

Univariate-Guided Sparse Regression

Sourav Chatterjee^{1,3}

Trevor Hastie^{2,3}

Robert Tibshirani^{2,3}

¹Department of Mathematics

²Department of Biomedical Data Science

³Department of Statistics

Stanford University

Abstract

In this paper, we introduce “UniLasso”— a novel statistical method for regression. This two-stage approach preserves the signs of the univariate coefficients and leverages their magnitude. Both of these properties are attractive for stability and interpretation of the model. Through comprehensive simulations and applications to real-world datasets, we demonstrate that UniLasso outperforms Lasso in various settings, particularly in terms of sparsity and model interpretability. We prove asymptotic support recovery and mean-squared error consistency under a set of conditions different from the well-known irrepresentability conditions for the Lasso. Extensions to generalized linear models (GLMs) and Cox regression are also discussed.

1 Introduction

High-dimensional regression and classification problems are ubiquitous across fields such as genomics, finance, and social sciences. In these settings, the Lasso (Least Absolute Shrinkage and Selection Operator) has emerged as a widely-used methodology due to its ability to simultaneously perform variable selection and regularization, yielding sparse and interpretable models. Despite its widespread use, Lasso has certain limitations, including sensitivity to correlated predictors and inclusion of spurious features in the final model. There have been many related proposals including MC+ (Zhang 2010), the elastic net (Zou & Hastie 2005), the adaptive Lasso (Zou 2006), and SparseNet (Mazumder, Friedman & Hastie 2011).

To address these challenges, we propose UniLasso, a novel regression methodology. In this work, we aim to achieve two primary goals: predictive accuracy and (especially) interpretability. To achieve this, UniLasso integrates marginal (univariate) information into a coherent multivariate framework: our method preserves the signs of the univariate coefficients and leverages their magnitude. This approach not only enhances predictive accuracy but also provides insights into the relative importance of predictors.

Here is an example, taken from unpublished work with collaborators of the third author. (Since the work is unpublished, we do not give background details). The dataset is from cancer proteomics, with 559 biomarkers measured on 81 patients, 20 healthy and 61 with cancer. We divided the data into 70% training and 30% test, and applied both the lasso and uniLasso to the training set. Both methods performed well, giving cross-validation error of about 5%, and 4 and 3 errors respectively out of 30 test samples. Table 1 shows the results of lasso and uniLasso applied to this data, using cross-validation to choose the model. In the top table there are 2 sign changes from univariate to multivariate: #3 goes from mildly protective to a negative indicator, while # 7 goes in the opposite direction.

By design, uniLasso in the bottom table has no sign changes. Further, its univariate coefficients are much larger (in absolute value) than those of the lasso. This makes it a more credible and stable model. We investigate stability in Section 10.1.

The remainder of this paper is organized as follows. Section 2 provides a detailed description of the

UniLasso methodology. We examine the relationship between UniLasso and the Adaptive Lasso in Section 3. We compare UniLasso to least squares and Lasso in the well-known diabetes data set in Section 4: here we show the utility of unregularized uniLasso as an alternative to least squares. In Section 5 we look at the orthonormal design case, where we can derive explicit an expression for the UniLasso solution. Section 6 presents some theoretical results on support recovery and mean square error consistency. Section 7 presents simulation studies comparing UniLasso to Lasso and other benchmark methods, while Section 8.1 examines uniReg, the unregularized ($\lambda = 0$) case. Section 9 applies UniLasso to real-world datasets, and in Section 10 we present an “adversarial attack” view, and multiclass and GWAS examples. Section 11 explores the setting where additional data is available, not full data but just the univariate coefficients. In Section 12 we study whether cross-validation works properly in the UniLasso setting. We discuss UniLasso for GLMs and the Cox Survival Model, as well as computation in Sections 13 and 14. Lastly, Section 15 concludes with a discussion of future directions.

<i>Model chosen by Lasso</i>			
Biomarker	Univariate LS coefficient	Lasso coefficient	Sign change?
1	-12.709	-1.845	
2	25.269	37.492	
3	-0.34	6.245	X
4	-24.74	-11.262	
5	-17.966	-11.908	
6	-49.24	-18.919	
7	1.81	-6.265	X
8	22.909	13.995	
9	-37.646	-5.415	
10	13.024	0.79	
11	-10.141	-2.325	
12	-22.383	-12.745	

<i>Model chosen by uniLasso</i>		
Biomarker	Univariate LS coefficient	uniLasso coefficient
Biomarker	univariate	multivariate
A	58.924	4.982
B	25.269	6.286
C	-49.24	-13.667
D	73.844	38.448
E	69.152	23.034
F	-54.94	-9.261
G	77.688	9.458

Table 1: *Proteomics study. Top table: Univariate least squares coefficients and lasso coefficients, for model chosen by Lasso. Bottom table: same for model chosen by uniLasso. All biomarkers are different in the two tables, with the exception of #2 and #6 in the first table corresponding to B and C in the second.*

2 Univariate-guided lasso

2.1 Our proposal

We assume the standard supervised learning setup: we have training features and target $X_{n \times p}$, $y_{n \times 1}$. For now we assume that y is quantitative and we fit a linear model using squared error loss; later we discuss the binomial and other GLM families, as well as the Cox model.

Our procedure has 3 simple steps, which we motivate here. For interpretability and prediction accuracy, we preprocess the features in Step 1, multiplying them by a robust version of the univariate least-squares coefficient estimates. In Step 2 we fit a **non-negative** lasso. Together these ensure that

- (a) *the signs of the final coefficients agree with the signs of the univariate coefficients (or they are zero);*
- (b) *features with larger univariate coefficients will tend to have larger coefficients in the final model.*

Here is the uniLasso algorithm.

UniLasso algorithm

1. For $j = 1, 2, \dots, p$, fit the p separate univariate least-squares models $\hat{\eta}_j(x_j) = \hat{\beta}_{0j} + \hat{\beta}_j x_j$ to the response y . For $j = 1, 2, \dots, p$ and each $i \in 1, 2, \dots, n$, compute the leave-one-out (LOO) fitted value $\hat{\eta}_j^{-i} = \hat{\beta}_{0j}^{-i} + \hat{\beta}_j^{-i} x_{ij}$, resulting in a new $n \times p$ feature matrix $F = \{\hat{\eta}_j^{-i}\}$.
2. Fit the Lasso — with an intercept, no standardization, and non-negativity constraints — to target y and these LOO fits as features, giving coefficients $\hat{\theta} = (\hat{\theta}_0, \hat{\theta}_1, \dots, \hat{\theta}_p)^\top$:

$$\underset{\theta}{\text{minimize}} \left\{ \frac{1}{n} \sum_{i=1}^n (y_i - \theta_0 - \sum_{j=1}^p \theta_j \hat{\eta}_j^{-i})^2 + \lambda \sum_{j=1}^p |\theta_j| \right\} \quad \text{subject to } \theta_j \geq 0, j = 1, \dots, p. \quad (1)$$

and return the composite model

$$\hat{\eta}(x) = \hat{\theta}_0 + \sum_{j=1}^p \hat{\theta}_j \hat{\eta}_j(x_j). \quad (2)$$

Since each of the constituent models $\hat{\eta}_j(x_j)$ are linear, this final model can be written

$$\hat{\eta}(x) = \hat{\gamma}_0 + \sum_{j=1}^p \hat{\gamma}_j x_j, \quad (3)$$

with $\hat{\gamma}_j = \hat{\theta}_j \hat{\beta}_j$, and $\hat{\gamma}_0 = \hat{\theta}_0 + \sum_{\ell=1}^p \hat{\theta}_\ell \hat{\beta}_{0\ell}$.

This procedure is computationally convenient: Step 1 uses an efficient LOO formula, and then in Step 2, we can apply any efficient ℓ_1 solver. Here we used the R language package `glmnet`. Specifically, we use the function `cv.glmnet` to estimate the Lasso path parameter, and have all of the functionality for `glmnet` at our disposal. We provide a function `cv.UniLasso` that implements this approach.

Note that we *do not* standardize the features before applying the non-negative Lasso in Step 2. From our knowledge of multiple linear regression, the first constraint — agreement between univariate and multivariate signs — may seem like an unreasonable restriction. However our belief is that in high-dimensions there are likely to be a multitude of different models that have about the same MSE as

the “optimal” one chosen by the Lasso. Hence it can make sense to choose one that is interpretable and sparser than that of lasso.

These properties mean, for example, if the feature “age” has a positive univariate coefficient — indicating increasing risk of the outcome variable (such as Alzheimer’s disease), it will have a positive (or zero) coefficient in the final lasso model. And if age is strongly significant on its own, it is more likely to be chosen in the multivariate model.

These conjectures are borne out by our numerical studies. In simulations with varying problems sizes and SNR, and a number of real datasets (see Section 7), in almost every case UniLasso did no worse than the Lasso in terms of out-of-sample MSE, and delivered a substantially sparser model. These real datasets were not “cherry picked” but were the first group of datasets that we tested. We note that Meinshausen (2012) studies sign-constrained least squares estimation for high-dimensional regression, which relates to our non-negativity constraint in our second step.

Remark A. In equation (1), the first term has a multiplier of $1/n$. In other descriptions of the Lasso, the factor may be set to $1/2$; the `glmnet` package uses $1/2n$. The different choices lead to equivalent problems and just differ in the scaling of the path parameter λ . In the theory of Section 6 we assume that the $1/n$ multiplier is used.

Remark B. The UniLasso procedure applies seamlessly to other GLMs as well the Cox model. Indeed, all of the models covered by the `glmnet` package are at our disposal. All we need are the separate fitted linear models and their LOO fit vectors in step 1, and then `glmnet` can be directly applied. The only challenge is find an (approximate) LOO formula for the given family. We give details of these computations in the Section 14.

Remark C. UniLasso can be thought of as a version of Stacked regression, or Stacking (Wolpert 1992, Breiman 1996). Stacking is a two-stage procedure for combining the predictions from a set of models (“learners”), in order to get improved predictions. It works as follows. A set of base models is trained on the training data; these models can be of the same type (e.g. gradient boosting) or different types (e.g., linear regression, decision trees, etc.). Each model generates predictions, capturing specific aspects of the relationship between the predictors and the target variable. A meta-model is then trained to combine the predictions of the base models into a final prediction. This meta-model learns how to weigh and integrate the outputs of the base models to minimize the overall prediction error. UniLasso is a special case of stacking where the individual learners are simple univariate regressions.

Remark D. Why do we use the LOO estimates $\hat{\eta}_j^{-i}$ as features in step 2, instead of the least squares estimates $\hat{\eta}_j^i = \hat{\eta}_j(x_{ij})$? In traditional stacking this is essential because the individual learners can be of very different complexity. In UniLasso, one would think that the learners (univariate regression) all have the same complexity so that the LOO versions are not needed. Indeed, the theory in Section 6 is based on the LOO estimates but holds equally well for the usual (non-LOO) estimates. However in practice we have found that the LOO estimates often lead to better performance and hence we use them. With the use of the usual (non-LOO) univariate estimates in uniLasso, the resulting estimator is closely related to the adaptive lasso and we explore this connection in Section 3.

Remark E. The non-negative garotte (Breiman 1995) is another closely related method. It minimizes $\sum_i (y_i - \sum_j c_j \hat{\beta}_j x_{ij})^2$ subject to $c_j \geq 0$ for all j , and $\sum c_j \leq s$, where $\hat{\beta}_j$ are the usual (multivariable) LS estimates. This is different from our proposal in an important way— our use of univariate LOO estimates in the first step. The univariate coefficients lead to materially different solutions and also allow application to the $p > n$ scenario.

2.2 UniLasso with no-regularization

We get an interesting special case of UniLasso if we set $\lambda = 0$, so that there is no ℓ_1 regularization in Step 2. That is, we solve the following::

$$\text{minimize}_{\theta} \left\{ \frac{1}{n} \sum_{i=1}^n \left(y_i - \theta_0 - \sum_{j=1}^p \theta_j (\hat{\beta}_{0j}^{-i} + \hat{\beta}_j^{-i} x_{ij}) \right)^2 \right\} \quad \text{with } \theta_j \geq 0 \forall j. \quad (4)$$

We call this “UniReg” for univariate-guided regression and it represents an interesting alternative to the usual least squares estimates. For example, the non-negative constraint still can provide sparsity even though there is no ℓ_1 penalty. And it gives a unique solution even if $p > n$. The standard error and distribution of the estimated coefficients can be estimated via the bootstrap. We compare UniReg to least squares in the diabetes example of Section 4, and study it in more detail in Section 8.1.

2.3 Two examples

We generated data with 300 observations and 1000 features, the feature covariance having an $AR(1)$ form with $\rho = 0.8$. The coefficient vector had 100 non-zero values, uniformly distributed between 0.5 and 2.0, positioned at every other feature. Finally the error variance was 225 yielding an SNR of about 1.5.

The top panels of Figure 1 show the CV and test error curves for the Lasso and UniLasso. We can think of this as a “home court” setting for **UniLasso** where the pairwise correlations between features is positive, as are the true coefficients. This connection is explained in the theorems of Section 6. We see that **UniLasso** has test error a little below that of Lasso, with a smaller active set. The bottom panels shows the true positives (green) and false positives (red). UniLasso yields more true positives and fewer false positives.

Table 2 shows the result of 100 simulations from this setup. We see that same pattern emerges. Note that the table shows MSE, rather than PSE as in Figure 1; the two quantities differ in expectation by $\sigma^2 = 225$. Notice that UniLasso has much higher TPR and much lower FPR than Lasso.

	MSE-Lasso	MSE-UniLasso	Support-Lasso	Support-UniLasso
mean	294.17	282.27	72.00	46.00
se	1.46	1.33	1.78	0.82
	TPR-Lasso	FPR-Lasso	TPR-UniLasso	FPR-UniLasso
mean	0.33	0.67	0.49	0.51
se	0.01	0.01	0.01	0.01

Table 2: *Test MSE, support, TPR (True Positive Rate) and FPR (False Positive Rate) for 100 simulations from the setting of Figure 1*

For the one realization from the same setup but taking $n = 300, p = 100$, Figure 2 shows the number of nonzero coefficients for lasso and uniLasso as a function of λ . The right end of the plot represents least squares (for the lasso) and uniReg (for uniLasso). Least squares of course uses all $p = 100$ features for $\lambda = 0$, but remarkably uniReg uses fewer than 20. It seems that the LOO feature scaling in step 1 and non-negative constraints in step 2 provide implicit regularization.

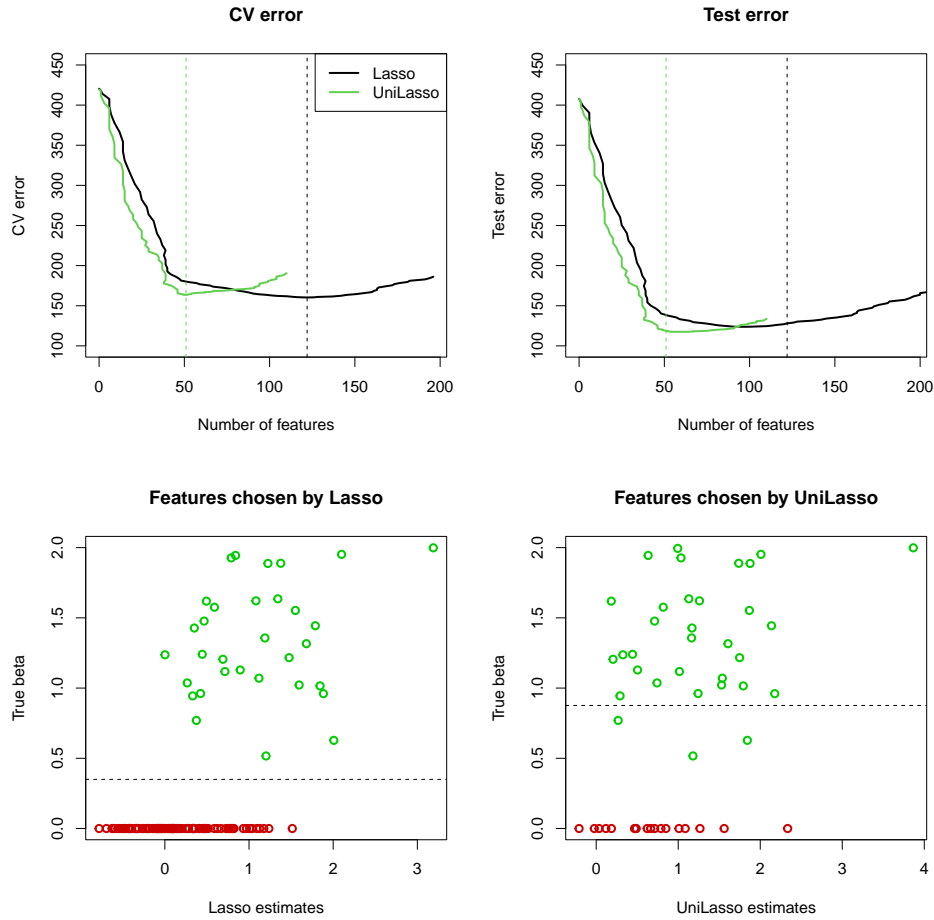


Figure 1: Results for homecourt example. Top panel; CV and Test set prediction error. The dotted vertical lines shows the values of λ chosen by CV for each model. Bottom panels: true positives (green) and false positives (red). Horizontal dotted lines: mean value of true nonzero coefficients for the chosen support set (all of which are positive).

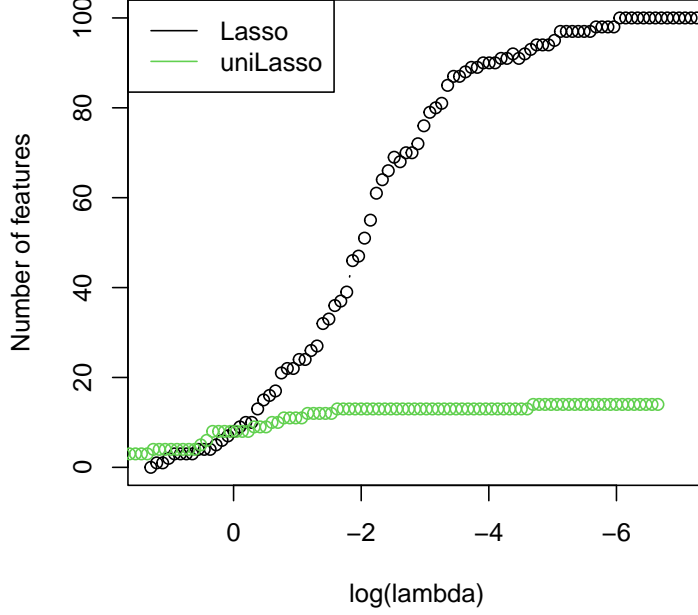


Figure 2: *Lasso and uniLasso: size of support set as $\lambda \rightarrow 0$ for low-SNR setting with $n = 300, p = 100$.*

The next example shows a case where UniLasso performs poorly. Here $n = 300$, $p = 1000$, and the pairwise correlations between the features and the signs of the 100 true coefficients are both about half positive and negative. The SNR is 4.5. In Section 6 we see that **UniLasso does well when the population coefficients for positively correlated features have the same sign**, an assumption clearly violated in this example. Fortunately, in any given example, we can assess the relative merits of the Lasso and UniLasso via cross-validation.

3 Relationship of UniLasso to the Adaptive Lasso

The adaptive Lasso (Zou 2006) is defined by

$$\hat{\beta}^* = \operatorname{argmin} \frac{1}{2} \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p \hat{w}_j |\beta_j| \quad (5)$$

where $\hat{w}_j = 1/|\hat{\beta}_j|^\nu$ and $\hat{\beta}$ is any root- n consistent estimate of β , for example the least squares estimates.

The original paper assumed $n > p$; with $p > n$, one cannot use the least-squares estimates. One possibility is to use an initial estimate (from say Ridge or Lasso), solve the adaptive Lasso, and iterate. Another option is to use the univariate least-squares estimates: Huang, Ma & Zhang (2008) consider a combination of these to achieve some theoretical guarantees.

Let $\hat{\beta}_{j,u}$ be the univariate least squares coefficients. Consider the adaptive Lasso with weights $1/|\hat{\beta}_{j,u}|$. Then it is easy to show that this procedure is equivalent to uniLasso with $\hat{\eta}_j^i$ replacing their LOO versions $\hat{\eta}_j^{-i}$, and removing the non-negativity constraint in the second stage.

This procedure does not share some of the properties of uniLasso. In particular, the final coefficients may not have the same signs as the univariate coefficients. Importantly, in our simulations of

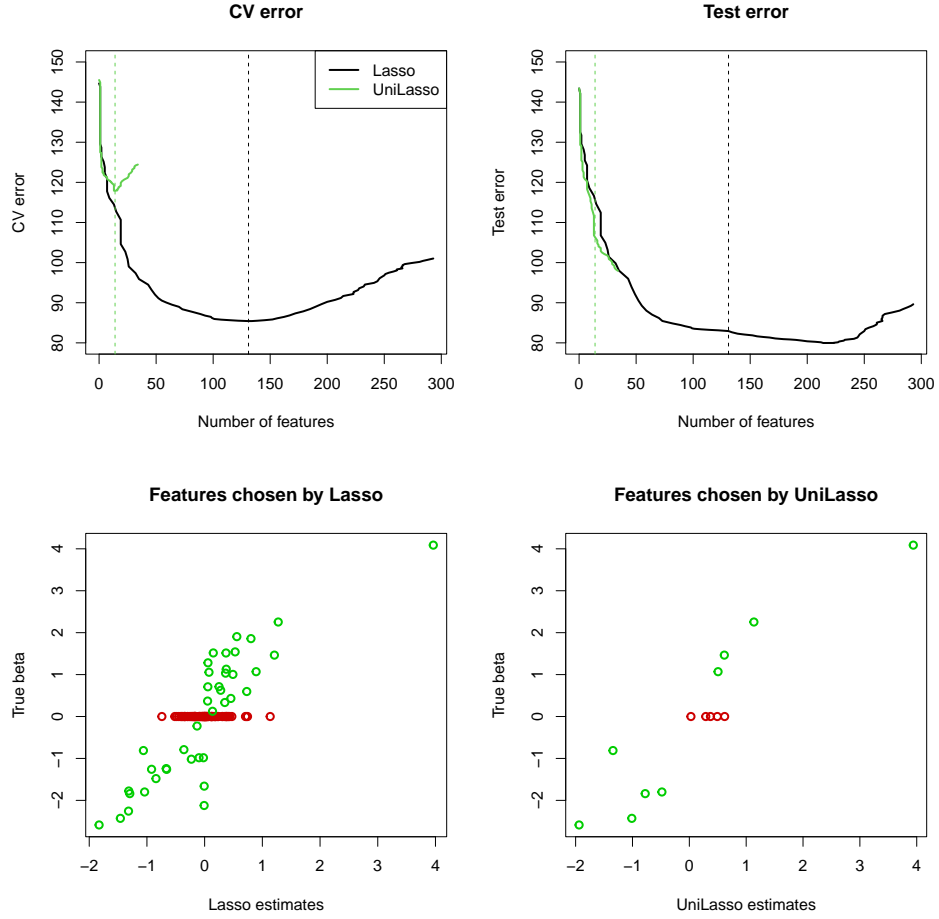


Figure 3: Results for simulated high SNR example, where the true number of nonzero coefficients is 100 and $p = 1000$. Top panel: CV and test set prediction error. The dotted vertical lines shows the values of λ , chosen by CV for each model. Bottom panels: true positives (green) and false positives (red).

Section 7, it tends to yield less sparse models and sometimes much higher MSE.

Alternatively, one could include the non-negativity constraints in this version of the adaptive lasso. The following result shows that we obtain a procedure equivalent to uniLasso, except for the use of LOO estimates in step 1.

Proposition 1. Let $(\hat{\beta}_{0j}, \hat{\beta}_j)$ be the univariate least squares coefficients for variable j in a p variable linear model with data $\{(x_i, y_i)\}_1^n$. Let $\eta_j^i = \hat{\beta}_{0j} + \hat{\beta}_j x_{ij}$ be the univariate linear fit for variable j and observation i .

Then the following two problems are equivalent:

$$\min_{\theta} \sum_{i=1}^n (y_i - \theta_0 - \sum_{j=1}^p \eta_j^i \theta_j)^2 + \sum_{j=1}^p |\theta_j| \text{ s.t. } \theta_j \geq 0 \quad \forall j \quad (6)$$

$$\min_{\gamma} \sum_{i=1}^n (y_i - \gamma_0 - \sum_{j=1}^p x_{ij} \gamma_j)^2 + \sum_{j=1}^p \frac{|\gamma_j|}{|\hat{\beta}_j|} \text{ s.t. } \text{sign}(\gamma_j) = \text{sign}(\hat{\beta}_j) \quad \forall j \quad (7)$$

The exact equivalence is obtained with $\hat{\gamma}_j = \hat{\theta}_j \hat{\beta}_j$, and $\hat{\gamma}_0 = \hat{\theta}_0 + \sum_{\ell=1}^p \hat{\theta}_\ell \hat{\beta}_{0\ell}$.

The proof is in the Appendix. However our experiments with this version of adaptive lasso produced similar results to the one that does not enforce sign constraints,

Here is an example. Figure 3 shows the results from one realization of the “low-SNR” setting of Section 7. We see that **both versions of the adaptive lasso overfit, and cross-validation fails to detect this.** On the other hand, uniLasso chooses a null model and achieves a much lower MSE. The use of the usual (non-LOO) univariate coefficients in the adaptive lasso causes the problem: it allows the procedure to overfit and cross-validation fails. The reason is that for computational efficiency, cross-validation is not wrapped around both steps (estimation of univariate coefficients and the lasso fit) but is only applied to the lasso fit.

By using LOO fits, uniLasso avoids this problem, because step 1 is internally validated. We note that this poor performance of adaptive lasso does not always occur in this setting, but occurs often enough to hurt its average performance, especially in support size.

Finally we note that Candès, Wakin & Boyd (2008) propose an iterated version of the adaptive Lasso, in which the current solutions are used to define weights for the next iteration. They show that the method can enhance sparsity, especially for signal recovery.

4 Application to the diabetes data

Used in Efron, Hastie, Johnstone & Tibshirani (2004), this dataset consists of ten baseline variables: age, sex, body mass index, average blood pressure and six blood serum measurements. The measurements were obtained for each of $n = 442$ diabetes patients, as well as the response of interest, a quantitative measure of disease progression one year after baseline.

Table 3 shows the full least squares fit on the left, and the Lasso and uniLasso coefficients in the middle panel. The uniLasso coefficients for both the unregularized case (UniReg, $(\lambda = 0)$ and the cv choice of λ are shown; they are very similar. We did 200 half-samplings of the data, training on the first half and computing test error on the second half. The two rightmost columns of Table 3 shows the frequency of selection over the 200 realizations.

It is interesting to first compare the unregularized models least squares and UniReg. The biggest surprise is the sex coefficient: it is negative and highly significant in least squares but is not even chosen by uniLasso! It turns out that sex on its own has a positive coefficient and is highly significant, and this explains why uniLasso did not choose it. Now what causes the sign change for sex? Due to

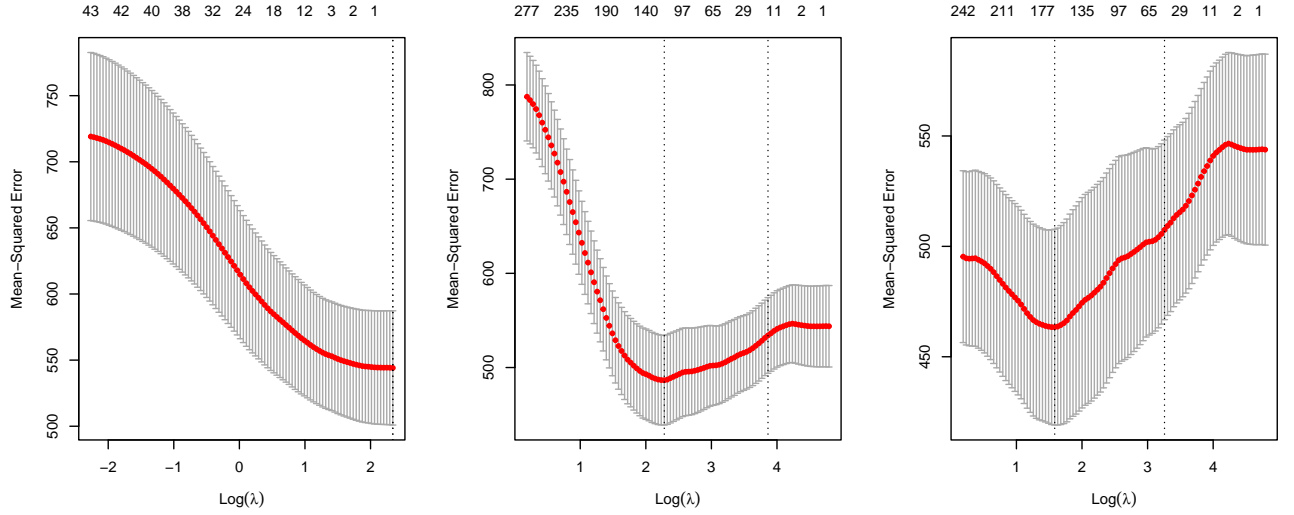


Figure 4: *CV curves from a low-SNR simulation with $n = 300, p = 1000$. The test MSE for each chosen model is given above each panel.*

	Least Squares				Lasso	UniReg	UniLasso	Lasso-freq	UniLasso-freq
	Estimate	Std. Error	t value	$\Pr(> t)$					
age	-0.01	0.04	-0.17	0.87	0.00	0.00	0.00	124.00	4.00
sex	-0.15	0.04	-3.92	0.00	-0.12	0.00	-0.00	196.00	4.00
bmi	0.32	0.04	7.81	0.00	0.32	0.34	0.33	200.00	200.00
bp	0.20	0.04	4.96	0.00	0.18	0.16	0.16	200.00	197.00
s1	-0.49	0.26	-1.90	0.06	-0.06	0.00	0.00	152.00	0.00
s2	0.29	0.21	1.41	0.16	0.00	0.00	0.00	60.00	0.00
s3	0.06	0.13	0.47	0.64	-0.14	-0.12	-0.10	184.00	191.00
s4	0.11	0.10	1.10	0.27	0.00	0.00	0.00	82.00	15.00
s5	0.46	0.11	4.37	0.00	0.32	0.30	0.33	200.00	200.00
s6	0.04	0.04	1.02	0.31	0.03	0.01	0.01	163.00	96.00

Table 3: *Least squares, Lasso and UniLasso results on full data*

the correlation between bmi and sex, adding sex to the model containing bmi has a strong effect. In addition, there is a moderately strong interaction between bmi and sex, with a negative coefficient. Hence the strong negative coefficient of sex in the LS model is really reflecting the relationship between bmi and sex. In summary, looking at both LS and unregularized uniLasso can help us to understand some hidden effects.

Turning now to lasso and unLasso with regularization, Table 4 shows the test error: Least Squares (LS) has error about 1% lower than Lasso, and 6.5% lower than UniLasso. However as seen in Table 4, UniLasso displays much smaller average model size. Further in Table 5 we see that age appears in about 60% of the models chosen by LS and the lasso, and about half the time, its coefficient is negative. Features s2 and s3 show similar behavior.

Figure 4 shows the distribution of coefficients over 200 half-samplings. Those for UniLasso are much sparser and always have the same sign. In this example, the choice between Lasso and UniLasso depends on whether prediction error or model stability and interpretation are more important.

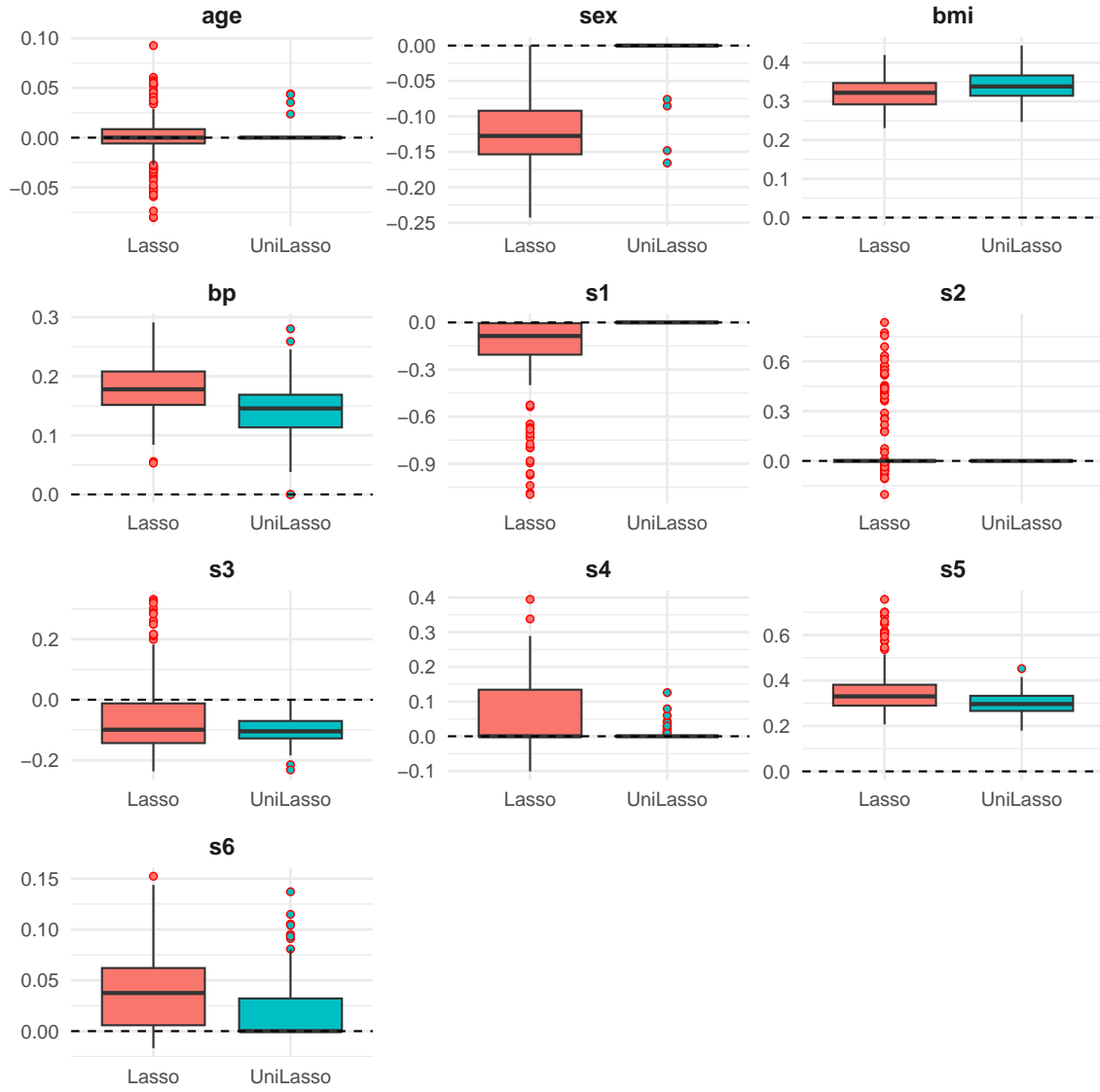


Figure 5: *Coefficients of diabetes predictors over 200 half-samplings.*

	LS	Lasso	UniLasso
mean	51.72	52.20	55.22
se	0.44	0.44	0.45
support size	10.00	7.80	4.54

Table 4: *Average test error and support over 200 half-samples*

Feature	Least Squares			Lasso		
	-1	0	1	-1	0	1
age+	108	0	92	60	76	64
s2+	17	0	183	28	140	32
s3−	58	0	142	158	16	26

Table 5: *Three features with the most varying coefficient signs over the 200 half-samples. Their univariate signs are +, +, − respectively.*

5 Analysis of UniLasso with Orthonormal Columns

In this section, in the special case of orthonormal features, we derive explicit formula for the UniLasso coefficients. Figure 5 shows the Lasso and UniLasso paths for a simulated example with an orthonormal feature matrix. They look quite different, with the UniLasso path being considerable sparser.

It is well-known that in this setting, the Lasso coefficients are simply soft-thresholded versions of the univariate least squares estimates. This is (approximately) true of UniLasso, but with a different thresholding function, as we now show.

Suppose X is orthonormal so that each column x_j satisfies $\|x_j\|_2^2 = 1$ and $x_i^T x_j = 0$ for $i \neq j$. Assume also that $\bar{y} = 0$ and $\bar{x}_j = 0$ for each j . Note that the least squares coefficients are $\hat{\beta}_{0j} = 0$ and $\hat{\beta}_j = x_j^T y$.

Here we use the actual fits $\hat{\eta}_j^i$ rather than the LOO fits $\hat{\eta}_j^{-i}$ in the second stage — an approximation that simplifies the derivation, and often gives very similar solutions to UniLasso. In this case we have $\hat{\eta}_j^i = \hat{\beta}_j x_{ij}$, since $\hat{\beta}_{0j} = 0$. We can ignore θ_0 , which will be zero since $\bar{y} = 0$ and all the $\hat{\eta}_j = \hat{\beta}_j x_j$ have means zero.

The coefficients $\hat{\theta}_j$ are determined by solving:

$$\min_{\theta_j} \frac{1}{2} \|y - \sum_{\ell=1}^p \theta_\ell \hat{\beta}_\ell x_\ell\|_2^2 + \lambda \|\theta\|_1$$

We have the derivative w.r.t. θ_j :

$$-\hat{\beta}_j x_j^\top \left(y - \sum_{\ell=1}^p \theta_\ell \hat{\beta}_\ell x_\ell \right) + \lambda \cdot \text{sign}(\theta_j) = 0$$

Since the x_j are orthogonal and unit norm, and using $x_j^\top y = \hat{\beta}_j$, we get $\hat{\beta}_j^2 \theta_j = \hat{\beta}_j^2 - \lambda \cdot \text{sign}(\theta_j)$, and hence

$$\theta_j = \left(1 - \frac{\lambda}{\hat{\beta}_j^2} \right)_+.$$

Hence the final coefficients for x_j are

$$\hat{\gamma}_j = \hat{\theta}_j \cdot \hat{\beta}_j = \hat{\beta}_j \left(1 - \frac{\lambda}{\hat{\beta}_j^2} \right)_+ = \text{sign}(\hat{\beta}_j) \left(|\hat{\beta}_j| - \frac{\lambda}{|\hat{\beta}_j|} \right)_+ \quad (8)$$

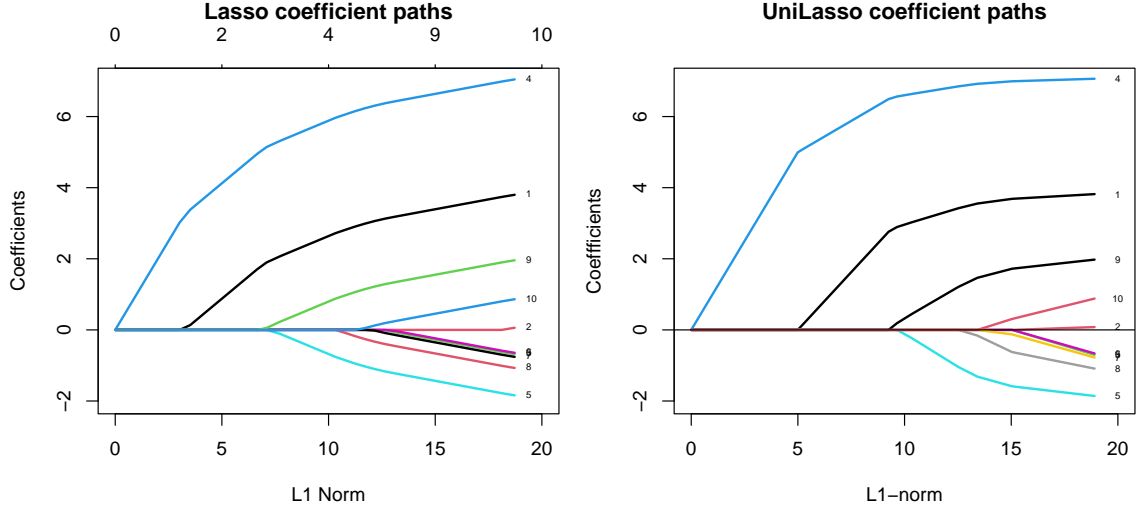


Figure 6: *Coefficient paths for Lasso and UniLasso, in a simulated example with 10 orthonormal features.*

The last expression can be compared with the similar expression for the Lasso in this situation:

$$\text{sign}(\hat{\beta}_j) \left(|\hat{\beta}_j| - \lambda \right)_+ . \quad (9)$$

Figure 5 shows the shrinkage functions for ridge regression, Lasso, and UniLasso. Ridge uses proportional shrinkage, while Lasso translates all coefficient to zero by the same amount. UniLasso is similar to Lasso, except that the larger coefficients are shrunk less than the smaller ones.

The SparseNet procedure (Mazumder et al. 2011) uses a thresholding function that has a roughly similar shape to that of UniLasso, but its objective is not convex. This shrinkage pattern is somewhere between Lasso and ℓ_0 or best subset selection. UniLasso achieves this without losing convexity, a significant computational advantage.

6 Theoretical analysis

In this section we study the support recovery and mean-squared error properties of UniLasso. Let X_1, \dots, X_p be square-integrable random variables with nonzero variance, defined on the same probability space, and let

$$Y = \gamma_0 + \sum_{j \in S} \gamma_j X_j + \epsilon,$$

where S is a subset of $\{1, \dots, p\}$, ϵ is a mean zero random variable that is independent of the X_j 's, and the $(\gamma_j)_{j \in S}$ are nonzero coefficients. We will refer to S as the *support*. Assume that Y also has nonzero variance. Our data consists of i.i.d. random vectors $(Y_i, X_{i,1}, \dots, X_{i,p})$, $i = 1, \dots, n$ (where $n \geq 2$), each having the same distribution as (Y, X_1, \dots, X_p) .

The UniLasso algorithm with penalty parameter $\lambda > 0$ goes as follows. For each $1 \leq i \leq n$ and $1 \leq j \leq p$, let

$$\hat{\beta}_{i,j} := \frac{\frac{1}{n-1} \sum_{k \neq i} Y_k X_{k,j} - \left(\frac{1}{n-1} \sum_{k \neq i} Y_k \right) \left(\frac{1}{n-1} \sum_{k \neq i} X_{k,j} \right)}{\frac{1}{n-1} \sum_{k \neq i} X_{k,j}^2 - \left(\frac{1}{n-1} \sum_{k \neq i} X_{k,j} \right)^2}$$

be the regression coefficient from the univariate regression of Y on X_j omitting observation i , and let

$$\hat{\alpha}_{i,j} := \frac{1}{n-1} \sum_{k \neq i} Y_k - \frac{\hat{\beta}_{i,j}}{n-1} \sum_{k \neq i} X_{k,j}$$

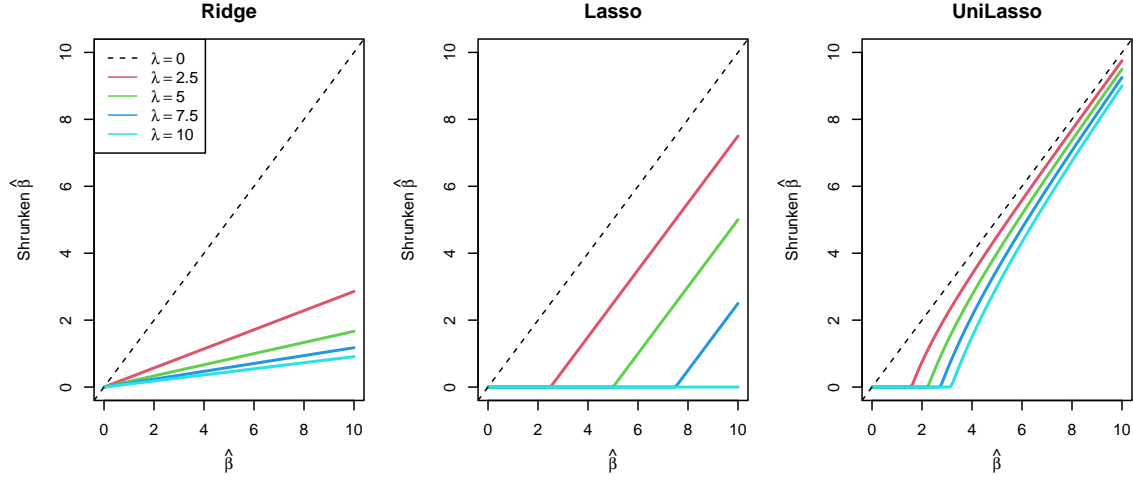


Figure 7: *Shrinkage functions for ridge regression, Lasso and UniLasso, in a simulated example with 10 orthonormal features.*

be the intercept term. Note that $\hat{\beta}_{i,j}$ is a consistent estimate of

$$\beta_j := \frac{\text{Cov}(Y, X_j)}{\text{Var}(X_j)},$$

and $\hat{\alpha}_{i,j}$ is a consistent estimate of

$$\alpha_j := E(Y) - \beta_j E(X_j).$$

Then, let

$$\hat{Y}_{i,j} := \hat{\alpha}_{i,j} + \hat{\beta}_{i,j} X_{i,j}$$

be the predicted value of Y_i from this univariate regression. Next, obtain the estimates $\hat{\theta}_j$, $j = 0, \dots, p$, by minimizing

$$L(\theta_1, \dots, \theta_p) = \frac{1}{n} \sum_{i=1}^n (Y_i - \theta_0 - \theta_1 \hat{Y}_{i,1} - \dots - \theta_p \hat{Y}_{i,p})^2 + \lambda \sum_{j=1}^p \theta_j$$

subject to the constraint that $\theta_j \geq 0$ for each $1 \leq j \leq p$. Finally, define

$$\hat{\gamma}_j := \hat{\theta}_j \hat{\beta}_j,$$

to be the UniLasso estimate of γ_j for $1 \leq j \leq p$, where

$$\hat{\beta}_j := \frac{\frac{1}{n} \sum_{k=1}^n Y_k X_{k,j} - (\frac{1}{n} \sum_{k=1}^n Y_k)(\frac{1}{n} \sum_{k=1}^n X_{k,j})}{\frac{1}{n} \sum_{k=1}^n X_{k,j}^2 - (\frac{1}{n} \sum_{k=1}^n X_{k,j})^2}$$

is the regression coefficient from the univariate regression of Y on X_j , and let

$$\hat{\gamma}_0 := \hat{\theta}_0 + \sum_{j=1}^p \hat{\theta}_j \hat{\alpha}_j,$$

where

$$\hat{\alpha}_j := \frac{1}{n} \sum_{k=1}^n Y_k - \hat{\beta}_j \frac{1}{n} \sum_{k=1}^n X_{k,j}$$

is the intercept term from the univariate regression of Y on X_j .

The following theorem shows, roughly speaking, that if (1) $\text{sign}(\gamma_j) = \text{sign}(\beta_j)$ for each $j \in S$, (2) the penalty parameter λ is bigger than $|\beta_j|$ for all $j \notin S \cup \{0\}$, and (3) both $\log p$ and $\log n$ are small compared to $n\lambda^2$, then with high probability, $\hat{\gamma}_j = 0$ for all $j \notin S \cup \{0\}$ and $\hat{\gamma}_j = \gamma_j + O(\lambda)$ for all $j \in S \cup \{0\}$.

Theorem 1. *Suppose that:*

1. $\gamma_j \beta_j > 0$ for each $j \in S$.
2. The covariance matrix of $(X_j)_{j \in S}$ is nonsingular with minimum eigenvalue η .
3. There is a positive constant C_0 such that $\text{Var}(Y) \geq C_0$ and $\text{Var}(X_j) \geq C_0$ for each $j \in S$.
4. There are positive constants C_1 and C_2 such that for each $t \geq 0$ and $1 \leq j \leq p$, $\mathbb{P}(|Y| \geq t)$, $\mathbb{P}(|\epsilon| \geq t)$ and $\mathbb{P}(|X_j| \geq t)$ are bounded above by $C_1 e^{-C_2 t^2}$.

Let $M_1 := \max_{j \in S \cup \{0\}} |\gamma_j|$, $M_2 := \min_{j \in S} |\beta_j|$ and $M_3 := \max_{j \in S} |\beta_j|$. Then there are positive constants K_1, K_2, K_3, K_4, K_5 depending only on $C_0, C_1, C_2, \eta, M_1, M_2, M_3$ and $|S|$ such that if

$$K_1 \max_{j \notin S \cup \{0\}} |\beta_j| \leq \lambda \leq K_2,$$

then

$$\begin{aligned} & \mathbb{P}(\hat{\gamma}_j = 0 \text{ for all } j \notin S \cup \{0\} \text{ and } |\hat{\gamma}_j - \gamma_j| \leq K_3 \lambda \text{ for all } j \in S \cup \{0\}) \\ & \geq 1 - K_4 p n e^{-K_5 n \lambda^2}. \end{aligned}$$

The above theorem is roughly comparable to the available results for the Lasso. A close comparison would be, for instance, (Hastie, Tibshirani & Wainwright 2015, Theorem 11.3). Like Theorem 1, this theorem also assumes a lower bound on the covariance matrix of the covariates in the support, and the maximum difference between $\hat{\gamma}_j$ and γ_j for $j \in S$ is of order λ . However, there is one key difference. The results about Lasso, including the one cited above, require a condition known as *mutual incoherence* or *irrepresentability*. Roughly speaking, it means that if we regress X_k for some $k \notin S$ on $(X_j)_{j \in S}$, the regression coefficients should be small. Notably, our Theorem 1 requires no such relation to hold between the covariates inside and outside the support. All we need is that the univariate regression coefficients of Y on covariates outside the support are small.

Having said that, we make clear that assumption (1) above is a crucial one: namely that γ_j and β_j have the same sign for each $j \in S$. The next result gives a natural sufficient condition under which this holds.

Theorem 2. *Let $\delta_{j,k}$ denote the population value of the univariate regression of X_j on X_k . Suppose that for every pair of distinct indices $j, k \in S$, $\delta_{j,k} \geq 0$ if γ_j and γ_k have the same sign, and $\delta_{j,k} \leq 0$ if γ_j and γ_k have opposite signs. Then β_j is nonzero and has the same sign as γ_j for each $j \in S$. Moreover, $|\beta_j| \geq |\gamma_j|$ for each $j \in S$.*

The last result, partly due to Ryan Tibshirani, gives a necessary and sufficient condition for equality of signs.

Theorem 3. *Suppose that $Y = \sum_{j \in S} \gamma_j X_j + \epsilon$, where all the γ_j 's are nonzero, and ϵ has zero mean, finite variance, and is independent of $(X_j)_{j \in S}$. Let β_j be the coefficient of X_j in the univariate (population) regression of Y on X_j . Let $\delta_{k,j}$ be the coefficient of X_j in the univariate (population) regression of X_k on X_j . For each j , let A_j be the set of $k \in S$ for which $\text{sign}(\delta_{k,j}) = \text{sign}(\gamma_k \gamma_j)$ or $\delta_{k,j} = 0$. Then $\beta_j \gamma_j \geq 0$ if and only if*

$$\sum_{k \notin A_j} |\gamma_k \delta_{k,j}| \leq \sum_{k \in A_j} |\gamma_k \delta_{k,j}|.$$

In particular, if $A_j = S$, then this holds.

The proofs of these results are in the Appendix.

7 Some simulation results

Figure 8 shows the results of a comparative study of UniLasso with Lasso and two other sparse regression methods. Shown are means and standard errors over 100 simulations.

The methods are:

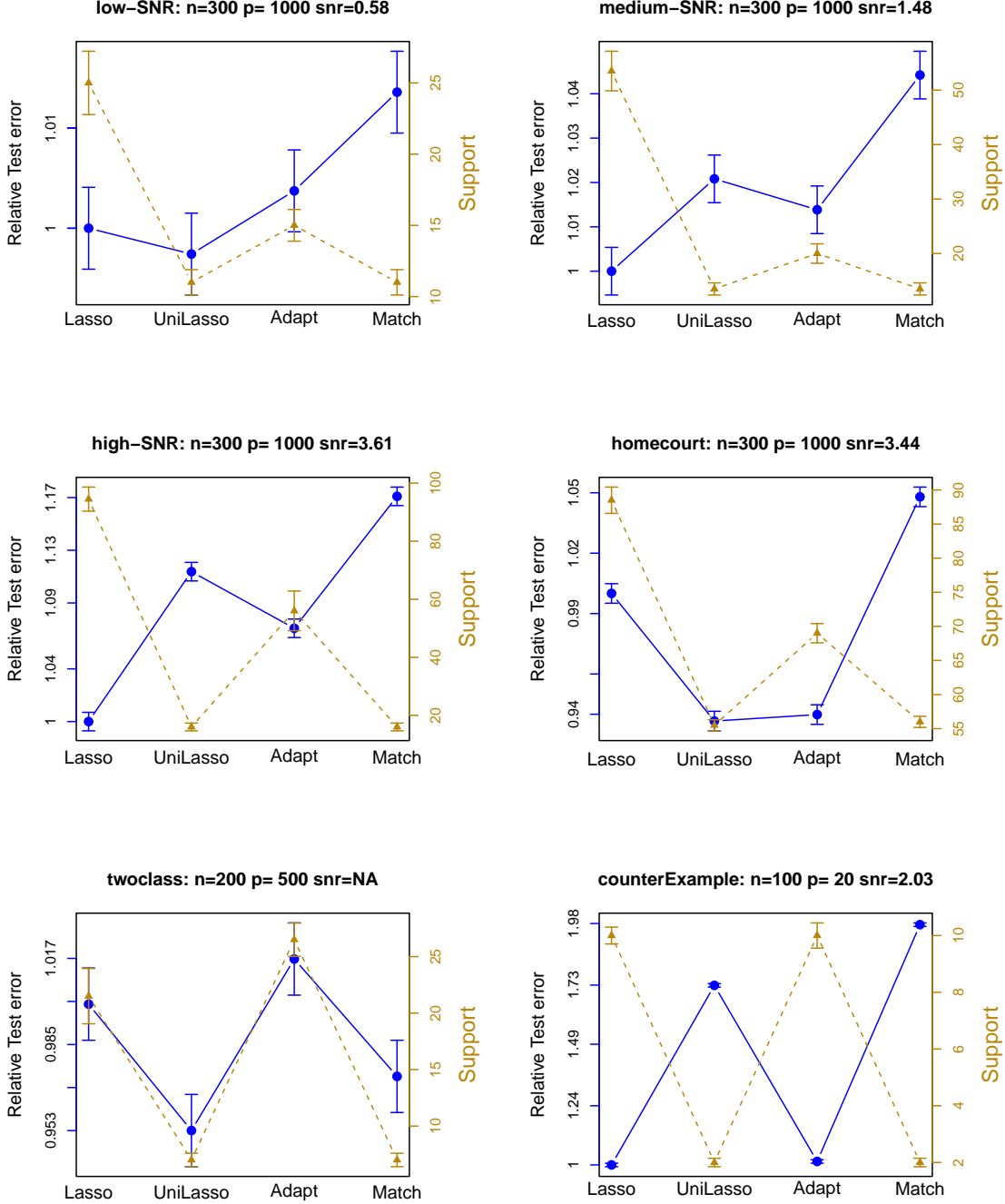
- Lasso (using `glmnet`),
- UniLasso
- Adaptive Lasso using penalty weights $1/|\hat{\beta}_j^{uni}|$. These first three methods use CV to estimate the value of λ that minimizes test error.
- Matching: a variant of the Lasso, where we increase the λ parameter from the CV-based optimal value, until the support size matches that of UniLasso.

The simulation scenarios are:

1. **Low, medium and high SNR:** Here $n = 300, p = 1000$, $N(0, 1)$ features with pairwise correlation 0.5; The nonzero regression coefficients are $N(0, 1)$; Gaussian errors with SD σ chosen to produce low < 1 , medium (≈ 1), or high (> 2) SNRs.
2. **Homecourt:** As in #1, but the feature covariance is AR(1) with $\rho = 0.8$ and the non-zero regression coefficients are $U(0.5, 2)$, and are positioned on every other feature (1, 3, ... 99).
3. **Two-class:** The feature covariance is in #2, with $n = 200, p = 500$ and a binary target y . The first 20 features are shifted in the $y = 1$ class by 0.5 units.
4. **Counter-example:** $n = 100, p = 20$, $x_1 \sim N(0, 1), x_2 = x_1 + N(0, 1), \beta = (1, -.5, 0, 0, \dots 0)$, error SD = 0.5.

In the first 3 scenarios, 10% of the true coefficients are non-zero. Figure 8 shows the test set error (relative to the Lasso) and support size (number of non-zero coefficients)

Figure 8: Blue lines: relative test error (Gaussian) , misclassification rate (Binomial) rate or deviance (Cox), with support size superimposed in gold.



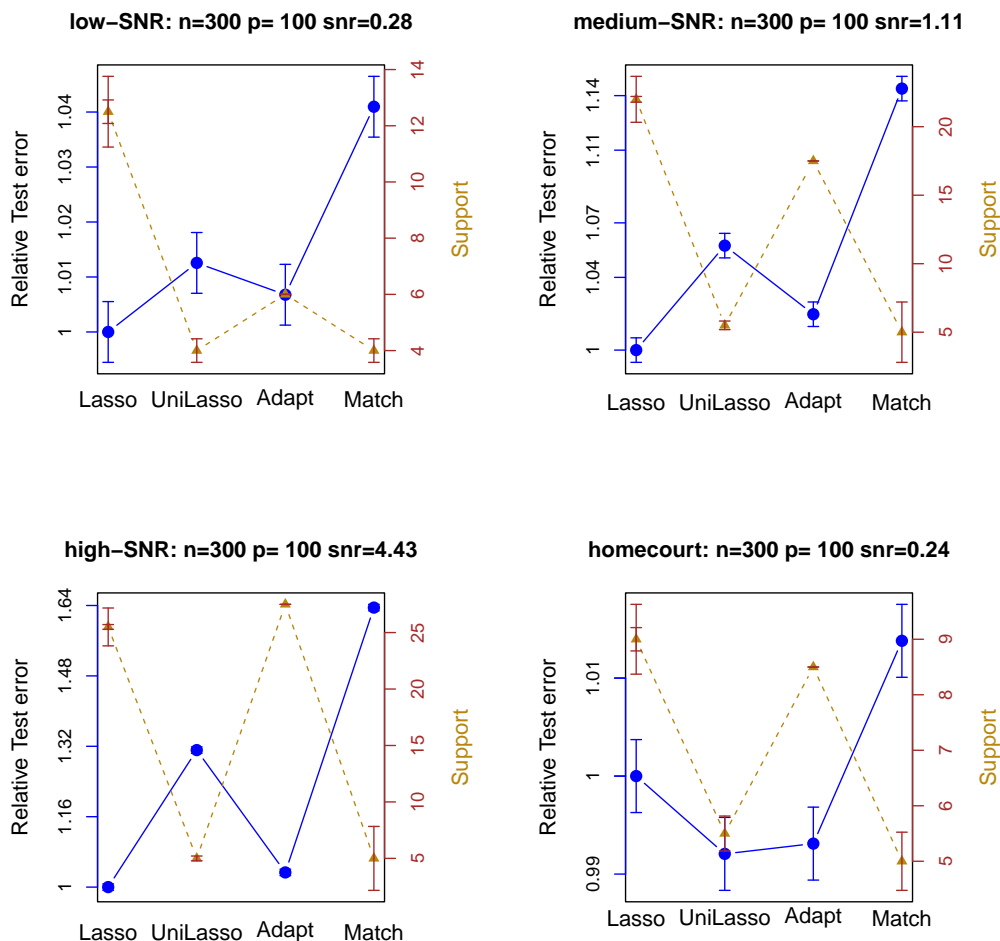
Overall we see that UniLasso shows MSE similar to that of Lasso, with often a much smaller support. The same general pattern holds with the real datasets.

Exceptions to this are the high-SNR setting where Lasso wins handily and the homecourt setting designed to exploit UniLasso's strength (recall Theorem 2). In neutral court, the regression coefficients are allowed to have arbitrary signs and the advantage of UniLasso goes away. The "counter-example" setting is the classic case where features have positive correlation but the true regression coefficients have opposite signs; not surprisingly UniLasso does poorly. Neither the adaptive Lasso or the matching method show much advantage, compared to Lasso and UniLasso .

Of course in practice one has cross-validation to help determine which method is preferred in a given example.

Figure 9 shows the corresponding results for the simulated examples, for $n = 300, p = 100$.

Figure 9: $N > p$ case. Blue lines: relative test error (Gaussian) misclassification rate (Binomial) or deviance (Cox), with support size superimposed in gold.



8 The unregularized setting: uniReg

8.1 Simulated data

Here we consider the case where $\lambda = 0$, so that there is no ℓ_1 penalty, as detailed in equation (4). While the main use case has $n > p$, we note that uniReg can be defined for $p > n$ by taking $\lambda \rightarrow 0$ in uniLasso. Computationally, this will approximately give the solution with minimum ℓ_1 norm, in the same way that glmnet gives the least squares solution with minimum ℓ_1 norm.

Conveniently even without the ℓ_1 penalty, the non-negativity constraint still promotes sparsity in the solution. In Table 6 we compare least squares with uniReg in a subset of the simulated settings defined in Section 7. UniReg performs remarkably well, often winning in MSE and yielding consistently sparser models.

Table 6: Comparison of least squares with uniReg in a subset of simulated settings defined earlier.

$N = 300, p = 100$							
MSE							
	setting	family	n	p	SNR	mse-LS	mse-uniReg
1	low-SNR	gaussian	300	100	0.318	97.678	69.655
2	medium-SNR	gaussian	300	100	1.272	24.428	18.559
3	high-SNR	gaussian	300	100	5.087	6.104	5.82
4	homecourt	gaussian	300	100	0.228	149.683	104.979
Support							
	setting	family	n	p	SNR	supp-LS	supp-uniReg
1	low-SNR	gaussian	300	100	0.318	100	8
2	medium-SNR	gaussian	300	100	1.272	100	9
3	high-SNR	gaussian	300	100	5.087	100	7
4	homecourt	gaussian	300	100	0.228	100	13
$N = 300, p = 30$							
MSE							
	setting	family	n	p	SNR	mse-LS	mse-uniReg
1	low-SNR	gaussian	300	30	0.092	71.339	65.171
2	medium-SNR	gaussian	300	30	0.369	17.836	16.489
3	high-SNR	gaussian	300	30	1.475	4.459	4.178
4	homecourt	gaussian	300	30	0.059	111.771	101.542
Support							
	setting	family	n	p	SNR	supp-LS	supp-uniReg
1	low-SNR	gaussian	300	30	0.092	30	4
2	medium-SNR	gaussian	300	30	0.369	30	4
3	high-SNR	gaussian	300	30	1.475	30	4
4	homecourt	gaussian	300	30	0.059	30	4
$N = 300, p = 1000$							
MSE							
	setting	family	n	p	SNR	mse-LS	mse-uniReg
1	low-SNR	gaussian	300	1000	0.578	842.191	510.837
2	medium-SNR	gaussian	300	1000	1.48	374.915	249.076
3	high-SNR	gaussian	300	1000	3.613	186.171	147.276
4	homecourt	gaussian	300	1000	1.53	577.852	420.508
Support							
	setting	family	n	p	SNR	supp-LS	supp-uniReg
1	low-SNR	gaussian	300	1000	0.578	1000	24.5
2	medium-SNR	gaussian	300	1000	1.48	1000	27
3	high-SNR	gaussian	300	1000	3.613	1000	27
4	homecourt	gaussian	300	1000	1.53	1000	135.5

8.2 Real data

We compared least squares and uniReg on the regression datasets from the UCI database. There were 23 datasets in all. Some had multiple targets: we treated each target separately and averaged the results. We randomly sampled each dataset into a 70/30 training/test split, and report the test set MSE and support in Figure 8.2. One dataset had a small sample ($n = 60$) which produced a very large MSE from least squares; we omit this from the plot for visibility.

The two methods perform similarly in MSE; but uniLasso has much smaller support, averaging about 1/3 that of the least squares support (which has the number of features p).

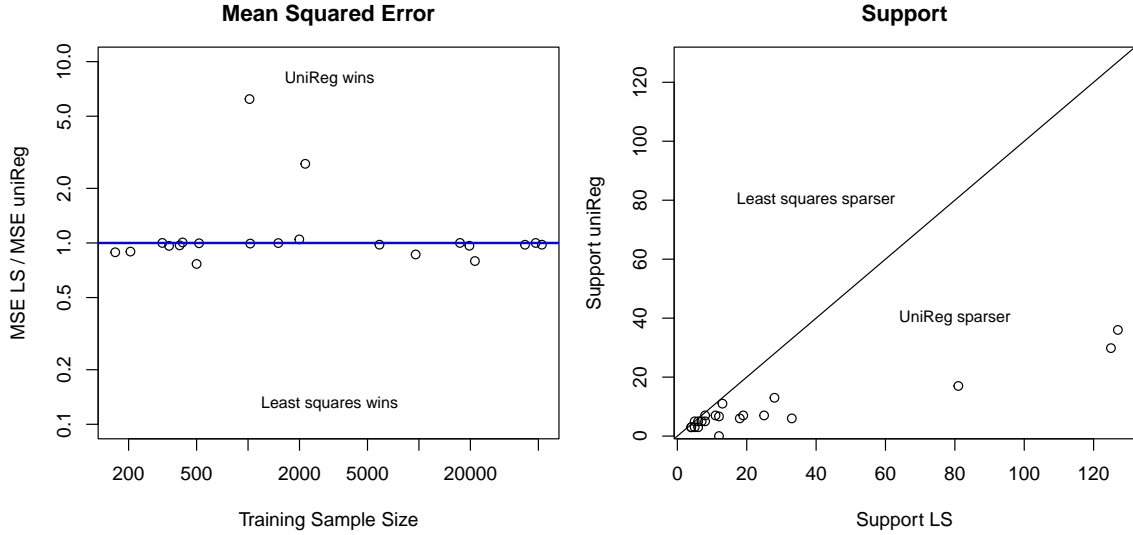


Figure 10: *LS vs uniReg on regression datasets from the UCI database. All datasets have $n > p$. The left plot shows the MSE ratio on the log scale. The right panel shows the support sizes of the chosen models.*

Dataset	n	p	Feature type	Outcome type
Spam	2301	57	mixed	binary
NRTI	1005	211	binary mutations	drug effectiveness (quantitative)
Leukemia	72	7129	gene expression (quantitative)	binary
DLBCL	240	1000	gene expression (quantitative)	survival
Breast Cancer	157	1000	gene expression (quantitative)	survival
Ovarian	190	2238	Mass spec peak intensities	binary

Table 7: *Summary of datasets used in our study.*

9 Real data examples for uniLasso

Table 7 gives a high level summary of six datasets that we examined in this study. Figure 9 shows the results: these are test error and support size averaged over 100 train/test 50-50 splits.

We see the same general trend as in the simulated examples: UniLasso tends to give test error similar to that of Lasso, with often smaller support size.

10 More examples

10.1 UniLasso and adversarial attacks

In machine learning terminology, an adversarial attack creates an altered input that leads to a large error in the model’s prediction. In image recognition problems, the attack also produces an image that is imperceptible to humans. In statistical terms, an adversarial attack creates features in the test data that come from a different distribution than the training data,

A common way of creating an adversarial attack is to add noise in the direction of the gradient of the objective function (e.g. Kong & Ge (2023)). We investigate this here in the context of the proteomics example of Table 1. We carried out the following process:

- (a) Randomly split the data into a 70/30 train test split.

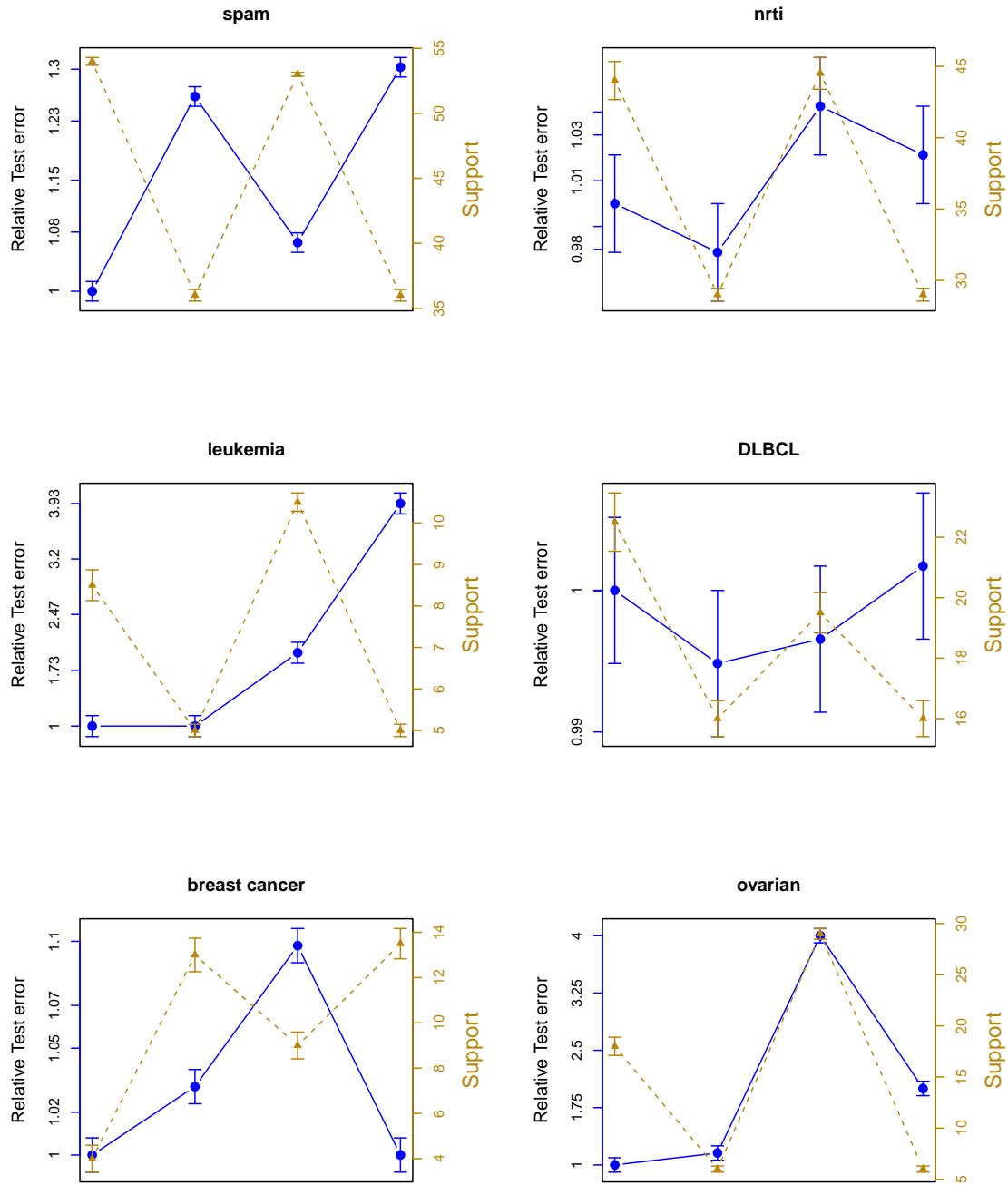


Figure 11: *Real data examples: Blue lines: relative test error (Gaussian) misclassification rate (Binomial) or deviance (Cox), with support size superimposed in gold.*

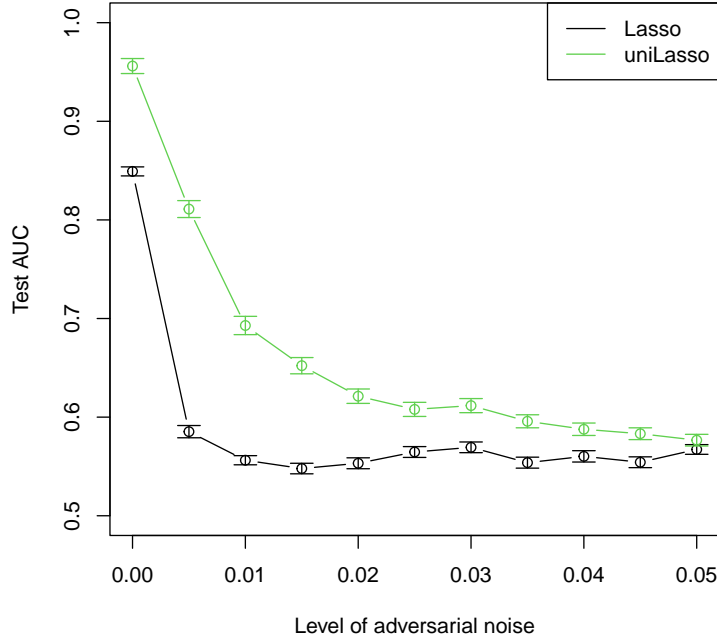


Figure 12: *Results of adversarial attacks in the proteomics example of Table 1: test set AUC*

- (b) Fit lasso and uniLasso to the training data
- (c) Add varying levels of Gaussian noise to the test features in the direction of the (sub) gradient of each model from (b), and compute the test set AUC.

We repeated this 200 times and plot the results in Figure 12. We see that uniLasso is less affected by the attack than the lasso, presumably because the univariate guidance disallows sign changes and the individual features that are selected are stronger on their own.

10.2 Multiclass setting and small round blue cell cancers

The multinomial approach to the multiclass problem does not generalize easily to UniLasso. The reason is that the coefficients for each class are only determined up to a shift, and hence a non-negativity constraint does not make sense. Instead we take a “one-versus rest” (OVR) approach, applying UniLasso binary classifier, to classify each class from the others. Then we take the class with the highest probability as the predicted class.

We applied this to a dataset on childhood cancer, with 63 training and 25 test samples in 4 classes. (Khan, Wei, Ringner, Saal, Ladanyi, Westermann, Berthold, Schwab, Antonescu, Peterson, & Meltzer 2001) With the original train/test split we get zero test errors with glmnet. We repeated the train/test split 50 times and obtained these results:

	mean errors out of 25	sd	mean(supp)	sd(supp)
Lasso multinomial	0.96	0.03	18.68	0.08
UniLasso OVR	0.18	0.01	27.02	0.05
Lasso OVR	0.22	0.01	58.08	0.08

10.3 A GWAS example

The data here come from the Kaggle site

Number of Errors	0	1	2	3	4	7
Lasso multinomial	27	9	8	4	1	1
UniLasso OVR	41	9				
Lasso OVR	39	11				

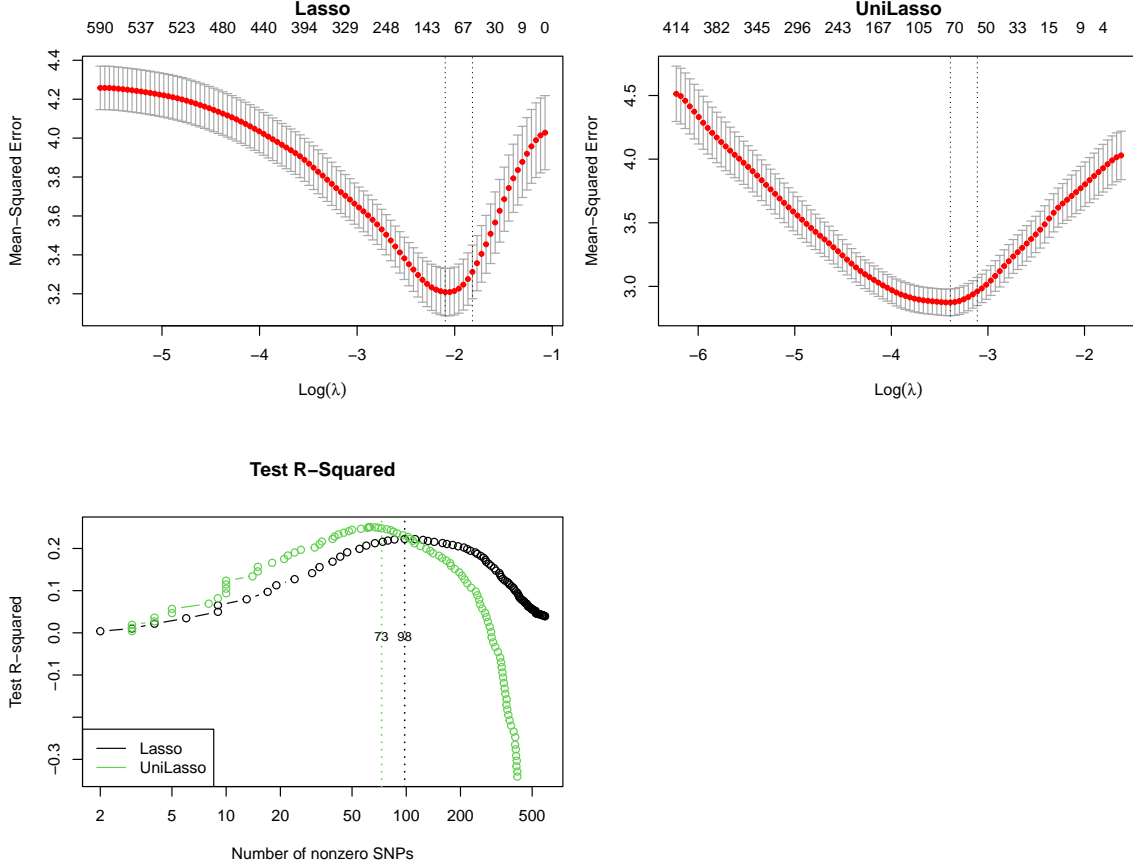


Figure 13: *GWAS example: CV and test set R-squared curves*

<https://www.kaggle.com/datasets/seascape/snp-dataset-for-gwas>

It is described as a simulated dataset comparable to the Illumina 650K human array, for SNP genotyping. There are 1000 individuals, and Genotypes (0, 1 or 2) for 482906 markers. For illustration, we selected the 10,000 markers with highest variance and divided the data into training and test sets each of size 500. The target measure is quantitative. In Figure 10.3 we see that UniLasso delivers a model with slightly higher test R^2 using fewer SNPs. Table 8 summarizes the signs of the Lasso and UniLasso coefficients in final chosen models. Interestingly, ignoring the zeroes, there are no sign disagreements!

	uni<0	uni=0	uni>0
lasso<0	33	13	0
lasso=0	1	9901	0
lasso>0	0	13	39

Table 8: *GWAS example: signs of Lasso and UniLasso coefficients in final chosen models.*

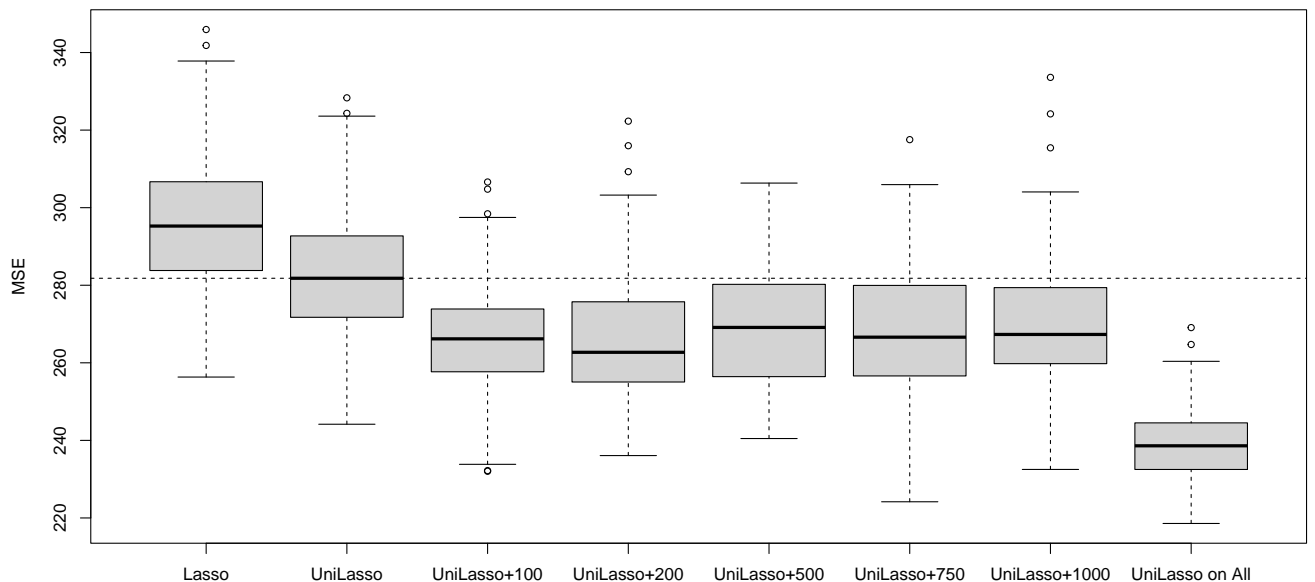


Figure 14: Results from an experiment examining the use of external data with uniLasso. See text for details.

11 The use of external univariate scores

Consider the setting where we have our training set T and also external data E from the same domain (e.g. disease). We assume that E does not contain raw data but only summary results. Specifically, E contains just univariate coefficients and standard errors for each feature. This setting occurs fairly often in bio-medical settings where investigators are not willing to share their raw data, but do publish and share summary results.

We demonstrate here how uniLasso can make productive use of external scores. The idea is to use the univariate coefficients from E , rather than computing LOO estimated from the training set. Then in step 2 of uniLasso, we proceed as usual.

To investigate this scheme, we generated data from our ‘homecourt’ setting of Section 7 ($n = 300, p = 1000, SNR = 1.5$), as well as 1000 extra samples from the same distribution. Figure 11 shows the test set MSE over 200 simulations from the following strategies:

- the test set MSE for the lasso, and uniLasso, both applied just to the training set
- uniLasso+ m , where the univariate scores derived from the m extra samples are used in place of the LOO estimates
- uniLasso+all- uniLasso trained on the combined 1300 samples.

The last estimate is there for reference: it is not available without access to the raw extra data.

We see that adding the scores from just 100 external observations improves the performance of uniLasso. Perhaps surprisingly, we don’t see any improvement with additional scores. The rightmost boxplot shows that access to the full external data (not just the scores) would lead to further improvement.

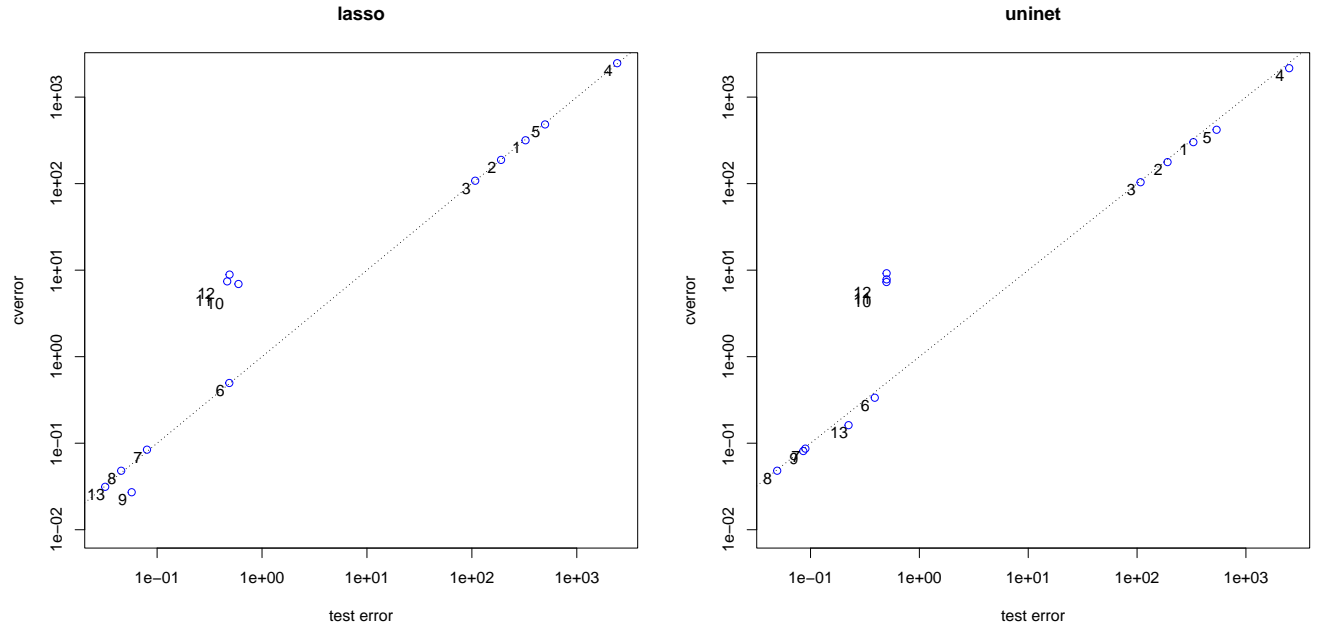


Figure 15: *CV error at the selected value of λ versus the test error, for the Lasso (left) and UniLasso (right). Each datapoint represents one of our simulated or real datasets studied earlier. Both CV estimates are quite good, with both over-estimating the test error in a few cases.*

12 Does CV work in our two stage UniLasso procedure?

In the UniLasso algorithm we use the target y in both steps, but for (significant) computational convenience, we only apply cross-validation in the 2nd step. Thus one should be concerned that cross-validation may not perform well here. In the simulations of Section 7 we used this form of cross-validation to choose the λ tuning parameter and it seems to have done a reasonably good job.

A related question is whether the error reported by CV is a good estimate of test error, especially at the selected value of λ . Figure 12 shows the CV error at the selected value of λ versus the test error, for the Lasso (left) and UniLasso (right). We see that both CV estimates are quite good, with both over-estimating the test error in a few cases. It seems clear that our use of cross-validation in UniLasso is not a major concern..

13 UniLasso for GLMs and the Cox Survival Model

Our discussion of uniLasso has focussed on squared-error loss. However, we can use the same idea in any scenario where we are able to fit a Lasso model and produce a scalar prediction function $\eta(x)$. This includes in particular all generalized linear models, and the Cox proportional hazards model. These models are included in `glmnet`.

The steps are almost the same as before (we will discuss for GLMs):

1. Fit the p univariate GLMs, and produce the linear predictor functions $\hat{\eta}_j(x_j)$. Compute the LOO predictions $\hat{\eta}_j^{-i}$ for the n training observations for each of these.
2. Using these LOO predictions as features, fit a non-negative Lasso GLM to the response, yielding coefficients $\hat{\theta} = (\hat{\theta}_0, \hat{\theta}_1, \dots, \hat{\theta}_p)^\top$.

3. Return the composite estimated linear predictor

$$\hat{\eta}(x) = \hat{\theta}_0 + \sum_{j=1}^p \hat{\theta}_j \hat{\eta}_j(x_j), \quad (10)$$

which collapses as before to a linear model $\hat{\eta}(x) = \hat{\gamma}_0 + \sum_{j=1}^p \hat{\gamma}_j x_j$.

Step 1 can be computationally challenging. Estimating the p univariate functions $\hat{\eta}_j(x_j)$ is fairly straightforward, since each require a few very simple Newton steps. The LOO estimates are more challenging. While for squared-error loss, we have simple formulas for computing these exactly, this is not the case for nonlinear models. However, there are good approximations available that are computationally manageable. We discuss all the computational aspects in the next section.

14 Efficient Computations for UniLasso

14.1 Least Squares

We first discuss the computations for squared-error loss. For feature j we have to fit a univariate linear model, and then compute its LOO predictions. We are required to fit the model $\eta_j(x_j) = \beta_{0j} + \beta_j x_j$ using the data $\{x_{ij}, y_i\}_1^n$. This task is simplified if we standardized x_j to have mean zero and unit variance: $z_j = (x_j - \bar{x}_j)/s_j$, where $\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{ij}$, and $s_j^2 = \frac{1}{n} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2$. Then the least squares estimates using z_j are $(\hat{\delta}_{0j}, \hat{\delta}_j) = (\bar{y}, \frac{1}{n} \sum_{i=1}^n z_{ij} y_i)$. These are mapped back to least squares estimates for x_j via $(\hat{\beta}_{0j}, \hat{\beta}_j) = (\hat{\delta}_{0j} - \hat{\delta}_j \bar{x}_j/s_j, \hat{\delta}_j/s_j)$.

This standardized form is particularly useful for computing the LOO fits, since fits are invariant under affine transformations of the features. We have the classical formula for the LOO residual in linear regression

$$y_i - \hat{\eta}_j^{-i} = \frac{y_i - \hat{\eta}_j^i}{1 - H_{ii}}, \quad (11)$$

where H_{ii} is the i th diagonal entry of the *hat* matrix. It is straightforward to show that using the z_j gives $H_{ii} = 1/n + z_{ij}^2/n$. Now simple algebra backs out an expression for $\hat{\eta}_j^{-i}$.

These operations can be performed efficiently for all i and p simultaneously using matrix operations in **R**, without the need for any loops. In particular we can perform *Hadamard* (elementwise) arithmetic operations (eg multiplication) of like-sized matrices in single operations.

14.2 Generalized Linear Models

The discussion here also applies to the Cox model. Fitting a GLM by maximum likelihood is typically done using a Newton Algorithm. This iterative algorithm amounts to making a quadratic approximation to the negative log-likelihood at the current solution, and then solving the quadratic problem to get the updated solution. For GLMs this can be cast as *Iteratively Reweighted Least Squares* (IRLS).

Given the fitted linear predictor vector $\eta_j^{(\ell)}$ at iteration ℓ , one forms a *working* response vector $z_j^{(\ell)}$ that depends on $\eta_j^{(\ell)}$ and y and other properties of the particular GLM family, and an observation weight vector $w_j^{(\ell)}$, and fits an updated model $\eta_j^{(\ell+1)}$ by weighted least squares of $z_j^{(\ell)}$ on x_j with weights $w_j^{(\ell)}$. Usually 4 iterations are sufficient.

Once again we can fit all the univariate GLMs using matrix operations at the same time — i.e. we perform a each IRLS step simultaneously for all p univariate GLMS. The expressions are only slightly more complex than in the unweighted case.

What about the LOO fits $\hat{\eta}_j^{-i}$ for each univariate GLM? Here we make an approximation, as recommended by (Rad & Maleki 2020), which amounts to using the final weighted least squares IRLS

iteration when fitting the models $\hat{\eta}_j(x_j)$. A simple formula is available there, similar to what we got before, except a bit more detailed to accommodate observation weights. Again we can use Hadamard matrix operations to do this simultaneously for all i and j .

For the Cox model, we make a further approximation. The implied weight matrix in IRLS is not diagonal. We simply use the diagonal which leads to an approximate Newton algorithm. Note that for the Cox model the intercept is always zero.

14.3 Software in R

We have written an R function `cv.UniLasso`, whose arguments mirror `cv.glmnet`. This function does the following, currently for "binomial", "gaussian" and "cox" families. It does all the work outlined above, and performs the second stage `cv.glmnet` using the (approximate) LOO $\hat{\eta}_j^{-i}$ as features. A `cv.glmnet` object contains information about suggested values of λ , as well as a fitted `glmnet` model fit to all the data with solutions at all values of λ . This object is referenced when one predicts from a `cv.glmnet` object. In this case we make sure that the coefficients that are stored on this objects for each λ are the collapsed versions as in (10).

Hence `cv.UniLasso` returns a bona-fide `cv.glmnet` object, for which there are a number of methods for plotting, printing and making predictions.

`cv.UniLasso` has three important arguments:

1. `loo = TRUE`: by default it uses the $\hat{\eta}_j^{-i}$ as features. If this is set to `FALSE` it will use the univariate fitted values $\hat{\eta}_j^i$ instead.
2. `lower.limits = 0`: this default choice guarantees that the $\hat{\theta}_j$ are all non-negative. We recommend this in order to get a sparser and more interpretable model.
3. `standardize = FALSE`: this default is the most sensible to use, since part of the point here is to boost strong variables through their scale. Standardizing would undo that.

15 Discussion

In this paper we have introduced a novel method for sparse regression that “stacks” univariate regressions with a non-negative form of the Lasso. The procedure has interesting properties, namely that the final features weights have the same signs as the univariate coefficients and tend to be larger when the univariate coefficients are large. The test set MSE of the method tends to be similar to that of the Lasso, with substantially smaller support and lower false positive rate.

There are many possible extensions for this work. For example, rather than starting with raw features, one could apply UniLasso to derived features such as trees from random forests or gradient boosting. This is a topic for future study.

Draft versions of the UniLasso package can be found at

<https://github.com/trevorhastie/uniLasso> (R)

<https://github.com/sophial05/uni-lasso> (Python)

The Python package was written by Sophia Lu.

Disclosure Statement

The authors have no conflicts of interest to declare.

Acknowledgments

The authors would like to thank Jerome Friedman, Alden Green, Sam Gross, Lucas Janson, Mert Pilanci, Jonathan Taylor, Ryan Tibshirani and Hui Zou for helpful discussions. We especially thank Ryan for posing and proving theorem 3., and Alden for the suggestion to investigate adversarial attacks in Section 10. S.C. was supported by NSF grants DMS-2113242 and DMS-2153654. T.H was supported by grant 5R01 EB 001988-21 from the National Institutes of Health; R.T. was supported by the NIH (5R01EB001988- 16) and the NSF (19DMS1208164).

Appendix

Proof of Theorem 1

We will need the following result (see, e.g., (Talagrand 2010, Theorem A.7.1)).

Lemma 1 (Bernstein's inequality). *Let Z_1, \dots, Z_n be i.i.d. random variables such that $E(Z_1) = 0$ and $E(e^{\beta|Z_1|}) \leq 2$ for some $\beta > 0$. Then for all $t \geq 0$,*

$$\mathbb{P}\left(\left|\frac{1}{n} \sum_{i=1}^n Z_i\right| \geq t\right) \leq 2 \exp\left(-\min\left\{\frac{n\beta^2 t^2}{4}, \frac{n\beta t}{2}\right\}\right).$$

Define random variables

$$A := \frac{1}{n} \sum_{i=1}^n Y_i^2, \quad B_j := \frac{1}{n} \sum_{i=1}^n X_{i,j}^2, \quad D_j := \frac{1}{n} \sum_{i=1}^n Y_i X_{i,j}.$$

As a corollary of Bernstein's inequality, we obtain the following.

Corollary 1. *There is a positive constant C_3 depending only on C_1 and C_2 such that for all $t \geq 0$,*

$$\mathbb{P}(|A - E(A)| \geq t) \geq 2e^{-C_3 n \min\{t^2, t\}},$$

and the same bound holds for $\mathbb{P}(|B_j - E(B_j)| \geq t)$ and $\mathbb{P}(|D_j - E(D_j)| \geq t)$ for all j .

Proof. Let $Z_i := Y_i^2 - E(A)$. Then $E(Z_i) = 0$, and for any $t \geq 0$,

$$\mathbb{P}(|Z_1| \geq t) \leq \mathbb{P}(Y_1^2 \geq t) \leq C_1 e^{-C_2 t}.$$

Thus, for any $\beta \in (0, C_2)$,

$$\begin{aligned} E(e^{\beta|Z_1|}) &= \int_0^\infty \beta e^{\beta t} \mathbb{P}(|Z_1| \geq t) dt \\ &\leq \int_0^\infty \beta e^{\beta t} C_1 e^{-C_2 t} dt \\ &= \frac{C_1 \beta}{C_2 - \beta}. \end{aligned}$$

Choosing β sufficiently small makes the upper bound ≤ 2 , allowing us to apply Bernstein's inequality. A similar argument works for B_j . For D_j , take $Z_i := Y_i X_{i,j} - E(D_j)$. Then note that $E(Z_i) = 0$, and for any $t \geq 2|E(D_j)|$,

$$\begin{aligned} \mathbb{P}(|Z_1| \geq t) &\leq \mathbb{P}(|Y_1 X_{1,j}| \geq t/2) \\ &\leq \mathbb{P}(Y_1^2 \geq t/2) + \mathbb{P}(X_{1,j}^2 \geq t/2) \\ &\leq 2C_1 e^{-C_2 t/2}. \end{aligned}$$

Since $|E(D_j)|$ can be bounded above by a number that depends only on C_1 and C_2 , this shows that there are constants C_3 and C_4 depending only on C_1 and C_2 such that for all $t \geq 0$,

$$\mathbb{P}(|Z_1| \geq t) \leq C_3 e^{-C_4 t}.$$

The proof is now completed by proceeding as before. □

We also obtain the following second corollary.

Corollary 2. *There are positive constants C_5 , C_6 and C_7 depending only on C_0 , C_1 and C_2 such that for any $t \in [0, C_5]$ and any i and j , $\mathbb{P}(|\hat{\beta}_{i,j} - \beta_j| \geq t)$ and $\mathbb{P}(|\hat{\alpha}_{i,j} - \alpha_j| \geq t)$ are bounded above by $C_6 e^{-C_7 n t^2}$.*

Proof. Fix i, j . Let

$$Q_1 := \frac{1}{n-1} \sum_{k \neq i} Y_k X_{k,j}, \quad Q_2 := \frac{1}{n-1} \sum_{k \neq i} Y_k,$$

$$Q_3 := \frac{1}{n-1} \sum_{k \neq i} X_{k,j}, \quad Q_4 := \frac{1}{n-1} \sum_{k \neq i} X_{k,j}^2.$$

Using the same approach as in Corollary 1 and the fact that $n-1 \geq n/2$ (because $n \geq 2$), we deduce the concentration inequality

$$\mathbb{P}(|Q_l - \mathbb{E}(Q_l)| \geq t) \leq 2e^{-Kn \min\{t^2, t\}} \quad (12)$$

for each $1 \leq l \leq 4$. Now, note that $\mathbb{E}(Q_4) = \mathbb{E}(X_j^2)$ and $\mathbb{E}(Q_3) = \mathbb{E}(X_j)$. Moreover, recall that $\text{Var}(X_j) = \mathbb{E}(X_j^2) - (\mathbb{E}(X_j))^2 \geq C_0 > 0$. Combining these observations with the above inequality, we see that there are positive constants K_1 and K_2 such that

$$\mathbb{P}(|Q_4 - Q_3^2| < C_0/2) \leq K_1 e^{-K_2 n}. \quad (13)$$

Since

$$\hat{\beta}_{i,j} = \frac{Q_1 - Q_2 Q_3}{Q_4 - Q_3^2},$$

the inequality (13) gives

$$\begin{aligned} \mathbb{P}(|\hat{\beta}_{i,j} - \beta_j| \geq t) &= \mathbb{P}\left(\left|\frac{Q_1 - Q_2 Q_3 - (Q_4 - Q_3^2)\beta_j}{Q_4 - Q_3^2}\right| \geq t\right) \\ &\leq \mathbb{P}(|Q_1 - Q_2 Q_3 - (Q_4 - Q_3^2)\beta_j| \geq C_0 t/2) + K_1 e^{-K_2 n}. \end{aligned} \quad (14)$$

Now take any $s > 0$, and suppose that $|Q_l - \mathbb{E}(Q_l)| < s$ for $1 \leq l \leq 4$. Then we also have

$$|Q_3^2 - (\mathbb{E}(Q_3))^2| \leq |Q_3 - \mathbb{E}(Q_3)|^2 + 2|\mathbb{E}(Q_3)||Q_3 - \mathbb{E}(Q_3)| < s^2 + K_3 s,$$

where K_3 depends only on C_1 and C_2 . Similarly,

$$|Q_2 Q_3 - \mathbb{E}(Q_2)\mathbb{E}(Q_3)| < K_4 s,$$

where K_4 depends only on C_1 and C_2 . Thus, we have

$$\begin{aligned} &|(Q_1 - Q_2 Q_3 - (Q_4 - Q_3^2)\beta_j) - (\mathbb{E}(Q_1) - \mathbb{E}(Q_2)\mathbb{E}(Q_3) - (\mathbb{E}(Q_4) - (\mathbb{E}(Q_3))^2)\beta_j)| \\ &< K_5 s + K_6 s^2, \end{aligned}$$

where K_5 and K_6 depend only on C_0 , C_1 and C_2 . But

$$\begin{aligned} &\mathbb{E}(Q_1) - \mathbb{E}(Q_2)\mathbb{E}(Q_3) - (\mathbb{E}(Q_4) - (\mathbb{E}(Q_3))^2)\beta_j \\ &= \mathbb{E}(YX_j) - \mathbb{E}(Y)\mathbb{E}(X_j) - (\mathbb{E}(X_j^2) - (\mathbb{E}(X_j))^2) \frac{\mathbb{E}(YX_j) - \mathbb{E}(Y)\mathbb{E}(X_j)}{\mathbb{E}(X_j) - (\mathbb{E}(X_j))^2} = 0. \end{aligned}$$

Plugging this into the previous display shows that

$$\mathbb{P}(|Q_1 - Q_2 Q_3 - (Q_4 - Q_3^2)\beta_j| \geq K_5 s + K_6 s^2) \leq \sum_{l=1}^4 \mathbb{P}(|Q_l - \mathbb{E}(Q_l)| \geq s) \quad (15)$$

Choosing s to solve $K_5 s + K_6 s^2 = C_0 t/2$, and combining equations (12), (14) and (15) yields the proof of the claimed concentration inequality for $\hat{\beta}_{i,j}$. The proof for $\hat{\alpha}_{i,j}$ is similar. We omit the details. \square

We are now ready to prove Theorem 1.

Proof of Theorem 1. Throughout this proof, K_1, K_2, \dots will denote positive constants that may depend only on C_0, C_1, C_2, θ and $|S|$ and nothing else, whose values may change from line to line. Also, we will denote by $o(1)$ any random variable Z such that

$$\mathbb{P}(|Z| \geq t) \leq K_1 n e^{-K_2 n t^2}$$

for all $t \in [0, K_3]$ (for some K_1, K_2, K_3 according to the above convention), and by $O(1)$ any random variable Z such that $|Z - K_1| = o(1)$ for some K_1 . Define

$$\tilde{L}(\theta_0, \theta_1, \dots, \theta_p) := \frac{1}{n} \sum_{i=1}^n (Y_i - \theta_0 - \theta_1(\hat{Y}_{i,1} - \hat{\alpha}_1) - \dots - \theta_p(\hat{Y}_{i,p} - \hat{\alpha}_p))^2 + \lambda \sum_{j=1}^p \theta_j.$$

Note that $(\hat{\theta}'_0, \hat{\theta}_1, \dots, \hat{\theta}_p)$ minimizes \tilde{L} under $\hat{\theta}_j \geq 0$ for all $1 \leq j \leq p$ if and only if $(\hat{\theta}_0, \hat{\theta}_1, \dots, \hat{\theta}_p)$ minimizes L under the same constraint, where

$$\hat{\theta}_0 = \hat{\theta}'_0 - \sum_{j=1}^p \hat{\theta}_j \hat{\alpha}_j.$$

Note that this means $\hat{\gamma}_0 = \hat{\theta}'_0$. We will henceforth consider this modified optimization problem. Note that for each $1 \leq j \leq p$, and for any $\theta_0 \in \mathbb{R}$ and $\theta_1, \dots, \theta_p \geq 0$,

$$\begin{aligned} \tilde{L}_j &:= \frac{\partial \tilde{L}}{\partial \theta_j} \\ &= -\frac{2}{n} \sum_{i=1}^n (Y_i - \theta_0 - \theta_1(\hat{Y}_{i,1} - \hat{\alpha}_1) - \dots - \theta_p(\hat{Y}_{i,p} - \hat{\alpha}_p))(\hat{Y}_{i,j} - \hat{\alpha}_j) + \lambda. \end{aligned} \quad (16)$$

By an application of the Cauchy-Schwarz inequality, we get

$$\begin{aligned} &\left| \frac{1}{n} \sum_{i=1}^n (Y_i - \theta_0 - \theta_1(\hat{Y}_{i,1} - \hat{\alpha}_1) - \dots - \theta_p(\hat{Y}_{i,p} - \hat{\alpha}_p))(\hat{Y}_{i,j} - \hat{\alpha}_j) \right| \\ &\leq \left(\frac{1}{n} \sum_{i=1}^n (Y_i - \theta_0 - \theta_1(\hat{Y}_{i,1} - \hat{\alpha}_1) - \dots - \theta_p(\hat{Y}_{i,p} - \hat{\alpha}_p))^2 \right)^{1/2} \left(\frac{1}{n} \sum_{i=1}^n (\hat{Y}_{i,j} - \hat{\alpha}_j)^2 \right)^{1/2} \\ &\leq \sqrt{\tilde{L}(\theta_0, \dots, \theta_p)} \left(\frac{1}{n} \sum_{i=1}^n (\hat{Y}_{i,j} - \hat{\alpha}_j)^2 \right)^{1/2}. \end{aligned}$$

Now, note that

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n (\hat{Y}_{i,j} - \hat{\alpha}_j)^2 &\leq \frac{2}{n} \sum_{i=1}^n (\hat{Y}_{i,j} - \hat{\alpha}_{i,j})^2 + \frac{2}{n} \sum_{i=1}^n (\hat{\alpha}_{i,j} - \hat{\alpha}_j)^2 \\ &\leq \frac{2}{n} \sum_{i=1}^n \hat{\beta}_{i,j}^2 X_{i,j}^2 + \frac{4}{n} \sum_{i=1}^n (\hat{\alpha}_{i,j} - \alpha_j)^2 + 4(\hat{\alpha}_j - \alpha_j)^2 \\ &\leq \left(\frac{2}{n} \sum_{i=1}^n X_{i,j}^2 \right) \max_{1 \leq i \leq n} \hat{\beta}_{i,j}^2 + 4 \max\{(\hat{\alpha}_j - \alpha_j)^2, \max_{1 \leq i \leq n} (\hat{\alpha}_{i,j} - \alpha_j)^2\}. \end{aligned}$$

Let M_j^β and M_j^α denote the two maxima on the right. Combining the previous three displays, we get that for $\theta_0 \in \mathbb{R}$ and $\theta_1, \dots, \theta_p \geq 0$,

$$\tilde{L}_j(\theta_0, \dots, \theta_p) \geq \lambda - \sqrt{\tilde{L}(\theta_0, \dots, \theta_p)(2B_j M_j^\beta + 4M_j^\alpha)}. \quad (17)$$

Take any $j \in \{1, \dots, p\} \setminus S$. Suppose that $\hat{\gamma}_j \neq 0$. Then $\hat{\theta}_j > 0$. This implies that

$$\tilde{L}_j(\hat{\gamma}_0, \hat{\theta}_1, \dots, \hat{\theta}_p) = 0,$$

because otherwise, we can slightly perturb $\hat{\theta}_j$ while maintaining the nonnegativity constraint and decreasing the value of L . By inequality (17), this implies that if $\hat{\theta}_j > 0$, then we must have

$$\tilde{L}(\hat{\gamma}_0, \hat{\theta}_1, \dots, \hat{\theta}_p)(2B_j M_j^\beta + 4M_j^\alpha) \geq \lambda^2.$$

But note that

$$\tilde{L}(\hat{\gamma}_0, \hat{\theta}_1, \dots, \hat{\theta}_p) \leq \tilde{L}(0, 0, \dots, 0) = A.$$

Thus, we conclude that if $\hat{\theta}_j > 0$, then

$$\lambda^2 \leq A(2B_j M_j^\beta + 4M_j^\alpha).$$

Thus,

$$\begin{aligned} \mathbb{P}(\hat{\theta}_j > 0) &\leq \mathbb{P}(A > 2E(A)) + \mathbb{P}(B_j > 2E(B_j)) \\ &\quad + \mathbb{P}\left(M_j^\beta \geq \frac{\lambda^2}{16E(A)E(B_j)}\right) + \mathbb{P}\left(M_j^\alpha \geq \frac{\lambda^2}{16E(A)}\right). \end{aligned}$$

By Corollary 1, the first two probabilities on the right are bounded above by $K_1 e^{-K_2 n}$. Next, note that if

$$K_3 |\beta_j| \leq \lambda \leq K_4, \quad (18)$$

for some sufficiently small K_3 and K_4 , then by Corollary 2,

$$\begin{aligned} \mathbb{P}\left(M_j^\beta \geq \frac{\lambda^2}{16E(A)E(B_j)}\right) &\leq n\mathbb{P}\left(\hat{\beta}_{1,j}^2 \geq \frac{\lambda^2}{16E(A)E(B_j)}\right) \\ &\leq n\mathbb{P}(|\hat{\beta}_{1,j} - \beta_j| \geq K_5 \lambda) \leq K_6 n e^{-K_7 n \lambda^2}. \end{aligned}$$

Similarly, the same bound holds for the tail probability of M_j^α . Combining, we get that under the condition (18),

$$\mathbb{P}(\hat{\theta}_j > 0) \leq K_8 n e^{-K_9 n \lambda^2},$$

and thus,

$$\mathbb{P}(\hat{\gamma}_j \neq 0 \text{ for some } j \notin S \cup \{0\}) \leq \mathbb{P}(E^c) \leq K_8 p n e^{-K_9 n \lambda^2}, \quad (19)$$

where E denotes the event that $\hat{\theta}_j = 0$ for all $j \notin S \cup \{0\}$. Suppose that E happens. Take any $k \in S$. If $\hat{\theta}_k \neq 0$, then $\tilde{L}_k(\hat{\gamma}_0, \hat{\theta}_1, \dots, \hat{\theta}_p) = 0$, whereas if $\hat{\theta}_k = 0$, then $\tilde{L}_k(\hat{\gamma}_0, \hat{\theta}_1, \dots, \hat{\theta}_p) \geq 0$ and $(\gamma_k - \hat{\gamma}_k)/\beta_k = \gamma_k/\beta_k > 0$. Thus, in either case, we have

$$\frac{\gamma_k - \hat{\gamma}_k}{\beta_k} \tilde{L}_k(\hat{\gamma}_0, \hat{\theta}_1, \dots, \hat{\theta}_p) \geq 0.$$

By the formula (16) for \tilde{L}_j , this shows that

$$\begin{aligned} \frac{\lambda(\gamma_k - \hat{\gamma}_k)}{2\beta_k} &\geq \left(\frac{\gamma_k - \hat{\gamma}_k}{\beta_k}\right) \frac{1}{n} \sum_{i=1}^n \left(Y_i - \hat{\gamma}_0 - \sum_{j \in S} \hat{\theta}_j (\hat{Y}_{i,j} - \hat{\alpha}_j)\right) (\hat{Y}_{i,k} - \hat{\alpha}_k) \\ &= \frac{\gamma_k - \hat{\gamma}_k}{n\beta_k} \sum_{i=1}^n \left[\gamma_0 - \hat{\gamma}_0 + \sum_{j \in S} (\gamma_j X_{i,j} - \hat{\theta}_j (\hat{Y}_{i,j} - \hat{\alpha}_j))\right] (\hat{Y}_{i,k} - \hat{\alpha}_k) \\ &\quad + \frac{\gamma_k - \hat{\gamma}_k}{n\beta_k} \sum_{i=1}^n \epsilon_i (\hat{Y}_{i,k} - \hat{\alpha}_k). \end{aligned} \quad (20)$$

Define

$$M := \max_{1 \leq i \leq n, j \in S} (|\hat{\beta}_{i,j} - \beta_j| + |\hat{\beta}_j - \beta_j|).$$

Then

$$|(\hat{Y}_{i,k} - \hat{\alpha}_k) - \beta_k X_{i,k}| = |\hat{\beta}_{i,k} - \beta_k| |X_{i,k}| \leq M |X_{i,k}|, \quad (21)$$

and since $\hat{\gamma}_j = \hat{\theta}_j \hat{\beta}_j$,

$$\begin{aligned} |\hat{\theta}_j(\hat{Y}_{i,j} - \hat{\alpha}_j) - \hat{\gamma}_j X_{i,j}| &= |\hat{\theta}_j \hat{\beta}_{i,j} X_{i,j} - \hat{\theta}_j \hat{\beta}_j X_{i,j}| \\ &= |\hat{\beta}_{i,j} - \hat{\beta}_j| |\hat{\theta}_j X_{i,j}| \leq M |\hat{\theta}_j| |X_{i,j}|. \end{aligned} \quad (22)$$

By (21) and (22), we have

$$\begin{aligned} &|\hat{\theta}_j(\hat{Y}_{i,j} - \hat{\alpha}_j)(\hat{Y}_{i,k} - \hat{\alpha}_k) - \hat{\gamma}_j \beta_k X_{i,j} X_{i,k}| \\ &\leq |\hat{\theta}_j(\hat{Y}_{i,j} - \hat{\alpha}_j) - \hat{\gamma}_j X_{i,j}| |\hat{Y}_{i,k} - \hat{\alpha}_k| + |\hat{\gamma}_j X_{i,j}| |(\hat{Y}_{i,k} - \hat{\alpha}_k) - \beta_k X_{i,k}| \\ &\leq M |\hat{\theta}_j| |X_{i,j}| (|\beta_k X_{i,k}| + M |X_{i,k}|) + M |\hat{\gamma}_j X_{i,j}| |X_{i,k}| \end{aligned} \quad (23)$$

Let $S' := S \cup \{0\}$. Define $X_0 \equiv 1$ and $X_{i,0} \equiv 1$ for $1 \leq i \leq n$. Combining equations (20), (21), (22) and (23), we get

$$\begin{aligned} &\frac{\lambda(\gamma_k - \hat{\gamma}_k)}{2\beta_k} - \left[(\gamma_k - \hat{\gamma}_k) \sum_{j \in S'} (\gamma_j - \hat{\gamma}_j) \left(\frac{1}{n} \sum_{i=1}^n X_{i,j} X_{i,k} \right) + (\gamma_k - \hat{\gamma}_k) \frac{1}{n} \sum_{i=1}^n \epsilon_i X_{i,k} \right] \\ &\geq -K_5(M + M^2) \left| \frac{\gamma_k - \hat{\gamma}_k}{n\beta_k} \right| \sum_{i=1}^n \left(|\gamma_0 - \hat{\gamma}_0| |X_{i,k}| \right. \\ &\quad \left. + \sum_{j \in S} (1 + |\hat{\theta}_j| + |\hat{\gamma}_j|) (|X_{i,j}| + |\epsilon_i|) |X_{i,k}| \right). \end{aligned} \quad (24)$$

Next note that $\tilde{L}_0(\hat{\gamma}_0, \hat{\theta}_1, \dots, \hat{\theta}_p) = 0$, where \tilde{L}_0 denotes the derivative of \tilde{L} is the zeroth coordinate. This means that

$$\begin{aligned} 0 &= \frac{1}{n} \sum_{i=1}^n \left(Y_i - \hat{\gamma}_0 - \sum_{j \in S} \hat{\theta}_j (\hat{Y}_{i,j} - \hat{\alpha}_j) \right) \\ &= \frac{1}{n} \sum_{i=1}^n \left[\gamma_0 - \hat{\gamma}_0 + \sum_{j \in S} (\gamma_j X_{i,j} - \hat{\theta}_j (\hat{Y}_{i,j} - \hat{\alpha}_j)) \right] + \frac{1}{n} \sum_{i=1}^n \epsilon_i. \end{aligned}$$

Proceeding as before (and recalling that $X_{i,0} = 1$), this gives

$$\begin{aligned} &(\gamma_0 - \hat{\gamma}_0) \sum_{j \in S'} (\gamma_j - \hat{\gamma}_j) \left(\frac{1}{n} \sum_{i=1}^n X_{i,j} X_{i,0} \right) + (\gamma_0 - \hat{\gamma}_0) \frac{1}{n} \sum_{i=1}^n \epsilon_i X_{i,0} \\ &\leq \frac{M |\gamma_0 - \hat{\gamma}_0|}{n} \sum_{j \in S} |\hat{\theta}_j X_{i,j}|. \end{aligned} \quad (25)$$

Let $\Delta := \max_{j \in S'} |\hat{\gamma}_j - \gamma_j|$. For each $j \in S$, let $\theta_j := \gamma_j / \beta_j$. Recall the quantity M_2 from the statement of the theorem. Note that if

$$M < \frac{1}{2} M_2, \quad (26)$$

we get that for each $j \in S$,

$$\begin{aligned} |\hat{\theta}_j - \theta_j| &= \left| \frac{\hat{\gamma}_j}{\hat{\beta}_j} - \frac{\gamma_j}{\beta_j} \right| \\ &\leq \frac{|\hat{\gamma}_j - \gamma_j|}{|\hat{\beta}_j|} + \gamma_j \left| \frac{1}{\hat{\beta}_j} - \frac{1}{\beta_j} \right| \\ &\leq K_6(\Delta + M). \end{aligned}$$

Combining this observation with the inequalities (24) and (25), and summing over $k \in S'$, we get that

$$\sum_{j,k \in S'} \hat{\sigma}_{j,k} (\gamma_j - \hat{\gamma}_j) (\gamma_k - \hat{\gamma}_k) \leq K_7 \lambda \Delta + K_8 \Delta Q_1 + K_9 (M + M^2) \Delta (\Delta + M) Q_2, \quad (27)$$

where

$$\hat{\sigma}_{j,k} := \frac{1}{n} \sum_{i=1}^n X_{i,j} X_{i,k}, \quad Q_1 := \max_{j \in S'} \left| \frac{1}{n} \sum_{i=1}^n \epsilon_i X_{i,j} \right|, \quad Q_2 := \max_{j \in S'} \frac{1}{n} \sum_{i=1}^n X_{i,j}^2.$$

Let $\hat{\eta}$ be the smallest eigenvalue of the positive semidefinite matrix $(\hat{\sigma}_{j,k})_{j,k \in S'}$, so that

$$\sum_{j,k \in S'} \hat{\sigma}_{j,k}(\gamma_j - \hat{\gamma}_j)(\gamma_k - \hat{\gamma}_k) \geq \hat{\eta} \sum_{j \in S'} (\hat{\gamma}_j - \gamma_j)^2 \geq \hat{\eta} \Delta^2.$$

Combining this with equation (27) and rearranging, we get

$$(\hat{\eta} - K_9(M + M^2)Q_2)\Delta^2 \leq (K_7\lambda + K_8Q_1 + K_9(M + M^2)MQ_2)\Delta$$

If the coefficient of Δ^2 on the left is positive, this gives

$$\Delta \leq \frac{K_7\lambda + K_8Q_1 + K_9(M + M^2)MQ_2}{\hat{\eta} - K_9(M + M^2)Q_2}.$$

Now, using Bernstein's inequality, it is easy to show that $Q_1 = o(1)$ and $Q_2 = O(1)$ (following the conventions introduced at the beginning of the proof). By Corollary 2, we know that $M = o(1)$. Again by Bernstein's inequality, $\hat{\sigma}_{j,k} = E(X_j X_k) + o(1)$ for each $j, k \in S$. From this, it is easy to deduce via standard matrix inequalities that $\hat{\eta} = \eta_0 + o(1)$, where η_0 is the minimum eigenvalue of $(E(X_j X_k))_{j,k \in S'}$. Recalling that $X_0 = 1$, we see that this is equal to the minimum eigenvalue η of the covariance matrix of $(X_j)_{j \in S}$. Combining all of these, it is now straightforward that under the condition (18),

$$\mathbb{P}(E \cap \{\Delta > K_{10}\lambda\}) \leq K_{11}n e^{-K_{12}n\lambda^2}.$$

Together with equation (19), this completes the proof. \square

Proof of Theorem 2. Note that since $Y = \gamma_0 + \sum_{j \in S} \gamma_j X_j + \epsilon$, and ϵ is independent of the X_j 's, we have

$$\begin{aligned} \beta_j &= \frac{\text{Cov}(Y, X_j)}{\text{Var}(X_j)} \\ &= \frac{\sum_{k \in S} \gamma_k \text{Cov}(X_k, X_j)}{\text{Var}(X_j)} \\ &= \gamma_j + \sum_{k \in S \setminus \{j\}} \gamma_k \frac{\text{Cov}(X_k, X_j)}{\text{Var}(X_j)} = \gamma_j + \sum_{k \in S \setminus \{j\}} \gamma_k \delta_{k,j}. \end{aligned}$$

Since $\gamma_j \neq 0$, we may divide throughout by γ_j and get

$$\frac{\beta_j}{\gamma_j} = 1 + \sum_{k \in S \setminus \{j\}} \frac{\gamma_k}{\gamma_j} \delta_{k,j}.$$

Now, by assumption, $\delta_{k,j} \geq 0$ whenever $\gamma_k/\gamma_j > 0$, and $\delta_{k,j} \leq 0$ whenever $\gamma_k/\gamma_j < 0$. Thus,

$$\sum_{k \in S \setminus \{j\}} \frac{\gamma_k}{\gamma_j} \delta_{k,j} \geq 0.$$

This shows that $\beta_j/\gamma_j \geq 1$. In particular, β_j is nonzero and has the same sign as γ_j . \square

Proof. of Proposition 1.

If you substitute $\eta_j^i = \hat{\beta}_{0j} + \hat{\beta}_j x_{ij}$ in the objective in (6) you get

$$\sum_{i=1}^n (y_i - (\theta_0 + \sum_{j=1}^p \hat{\beta}_{0j}) - \sum_{j=1}^p x_{ij} \hat{\beta}_j \theta_j)^2 + \sum_{j=1}^p |\theta_j|$$

Now you change variables to $\gamma_0 = \theta_0 + \sum_{j=1}^p \hat{\beta}_{0j}$ and $\gamma_j = \hat{\beta}_j \theta_j$.

The result is immediate, since

$$|\theta_j| = \frac{|\gamma_j|}{|\hat{\beta}_j|}$$

and

$$\{\text{sign}(\gamma_j) = \text{sign}(\hat{\beta}_j) \forall j\} \equiv \{\theta_j \geq 0 \forall j\}$$

Proof of Theorem 3

Proof. Note that

$$\begin{aligned} \beta_j &= \frac{\text{Cov}(Y, X_j)}{\text{Var}(X_j)} \\ &= \sum_{k \in S} \gamma_k \frac{\text{Cov}(X_k, X_j)}{\text{Var}(X_j)} \\ &= \sum_{k \in S} \gamma_k \delta_{kj}. \end{aligned}$$

Thus,

$$\begin{aligned} \frac{\beta_j}{\gamma_j} &= \sum_{k \in S} \frac{\gamma_k}{\gamma_j} \delta_{kj} \\ &= \sum_{k \in A_j} \left| \frac{\gamma_k}{\gamma_j} \delta_{kj} \right| - \sum_{k \notin A_j} \left| \frac{\gamma_k}{\gamma_j} \delta_{kj} \right| \\ &= \frac{\sum_{k \in A_j} |\gamma_k \delta_{kj}| - \sum_{k \notin A_j} |\gamma_k \delta_{kj}|}{|\gamma_j|}. \end{aligned}$$

Since $\beta_j \gamma_j \geq 0$ if and only if $\beta_j / \gamma_j \geq 0$, this proves the claim. \square

References

- Breiman, L. (1995), ‘Better subset selection using the non-negative garotte’, *Technometrics* **37**, 738–754.
- Breiman, L. (1996), ‘Stacked regressions’, *Machine Learning* **24**, 51–64.
- Candes, E. J., Wakin, M. B. & Boyd, S. P. (2008), ‘Enhancing sparsity by reweighted l1 minimization’, *Journal of Fourier Analysis and Applications* **14**(5), 877–905.
- Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2004), ‘Least angle regression’, *Annals of Statistics* **32**(2), 407–499. With discussion, and a rejoinder by the authors.
- Hastie, T., Tibshirani, R. & Wainwright, M. (2015), *Statistical Learning with Sparsity: The Lasso and Generalizations*, Chapman & Hall/CRC.
- Huang, J., Ma, S. & Zhang, C. (2008), ‘Adaptive lasso for sparse high-dimensional regression models’, *Statistica Sinica* **18**(4), 1603–1618.
- Khan, J., Wei, J., Ringner, M., Saal, L., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C., Peterson, C., & Meltzer, P. (2001), ‘Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks’, *Nature Medicine* **7**, 673–679.
- Kong, X. & Ge, Z. (2023), ‘Adversarial attacks on regression systems via gradient optimization’, *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **53**(12), 7827–7839.
- Mazumder, R., Friedman, J. & Hastie, T. (2011), ‘Sparsenet: Coordinate descent with non-convex penalties’, *J. Amer. Statist. Assoc.* **106**.
- Meinshausen, N. (2012), ‘Sign-constrained least squares estimation for high-dimensional regression’, *Electronic Journal of Statistics* **7**.

- Rad, K. R. & Maleki, A. (2020), ‘A scalable estimate of the out-of-sample prediction error via approximate leave-one-out cross-validation’, *Journal of the Royal Statistical Society Series B: Statistical Methodology* **82**(4), 965–996.
URL: <https://doi.org/10.1111/rssb.12374>
- Talagrand, M. (2010), *Mean field models for spin glasses: Volume I: Basic examples*, Springer Science & Business Media.
- Wolpert, D. (1992), ‘Stacked generalization’, *Neural Networks* **5**, 241–259.
- Zhang, C.-H. (2010), ‘Nearly unbiased variable selection under minimax concave penalty’, *The Annals of Statistics* **38**(2), 894 – 942.
URL: <https://doi.org/10.1214/09-AOS729>
- Zou, H. (2006), ‘The adaptive lasso and its oracle properties’, *Journal of the American Statistical Association* **101**, 1418–1429.
- Zou, H. & Hastie, T. (2005), ‘Regularization and variable selection via the elastic net’, *Journal of the Royal Statistical Society Series B.* **67**(2), 301–320.