as OLS with larger σ.

Finally, figure 4b plots the MSPE relative to OLS. While OLS does better for smaller σ, regularization is needed for noisy problems, and doing ridge regression and/or choosing K improved on MSPE. It appears that the two-step procedure did the best among all the methods, for all but the noisiest case.

2.3 LASSO Simulation Study

Like in the ridge regression analysis, we use a simulation study to examine the predictive performance of choosing K in addition to λ when running LASSO.

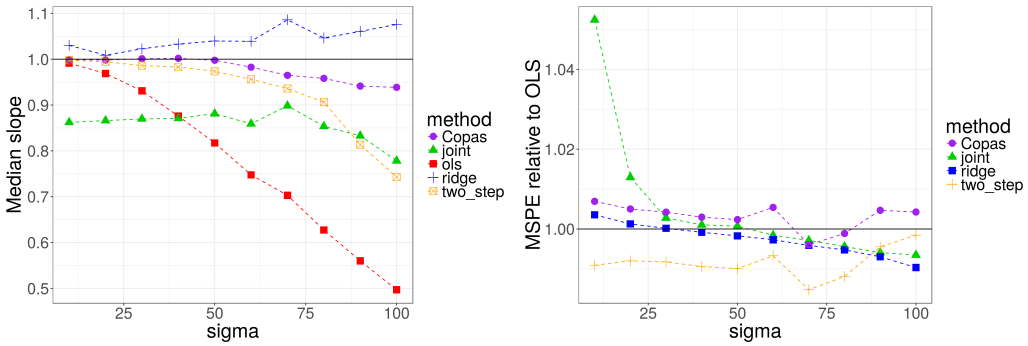


Figure 4: (a) Median slope$(y_{new}, \hat{y})$ over 500 trials, as a function of the noise σ, (b) median MSPEs on the test set, relative to OLS, as a function of the noise σ.

2.3.1 Simulation Setup

The set-up is similar to Copas (1983), except the number of features p is larger than the number of observations in our training set n, and we enforce the true β to be sparse. We pick a matrix $X_{train} \in \mathbb{R}^{n \times p}$, and $\beta_{full} \in \mathbb{R}^p$, to be fixed for the remainder of this analysis. Let s be the sparsity parameter, and let β_s be β_{full} but with its first $p - s$ entries set to 0. We then generate $y_{train} \sim \mathcal{N}(X_{train}\beta_s, \sigma^2 I_{n \times n})$. We set the size of our training set n to 200, and set p to 1000. We subsequently draw a test set, whose features x_{new} are drawn from a zero mean multivariate normal distribution with covariance given by the empirical covariance of X_{train}; y_{new} are then drawn $\mathcal{N}(x_{new}\beta_s, \sigma^2)$. We set the size of our test set to be 1000. We again examine slope$(y_{new}, \hat{y})$ and MSPE$(y_{new}, \hat{y})$ across many trials of this data generation procedure.

2.3.2 Simulation Results

Using similar reasoning as in the ridge regression case, whether our LASSO regression predictor $\hat{y} = x^T \hat{\beta}_\lambda^{lasso}$ needs to be shrunk or inflated depends on our choice of λ. In figure 5a, where we show slope$(y_{new}, \hat{y})$ as a function of λ. When λ is small, we are in the OLS regime, and our slope is less than 1 as expected, and shrinkage is needed; when λ is large, the slope is greater than 1, and inflation is needed. For this particular experiment, cross-validation chose a λ that returned a slope greater than 1, and inflation is needed for better estimates of y_{new}.

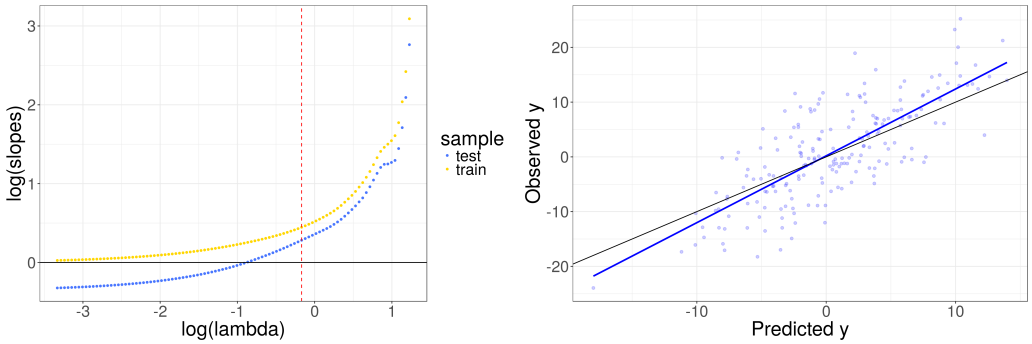


Figure 5: (a) Log(slope) of y_{new} on $\hat{y} = x_{new}^T \hat{\beta}_\lambda^{lasso}$ as a function of $\log(\lambda)$. Vertical red line is the λ chosen by cross-validation. Yellow are slopes in the training sample, blue is slope in the test sample. (b) Scatterplot of $(y_{new}, \hat{y})$ in the test sample, where $\hat{y}$ computed with the LASSO objective with the same λ chosen by cross-validation in (a), and the blue line is the regression line. Black line is the 45 degree line.

Varying noise

We investigate if this is a consistent occurrence by repeating this experiment 500 times. Moreover, we see if varying the noise parameter σ has any effect. In total, we run this experiment 500 times for each value of σ, and our results are displayed in figure 6. Figure 6a shows the median slope across 500 trials for varying σ. We see that choosing λ and K either jointly or in two steps

results in smaller slope, for all σ. However, for small σ, choosing K results in a slope less than 1, and we have over-compensated. Figure 6b displays the relative test set MSPE of choosing λ and K over the MSPE to choosing just λ. It does not seem that choosing K in addition to λ improves MSPE, and it does worse when σ is small.

In all cases, running LASSO for feature selection and then applying the Copas procedure to the selected features gave the worst results. This perhaps is unsurprising given that LASSO was unable to recover the true non-zero coefficients, so the Copas procedure cannot work well.

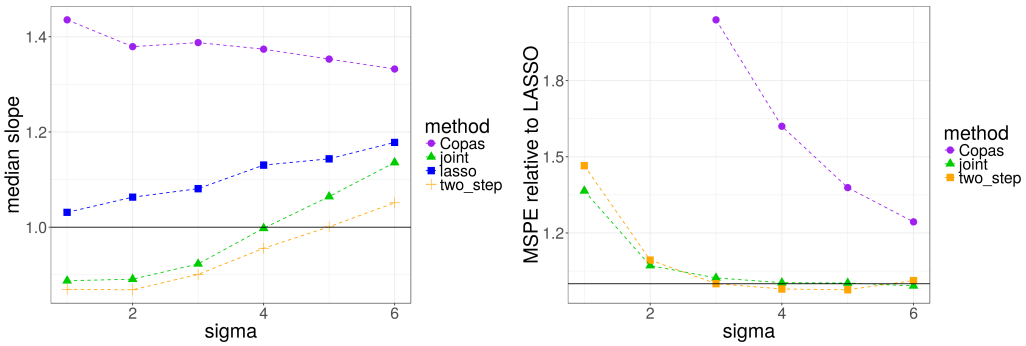


Figure 6: (a) Median slope$(y_{new}, \hat{y})$ over 500 trials, as a function of the noise σ. (b) The median relative test set MSPEs to the traditional LASSO procedure, as a function of the noise σ.

Varying sparsity

Next, we examine the performance of our procedure relative to the traditional LASSO procedure as the sparsity varies. In figure 7a, we again compare the slopes; in figure 7b, we compare the MSPEs. It appears that selecting K either jointly or in two stages with λ does not yield significant improvements for the sparsity levels we examined. Running LASSO for feature selection and then applying the Copas procedure again did the worst, and its performance is particularly bad when the number of non-zero coefficients gets larger.

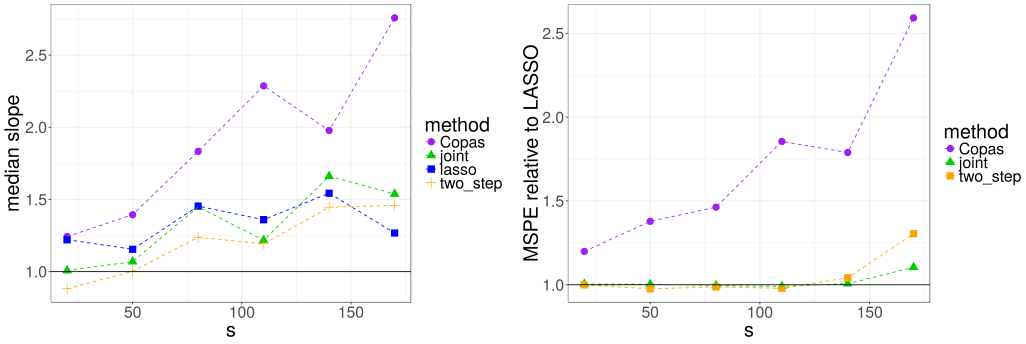


Figure 7: (a) Median slope$(y_{new}, \hat{y})$ over 500 trials, as a function of number of nonzero coefficients s. (b) The median relative MSPEs of our method to the traditional LASSO procedure, as a function of the noise σ.

2.4 Single-response analysis of fMRI data

For each of the twenty voxels, we predict its response to the image a subject was viewing at the time. Features were extracted from an image using a Gabor wavelet tranformation, resulting in $p = 10921$ features. We used $n = 1225$ observations in our training set, and held out $N = 525$ observations in our test set. We predict the response first using LASSO, where λ was selected using cross-validation on the training set. We compare this performance with using LASSO but where we select K jointly with λ (again using cross-validation on the training set).

The results are shown in figure 8, which display the slopes and MSPEs on the test set. We see that for 13 of the 20 voxels, choosing K jointly with λ resulted in slopes closer to 1; there are some

voxels, like voxel 16, where our method gave a slope further from one. MSPEs for both methods in all the voxels however, were rather similar.

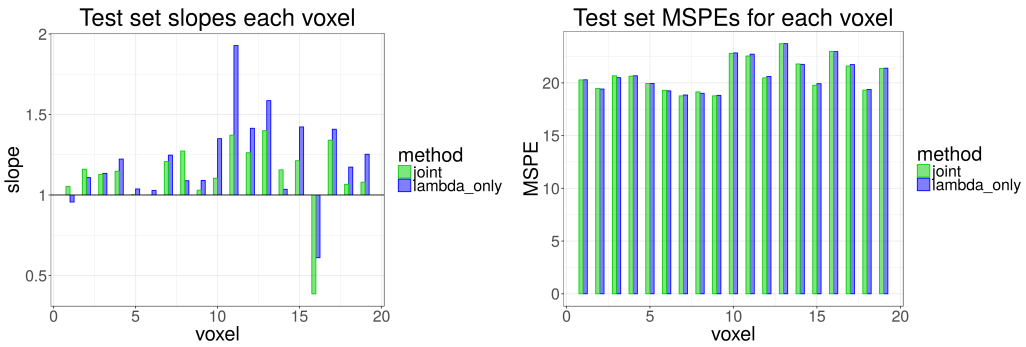


Figure 8: (a) The slope slope$(y_{new}, \hat{y})$ each of the twenty voxels. Blue are slopes when predicting using LASSO with only λ selection, while green are slopes when predicting using our method of jointly selecting K and λ. (b) The MSPEs on the test data, again comparing the two methods. (voxel 20 was ignored in this plot, as both LASSO and and our method returned poor fits relative to the other voxels.)

As a particular example, in figure 9, we look at the scatterplot of y_{new} and $\hat{y}$ for voxel 15. We can visually inspect that LASSO overshrinks the predictors (blue line), and choosing K ameliorates this over-shrinkage by re-inflating this prediction (green line).

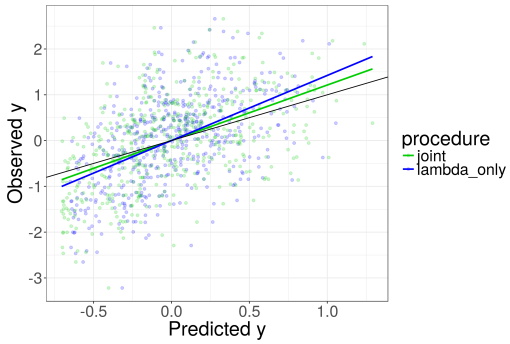


Figure 9: Scatterplot of $(y_{new}, \hat{y})$ in the test sample of voxel 15. Blue points are $\hat{y}$ computed with the LASSO objective, only choosing λ. Green points are $\hat{y}$ computed with LASSO with K selection joint with λ. Black line is the 45 degree line.

We therefore conclude that for this particular data application, LASSO over-shrinks the predictors, and it appears that choosing K jointly with λ in running LASSO can help pull the slopes back towards one.

For this particular analysis, we have treated each voxel as a separate regression problem. However, we know that the responses of these voxels are correlated, and in the next section, we will consider shrinkage for multi-response regression in a way that takes advantage of the relationship between these 20 voxels.

3 Multi-Response Prediction in High Dimensions

Consider the problem of predicting the fMRI response to natural images for $q = 20$ different voxels in the primary visual cortex (V1). Moreover, adjacent voxels will be strongly correlated, and we would like to take advantage of these correlations to improve predictive accuracy. We begin by reviewing a method for exploiting correlations in multi-response linear regression where $p < n$.

3.1 Curds & Whey & Stein

As before, X is an $n \times p$ centered data matrix. Now we have q different responses, $Y = (y^{(k)})_{k=1}^q$, each in $\mathbb{R}^n$, so Y is an $n \times q$ matrix. Each response is linearly related to the covariates, so

$$Y = XA + E, \tag{14}$$

where A is a $p \times q$ matrix of pq regression coefficients, and E is an $n \times q$ matrix of independent draws from $\mathcal{N}(0, \sigma^2)$. The OLS estimate is $\widehat{A} = (X^\mathsf{T}X)^{-1}X^\mathsf{T}Y$. One way to assess its accuracy is via total squared prediction error (PTSE) on a new sample pair $x \sim \mathcal{N}(0, V)$ and $y = xA + \varepsilon$:

$$\mathbb{E}\left\|y - x\widehat{A}\right\|_2^2 = q\sigma^2 + \sum_{k=1}^q \mathbb{E}\left[(A_k - \widehat{A}_k)^\mathsf{T} V (A_k - \widehat{A}_k)\right]. \tag{15}$$

The previous display is the multi-response analogue of the bias-variance decomposition in (3) (note we omit an intercept), indicating that we could do better on each problem (and hence in PTSE) by applying Copas (1983) shrinkage on each problem separately. This amounts to shrinking $\widehat{A}$ by a $q \times q$ diagonal matrix:

$$\widetilde{A} = \widehat{D}\widehat{A}, \text{ where } \widehat{D}_{kk} = \left(1 - \frac{(p-2)(\widehat{\sigma}^2/n)\nu}{(\nu+2)\widehat{A}_k^\mathsf{T} S \widehat{A}_k}\right), \tag{16}$$

where $\widehat{\sigma}^2$ is a pooled estimate of variance. Perhaps we can do even better in terms of PTSE by using a *non-diagonal* shrinking $q \times q$ matrix $\widehat{B}$ and taking $\breve{A} = \widehat{B}\widehat{A}$. To leverage Stein-shrinkage with a shrinking matrix $\widehat{B}$, we require an extension of the James-Stein estimator, presented below:

Lemma 1. *(Efron & Morris, 1972) Suppose $\widehat{\Xi} = (\widehat{\Xi}_1 \cdots \widehat{\Xi}_p)^\mathsf{T}$ is a $p \times q$ matrix with $q + 1 < p$ and $\widehat{\Xi}_j \overset{ind}{\sim} \mathcal{N}(\Xi_j, I_q)$. Define $\mathcal{S} = \widehat{\Xi}^\mathsf{T}\widehat{\Xi}$, $\nu = p - q - 1$, and*

$$\breve{\Xi}_j := \left(I - \nu\mathcal{S}^{-1}\right)\widehat{\Xi}_j. \tag{17}$$

Then **for all** Ξ *the shrunk matrix $\breve{\Xi}$ dominates the MLE $\widehat{\Xi}$ in the Frobenius norm:*

$$\mathbb{E}\|\breve{\Xi} - \Xi\|_F^2 < \mathbb{E}\|\widehat{\Xi} - \Xi\|_F^2. \tag{18}$$

Note that $V^{1/2}\widehat{A}_k \sim \mathcal{N}(V^{1/2}A_k, (\sigma^2/n)I_p)$, and since each error term E_{ik} is assumed independent above, $V^{1/2}\widehat{A} \sim \mathcal{N}(V^{1/2}A, (\sigma^2/n))I_{p\times q})$. To apply the lemma, let $\widehat{\Xi} = \sqrt{n/\sigma^2}V^{1/2}\widehat{A}$ and similarly for Ξ, so the estimation error term in equation (15) can be written as $\mathbb{E}\|\widehat{\Xi} - \Xi\|_F^2$. Blithely assuming $V = n^{-1}X^\mathsf{T}X$ as in Section 1.1, we improve upon PTSE with

$$\breve{A} := \left(I - \nu V^{-1}\left((n/\sigma^2)\widehat{A}^\mathsf{T}V\widehat{A}\right)^{-1}\right)\widehat{A} \tag{19}$$

$$= \left(I - \sigma^2 rn(X^\mathsf{T}X)^{-1}\widehat{Q}^{-1}(Y^\mathsf{T}Y)^{-1}\right)\widehat{A} \tag{20}$$

$$\approx \left\{(1-r)I + r\widehat{Q}^{-\mathsf{T}}\right\}^{-1}\widehat{A} \tag{21}$$

where $\widehat{Q} := (Y^\mathsf{T}Y)^{-1}Y^\mathsf{T}X(X^\mathsf{T}X)^{-1}X^\mathsf{T}Y$ is the CCA matrix and $r = \frac{\nu}{n} = \frac{p-q-1}{n}$. The last step follows from plugging in a residual estimate of $\sigma^2 I \approx (Y - X\widehat{A})^\mathsf{T}(Y - X\widehat{A})/(n-p)$ and applying Woodbury's formula. This gives rise to the *Curds & Whey* method of Breiman & Friedman (1997):

1. (Run CCA) Factor $\widehat{Q} = \widehat{T}^{\mathsf{T}}\widehat{C}^2\widehat{T}^{-1}$, where $\widehat{C}^2$ is diagonal and $\widehat{T}$ forms a basis.
2. Set $\widehat{B} = \left\{(1-r)I + r\widehat{Q}^{-\mathsf{T}}\right\}^{-1} = \widehat{T}^{-1}\widehat{D}\widehat{T}$, where $\widehat{D}_{kk} = \frac{\widehat{C}^2_{kk}}{\widehat{C}^2_{kk}+r(1-\widehat{C}^2_{kk})}$. Define the Curds & Whey coefficients $\breve{A} = \widehat{B}\widehat{A}$. Predict $\breve{y} = x\breve{A}$ on new data.

As indicated in Figure 10 below, this closed-form method for Curds & Whey is in some cases worse than OLS, since this method uses the entire sample to estimate canonical correlations and hence is prone to overfitting. We can instead use cross-validation to select the diagonal matrix in Curds & Whey, which turns out to be the solution to a QP:

$$\widehat{D} := \text{pmax}(\text{diag}(M^{-1}u), 0) \text{ where } u = \sum_n (y_n^{\mathsf{T}}\widehat{T}_{\setminus n}^{-1}) \circ r_n, M = \sum_n \left(\widehat{T}_{\setminus n}^{-\mathsf{T}}\widehat{T}_{\setminus n}^{-1}\right) \circ r_n r_n^{\mathsf{T}} \text{ and } r_n = \widehat{T}_{\setminus n}\widehat{y}_n^{\setminus n},$$

where $\setminus n$ means the quantity is estimated with the n^{th} datapoint excluded. This works best in the top-right facet of the figure, which approaches the high dimensional case. This method is also the most extensible as it relies the least on distributional assumptions. Still, it's interesting to note how applying shrinkage à la Copas (1983) separately on each of the q problems performs comparably to Curds & Whey, as the latter is estimating the best linear predictor of y on $\widehat{y}$ and the former does not share strength across problems (beyond the pooled estimate of the error variance). Neither method improves much in relative TSE when q increases.

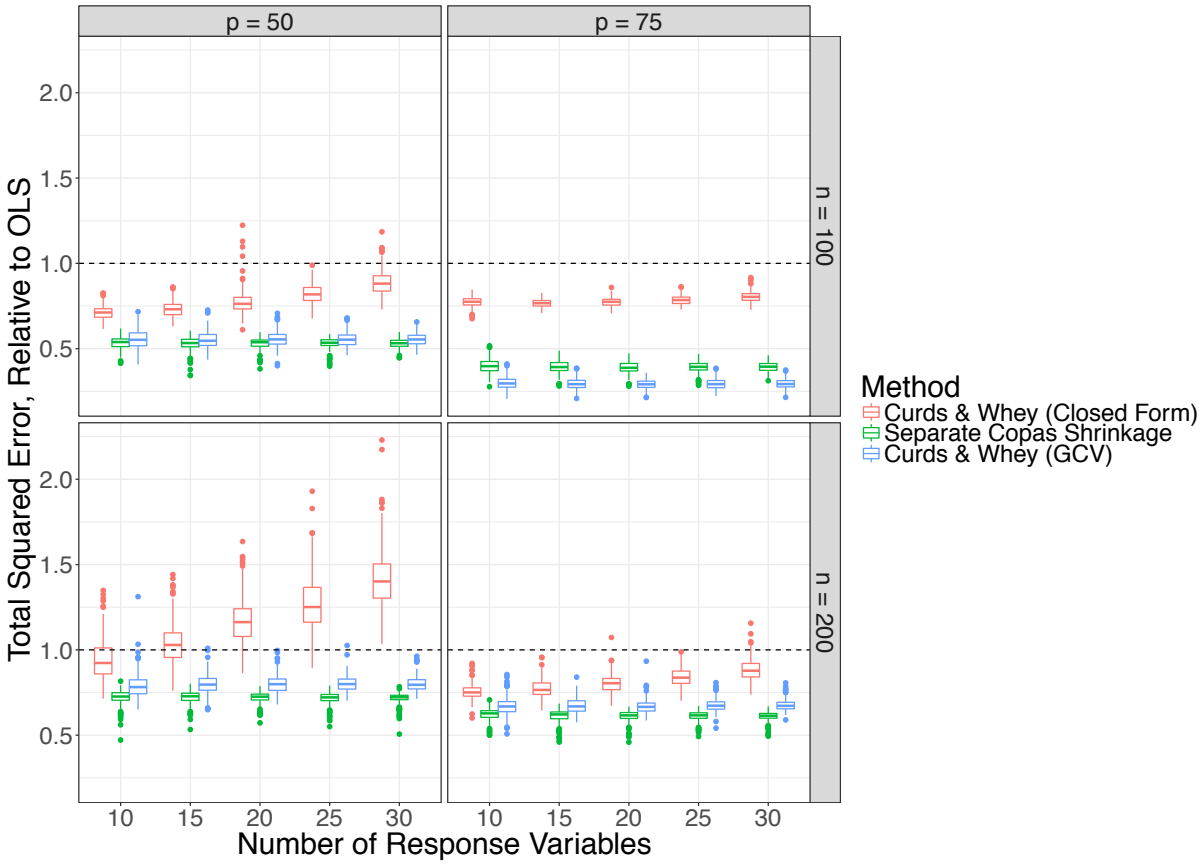


Figure 10: Simulation-based comparison of multi-response shrinkage methods in moderate dimensions p relative to the sample size n. The simulation design follows Breiman & Friedman (1997), with fixed, unit signal-to-noise ratio. The horizontal axis is the number of responses q that we are modeling, and the vertical axis is the ratio $\frac{\sum_k (A_k - \widetilde{A}_k)^{\mathsf{T}} V (A_k - \widetilde{A}_k)}{\sum_k (A_k - \widehat{A}_k)^{\mathsf{T}} V (A_k - \widehat{A}_k)}$ of PTSE (modulo constants) for each method relative to OLS. Each box plot was created using 250 runs.

3.2 Extensions to Sparse Modeling

When the number of predictors p exceeds the sample size n, we face two major problems. First, as before, OLS is not well-defined, since $X^\mathsf{T}X$ is not invertible; second, CCA is also not well-defined for the same reason. Motivated by our mulitple-voxel fMRI response problem, we will assume A is sparse. The first issue can be overcome by solving LASSO separately on each problem:

$$\widehat{A}_{\text{lasso}} := \arg\min_{A'} \sum_{k=1}^{q} \left[\|y^{(k)} - XA'_k\|_2^2 + \lambda_k \|A'_k\|_1 \right], \tag{22}$$

This will be our baseline predictor, as it does not attempt to exploit any relationship between the q related problems. Each λ_k is selected via cross-validation.

3.2.1 Subset Selection then Curds & Whey

One extreme is where the sparsity patterns of the $(A_k)_{k=1}^q$ are related so that the union of their support is not much larger than the largest support. Then it makes sense to replace the separable regularization term $\sum_{k=1}^q \sum_{j=1}^p \lambda_k |A'_{jk}|$ in equation (22) with $\lambda \sum_{j=1}^p \max_{k=1:q} |A'_{jk}| = \lambda \sum_{j=1}^p \|(A')^j\|_\infty$ to select features jointly across all of the problems. Obozinski et al (2006) relax this complete feature-sharing scenario by replacing the ℓ_∞ norm with ℓ_2:

$$\widehat{A}_{\text{multitask-lasso}} := \arg\min_{A'} \sum_{k=1}^{q} \|y^{(k)} - XA'_k\|_2^2 + \lambda \sum_{j=1}^{p} \|(A')^j\|_2. \tag{23}$$

This method (which can be run in `glmnet` by setting `family = "mgaussian"`) already shares strength across the different problems, but if we are using equation (23) to select a sparse subset $\widehat{S}$ of s features jointly for all q problems, then we can run the Curds & Whey procedure on $X_{\widehat{S}} \in \mathbb{R}^{n \times s}$ and Y. This requires $q+1 < s < n$, so we need to constrain λ when doing CV. We call the resulting estimate $\widehat{A}_{\text{ML-CW}}$ for Multitask-LASSO then Curds & Whey. This method relies heavily on feature selection, which requires a much stronger signal than getting good predictions with LASSO. The benefit is that pooling effectively affords us more observations.

3.2.2 Sparse CCA

Given an $n \times p$ matrix U and an $n \times q$ matrix W with $U^\mathsf{T}U = I_p$ and $W^\mathsf{T}W = I_q$, when $n < p$ one way to make progress is to assume the canonical u-vectors are sparse and impose an ℓ_1 constraint:

$$\min_{\substack{u,w \\ \|u\|_2 \vee \|w\|_2 \le 1 \\ \|u\|_1 \le c\sqrt{p}}} u^\mathsf{T}U^\mathsf{T}Ww, \tag{24}$$

for $c \in [0, 1]$. Note if $c = 1$ then the ℓ_1 constraint is vacuous and this is equivalent to CCA. This assumption that the U-canonical directions are sparse makes sense in the context of our fMRI study, since the fMRI responses only 'care' about a small number of Gabor wavelets. We use the implementation of Witten et al (2009) to obtain a y-canonical basis $\widehat{T}^\mathsf{T}$ and then pool the LASSO predictions with $\widehat{T}^{-1}\widehat{D}\widehat{T}y$ (here $\widehat{D}$ can be obtained with CV as before but can have elements bigger than 1).

3.2.3 Sparse Precision Estimation

The CCA matrix $\widehat{Q} := (Y^\mathsf{T}Y)^{-1}Y^\mathsf{T}X(X^\mathsf{T}X)^{-1}X^\mathsf{T}Y$ can be thought of as estimating $\Sigma_{yy}^{-1}\Sigma_{yx}\Theta\Sigma_{xy}$, where $\Theta = \Sigma_{xx}^{-1}$. In the simulations of Breiman & Friedman (1997), the covariates are drawn from an $\text{AR}_p(1)$ Gaussian graphical model $(\Sigma_{xx})_{ij} = r^{|i-j|}$, which is a Markov model, and hence the inverse covariance or *precision* matrix Θ is tridiagonal (by Hammersley-Clifford). If we knew of this sparse structure a-priori we could bring high-dimensional sparse precision matrix estimation to bear on constructing a plug-in estimate for $\widehat{Q}$. This seems relevant to the Gabor wavelet features of natural images, but maybe not the most direct way to take advantage of structure.

3.2.4 Best Linear Predictor via Cross-Validation

The goal of each of the previous extensions of the Curds & Whey procedure was to take weighted combinations of the q separate LASSO predictions $\widehat{y}$ to share strength across problems. That is, we take $\widetilde{y} = \widehat{B}\widehat{y}$ where $\widehat{B}$ is a linear predictor for y based on $\widehat{y}$. Ideally, we would use the best linear predictor $B^* = \arg\min_B \mathbb{E}\|y - B\widehat{y}\|_2^2$, and we can construct an unbiased estimate of this using cross-validation. The recipe is straightforward:

1. Split the training data into $\mathcal{V} = 5$ equally sized subsets. For each $v = 1 : 5$,
2. Fit q LASSO problems separately using all the data except group v. Let $\widehat{y}(v)$ denote the predictions on the held-out data, and $y(v)$ the true values on that subset.
3. Fit q OLS problems separately. For $k = 1 : q$, regress $y_k(v)$ on $\widehat{y}(v)$. Write the regression coefficients in a $q \times q$ matrix $\widehat{B}(v)$. Take $\widehat{B} = \frac{1}{\mathcal{V}} \sum_{v=1}^{\mathcal{V}} \widehat{B}(v)$.

In small datasets, this will likely be highly variable, which can be mitigated to an extent with the choice of $\mathcal{V}$. We refer to this method as Best Linear Predictor via Cross-Validation (BLP-CV).

3.2.5 Simulations

In Figure 11 below, we compare each of these extensions to LASSO in terms of PTSE. The simulation design again follows Breiman & Friedman (1997); specifically, we use their method for generating a block of s non-zero coefficients in each problem, and then append $p - s$ zero coefficients.

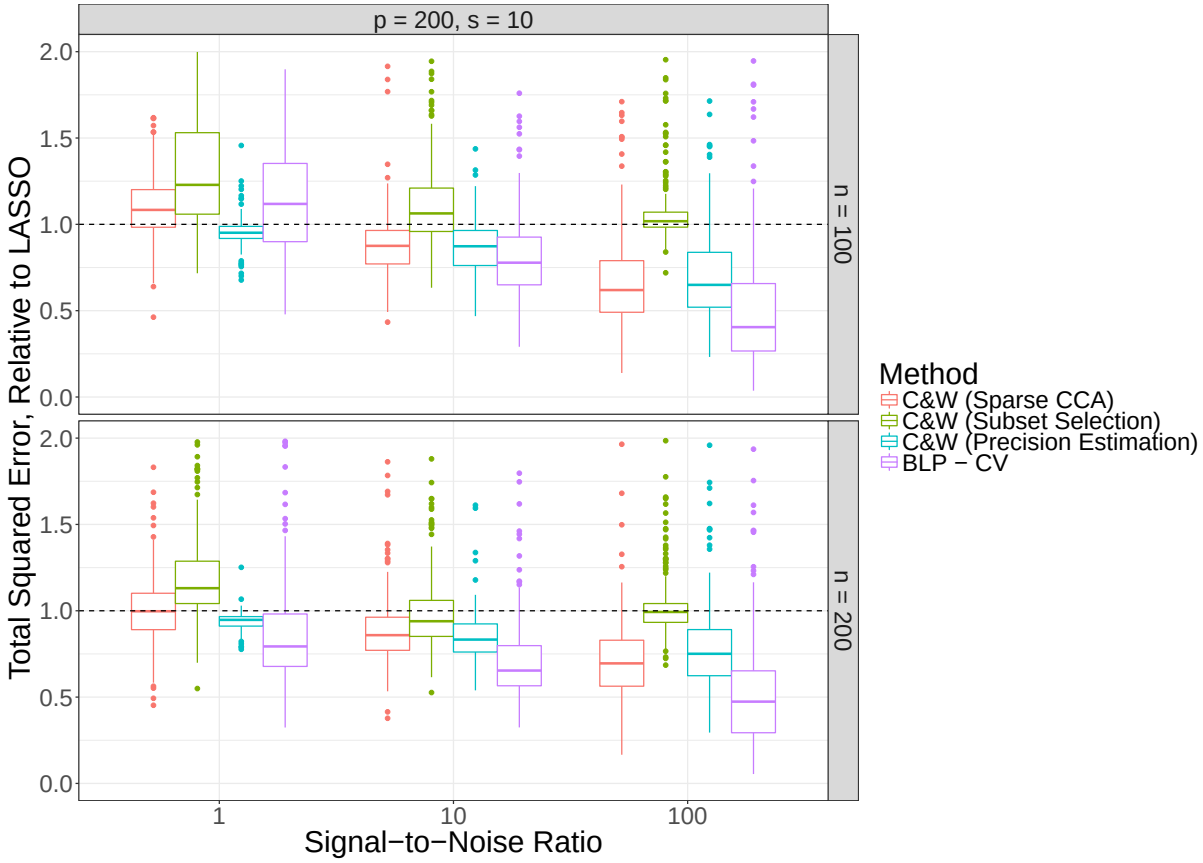


Figure 11: Simulation-based comparison of multi-response ($q = 5$) shrinkage methods in high dimensions p relative to the sample size n. The setup is the same as Figure 10 with additional $p - s$ zeros added. Again, each box plot was created using 250 runs. $< 1\%$ of runs went above 2 on the vertical axis.

Note since each method in Figure 11 returns a $q \times p$ matrix of the form $\widehat{B}\widehat{A}_{\text{lasso}}$, each is strictly slower to compute than LASSO, but the computational cost of estimating $\widehat{B}$ also varies. The Sparse Precision Estimation method is by far the fastest, since it assumes the model $(\Sigma_{xx})_{ij} = r^{|i-j|}$, estimates r via autocorrelation, and then computes the tri-diagonal inverse in closed form. Since this model is correct in our simulations, this yields a fairly reliable plugin estimate of the CCA matrix, even in high dimensions.

The next fastest method is running Curds & Whey on a subset of the covariates. This in part because, rather than running (23), we estimate the support directly from $\widehat{A}_{\text{lasso}}$. As implemented, subset selection does not do consistently better than LASSO, even when the SNR is **obscenely high**. This is surprising in light of Figure 10, since if we knew the support exactly, this method would even beat OLS run on the support (which definitely beats LASSO in prediction error). This result appears to underscore the difficulties of subset selection.

The matrix algebra in computing the ordinary CCA matrix ($q < p < n$) should be $\mathcal{O}(qp^2)$. The cost of this implementation of sparse CCA is unclear. The algorithm runs for a fixed number of iterations (15), and each update has a single matrix-vector product $\mathcal{O}(qs)$ where $s \ll p$ is the sparsity level as well as a grid search. In terms of wall-clock time, the Sparse CCA approach is the third fastest. As in the GCV approach to Curds & Whey, we run Sparse CCA $\mathcal{V} + 1 = 6$ times, and that's the bulk of the computational cost.

The BLP-CV approach is a clear front-runner in terms of relative improvement over LASSO but also computational cost. The algorithm takes roughly $\mathcal{V} + 1 = 6$ times as long to run as LASSO. In the top left of Figure 11 (relatively low signal and sample size), LASSO beats BLP-CV most of the time, so it's not always true that the added computational cost will be worth it.

3.3 Application to fMRI Study

A single subject observed a total of $n = 1750$ images, each of shape 128×128. After pre-processing each image using a Gabor wavelet pyramid, $p = 10921$ covariates remain, and we observe the fMRI response in $q = 20$ voxels. In Figure 12 we display two intermediate layers of the Gabor wavelet pyramid. Most (over 8000) of the features are on the finest level of the pyramid, which are localized to patches a few pixels wide.

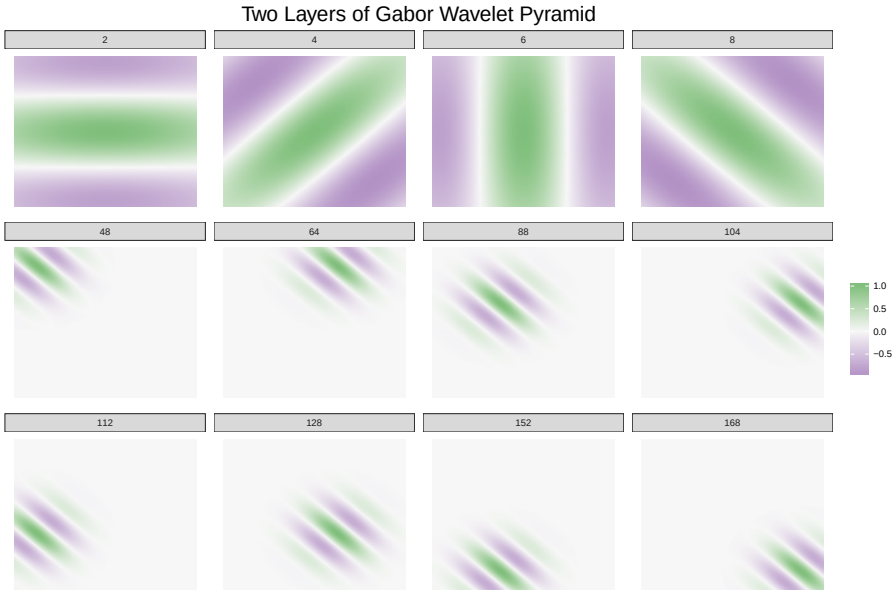


Figure 12: Understanding the features by the transforms they represent. Each feature is a coefficient from the image convolved with a wavelet basis element. Up in the pyramid, the wavelets depend on the entire 128×128 image; toward the bottom wavelets are localized to a small patch.

In Figure 14 below we note that the fMRI responses for different voxels are indeed (positively) correlated, with one block of voxels having much stronger correlations than the others.

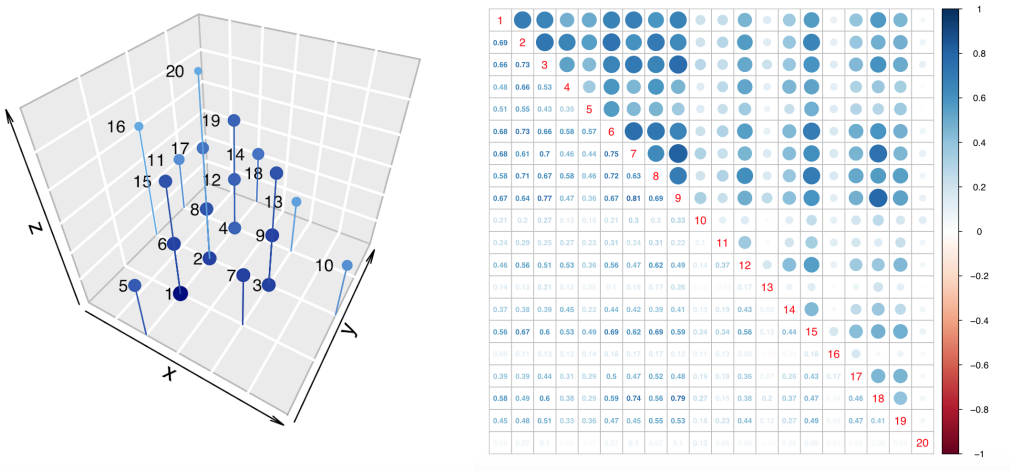


Figure 13: Left: Each voxel in relative spatial coordinates, colored (and sized) proportional to the correlation between that voxel's fMRI responses and those of voxel 1. Right: pairwise correlations between voxels. The first row of this thus gives the intensities for the 3D-plot on the left.

In keeping with Kay et al (2008) who published this data, we use the correlation between the predicted responses $\widehat{y}$ and the observed responses y on test set to measure the performance of our models. In Figure 14, we find that estimating the matrix $\widehat{B}$ via cross validation does not appear to help much by this measure.

Figure 14: Model comparison using the test set. We report the correlation between fitted and observed fMRI response values for all voxels.

4 Conclusion

In the single-response high-dimensional regression case, we explored ways to do shrinkage inspired by Copas (1983) in the context of LASSO and ridge regression. We use cross validation to jointly choose a regularization parameter λ and a shrinking factor K. On the fMRI data this approach had negligible improvement in terms of MSPE. While we argued that this additional parameter cannot hurt *in theory*, we believe the added parameter is an unnecessary degree of freedom after the shrinkage provided by λ.

In the multi-response high-dimensional regression case, we provided a Stein-shrinkage interpretation of the curds & whey method, and found through simulations that curds & whey with cross validation works well in moderate dimensions, but noted that CCA is not well defined when $p > n$. We explored possible extensions of curds & whey to the high dimensional case, and found that the high dimensional precision estimate always outperformed LASSO, although it relied on distributional assumptions for the covariates. By contrast, directly estimating the best linear predictor B^* via CV tended to improve predictions relative to LASSO and does not rely on distributional assumptions. In the fMRI application, the BLP-CV method allowed us to share strength across voxels, as indicated in Figure 14.

5 References

Efron, B., & Morris, C. (1972). Empirical Bayes on vector observations: An extension of Stein's method. *Biometrika*, 59(2):335–347.

Copas, J. B. (1983). Regression, prediction and shrinkage. *Journal of the Royal Statistical Society*, 311–354.

Breiman, L., & Friedman, J. H. (1997) Predicting multivariate responses in multiple linear regression. *Journal of the Royal Statistical Society*, 59(1):3–54.

Obozinski, G., Taskar, B., & Jordan, M. I. (2006) Multi-task feature selection. Technical report, *Department of Statistics, University of California, Berkeley.*

Kay, K., Naselaris, T., Prenger, R. J., & Gallant, J. (2008) Identifying natural images from human brain activity. *Nature*, 452(7185):352–355.

Witten, D. M., Tibshirani, R., & Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534.