1 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.

1.1 Choosing K

Recall the ridge regression objective given in (**??**), 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,

$$\hat{\beta}_\lambda^{lasso} = \arg\min_\beta \|y - X\beta\|^2 + \lambda\|\beta\|_1 \tag{1}$$

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 $K x_{new}^T \hat{\beta}_\lambda^{ridge}$ (or $K x_{new}^T \hat{\beta}_\lambda^{lasso}$).

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, call it $\tilde{K}$. 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

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

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.

1.2 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 (**??**) to solve $\hat{\beta}_\lambda^{ridge}$, and choosing λ with ten-fold cross-validation.

1.2.1 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 $\hat{y}$,

$$\text{MSPE}(y_{new}, \hat{y}) = \frac{1}{N}\sum_{i=1}^{N}(y_{new,i} - \hat{y}_i)^2 \tag{3}$$

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

$$\text{slope}(y_{new}, \hat{y}) = \frac{(y_{new} - \bar{y}_{new})^T(\hat{y} - \bar{\hat{y}})}{\|\hat{y} - \bar{\hat{y}}\|_2^2} \tag{4}$$

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.

1.2.2 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 $K > 1$).

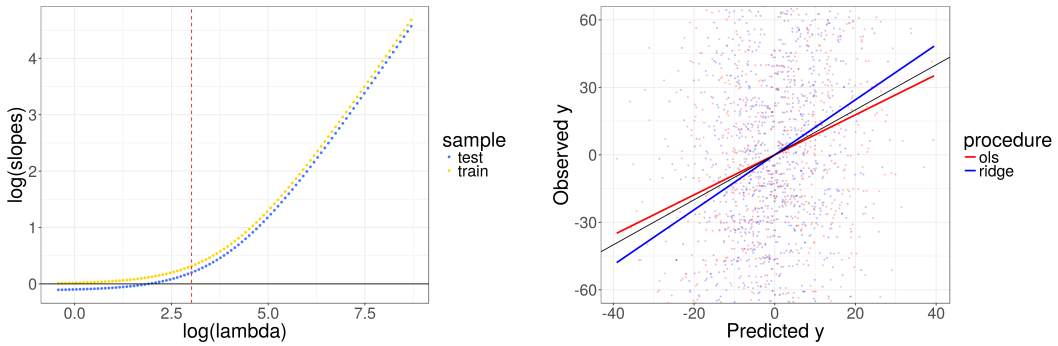
This is demonstrated in figure 1a, where we show $\text{slope}(y_{new}, \hat{y}_{ridge})$ as a function of λ. As discussed, when λ is small, we are in the OLS regime, and our slope is less than 1, i.e. we need to further shrink our predictions; when λ is large, the slope is greater than 1, and hence 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.

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 $\text{slope}(y_{new}, \hat{y})$, 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, $\text{slope}(y_{new}, \hat{y})$ is less than 1, as predicted by Copas (1983), and shrinkage is clearly needed. The distribution of $\text{slope}(y_{new}, \hat{y})$ 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



(a) (b)

Figure 1: (a) Log(slope) of y on $\hat{y} = x^T \hat{\beta}_\lambda^{ridge}$ as a function of $\log(\lambda)$. 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 test sample. Black line is the identity $y = x$ line. Red and blue lines are the regression lines of y_{new} on $\hat{y}_{OLS}$ and y_{new} on $\hat{y}_{ridge}$, respectively. The particular λ used to compute $\hat{y}_{ridge}$ is the same as the λ displayed in (a).

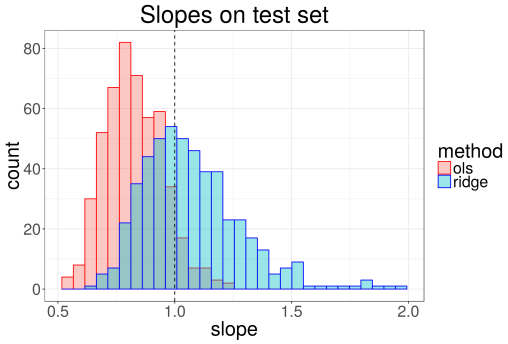


Figure 2: Histogram of slope$(y_{new}, \hat{y})$ over 500 trials. Red are slopes when $\hat{y}$ is obtained by OLS. Blue are slopes when $\hat{y}$ are obtained by ridge regression and cross-validation selected λ.

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. 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.

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.

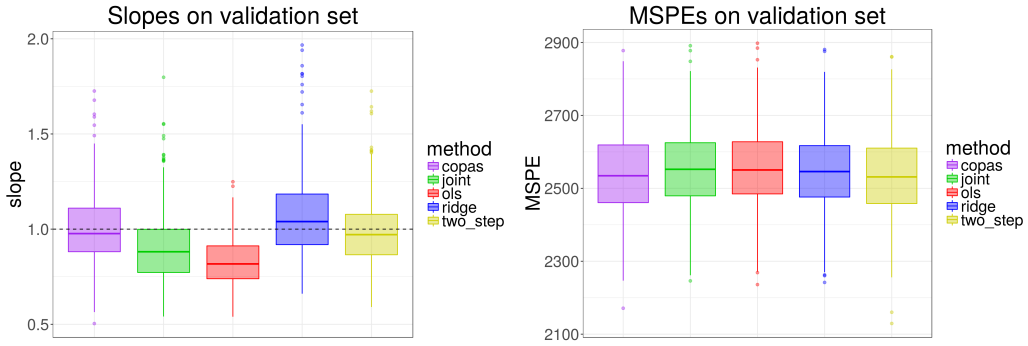


Figure 3: (a) Distribution of slope$(y_{new}, \hat{y})$ for $\hat{y}$ computed with OLS, OLS with Copas K, ridge regression, joint (λ, K) selection, and two-step (λ, K) selection; (b) Distribution of MSPEs in the test set, relative to OLS. In this plot we set $\sigma = 50$

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. The two-step procedure had the smallest median MSPE among all the methods, for all but the noisiest case.

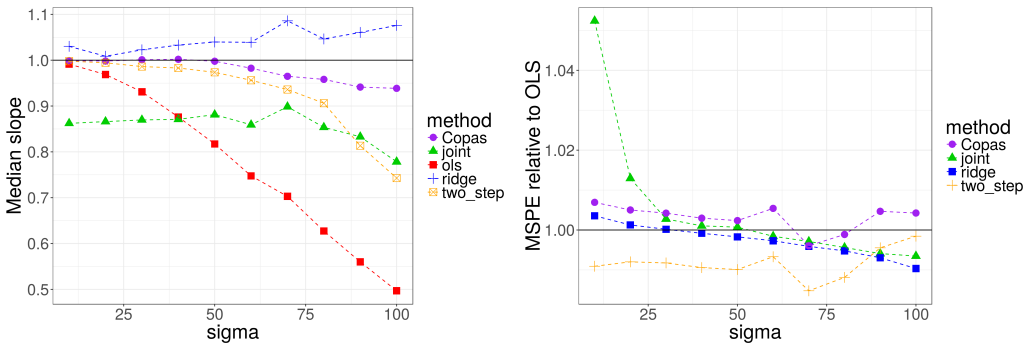


Figure 4: (a) Median slope$(y_{new}, \hat{y})$ over 500 trials, as a function of the noise σ, (b) median MSPEs on the test set, relative to OLS, as a function of the noise σ.

1.3 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.

1.3.1 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.

1.3.2 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 inflating our predictions may provide better estimates of y_{new}.

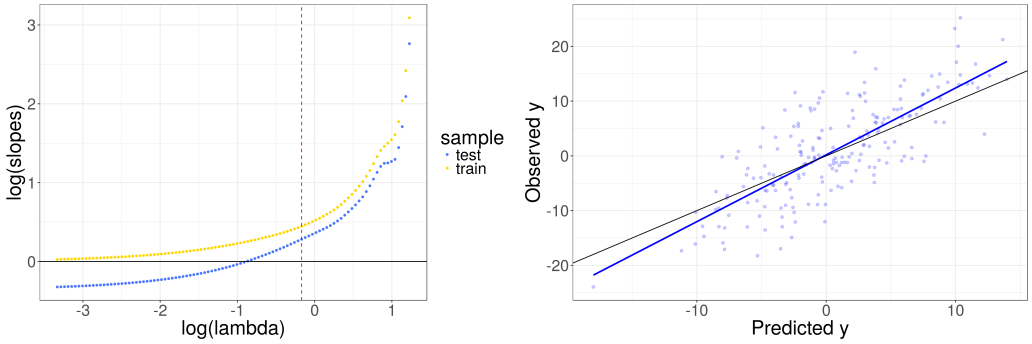


Figure 5: (a) Log(slope) of y on $\hat{y} = x^T \hat{\beta}_{\lambda}^{lasso}$ as a function of log(λ). Vertical red line is the λ chosen by cross-validation. Yellow are slopes in the training sample, blue is slope in the test sample. (b) Scatterplot of $(y_{new}, \hat{y})$ in the test sample, where $\hat{y}$ 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 median slope, for all σ. However, for small σ, choosing K results in a median 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 did not work well.

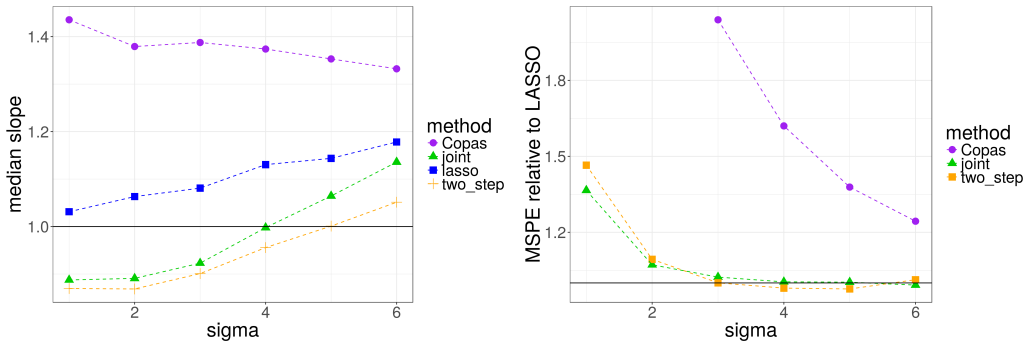


Figure 6: (a) Median slope$(y_{new}, \hat{y})$ over 500 trials, as a function of the noise σ. (b) The median relative test set MSPEs to the traditional LASSO procedure, as a function of the noise σ.

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 in MSPE 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.

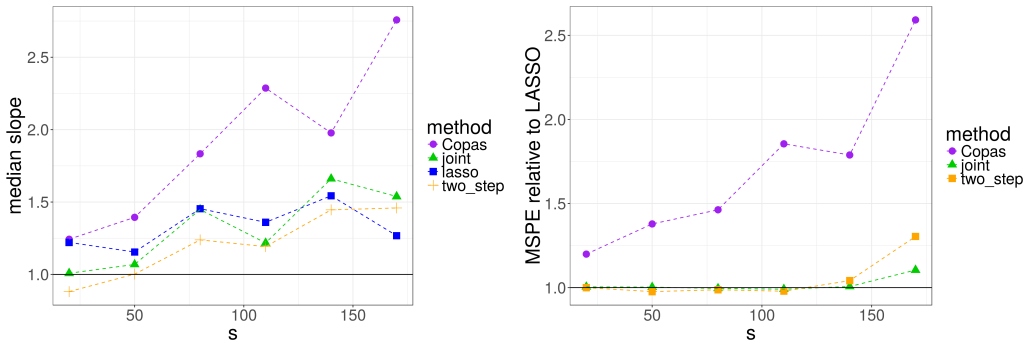


Figure 7: (a) Median slope$(y_{new}, \hat{y})$ over 500 trials, as a function of number of nonzero coefficients s. (b) The median relative MSPEs of our method to the traditional LASSO procedure, as a function of the noise σ.

1.4 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.

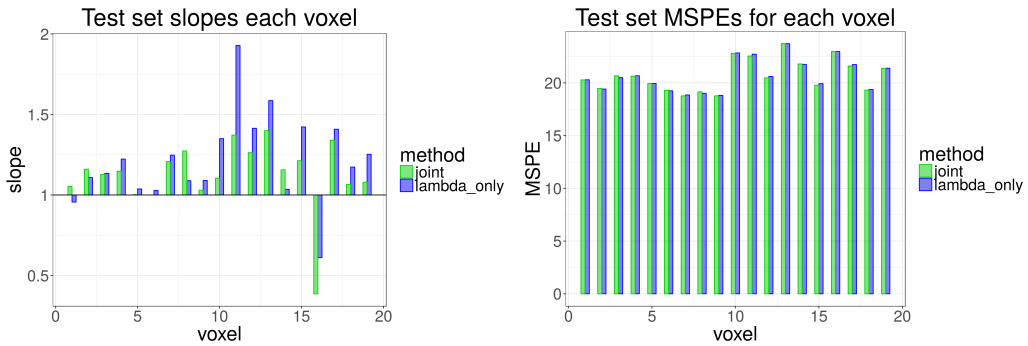


Figure 8: (a) slope$(y_{new}, \hat{y})$ for each of the twenty voxels. Blue are slopes when predicting using LASSO with only λ selection, while green are slopes when predicting using our method of jointly selecting K and λ. (b) The MSPEs on the test data, again comparing the two methods. (voxel 20 was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

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.

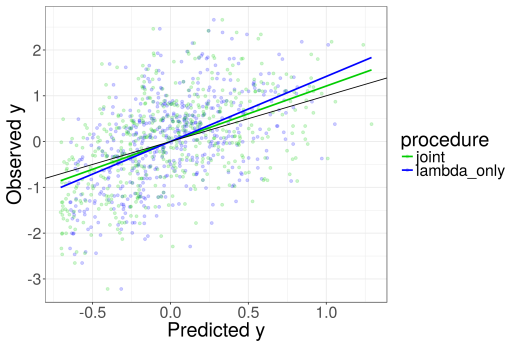


Figure 9: Scatterplot of $(y_{new}, \hat{y})$ in the test sample of voxel 15. Blue points are $\hat{y}$ computed with the LASSO objective, only choosing λ. Green points are $\hat{y}$ computed with LASSO with K selection joint with λ. Black line is the identity $y = x$ line.