Risk estimation for high-dimensional lasso regression

Darren Homrighausen[a] and Daniel J. McDonald[b]

[a]Department of Statistics, Colorado State University, Fort Collins, Colorado, USA;
[b]Department of Statistics, Indiana University, Bloomington, Indiana, USA

ARTICLE HISTORY
Compiled February 10, 2017

ABSTRACT
High-dimensional predictive models, those with more measurements than observations, require regularization to be well defined, perform well empirically, and possess theoretical guarantees. The amount of regularization, often determined by tuning parameters, is integral to achieving good performance. One can choose the tuning parameter in a variety of ways, such as through resampling methods or generalized information criteria. However, the theory supporting many regularized procedures relies on an estimate for the variance parameter, which is complicated in high dimensions. We develop a suite of information criteria for choosing the tuning parameter in lasso regression by leveraging the literature on high-dimensional variance estimation. We derive intuition showing that existing information-theoretic approaches work poorly in this setting. We compare our risk estimators to existing methods with an extensive simulation and derive some theoretical justification. We find that our new estimators perform well across a wide range of simulation conditions and evaluation criteria.

KEYWORDS
Model selection; tuning parameter selection; prediction; variance estimation

1. Introduction

Suppose we are given a data set, $Z_1, \ldots, Z_n$, of paired observations including a covariate $X_i \in \mathbb{R}^p$ and its associated response $Y_i \in \mathbb{R}$ such that $Z_i^\top = (X_i^\top, Y_i)$. Concatenating the covariates row-wise, we obtain the design matrix $\mathbb{X} = [X_1, \ldots, X_n]^\top \in \mathbb{R}^{n \times p}$. We assume that the relationship between the covariate and response is of the form

$$Y = \mathbb{X}\beta_* + \epsilon, \tag{1}$$

where $\epsilon \sim (0, \sigma^2 I)$, meaning the entries of ϵ are mean zero with uncorrelated components each having variance σ^2.

When $p > n$, estimation of the linear model requires some structural assumptions on β_* for learning algorithms to possess theoretical guarantees. A common approach in this scenario is to assume $\|\beta_*\|_q$ is small for some $q \geq 0$ and try to estimate β_* via

CONTACT D. J. McDonald. Email: dajmcdon@indiana.edu

penalized least squares. We will focus mainly on the lasso

$$\widehat{\beta}(\lambda) = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n} \|Y - \mathbb{X}\beta\|_2^2 + \lambda \|\beta\|_1 , \tag{2}$$

where $\lambda \geq 0$ is a tuning parameter and $\|\cdot\|_2$ and $\|\cdot\|_1$ are the ℓ_2- (Euclidean) and ℓ_1-norms respectively. Similar M-estimators with different penalties include, among others, ridge regression, the group lasso [1], and the smoothly clipped absolute deviation penalty [SCAD, 2]. Though the focus of this paper is on lasso, we will occasionally also reference ridge regression,

$$\begin{aligned} \widehat{\beta}_{ridge}(\lambda) &= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{n} \|Y - \mathbb{X}\beta\|_2^2 + \lambda \|\beta\|_2^2 \\ &= (\mathbb{X}^\top \mathbb{X} + \lambda I_p)^{-1} \mathbb{X}^\top Y, \end{aligned}$$

because it has a closed form which can provide intuition.

For lasso, by convexity there is always at least one solution to equation (2), although if $\operatorname{rank}(\mathbb{X}) < p$, there may be multiple minimizers [see 3, for details]. In this case, we refer to 'the' solution as the outcome of the particular minimization technique used [e.g. LARS, 4]. For ridge regression, a unique solution always exists for $\lambda > 0$, although, for λ small enough, numerical issues may intercede. We will also consider some modifications to equation (2) which attempt to eliminate the influence of tuning parameters (see Section 4.2 for a more detailed description).

The theoretical optimality properties that exist in the literature for penalized regression rely on appropriate tuning parameter selection. Under restrictions on the design matrix $\mathbb{X}$, the distribution of ϵ, and the sparsity pattern of β_*, [5] shows that, as long as the number of nonzero entries in β_* does not increase too quickly, the probability of making prediction errors with magnitude larger than $\sigma^2 \log(p)/n$ goes to zero if $\lambda_n = a\sigma\sqrt{\log(p)/n}$ for some constant a. Likewise, deviations in the distance between $\widehat{\beta}(\lambda)$ and β_* of order larger than $\sigma\sqrt{\log(p)/n}$ have small probability. While theoretical results of this type provide comfort that a data analyst's procedure will eventually perform well given sufficient data, they require the optimal λ_n which depends on unknown quantities such as σ^2, the noise distribution, and other constants.

In practice, many methods for empirically choosing λ given a fixed dataset have been proposed. These methods can be lumped into three broad categories: (1) generalized information criteria like AIC or BIC, (2) resampling procedures such as cross-validation or the bootstrap, and (3) reformulations of the lasso optimization problem (e.g. scaled sparse regression or $\sqrt{\text{lasso}}$).[1] In order to evaluate these approaches, we must be explicit as to the properties we desire in our final estimator: low prediction risk, parameter estimation consistency, correct model selection, or simply accurate estimates of the prediction risk.

The aim of this paper is to evaluate tuning parameter selection procedures for high-dimensional lasso regression. To this end we (1) introduce a suite of novel risk estimation methods that are simple to compute and perform well empirically, (2) contrast these new risk estimation methods with existing, superficially similar GIC-based methods, and, lastly, (3) provide a comprehensive simulation study over a broad range of data generating scenarios and estimation goals which compares our procedure to

[1] There is some overlap between these categories. For example, generalized cross-validation can be thought of as either a resampling procedure or an information criterion.

existing methods. This investigation both justifies our proposal and reveals deficiencies for current high-dimensional approaches while also suggesting interesting research directions, particularly the relationship between risk estimation and high-dimensional variance estimation.

In Section 2, we discuss two broad categories of procedures for tuning parameter selection: cross-validation and generalized information criteria. We demonstrate that there is a significant difference between using generalized information criteria in the low-dimensional ($p < n$) versus the high-dimentional ($p > n$) regimes. Section 3 motivates and introduces our proposed modification to Stein's unbiased risk estimation using plug-in estimators for σ^2 and the degrees-of-freedom for the lasso. It also discusses the different versions of modern high-dimensional variance estimators which we consider. In Section 4, we present a comprehensive simulation comparing our proposal with some existing alternatives. We focus on the performance of the lasso, but we include scaled-sparse regression, $\sqrt{\text{lasso}}$, and SCAD for comparison. We also demonstrate the methods on a genetics dataset. Finally, Section 5 gives a theoretical result, showing that under standard assumptions our proposed risk estimator converges to the true prediction risk at the parametric rate. Section 6 summarizes our recommendations and suggest possible avenues for further research.

Notation: For any vector $\beta \in \mathbb{R}^p$, we denote $\mathcal{S} = \mathcal{S}(\beta) = \{j : \beta_j \neq 0\}$ and $\mathbb{X}_{\mathcal{S}(\beta)}$ to be the columns of the design matrix selected by β. We write $\mathcal{S}_* = \mathcal{S}(\beta_*)$ and $s_* = |\mathcal{S}_*|$. Also, for any square matrix H, define the trace of H, $tr(H)$, to be the sum of the diagonal entries. Define the squared ℓ_2-prediction risk of a coefficient vector β to be

$$\mathrm{R}_\beta = n^{-1}\mathbb{E}\left\|\mathbb{X}\beta - \mathbb{X}\beta_*\right\|_2^2, \tag{3}$$

where the expectation is over the data $Z_1, \ldots, Z_n$. Likewise, we define the training error to be

$$\widehat{\text{train}}_\beta = n^{-1}\left\|\mathbb{X}\beta - Y\right\|_2^2.$$

Throughout this paper, if a procedure β is indexed by a tuning parameter λ, we will write, for example, $\widehat{\text{train}}_{\beta(\lambda)} \equiv \widehat{\text{train}}_\lambda$.

2. Existing tuning parameter selection methods

In this section, we discuss existing procedures for tuning parameter selection for lasso regression. In the context of regularized regression, risk estimation and tuning parameter selection are often used interchangeably because any risk estimator can be used to select tuning parameter(s). However, it is important for our exposition to belabor the distinction for two reasons: (1) not all tuning parameter selection procedures produce an estimate of the prediction risk, and (2) we may wish to evaluate the quality of the selection procedure by comparing model selection accuracy or parameter consistency, metrics which don't require a risk estimate anyway. That is, we may ask if $\sqrt{\text{lasso}}$, a tuning-free method which does not estimate the prediction risk, produces better estimates of β_* than the lasso with λ selected by cross-validation. As a preview of our results in Section 4.4, the answer to this question is generally no, but if we use GCV to select λ instead, then this conclusion is reversed. This section introduces existing tuning parameter selection procedures, some of which estimate the prediction risk—cross-validation, Stein's unbiased risk estimation (SURE), and information

criteria—while others do not.

2.1. *Cross-validation*

Frequently [for example 6–8], the recommended technique for selecting λ is through *K-fold cross-validation* (CV). Letting $V_n = \{v_1, \ldots, v_K\}$ be a partition of $\{1, \ldots, n\}$

$$CV(\lambda; V_n) = \frac{1}{K} \sum_{v \in V_n} \frac{1}{|v|} \sum_{r \in v} \left(Y_r - X_r^\top \widehat{\beta}^{(v)}(\lambda) \right)^2,$$

where $\widehat{\beta}^{(v)}(\lambda)$ is the lasso estimator in equation (2) with the observations in the validation set v removed, and $|v|$ indicates the cardinality of the set v. We define $\widehat{\lambda}_{CV} = \operatorname{argmin}_\lambda CV(\lambda; V_n)$. Common choices for K are $K = 10$ or $K = n$. Cross-validation was shown to perform correct model selection and lead to good prediction risk [9].

Several adaptations of cross-validation for use with the lasso have been proposed. One such method is Modified Cross-Validation [MCV, 10] which seeks to correct for a bias in CV induced by the lasso penalty. Generalized cross-validation [11, GCV] is a much older modification of cross-validation with some computational benefits. It can also be viewed as an information criterion, so we discuss it further in the next section.

2.2. *Generalized information criteria*

A common alternative to cross-validation is to minimize a generalized information criterion (GIC). Define the *degrees of freedom* [12] of the prediction $\widehat{Y} = \mathbb{X}\beta \in \mathbb{R}^n$ to be

$$\mathrm{df} = \frac{1}{\sigma^2} \sum_{i=1}^{n} \mathrm{Cov}(\widehat{Y}_i, Y_i),$$

where $\mathrm{Cov}(\widehat{Y}_i, Y_i) = \mathbb{E}\left[(\widehat{Y}_i - \mathbb{E}\widehat{Y}_i)(Y_i - \mathbb{E}Y_i) \right]$.

Referring to equation (1), if σ^2 is unknown and ϵ is Gaussian, then a GIC takes the form

$$\mathrm{info}(C_n, g) = \log\left(\widehat{\mathrm{train}}_\beta \right) + C_n \, g(\mathrm{df}), \tag{4}$$

where C_n depends only on n, and $g : [0, \infty) \to \mathbb{R}$ is a fixed function. This GIC form is frequently suggested in the literature for choosing λ in the lasso problem [for example 2, 13–16], with df replaced by an estimator $\widehat{\mathrm{df}}$. We defer discussion of how to form $\widehat{\mathrm{df}}$ for the lasso to Section 3. The choices $C_n = 2/n$ or $C_n = \log(n)/n$ with $g(x) = x$ are commonly referred to as AIC and BIC, respectively. Additionally, generalized cross-validation is defined as

$$\mathrm{GCV} = \frac{\widehat{\mathrm{train}}_\beta}{(1 - \mathrm{df}/n)^2}. \tag{5}$$

Written on the log scale, GCV takes the form of equation (4) with $g(x) = \log(1 - x/n)$ and $C_n = -2/n$.

While GIC-based tuning parameter selection has enjoyed good theoretical and empirical success in a broad range of applications, classical asymptotic arguments underlying GIC apply only for p fixed and rely on maximum likelihood estimates (or Bayesian posteriors) for all parameters including σ^2. More recent investigations have explored theoretical regimes in which p is allowed to increase, but the constraint $p < n$ is still enforced. [17] shows that the correct model is selected asymptotically even if $p \to \infty$ as long as $p/n \to 0$. Additionally, [16] investigates a variety GIC-based methods under increasing p, but again restricted to the case $p < n$.

Theoretical support for GIC breaks down in the high-dimensional setting. The most serious issue is that $\text{info}(C_n, g)$ from equation (4) is unusable without modification if $n < p$ because it is possible to achieve $\widehat{\text{train}}_\beta = 0$ and hence $\log(\widehat{\text{train}}_\beta) = -\infty$. Therefore, as $\lambda \to 0$, $\text{info}(C_n, g)$ will approach $-\infty$ unless $g(\text{df}) \to \infty$ faster, and $\lambda = 0$ will always be selected. Simply forcing $\lambda > \epsilon$ for some small positive ϵ often fails to remedy this situation in the sense that $\lambda = \epsilon$ is selected. Nonetheless, $\text{info}(C_n, g)$ is still commonly for use with the lasso, even in high-dimensional situations [e.g. 13].

To provide some intuition for this last claim, we provide the following trivial example which explores the behavior of AIC, BIC, and GCV for selecting the tuning parameter in a simple situation. We illustrate this problem with $\widehat{\beta}_{ridge}(\lambda)$, as these GIC then have a closed form.

Example 1. *Consider the following regression data set:*

$$Y = \frac{\sigma}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \qquad \textit{and} \qquad \mathbb{X} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & \sqrt{2} \\ 1 & -1 & 0 \end{bmatrix}.$$

In this no noise case, Y is a scalar multiple of a column of X.

For ridge regression, one can show that

$$\text{df}(\lambda) = \frac{3\lambda + 4}{(2+\lambda)(1+\lambda)},$$

$$\widehat{train}_\lambda = \frac{\sigma^2\lambda^2}{4}\left(\frac{1}{(2+\lambda)^2} + \frac{1}{(1+\lambda)^2}\right),$$

and so,

$$\begin{aligned} info(C_n, g) = &\log\left(\frac{\sigma^2\lambda^2}{4}\left(\frac{1}{(2+\lambda)^2} + \frac{1}{(1+\lambda)^2}\right)\right) \\ &+ C_n\, g\left(\frac{3\lambda+4}{(2+\lambda)(1+\lambda)}\right). \end{aligned}$$

For $0 < \lambda < 1$, $\frac{13\sigma^2\lambda^2}{144} \leq \widehat{train}_\lambda \leq \frac{5\sigma^2\lambda^2}{16}$, so $\log(\widehat{train}_\lambda) \to -\infty$ like $\log(\lambda)$ as $\lambda \to 0$. Hence, minimizing $info(C_n, g)$ will choose $\lambda = 0$ unless the second term increases at least as fast as $-\log(\lambda)$, that is we require constants c and C such that $g\left(\frac{3\lambda+4}{(2+\lambda)(1+\lambda)}\right) \geq C\log(1/\lambda)$ for all $\lambda < c$. We see immediately that AIC and BIC, which both have $g(x) \equiv x$, will always select $\lambda = 0$. This corresponds to reporting the unregularized, least squares solution.

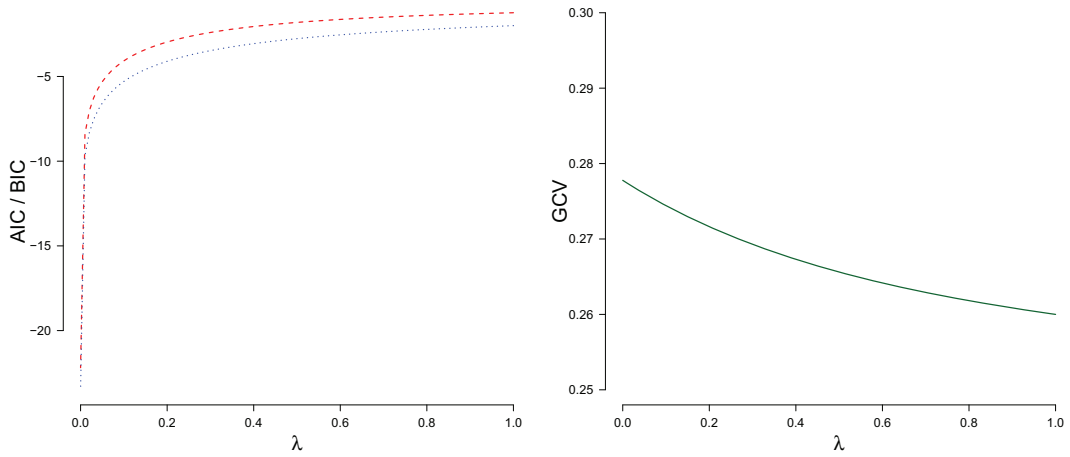

Figure 1. The left plot shows AIC (red, dashed) and BIC (blue, dotted) as we vary λ from 1×10^{-5} to 1 for the small numerical example. The right plot shows the same setup but GCV instead. Notice that, using AIC or BIC, we would always choose the unregularized, $\lambda = 0$ model while GCV leads us to select $\lambda = \infty$.

For GCV, the issue is a bit more subtle. In this example, as rank$(\mathbb{X}) = n = 2$, $-\log(1 - \mathrm{df}\,/n) \to \infty$ *and hence the rate that* $-\log(1 - \mathrm{df}\,/n)$ *goes to* ∞*, along with magnitude of the constants involved, determines which trivial solution,* $\lambda = 0$ *or* $\lambda \to \infty$*, is returned. In particular,*

$$\log\left(\frac{5\sigma^2}{9}\right) \geq GCV = \log\left(\frac{\sigma^2(2\lambda^2 + 6\lambda + 5)}{(2\lambda + 3)^2}\right)$$
$$\geq \lim_{\lambda \to \infty} GCV = \log\left(\frac{\sigma^2}{2}\right)$$

which means GCV will select $\lambda \to \infty$ *and* $\widehat{\beta} \to 0$.

In Figure 1, we plot AIC and BIC for $\lambda \in [1 \times 10^{-5}, 1]$ *(left plot) and GCV (right plot) for ridge regression on this dataset. Using AIC would have us report the unregularized model; that is using a least squares solution. We will illustrate how the lasso behaves with info*(C_n, g) *in greater detail below. Finally, we note that the behavior of GCV in this example is the opposite of what happens in the simulations we report below. There, the penalty term is unable to outweigh the training error term, and hence, the unregularized,* $\lambda = 0$*, solution is usually returned.*

3. Our procedure for tuning parameter selection via plug-in estimation

To remedy the pathological behavior of info(C_n, g) from equation (4) in the high-dimensional case, we propose to select λ in the lasso problem via unbiased risk estimation. Under the model in equation (1), the squared ℓ_2 prediction risk of a coefficient

vector β can be written

$$\begin{aligned}
\mathrm{R}_\beta &= n^{-1}\mathbb{E}\left\|\mathbb{X}\beta - \mathbb{X}\beta_*\right\|_2^2 \\
&= n^{-1}\mathbb{E}\left\|\mathbb{X}\beta - Y\right\|_2^2 - \sigma^2 + 2n^{-1}\sum_{i=1}^{n}\mathrm{Cov}(\widehat{Y}_i, Y_i), \\
&= n^{-1}\mathbb{E}\left\|\mathbb{X}\beta - Y\right\|_2^2 - \sigma^2 + 2n^{-1}\sigma^2\,\mathrm{df}\,.
\end{aligned}$$

Therefore, a suite of sensible estimators of the squared ℓ_2 prediction risk is produced via

$$\widehat{R}_\beta(\widehat{\sigma}^2, C_n) = n^{-1}\left\|\mathbb{X}\beta - Y\right\|_2^2 - \widehat{\sigma}^2 + C_n\widehat{\sigma}^2\widehat{\mathrm{df}}, \tag{6}$$

where C_n is a sequence of constants depending on n, $\widehat{\sigma}^2$ is an estimator of σ^2, and $\widehat{\mathrm{df}}$ is an estimator of df for the procedure under consideration. This general expression is commonly referred to as Stein's unbiased risk estimator [SURE, 18]. For simplicity, we will omit any arguments to $\widehat{R}$ that aren't directly relevant to the discussion at hand and write $\widehat{R}_\lambda \equiv \widehat{R}_{\beta(\lambda)}$ when β is indexed by the tuning parameter λ.

If $\mathbb{E}[\widehat{\sigma}^2\widehat{\mathrm{df}}] = \sigma^2\,\mathrm{df}$ and $\mathbb{E}[\widehat{\sigma}^2] = \sigma^2$ then $\widehat{R}_\beta(\widehat{\sigma}^2, C_n = 2n^{-1})$ is an unbiased estimator of R_β. For example, suppose that $n > p$, $\widehat{\beta}(0)$ is a least squares solution, and $\widehat{\sigma}^2 = (n-p)^{-1}\left\|Y - \mathbb{X}\widehat{\beta}(0)\right\|_2^2$ is the least squares estimator of σ^2. Then $\mathbb{E}[\widehat{\sigma}^2\widehat{\mathrm{df}}] = \sigma^2\,\mathrm{df}$ and $\widehat{R}_{\widehat{\beta}(0)}(\widehat{\sigma}^2, C_n = 2n^{-1})$ is the classical Mallow's Cp [19]. This follows as $\widehat{\beta}(0)$ is linear in Y and hence $\mathrm{df} = \widehat{\mathrm{df}} = tr(H) = \mathrm{rank}(\mathbb{X})$, where H is such that $\mathbb{X}\widehat{\beta}(0) = HY$.

As the lasso is not linear Y, we must use an estimate of df. [7, 20] show that for the lasso, the degrees of freedom of $\widehat{Y} = \mathbb{X}\widehat{\beta}(\lambda)$ is equal to $\mathbb{E}[\mathrm{rank}(\mathbb{X}_{\mathcal{S}(\lambda)})]$, suggesting the natural unbiased estimator $\widehat{\mathrm{df}} = \widehat{\mathrm{df}}(\lambda) = \mathrm{rank}(\mathbb{X}_{\mathcal{S}(\lambda)})$. This is the degrees of freedom estimator we use for both GIC and $\widehat{R}_\lambda$.

Though SURE is not in itself a new approach to selecting tuning parameters in the lasso problem, the literature at this point contains a major omission. When $\mathrm{rank}(\mathbb{X}) = n \leq p$, the choice of an estimator of the noise variance σ^2 is far from straightforward. For example, the lasso path algorithm in the `R` package `lars` avoids this issue. If $p < n$, it provides a Cp-like score, which is superficially similar to equation (6), with the least-squares variance estimator for the largest possible model as $\widehat{\sigma}^2$. Hence, it is unusable (and not produced) if $p > n$.

In the recent theoretical literature, results for high-dimensional tuning parameter selection assume σ^2 is known to get around the difficult task of high-dimensional variance estimation [21–24]. However, it is crucial to estimate σ^2 for $\widehat{R}_\lambda$ to work effectively in practice. To demonstrate this necessity, we perform a second small simulation to illustrate the poor behavior of $\widehat{R}(\sigma^2)$ when σ^2 is erroneously assumed known.

Example 2. *We generate draws according to the model in equation (1), such that $n = 30$, $p = 150$, and β_* has one nonzero coefficient drawn from the standard Laplace distribution. In Figure 2, we explore four methods for choosing λ for the lasso. Clockwise from top left these methods are $\widehat{R}_\lambda(\sigma^2 = 1)$, $\widehat{R}_\lambda(\widehat{\sigma}^2_{CV})$, $\widehat{R}_\lambda(\widehat{\sigma}^2_{RCV})$ (see Section 3.1 for definitions of these variance estimators), and lastly info($C_n = 2/n, g(x) = x$), which corresponds to AIC.*

As expected, $\widehat{R}_\lambda(\sigma^2 = 1)$ performs quite poorly when σ is far from 1. In this case,

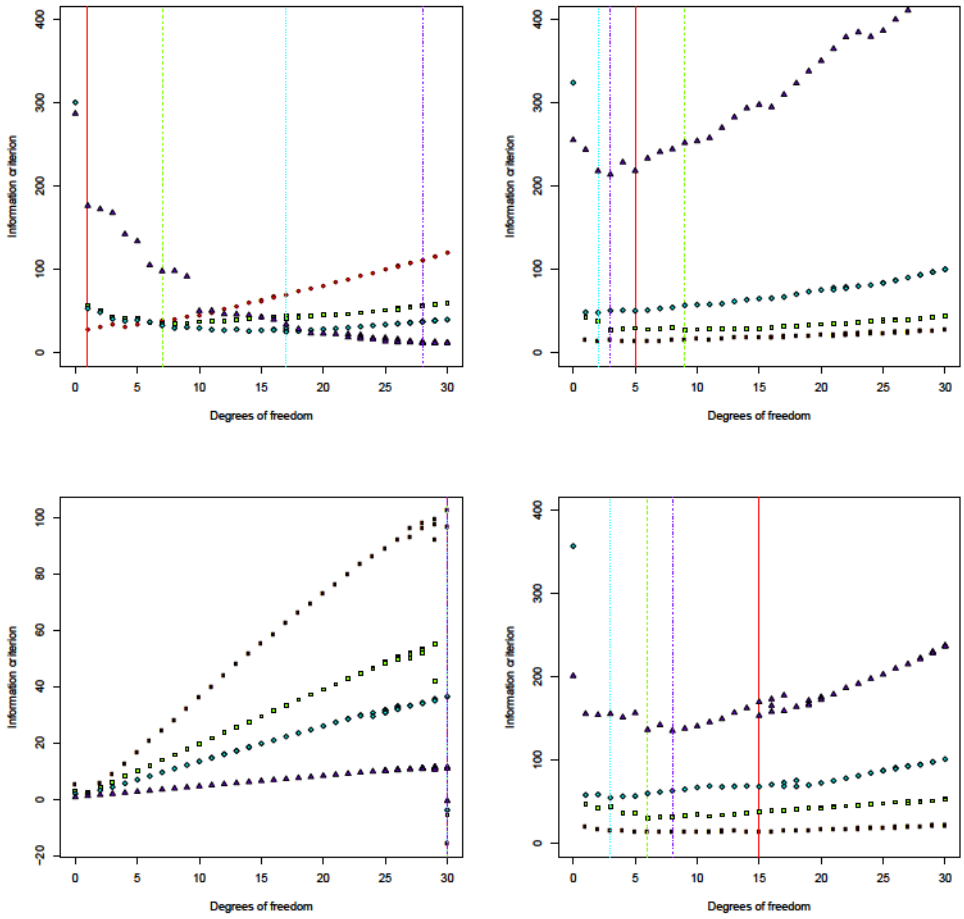

Figure 2. We use four different values for σ: $\sigma = 0.5$ (red, solid, circles), $\sigma = 1$ (green, dashed, squares), $\sigma = 1.5$ (cyan, dotted, diamonds), $\sigma = 5$ (violet, dash-dot, triangles). A vertical line is drawn at the minimizer. Risk estimation methods, clockwise from top left: $\widehat{R}(\sigma^2 = 1)$, $\widehat{R}(\widehat{\sigma}^2_{\mathrm{CV}})$, $\widehat{R}(\widehat{\sigma}^2_{\mathrm{RCV}})$, and info$(C_n = 2/n, g(x) = x)$. Notice that info$(C_n = 2/n, g(x) = x)$ always selects the unregularized model and $\widehat{R}(\sigma^2 = 1)$ depends significantly on σ.

the selected models have widely varying degrees of freedom, choosing highly non-sparse models despite there being only 1 non-zero true coefficient. Also, info$(C_n = 2, g(x) = x)$ continues to choose the unregularized solution, as predicted by the previous example, unless we arbitrarily constrain df *to be some value less than 30. The other two, $\widehat{R}(\widehat{\sigma}^2_{CV})$ and $\widehat{R}(\widehat{\sigma}^2_{RCV})$, perform much better. We now discuss both of these estimators. Occasionally in practice, researchers may not compute info$(C_n = 2, g(x) = x)$ for all λ. Instead, it is calculated from the most sparse to the least sparse solutions, and then cut off when info$(C_n = 2, g(x) = x)$ does not decrease. However, this procedure may not always work. In particular, for $\sigma = 5$, info$(C_n = 2, g(x) = x)$ is monotonically increasing, except for* $\mathrm{df} = 30$. *In other cases, info$(C_n = 2, g(x) = x)$ is not guaranteed to be convex, and this procedure will result in possibly ignoring better solutions.*

3.1. *High-dimensional variance estimation*

The literature on variance estimation in high dimensions is a quickly growing field. We use three high-dimensional variance estimators in our proposed risk estimator. A comprehensive evaluation of these estimators (and some others) is given by [25], but we note that the goal here is different: we do not wish to estimate σ^2 itself but rather wish to use it as an input to $\widehat{R}_\beta$, which can then be used to select tuning parameters or estimate R_β. It is not necessarily true that a good estimator of σ^2 leads to a good estimator of R_β.

The first two approaches start by finding $\widehat{\beta}(\widehat{\lambda}_{CV})$ by minimizing a K-fold cross-validation estimator of the risk to produce $\widehat{\lambda}_{CV}$ (see Section 2.1 for the details of cross validation) and finding a minimizer of equation (2) after inserting $\widehat{\lambda}_{CV}$. With this coefficient estimate, the squared ℓ_2-norm of the residuals can be used as a variance estimate, that is

$$\widehat{\sigma}^2_{CV} = \frac{1}{n - \widehat{\mathrm{df}}(\widehat{\lambda}_{CV})} \left\| Y - \mathbb{X}\widehat{\beta}(\widehat{\lambda}_{CV}) \right\|_2^2. \tag{7}$$

Alternatively, a restricted maximum likelihood-type method can be formed by examining the orthogonal complement of the projection onto the column space of $\mathbb{X}_{\mathcal{S}(\widehat{\lambda}_{CV})}$: $H^{\perp}_{CV}$. Using this projection we define

$$\widehat{\sigma}^2_{RMLE} = \frac{1}{tr(H^{\perp}_{CV})} \left\| H^{\perp}_{CV} Y \right\|_2^2 = \frac{1}{n - \widehat{\mathrm{df}}(\widehat{\lambda}_{CV})} \left\| H^{\perp}_{CV} Y \right\|_2^2.$$

The second equality follows because, for the lasso, $\mathrm{trace}(H^{\perp}_{CV}) = \mathrm{trace}(I - H_{CV}) = n - \mathrm{rank}\left(\mathbb{X}_{\mathcal{S}(\widehat{\lambda}_{CV})}\right)$, which implies $\mathrm{trace}(H^{\perp}_{CV}) = n - \widehat{\mathrm{df}}(\widehat{\lambda}_{CV})$. Hence these two variance estimators differ only in the size of the residuals. In fact, due to the nature of projections,

$$\left\| H^{\perp}_{CV} Y \right\|_2^2 \leq \left\| Y - \mathbb{X}\widehat{\beta}(\widehat{\lambda}_{CV}) \right\|_2^2.$$

Thus, it must hold that $\widehat{\sigma}^2_{RMLE} \leq \widehat{\sigma}^2_{CV}$ and $\widehat{R}(\widehat{\sigma}^2_{RMLE})$ penalizes model complexity less than $\widehat{R}(\widehat{\sigma}^2_{CV})$. In Section 4, our simulations show that, when choosing $C_n = 2/n$, $\widehat{R}(\widehat{\sigma}^2_{CV})$ results in lower prediction risk, better estimation consistency, and higher precision, while $\widehat{R}(\widehat{\sigma}^2_{RMLE})$ has better recall.

The third variance estimation method we consider is known as refitted cross-validation [RCV, 26]. After randomly splitting the data in half, $\mathbb{X}_{\mathcal{S}(\widehat{\lambda}_{CV})}$ is formed on the first half and $\widehat{\sigma}^2_1$ is formed via equation (7), using the Y and $\mathbb{X}$ values from the second half. The procedure is then repeated, exchanging the roles of the halves, producing $\widehat{\sigma}^2_2$. A final estimate is formed via $\widehat{\sigma}^2_{RCV} = (\widehat{\sigma}^2_1 + \widehat{\sigma}^2_2)/2$.

In a comprehensive simulation study, [25] finds that $\widehat{\sigma}^2_{CV}$ is the most reliable estimator for σ^2 out of those cited above, although, as pointed out by [26], it appears to have a negative bias whereas $\widehat{\sigma}^2_{RCV}$ does not. However, this doesn't mean that any of the above methods will necessarily produce superior performance as a plug-in variance estimator for risk estimation or tuning parameter selection.

Armed with any of the above high-dimensional variance estimators, we can form an

estimator of β_* via $\widehat{\beta}(\widehat{\lambda})$, where

$$\widehat{\lambda} = \underset{\lambda}{\operatorname{argmin}} \widehat{R}_\lambda(\widehat{\sigma}, C_n). \tag{8}$$

As discussed above, tuning parameter selection procedures based on SURE or information criteria have no theoretical justification when the variance is unknown and $p > n$. In the next section, we present a comprehensive empirical investigation of the performance of the lasso with tuning parameter selected by the aforementioned methods. Additionally, we include comparisons to other modified lasso-type methods for completeness.

4. Empirical evaluation

In the remainder of this paper, we evaluate our proposed risk estimation methods for the purposes of choosing the tuning parameter λ for lasso. We consider only the high-dimensional setting and evaluate success using several criteria such as prediction risk and model selection. We first perform a comprehensive simulation and then present results from a real-world application involving survival times as a function of gene expression data.

4.1. *Simulation parameters*

For our simulations, we consider a wide range of possible conditions by varying the correlation in the design, ρ; the number of measurements, p; the sparsity, α; and the signal-to-noise ratio, SNR. In all cases, we let $n = 200$ (similar results hold for $n = 100$).

The design matrices, $\mathbb{X} \in \mathbb{R}^{n \times p}$, are produced by concatenating independent and identically distributed rows with mean zero and correlations introduced by an autoregressive model: $\mathrm{Cov}(X_{ij}, X_{ik}) = \rho^{|j-k|}$. For these simulations, we consider correlations $\rho = 0.1$, 0.5, and 0.8.

For sparsity, we define $s_* = \lfloor n^\alpha \rfloor$ and generate the s_* non-zero elements of β_* from a Laplace distribution with parameter 1, which matches a Bayesian interpretation of the lasso. We let α be 0.4 or 0.7, which corresponds to 8 or 40 non-zero elements, respectively. We vary σ^2 so that the signal-to-noise ratio, defined to be $\mathrm{SNR} = n^{-1}\beta_*^\top \mathbb{E}[\mathbb{X}^\top \mathbb{X}]\beta_*/\sigma^2$, is 0.1, 1, or 10. Note that as SNR increases the observations go from a high-noise and low-signal regime to a low-noise and high-signal one. We let $p = 400$ or $p = 1500$.

Lastly, we consider two different noise distributions, $\epsilon_i \sim N(0, 1)$ and $\epsilon_i \sim 3^{-1/2} t(3)$. Here $t(3)$ indicates a t distribution with 3 degrees of freedom and the $3^{-1/2}$ term makes the variance equal to 1 and the ϵ_i are independent. As the results for these noise distributions are quite similar, we only present the Gaussian simulations. Furthermore, while we have simulated all combinations of these parameters and distributions, we include only a subset here for brevity.

4.2. *Modified lasso-type methods*

For a more complete comparison, we include in our simulations some variations on the lasso estimator that have been proposed.

First, [27] develops 'scaled sparse regression' (SSR), which uses the fact that the optimal choice of λ for lasso is asymptotically proportional to σ. By recasting the lasso problem as

$$\widehat{\beta}_{SSR} = \underset{\beta,\sigma}{\operatorname{argmin}} \frac{1}{2n\sigma} \|Y - \mathbb{X}\beta\|_2^2 + \frac{(1-a)\sigma}{2} + M \|\beta\|_1 ,$$

and fixing M and a, the authors develop theory for "tuning parameter free" lasso with simultaneous variance estimation. Though this is a promising approach, the objective function is not convex, hence the variance and the lasso solution are iteratively computed and the solutions tend to depend on the starting values. Nonetheless, SSR enjoys attractive theoretical properties.

Alternatively, [28] suggests the $\sqrt{\text{lasso}}$, or "square root lasso," as a modification of the lasso problem

$$\widehat{\beta}_{\sqrt{\text{lasso}}} = \underset{\beta}{\operatorname{argmin}} \frac{1}{\sqrt{n}} \|Y - \mathbb{X}\beta\|_2 + \frac{\lambda_n}{n} \|\beta\|_1 . \tag{9}$$

Appealing to asymptotic arguments, they show that the minimizer of equation (9) achieves near oracle performance if $\lambda_n = c\sqrt{n}\Phi^{-1}(1-\alpha/(2p))$, which does not depend on σ. Here, Φ^{-1} is the quantile function for the standard Gaussian distribution.

We also consider the Smoothly Clipped Absolute Deviation Penalty [2]:

$$\widehat{\beta}_{SCAD} = \underset{\beta}{\operatorname{argmin}} \frac{1}{2n} \|Y - \mathbb{X}\beta\|_2^2 + \sum_{j=1}^{p} g_\lambda(|\beta_j|),$$

where

$$g'_\lambda(\theta) = \lambda \left[\mathbf{1}(\theta \le \lambda) + \frac{(a\lambda - \theta)_+}{(a-1)\lambda} \mathbf{1}(\theta \ge \lambda) \right],$$

for some $a > 2$ and $\theta > 0$.

Lastly, our experiments show that GCV tends to dramatically under regularize in the lasso problem. Likewise, setting $C_n = \log(n)/n$ in $\widehat{R}_\beta(\widehat{\sigma}^2, C_n)$ tends to over regularize. Hence, we investigate a two-stage method whereby an intial screening is performed by selecting $\widehat{\lambda}_{GCV}$ and forming $\mathcal{S}_{\widehat{\lambda}_{GCV}}$. This often selects a very large model, typically with $|\mathcal{S}_{\widehat{\lambda}_{GCV}}| = n$. For the second stage, we use only the columns of $\mathbb{X}$ with indices in $\mathcal{S}_{\widehat{\lambda}_{GCV}}$ to compute $\widehat{R}_\beta(\widehat{\sigma}^2, C_n = \log(n)/n)$, which is minimized over λ to produce $\widehat{\lambda}$. Then, the output of this two-stage method is $\widehat{\beta}(\widehat{\lambda})$. We refer to this procedure as "2-stage" and do not report results for GCV alone as it is uniformly poor. This procedure is shown in Algorithm 1.

GCV's behavior is intimately connected to the rate at which the numerator, given by the training error, and the denominator, given by $(1 - \text{df}/n)^2$, go to zero as $\lambda \to 0$. In our simulations, the numerator goes to zero at a faster rate than the denominator and hence GCV tends to dramatically under-regularize. Additionally, by noting that $1/(1-x)^2 \approx 1 + 2x$, GCV is approximately the same as AIC. However, this approximation is only accurate for x near zero, which happens when df is forced to be small relative to n. In the classical case where $n \gg p$, this approximation is quite accurate, but in the high-dimensional problem, relatively larger df may explain some of the underperformance

Algorithm 1: 2-stage method for tuning parameter selection

Input: Design matrix $\mathbb{X}$, response Y, sequence of λ

1 Solve equation (2) for each λ;

2 Find $\widehat{\lambda}_{GCV}$ by minimizing equation (5);

3 Set $S_{\widehat{\lambda}_{GCV}}$ to be the non-zero elements of $\widehat{\beta}(\widehat{\lambda}_{GCV})$;

4 Compute $\widehat{R}_\beta(\widehat{\sigma}^2, C_n = \log(n)/n)$ using only the columns of $\mathbb{X}$ in $S_{\widehat{\lambda}_{GCV}}$ for each λ;

5 Select $\widehat{\lambda}_{\text{2-stage}}$ by minimizing $\widehat{R}_\beta(\widehat{\sigma}^2, C_n = \log(n)/n)$;

Output: Coefficient estimates $\widehat{\beta}(\widehat{\lambda}_{\text{2-stage}})$

Table 1. List of methods and abbreviations used in our empirical study

Abbreviation	Method
CV-10-Fold	10-fold cross validation
MCV	Modified Cross Validation
R-Oracle-2	$\widehat{R}_\beta(\sigma^2,\ C_n = 2/n)$
R-CV-2	$\widehat{R}_\beta(\widehat{\sigma}^2_{CV},\ C_n = 2/n)$
R-RMLE-2	$\widehat{R}_\beta(\widehat{\sigma}^2_{RMLE},\ C_n = 2/n)$
R-RCV-2	$\widehat{R}_\beta(\widehat{\sigma}^2_{RCV},\ C_n = 2/n)$
R-Oracle-logn	$\widehat{R}_\beta(\sigma^2,\ C_n = \log(n)/n)$
R-CV-logn	$\widehat{R}_\beta(\widehat{\sigma}^2_{CV},\ C_n = \log(n)/n)$
2-stage	Two-stage method using GCV then R-CV-logn
SCAD	Smoothly clipped absolute deviation
SSR	Scaled sparse regression
SQRT	$\sqrt{\text{lasso}}$
SQRT refitted	OLS estimation on the model selected with $\sqrt{\text{lasso}}$

of GCV as a tuning parameter selection method.

In the next section, we give more details about the numerical implementation of the methods considered in this paper to aid in reproducibility.

4.3. *Implementation of methods and notation*

For ease of reference, Table 1 displays all of the methods for which we present simulations. Since all of these methods rely on numerical optimization routines, it is important to discuss the particular implementation of the solvers used to generate $\widehat{\beta}(\lambda)$.

Two widely used implementations for lasso are `glmnet` [29], which uses coordinate descent and a grid of λ values, and `lars`, which leverages the piece-wise linearity of the lasso solution path. The package `glmnet` is much faster than `lars`, however, `glmnet` only examines a grid of λ values and returns an approximate solution at each λ (due to the iterative nature of the algorithm). Additionally, `glmnet` suffers from numerical stability issues for small λ values when $p > n$.

Because the `lars` path will necessarily change for different cross-validation folds, the grid-based nature of `glmnet` is more suited for use with cross-validation. For this reason, we use `glmnet` for CV-10-Fold and to find $\widehat{\sigma}^2_{CV}$, $\widehat{\sigma}^2_{RCV}$, and $\widehat{\sigma}^2_{RMLE}$.

With any high dimensional variance estimator $\widehat{\sigma}^2$, we need to compute $\widehat{\lambda} = \operatorname{argmin} \widehat{R}_\lambda(\widehat{\sigma}^2)$. We use `lars` to find the entire lasso solution path on all of the data to compute $\widehat{R}_\lambda(\widehat{\sigma}^2)$ and then report the minimizer $\widehat{\lambda}$ and $\widehat{\beta}(\widehat{\lambda})$.

To optimize the modified lasso problems' objective functions, we use the `R` package `scalreg` to fit SSR and the `R` package `flare` to fit the $\sqrt{\text{lasso}}$. For `scalreg`, we

choose the starting point for the iteration via the quantile method [30]. For `flare`, we set the tuning parameter to $\lambda = c\sqrt{n}\Phi^{-1}(1-\alpha/(2p))$ with $c = 1.1$ and $\alpha = 0.05$, as suggested by [28]. As $\sqrt{\text{lasso}}$ tended to pick the correct model but with overly regularized coefficient estimates, we will additionally examine a refitted version of $\sqrt{\text{lasso}}$ in which the unregularized least squares solution of Y on $\mathbb{X}_{\mathcal{S}(\widehat{\beta}_{\sqrt{\text{lasso}}})}$ is reported. In an attempt to get as close as possible to the global optimum, we decrease the `prec` (precision) option to 1×10^{-10} and increase `max.ite` (maximum iterations) to 1×10^{7}.

To fit SCAD, we use the package `ncvreg` [31] with default settings ($a = 3.7$) and choose λ via the built in CV function. We note that [2] suggests using either CV or an approximation to GCV which uses the trace of the projection matrix from the final iteration to form an estimate $\widehat{\text{df}}$. However, this matrix is a function of Y, so the calculated df is not unbiased. We therefore only report the default cross-validation-based method, and we note that subsequent work [14, 32] has carefully investigated information criteria using SCAD.

The ideal, or oracle, version of our method in equation (6) would use the known variance. We refer to this as the oracle risk estimator and note that it is unbiased. Obviously this is not a viable estimator in practice, but it is useful for normalizing comparisons in our simulation study. We provide two versions of this oracle estimator: $\widehat{R}_\beta(\sigma^2, C_n = 2/n)$ and $\widehat{R}_\beta(\sigma^2, C_n = \log(n)/n)$.

4.4. *Simulation results*

We present results for four different metrics based on different data analysis objectives. If the risk estimation methods are used to select tuning parameters, then the data analysts could be interested in the prediction risk, which evaluates how well we can predict a new Y given a new X; consistency, which measures how far the procedure $\widehat{\beta}$ is from β_*; or F-score, which considers how well a method does at model selection. Alternatively, when evaluating the success of a method, or when comparing it to another method, the risk estimate itself is of interest. We evaluate these four criteria in the following subsections. Table 1 shows the correspondence between the mathematical notation we have used so far, and the arabic letters used in the figures. When describing each figure, we will refer to different methods with the arabic letters for clarity.

4.4.1. Prediction risk

Prediction risk is an important criterion as it is often a major goal in modern data analysis applications. For these simulations, we approximate R_β in equation (3) with the average squared error over 5000 test observations and normalize it by subtracting σ^2, but continue to denote it R_β. We present boxplots for the log of the prediction risk of the selected models in Figure 3 and Figure 4 for SNR 0.1 and 10 respectively.

For low SNR, MCV, R-RMLE-2, SQRT, and SQRT refitted all perform noticeably worse than the competing methods. For high SNR, SCAD performs best, especially when $p = 1500$ and when the true vector β_* is non-sparse ($\alpha = 0.7$). Also, CV-10-Fold and R-CV-2 both perform somewhat better than R-RCV and R-CV-logn.

4.4.2. Consistency

The second performance metric we use examines the ability of $\widehat{\beta}(\widehat{\lambda})$ to produce accurate estimates of the true parameter β_*. We examine a normalized version of the deviation

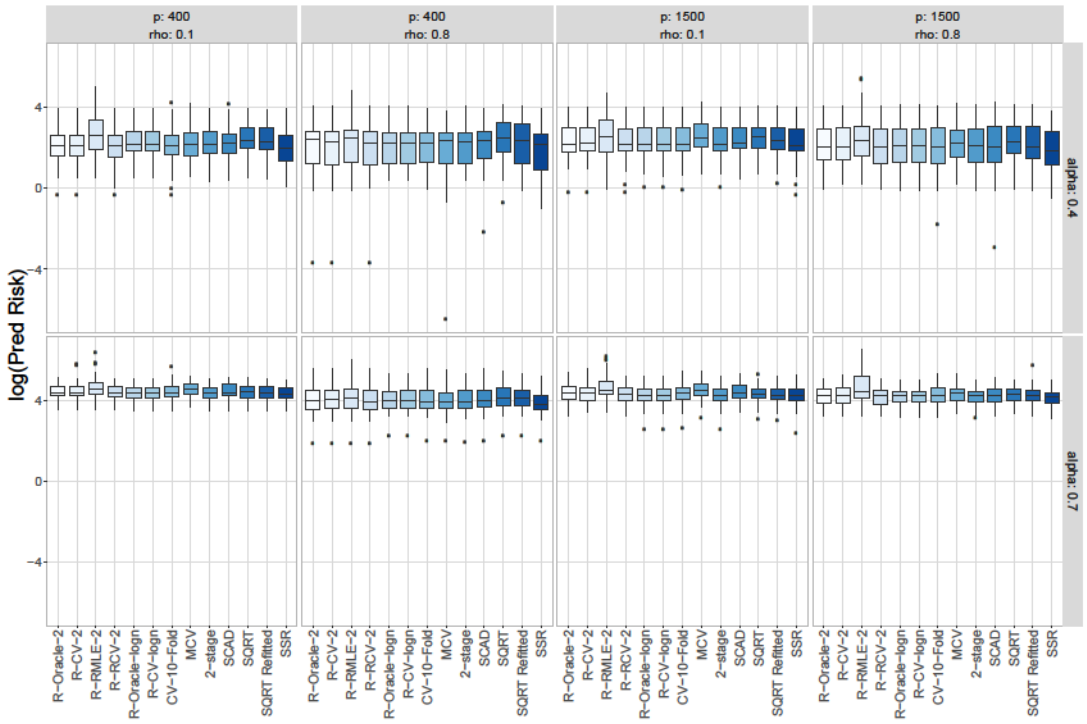

Figure 3. Comparison of log prediction risk for $SNR = 0.1$. Top row: $\alpha = 0.4$. Bottom row: $\alpha = 0.7$.

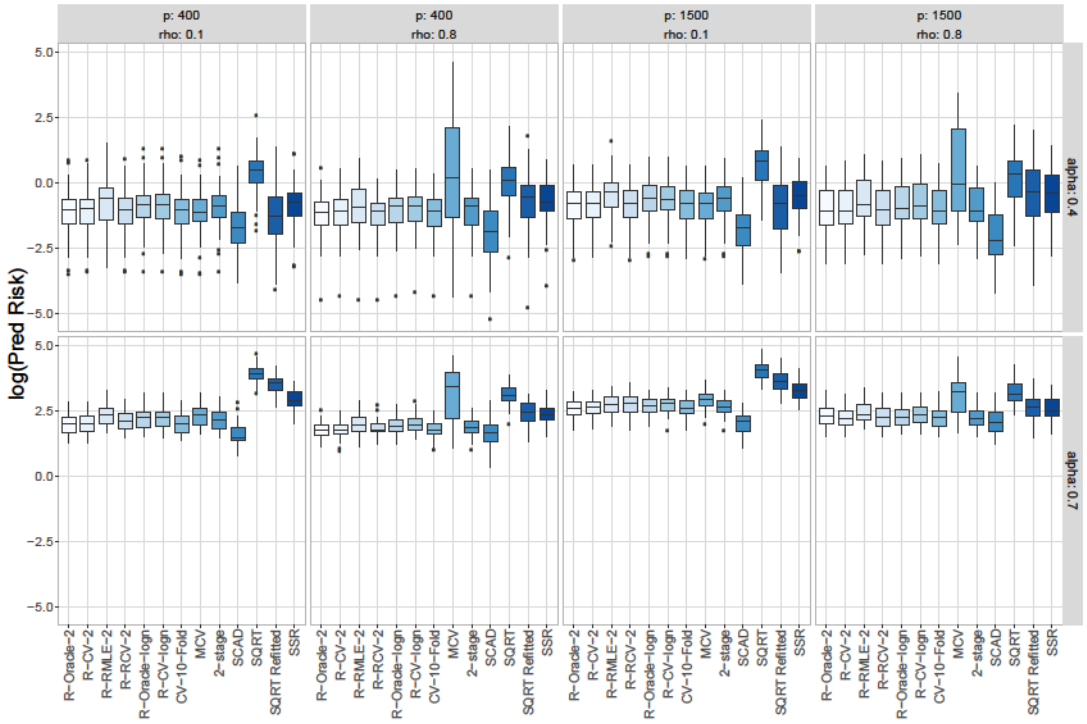

Figure 4. Comparison of log prediction risk for $SNR = 10$. Top row: $\alpha = 0.4$. Bottom row: $\alpha = 0.7$.

between the estimated coefficients and the size of the parameter:

$$C(\widehat{\beta}) = \frac{\mathbb{E}\left\|\widehat{\beta} - \beta_*\right\|_2^2}{14^{\|\beta_*\|_2^2}}.$$

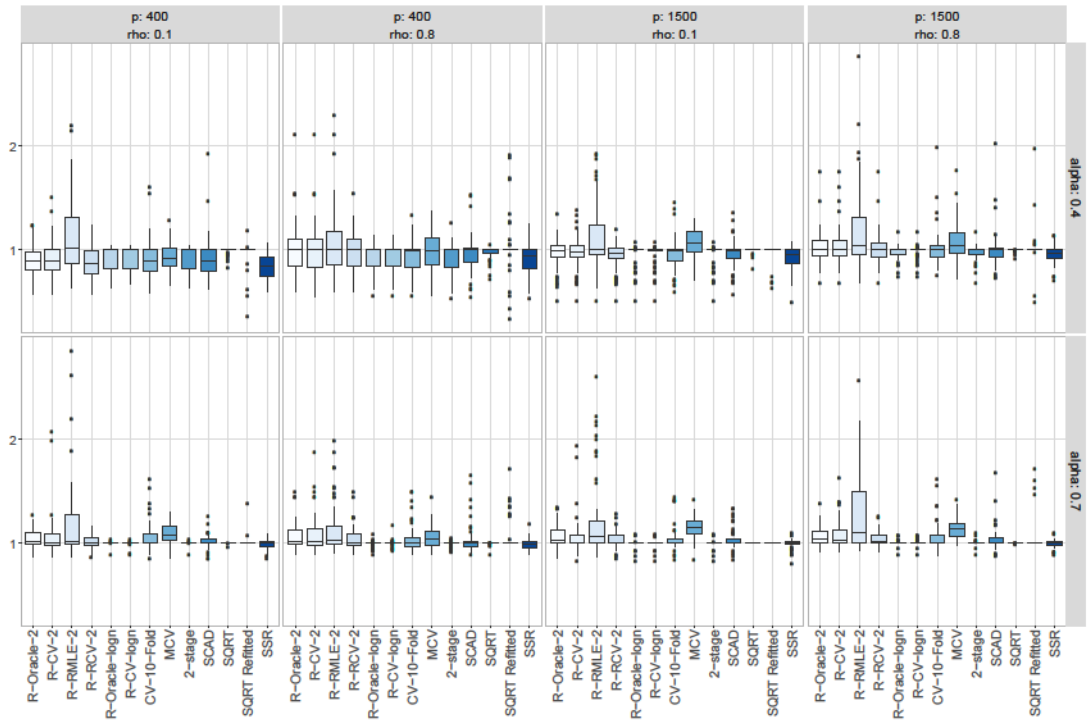

Figure 5. Comparison of consistency for $SNR = 0.1$. Top row: $\alpha = 0.4$. Bottom row: $\alpha = 0.7$.

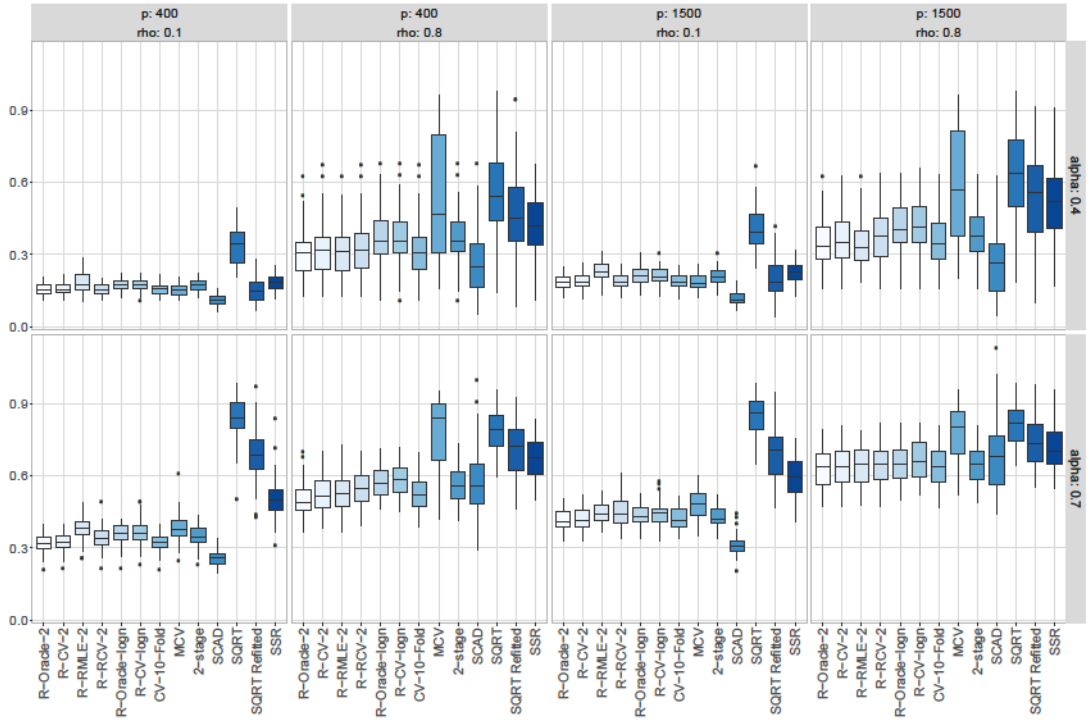

Figure 6. Comparison of consistency for $SNR = 10$. Top row: $\alpha = 0.4$. Bottom row: $\alpha = 0.7$.

Thus, smaller values are better, and values near 1 often represent overly sparse solutions as $\widehat{\beta} \equiv 0 \Rightarrow \mathbb{E}\left\|\widehat{\beta} - \beta_*\right\| = \|\beta_*\|$.

In the low-SNR regime (Figure 5), no procedure performs particularly well, as one would expect. The R-CV-logn, 2-stage, SQRT, and SQRT refitted nearly always select $\widehat{\beta} \equiv 0$ with occasional exceptions. This results in slightly better $C(\widehat{\beta})$ than the other methods. For the high-SNR regime (Figure 6), SCAD performs best, particularly in the sparse scenario ($\alpha = 0.4$) or when $p = 1500$ and $\rho = 0.1$. CV-10-Fold and R-CV-2 perform similarly to each other and are slightly better than the other methods. MCV, R-RMLE-2, SQRT, and SQRT refitted all perform rather poorly.

4.4.3. F-score

To examine the ability of these procedures to perform model selection directly, we define the precision and recall for a particular β to be respectively (recalling that $\mathcal{S} = \{j : |\beta_j| > 0\}$ and $|\mathcal{S}|$ is the number of elements in $\mathcal{S}$)

$$P(\mathcal{S}) = \frac{|\mathcal{S} \cap \mathcal{S}_*|}{|\mathcal{S}|} \qquad \text{and} \qquad R(\mathcal{S}) = \frac{|\mathcal{S} \cap \mathcal{S}_*|}{|\mathcal{S}_*|}.$$

To parsimoniously represent both precision and recall at the same time, we use the F-score (sometimes referred to as the $F1$-score), which is the harmonic mean of the precision and recall:

$$F(\mathcal{S}) = \frac{2R(\mathcal{S})P(\mathcal{S})}{R(\mathcal{S}) + P(\mathcal{S})} = \frac{2}{\frac{1}{R(\mathcal{S})} + \frac{1}{P(\mathcal{S})}}.$$

Observe that $F(\mathcal{S})$ is equal to one if and only if $R(\mathcal{S})$ and $P(\mathcal{S})$ are both equal to one and equal to zero if either $R(\mathcal{S})$ or $P(\mathcal{S})$ are equal to zero. Thus, higher values represent better performance. As an aside, the SQRT and SQRT refitted methods will have the same F-score (as they select the same model). We nonetheless plot both of the methods to maintain easier comparability to other figures.

For the low SNR case (Figure 7), no methods are consistently good. For the high SNR case (Figure 8), the 2-stage method, SSR, and R-CV-logn work well across all settings of α and ρ. When β_* is sparse ($\alpha = 0.4$), SQRT has good F-score performance, but it is one of the worst when α is large. The performance of SCAD has similar discrepancies: it one of the best performers when $\rho = 0.1$ and one of the worst when $\rho = 0.8$. This is potentially useful because ρ can be estimated by the data analyst before fitting the regression (as compared to the SNR or sparsity which cannot). Thus, one could use SCAD in the uncorrelated setting but avoid it when the design is highly correlated. It is notable that for F-score in the high SNR case only, R-CV-logn and 2-stage outperform CV-10-Fold, R-CV-2, and R-RCV-2.

4.4.4. Estimating the risk of the oracle linear model

Instead of using a risk estimate as a tool to empirically choose tuning parameters, sometimes it is important to directly estimate the risk of a procedure to evaluate or compare its performance. In this subsection, we investigate the risk estimation property of both K-fold CV and $\widehat{R}(\widehat{\sigma}, C_n)$ for a few choices of K and $\widehat{\sigma}^2$. As MCV, SSR, 2-stage, SCAD, and SQRT are model selection/estimation procedures and not risk estimators, we leave them out of this comparison. The goal here is to determine whether equation (6) can yield good risk estimates in the high-dimensional setting the same way that unbiased risk estimation can in the low-dimensional setting. Hence, we

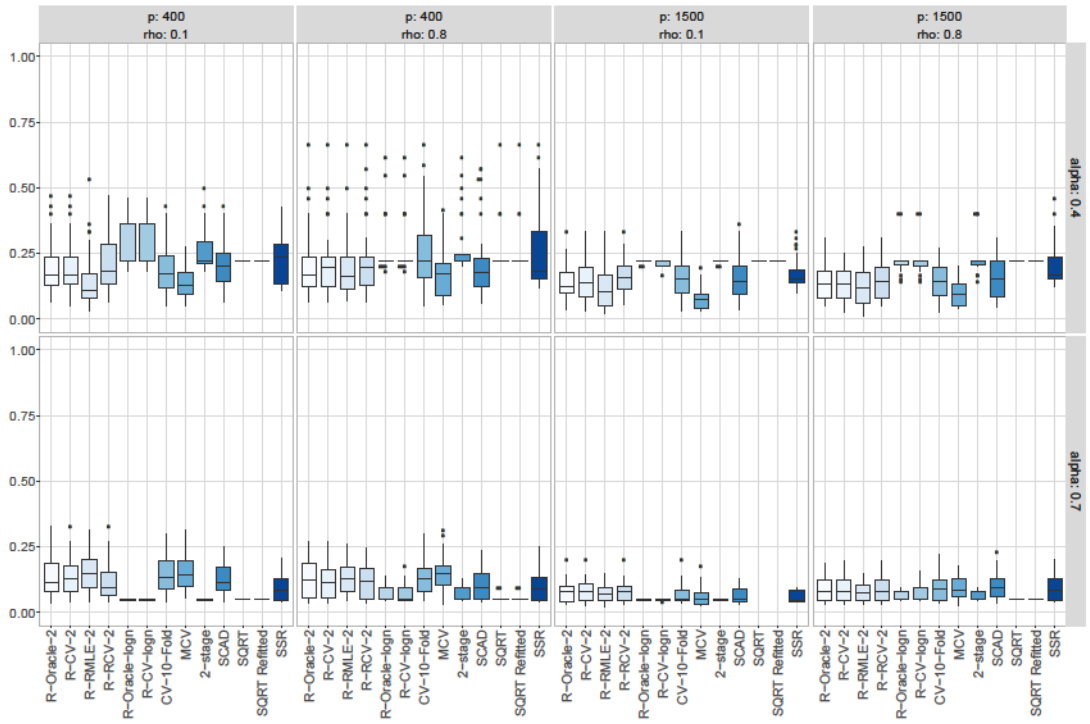

Figure 7. Comparison of F-score for $SNR = 0.1$. Top row: $\alpha = 0.4$. Bottom row: $\alpha = 0.7$.

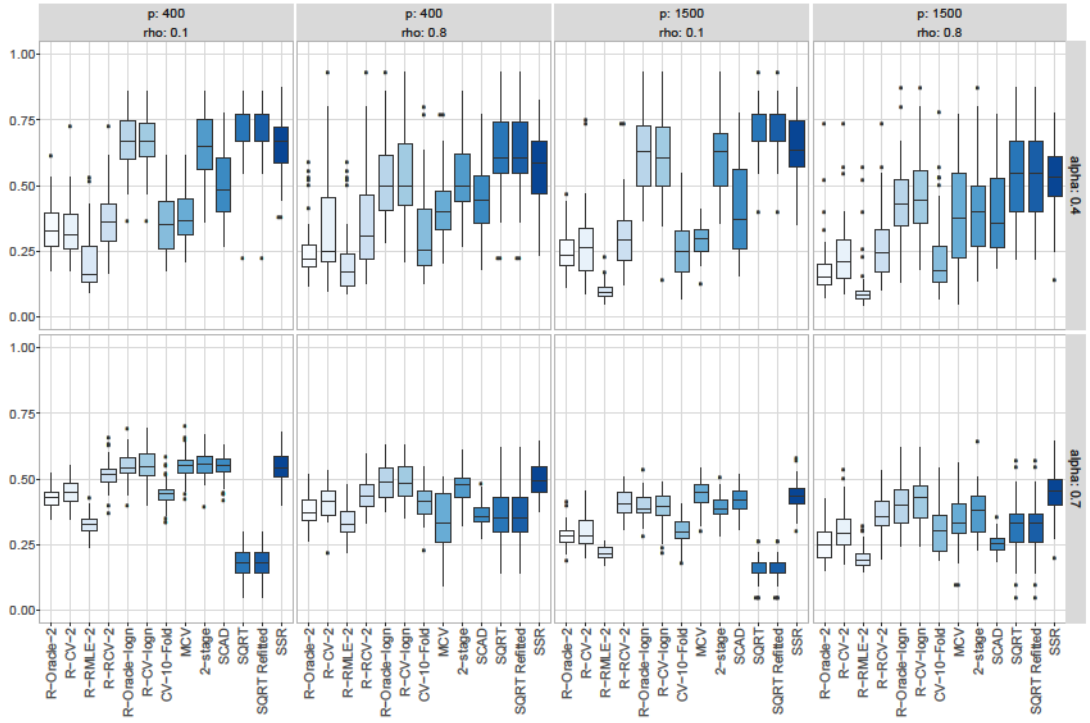

Figure 8. Comparison of F-score for $SNR = 10$. Top row: $\alpha = 0.4$. Bottom row: $\alpha = 0.7$.

set $C_n = 2/n$ as this would be the unbiased choice if either σ^2 is known and $\widehat{\mathrm{df}}$ is unbiased or $\widehat{\sigma}^2$ is unbiased and $\widehat{\mathrm{df}}$ doesn't depend on Y.

Table 2. The root mean squared error of all five risk estimators. Bold values indicate the best method(s) (those within .005 of the minimum) in each case.

snr	alpha	p	rho	CV-2-Fold	CV-10-Fold	R-CV-2	R-RCV-2	R-RMLE-2
0.1	0.4	400	0.1	0.125	**0.108**	**0.104**	**0.105**	**0.104**
0.1	0.4	400	0.8	0.118	**0.105**	**0.105**	**0.104**	**0.106**
0.1	0.4	1500	0.1	0.145	**0.107**	**0.106**	**0.104**	**0.106**
0.1	0.4	1500	0.8	0.128	**0.104**	**0.102**	**0.104**	**0.103**
0.1	0.7	400	0.1	0.447	0.147	0.131	**0.119**	0.158
0.1	0.7	400	0.8	0.405	0.122	**0.112**	**0.107**	0.132
0.1	0.7	1500	0.1	0.426	0.137	0.114	**0.100**	0.146
0.1	0.7	1500	0.8	0.413	0.150	0.118	**0.109**	0.156
10	0.4	400	0.1	0.142	0.115	**0.112**	0.115	**0.107**
10	0.4	400	0.8	0.142	0.099	**0.097**	**0.099**	**0.093**
10	0.4	1500	0.1	0.117	**0.101**	**0.103**	**0.099**	0.111
10	0.4	1500	0.8	0.139	**0.087**	**0.092**	0.163	**0.089**
10	0.7	400	0.1	0.463	**0.147**	0.160	0.465	0.227
10	0.7	400	0.8	0.393	**0.156**	0.161	0.308	0.165
10	0.7	1500	0.1	0.371	**0.110**	0.192	0.831	0.313
10	0.7	1500	0.8	0.440	**0.158**	0.272	0.588	0.198

Using, K-fold CV or $\widehat{R}$ to both choose $\widehat{\lambda}$ and evaluate the risk $\widehat{\beta}(\widehat{\lambda})$ conflates $\widehat{R}$'s performance at tuning parameter selection and risk estimation. Hence, for this evaluation only, we use as a β_*-estimation procedure the oracle least squares estimator. That is, we set

$$\widehat{\beta}_O = \underset{\beta}{\operatorname{argmin}} \|Y - \mathbb{X}_{\mathcal{S}_*}\beta\|_2^2$$

and then calculate $\widehat{R}_{\widehat{\beta}_O}(\widehat{\sigma}^2_{CV}, 2/n)$, $\widehat{R}_{\widehat{\beta}_O}(\widehat{\sigma}^2_{RCV}, 2/n)$, and $\widehat{R}_{\widehat{\beta}_O}(\widehat{\sigma}^2_{RMLE}, 2/n)$ where $\widehat{\sigma}^2$ is estimated with the relevant high-dimensional variance estimator. We also include 2-Fold CV and 10-Fold CV. This choice of β_* estimation procedure is still a function of the data, and hence is random, but it does not require the selection of a tuning parameter. It should, however, be in a neighborhood of β_*.

We find that for sparse models (Figure 9 and Figure 10, top rows), there is very little difference between these five procedures: all are unbiased on median, though 2-Fold CV has slightly larger variance. However, with less sparse models, 2-Fold CV greatly overestimates the risk, while 10-Fold CV is quite accurate. For high SNR and low sparsity, R-RCV-2 has a large upward bias, though it is otherwise quite accurate. For another take, Table 2 shows the squared difference between the risk estimate and the true risk (σ^2 in all cases), averaged across the simulation runs—the risk of the risk estimator. Looking down the table for low SNR, R-RCV-2 is the best method according to this metric, although for sparse models, 10-Fold CV and R-CV-2 are close behind in terms of MSE. This is because the small negative bias of R-RCV-2 is outweighed by the smaller variance it has relative to 10-fold CV and R-CV-2, which are relatively unbiased. With high SNR and dense models, R-RCV-2 is terrible with high positive bias and huge variance, worse than even 2-Fold CV. Note that R-RCV-2 uses a version of 2-Fold CV to estimate σ^2. Here, 10-Fold CV is easily the best, R-CV-2 has low bias, but relatively large variance, while R-RMLE-2 has a pronounced downward bias with small variance.

The poor performance of CV-2-Fold and R-RCV-2 (for dense, high SNR conditions) deserves additional comment. According to [25, Figure 9], the ability of $\widehat{\sigma}^2_{RCV}$ to estimate the variance deteriorates with increasing SNR, which is in line with our simulations. This is an area for further investigation as neither we nor [25] can provide a careful explanation for this phenomenon. One possibility is that splitting the

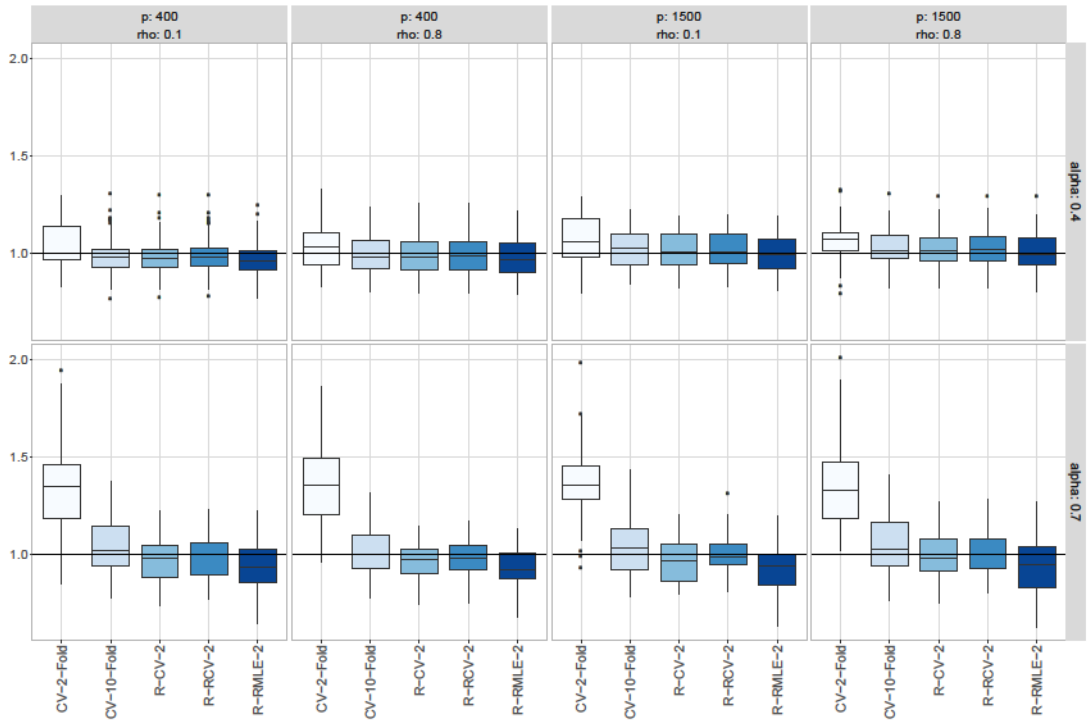

Figure 9. Comparison of risk estimation for $SNR = 0.1$. Top row: $\alpha = 0.4$. Bottom row: $\alpha = 0.7$.

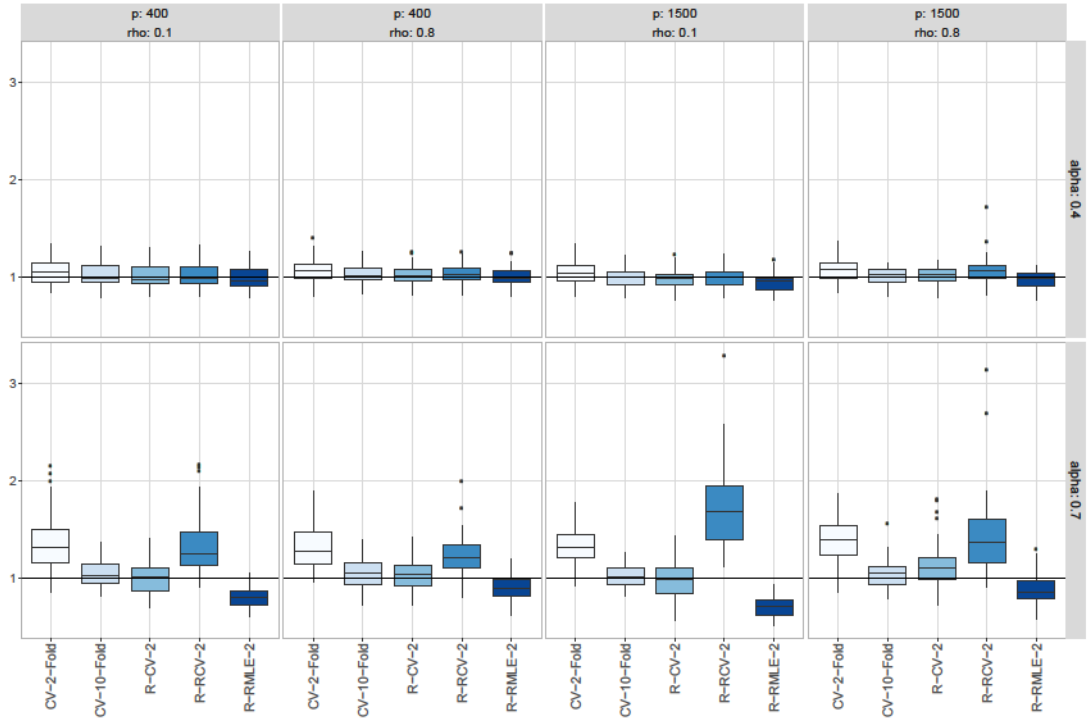

Figure 10. Comparison of risk estimation for $SNR = 10$. Top row: $\alpha = 0.4$. Bottom row: $\alpha = 0.7$.

data in half provides insufficient training data for accurate estimation and one or two additional splits may be sufficient to remedy the issue.

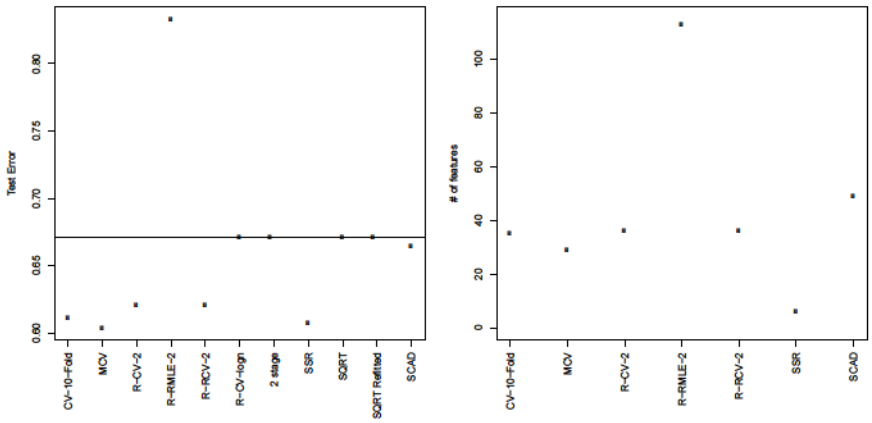

Figure 11. Analysis of leukemia patient survival times. Left plot: prediction risk on test data. The horizontal line indicates the risk of the identically zero estimator $\widehat{\beta}(\lambda) \equiv 0$. Right plot: number of selected genes for methods that report $\widehat{\beta}(\lambda) \neq 0$.

Another possible area for further investigation is the construction of a confidence interval for the risk estimator. As cross-validation averages over K folds in the training data, the variation of the prediction error on each fold can be used to form an informal confidence interval for the risk. This confidence interval can be useful in practice, for example when using the so-called "one standard error rule" [29]. The risk estimator in equation (6) does not rely directly on subsampling and hence does not by default produce a confidence interval. If the data analyst desires such an uncertainty estimate, a sensible, though computationally expensive, approach would be via the bootstrap.

4.5. *Data example: survival times for leukemia patients*

We examine a microarray data set consisting of diffuse large B-cell lymphoma (DLBCL) patients [33, 34]. This data set consists of measurements of 7399 genes made on 160 training patients and 80 test patients, matching the training and test split used by [34]. The response, Y, is the survival time for each patient which we transform as $\log(Y + 1)$ due to skewness.

Our results, which can be found in Figure 11 (left plot), are that many of the tuning parameter selection methods choose $\widehat{\lambda}$ such that $\widehat{\beta}(\widehat{\lambda}) \equiv 0$; that is, the identically zero vector. CV-10-Fold, MCV, R-CV-2, R-RCV-2, SSR, and SCAD produce non-trivial coefficient estimates that improve on the risk of the zero estimator while R-RMLE-2 produces a nontrivial coefficient estimate that is much worse than the zero estimator. For reference, the variance estimators $\widehat{\sigma}^2$ are approximately 0.23, 0.68, and 0.69 for $\widehat{\sigma}^2_{RMLE}$, $\widehat{\sigma}^2_{CV}$, and $\widehat{\sigma}^2_{RCV}$, respectively. Additionally, each method suggests dramatically different numbers of selected genes (Figure 11, right plot), ranging from 6 for SSR to 116 for R-RMLE-2. The intersection of the selected models for those methods which produce nontrivial coefficient estimates are genes 3822 and 4131, which may be reasonable candidates for further investigation.

5. Theoretical analysis

In this section, we provide a result demonstrating that, under a number of standard conditions, our risk estimator will produce a predictor whose performance is compa-

rable to that of the true model. For convenience, we define x_j to be the j^{th} column of $\mathbb{X}$ and X_{ij} to be the i, j entry of $\mathbb{X}$. Also, let $\mathcal{S} \subseteq \{1, 2, \ldots, p\}$ be an index set with $|\mathcal{S}|$ elements and define $\mathcal{S}^c$ to be complement: $\mathcal{S}^c := \{1, 2, \ldots, p\} \setminus \mathcal{S}$.

We define the following conditions.

Condition 1. Assume that $Y = \mathbb{X}\beta_* + \epsilon$

Condition 2. The ϵ_i are distributed i.i.d sub-Gaussian with variance σ^2. That is $\forall t \in \mathbb{R}$, $\mathbb{E}[\exp(t\epsilon_i)] \leq \exp\{\sigma^2 t^2/2\}$.

Condition 3. The design matrix $\mathbb{X}$ satisfies $C_X := \max_{1 \leq j \leq p} \|x_j\|_2^2 / n = O(1)$ for all p.

Condition 4. If $\beta \in \mathbb{R}^p$ is such that $\|\beta_{\mathcal{S}^c}\|_1 \leq L \|\beta_{\mathcal{S}}\|_1$, for some $L \geq 0$, then

$$\|\beta_{\mathcal{S}}\|_1^2 \leq \frac{|\mathcal{S}|}{\phi^2 n} \beta^\top \mathbb{X}^\top \mathbb{X} \beta,$$

where $\phi \equiv \phi(L) > 0$ is known as a compatibility constant.

These conditions are well-known and appear frequently in lasso-related theoretical results. We assume that the data is actually generated by a linear model, as was the case in our simulated analysis. We assume homoscedastic noise which has reasonable tails. Gaussian distributions satisfy Condition 2 as well as bounded distributions and other standard "light-tailed" distributions. Our goal will be to consider the standard high dimensional setting where $p \gg n$ and both approach infinity. Because of this, we need to ensure that as we add columns to the design matrix, larger and larger entries do not come to dominate the solution. Condition 3 says that the maximum column norm grows like its length but not with p. This condition can be eliminated without any difficulty, but it allows for easier interpretation of the result. Finally, we assume that the design matrix satisfies the so-called "compatibility condition" [35]. This allows us to relate the ℓ_1-norm of the coefficient vector with the L_2-norm of the predicted values for a collection of sufficiently sparse coefficient vectors. This condition is also related to the restricted eigenvalue condition [5, 36] which is an alternative.

We state the core result, showing an upper bound on the prediction loss of the lasso with tuning parameter chosen by $\widehat{R}$ versus the true coefficient vector β_*. Set $\Lambda = [\lambda_{\min},\ \lambda_{\max}]$ to be the optimization grid for the tuning parameter λ.

Theorem 1. *Assume Condition 1–Condition 4. Let $\delta > 0$ and $\Lambda = [\lambda_{\min},\ \lambda_{\max}]$. Set $\lambda_{\min} = 2\sigma\sqrt{\frac{2C_X(\log(p)+\delta)}{n}}$. Then, with probability at least $1 - 2e^{-\delta}$,*

$$\frac{1}{n}\left\|\mathbb{X}\beta_* - \mathbb{X}\widehat{\beta}(\widehat{\lambda})\right\|_2^2 \leq \left(\frac{2s_*}{\phi^2}\right)\left(9\lambda_{\max}^2 + \frac{8\sigma^2 C_X\left(\log(p)+\delta\right)}{n}\right) + \rho\left(4\sigma\sqrt{\frac{2C_X\left(\log(p)+\delta\right)}{n}}\right).$$

The first part of the upper bound depends on $\lambda_{\max}$. The second part depends on the penalty ρ. Results for the lasso with oracle tuning parameter deal only with an

upper bound that looks like

$$\frac{2s_*}{\phi^2}\frac{8\sigma^2 C_X\left(\log(p)+\delta\right)}{n}.$$

Therefore, for convergence, they examine the case where s_* goes to infinity as fast as possible. Thus, the lasso "works" as long as $s_* = o(n/\log(p))$. For our bound to be meaningful when s_* grows this quickly, we must have $\lambda_{\max} = O(\sqrt{\log(p)/n})$, the same order as $\lambda_{\min}$. That is, if s_* grows as fast as possible, we get a trivial Λ interval with the upper and lower bounds having the same order (though they can differ by an arbitrary constant). If, instead, s_* is constant, hence growing as slowly as possible, then we simply need $\lambda_{\max} = o(1)$.

Finally, we require $\rho\left(4\sigma\sqrt{\frac{2C_X(\log(p)+\delta)}{n}}\right)$ to go to zero at a similar rate. This of course depends on the penalty selected. For AIC, we require

$$\rho\left(4\sigma\sqrt{\frac{2C_X\left(\log(p)+\delta\right)}{n}}\right) = \frac{2}{n}\widehat{\sigma}^2\widehat{\mathrm{df}}\left(4\sigma\sqrt{\frac{2C_X\left(\log(p)+\delta\right)}{n}}\right).$$

Thus, if $\widehat{\sigma}^2 = O(1)$, then

$$\widehat{\mathrm{df}}\left(4\sigma\sqrt{\frac{2C_X\left(\log(p)+\delta\right)}{n}}\right) = o(n)$$

is sufficient. In particular, for $\widehat{\mathrm{df}} = O(s_*)$ gives convergence.

6. Discussion

In this paper, we investigate a large number of procedures for selecting λ in high-dimensional lasso problems. Our results supplement and elaborate upon those of [16] which apply to the low-dimensional setting ($p < n$). In general, the unbiased-risk-estimation methods we present perform consistently well across conditions. They exhibit many of the familiar properties from the AIC-vs.-BIC debate (BIC selects smaller models, AIC is better for prediction) as well as some variation across variance estimators due to estimation bias. Our simulations lead us to suggest a novel two-stage method (see Section 4.2 and Algorithm 1) that also performs consistently well and warrants further theoretical investigations.

Substantial theory exists for the optimal choice of the tuning parameter for the lasso and related methods. These results, however, depend both on unknown properties of the data generating process and unknown constants. Though there are many data-dependent methods for choosing the tuning parameters, there is a distinct lack of guidance in the literature about which method to use. This uncertainty is even more pronounced when faced with high-dimensional data where $p \gg n$.

We give examples that show that one commonly advocated approach, a generalized information criterion which has desirable theoretical properties in low dimensions, would necessarily choose the unregularized model with $\lambda = 0$ when $p > n$. Therefore, we propose a risk estimator motivated by Stein's unbiased risk estimation. This estimator requires three ingredients: an estimate of the degrees of freedom ($\widehat{\mathrm{df}}$), a constant

that may depend on n (C_n), and an estimator of the variance ($\widehat{\sigma}^2$). While the degrees of freedom for the lasso problem is well understood, the other two choices are much less so. In particular, high-dimensional variance estimation is a difficult problem in its own right.

6.1. *Overall recommendations*

In general, CV-10-Fold performs similarly to R-CV-2, which tends to outperform both R-RCV-2 and R-CV-logn. A notable exception is that R-CV-logn dramatically outperforms for model selection when in the high SNR regime. In all other cases, both CV-10-Fold and R-CV-2 should perform satisfactorily in practice relative to the other methods we examine.

For the oracle risk estimation methods, R-oracle-2 and R-oracle-logn, $\widehat{\sigma}^2_{CV}$ is a good estimator of σ^2 in practice and hence R-CV-2 and R-CV-logn behave very similarly to R-oracle-2 and R-oracle-logn, respectively. However, the variance estimator $\widehat{\sigma}^2_{RMLE}$ tends to dramatically underestimate σ^2 and hence R-RMLE-2 tends to underregularize. Also, though MCV performs the best on the genetics data set, it performed very poorly in the simulations. Hence, R-RMLE-2 and MCV should be avoided in practice.

SCAD performs well for both prediction risk and consistency, particularly when p is large and the true model is not sparse. On the other hand, SQRT refitted performs substantially better than SQRT and hence should be used as an additional step to SQRT in practice. However, SQRT refitted tends to underperform the other methods in our simulations.

In general, the SURE-based methods we develop perform quite well across different simulation conditions and evaluation metrics. The 2-stage method described in Section 4.3 also performs well and warrants further investigation. Standard 10-fold CV performs adequately while the behavior of scaled-sparse regression, $\sqrt{\text{lasso}}$ variants, and MCV depends strongly on the simulation condition. In particular, these modern methods often underperform the SURE-based methods presented in this paper.

Funding

Darren Homrighausen is supported by the National Science Foundation under grant DMS–1407543 and the Institute for New Economic Thinking; under grant INO14-00020. Daniel J. McDonald is supported by the National Science Foundation under grant DMS–1407439 and the Institute for New Economic Thinking under grant INO14-00020.

Biographical note

Darren Homrighausen is Assistant Professor of Statistics at Colorado State University. He has a bachelors degree from the University of Colorado in economics and math, and a and Ph.D. from Carnegie Mellon University in statistics. He has worked extensively on developing both methods and theory for solving various problems in astronomy and cosmology as well as investigating the prediction risk implications of empirical tuning parameter selection for lasso-type methods. More recently, he has become interested in examining the statistical implications of computational

approximations.

Daniel J. McDonald is Assistant Professor of Statistics and Adjunct Assistant Professor of Computer Science at Indiana University, Bloomington. His research interests involve the estimation and quantification of prediction risk, especially developing methods for evaluating the predictive abilities of complex dependent data. This includes the application of statistical learning techniques to time series prediction problems in the context of economic forcasting, as well as investigations of cross-validation and the bootstrap for risk estimation.

References

[1] Yuan M, Lin Y. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2006;68(1):49–67.

[2] Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American statistical Association. 2001;96(456):1348–1360.

[3] Tibshirani RJ. The lasso problem and uniqueness. Electronic Journal of Statistics. 2013; 7:1456–1490.

[4] Efron B, Hastie T, Johnstone I, et al. Least angle regression. The Annals of statistics. 2004;32(2):407–499.

[5] Bunea F, Tsybakov A, Wegkamp M. Sparsity oracle inequalities for the lasso. Electronic Journal of Statistics. 2007;1:169–194.

[6] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: Data mining, inference, and prediction. Springer Verlag; 2009.

[7] Zou H, Hastie T, Tibshirani R. On the degrees of freedom of the lasso. The Annals of Statistics. 2007;35(5):2173–2192.

[8] Hastie T, Tibshirani R, Wainwright MJ. Statistical learning with sparsity: the lasso and generalizations. Boca Raton, FL: CRC Press; 2015.

[9] Homrighausen D, McDonald DJ. Risk consistency of cross-validation for lasso-type procedures. forthcoming Statistica Sinica. 2016+;.

[10] Yu Y, Feng Y. Modified cross-validation for penalized high-dimensional linear regression models. Journal of Computational and Graphical Statistics. 2014;23(4):1009–1027.

[11] Golub GH, Heath M, Wahba G. Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics. 1979;21(2):215–223.

[12] Efron B. How biased is the apparent error rate of a prediction rule? Journal of the American Statistical Association. 1986;81(394):461–470.

[13] Bühlmann P, van de Geer S. Statistics for high-dimensional data: Methods, theory and applications. New York: Springer; 2011.

[14] Wang H, Li R, Tsai CL. Tuning parameter selectors for the smoothly clipped absolute deviation method. Biometrika. 2007;94(3):553–568.

[15] Tibshirani R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B (Statistical Methodology). 1996;58(1):267–288.

[16] Flynn CJ, Hurvich CM, Simonoff JS. Efficiency for regularization parameter selection in penalized likelihood estimation of misspecified models. Journal of the American Statistical Association. 2013;108:1031–1043.

[17] Wang H, Li B, Leng C. Shrinkage tuning parameter selection with a diverging number of parameters. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2009;71(3):671–683.

[18] Stein CM. Estimation of the mean of a multivariate normal distribution. The Annals of Statistics. 1981;9(6):1135–1151.

[19] Mallows CL. Some comments on c_p. Technometrics. 1973;15(4):661–675.

[20] Tibshirani RJ, Taylor J. Degrees of freedom in lasso problems. Annals of Statistics. 2012;

40:1198–1232.
[21] Chen J, Chen Z. Extended bic for small-n-large-p sparse glm. Statistica Sinica. 2012; 22(2):555.
[22] Zhang Y, Shen X. Model selection procedure for high-dimensional data. Statistical Analysis and Data Mining: The ASA Data Science Journal. 2010;3(5):350–358.
[23] Kim Y, Kwon S, Choi H. Consistent model selection criteria on high dimensions. The Journal of Machine Learning Research. 2012;13(1):1037–1057.
[24] Fan Y, Tang CY. Tuning parameter selection in high dimensional penalized likelihood. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2013; 75(3):531–552.
[25] Reid S, Tibshirani R, Friedman J. A study of error variance estimation in lasso regression. Statistica Sinica. 2016;26(35–67).
[26] Fan J, Guo S, Hao N. Variance estimation using refitted cross-validation in ultrahigh dimensional regression. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2012;74(1):37–65.
[27] Sun T, Zhang CH. Scaled sparse linear regression. Biometrika. 2012;99(4):879–898.
[28] Belloni A, Chernozhukov V, Wang L. Square-root lasso: pivotal recovery of sparse signals via conic programming. Biometrika. 2011;98(4):791–806.
[29] Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. Journal of statistical software. 2010;33(1):1.
[30] Sun T, Zhang CH. Sparse matrix inversion with scaled lasso. Journal of Machine Learning Research. 2013;14:3385–3418.
[31] Breheny P, Huang J. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. Annals of Applied Statistics. 2011; 5(1):232–253.
[32] Zhang Y, Li R, Tsai CL. Regularization parameter selections via generalized information criterion. Journal of the American Statistical Association. 2010;105(489):312–323.
[33] Rosenwald A, Wright G, Chan WC, et al. The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. New England Journal of Medicine. 2002;346(25):1937–1947.
[34] Bair E, Tibshirani R. Semi-supervised methods to predict patient survival from gene expression data. PLoS Biol. 2004;2(4):e108.
[35] van de Geer SA, Bühlmann P. On the conditions used to prove oracle results for the lasso. Electronic Journal of Statistics. 2009;3:1360–1392.
[36] Bickel PJ, Ritov Y, Tsybakov AB. Simultaneous analysis of lasso and dantzig selector. The Annals of Statistics. 2009;37(4):1705–1732.

Appendix A. Proof of Theorem 1 and supporting results

Lemma 2 (Generalization of [13], Lemma 6.2)**.** *Define*

$$\mathcal{G} = \left\{ \max_{1 \le j \le p} 2|\epsilon^\top x_j|/n < M \right\}.$$

Suppose Condition 2 holds. For any $\delta > 0$, if

$$M := 2\sigma \sqrt{\frac{2C_X \left(\log(p) + \delta\right)}{n}}$$

then

$$\mathbb{P}(\mathcal{G}) \geq 1 - 2e^{-\delta}.$$

Proof. Define x_j to be the j^{th} column of $\mathbb{X}$ and recalling that X_{ij} is the j^{th} entry of the i^{th} covariate vector. Define

$$Z_j := \frac{2\epsilon^\top x_j}{n}.$$

Let $t \geq 0$ be given. Then, under Condition 2, we have

$$\mathbb{E}\left[\exp(tZ_j)\right] = \prod_{i=1}^{n} \mathbb{E}\left[\exp\left(\frac{2t\epsilon_i X_{ij}}{n}\right)\right] \leq \prod_{i=1}^{n} \exp\left(\frac{4t^2\sigma^2 X_{ij}^2}{2n^2}\right) = \exp\left(\frac{2t^2\sigma^2}{n^2}\left\|x_j\right\|_2^2\right).$$

Therefore,

$$\begin{aligned}
1 - \mathbb{P}\left(\mathcal{G}\right) &= \mathbb{P}\left(\max_j |Z_j| \geq M\right) \\
&\leq \sum_j \mathbb{P}\left(|Z_j| \geq M\right) \\
&\leq p \max_j \mathbb{P}\left(|Z_j| \geq M\right) \\
&\leq 2p \max_j \inf_t \exp(-tM) \exp\left(\frac{2t^2\sigma^2}{n^2}\left\|x_j\right\|_2^2\right) \\
&= 2p \inf_t \exp(-tM) \exp\left(\frac{2t^2\sigma^2}{n^2} \max_j \left\|x_j\right\|_2^2\right) \\
&= 2p \exp\left\{-\frac{n^2M^2}{8\sigma^2 \max_j \left\|x_j\right\|_2^2}\right\}.
\end{aligned}$$

Thus, for any $\delta > 0$, if we set

$$M := \sqrt{\frac{8 \max_j \left\|x_j\right\|_2^2 \sigma^2}{n^2}\left(\log(p) - \log(\delta)\right)}$$

then

$$\mathbb{P}\left(\mathcal{G}\right) \geq 1 - 2\delta.$$

Redefine $\delta \to e^{-\delta}$ and use $C_X \geq n^{-1} \max_j \left\|x_j\right\|_2^2$ to get the result. □

Lemma 3. *Define $\widehat{\lambda}$ as in equation (8). Set $\rho(\lambda) = C_n \widehat{\sigma}^2 \widehat{\mathrm{df}}(\lambda)$. Then for any $\lambda \geq 0$,*

$$\frac{1}{n}\left\|\mathbb{X}\beta_* - \mathbb{X}\widehat{\beta}(\widehat{\lambda})\right\|_2^2 + \lambda\left\|\widehat{\beta}(\lambda)\right\|_1 \leq \frac{2}{n}\epsilon^\top \mathbb{X}(\beta_* - \widehat{\beta}(\widehat{\lambda})) + \lambda\left\|\beta_*\right\|_1 + \rho(\lambda)$$

Proof.

$$\begin{aligned}\frac{1}{n}\left\|Y-\mathbb{X}\widehat{\beta}(\widehat{\lambda})\right\|_2^2+\lambda\left\|\widehat{\beta}(\lambda)\right\|_1 &\leq \frac{1}{n}\left\|Y-\mathbb{X}\widehat{\beta}(\widehat{\lambda})\right\|_2^2+\rho(\widehat{\lambda})+\lambda\left\|\widehat{\beta}(\lambda)\right\|_1\\ &\leq \frac{1}{n}\left\|Y-\mathbb{X}\widehat{\beta}(\lambda)\right\|_2^2+\rho(\lambda)+\lambda\left\|\widehat{\beta}(\lambda)\right\|_1\\ &\leq \frac{1}{n}\left\|Y-\mathbb{X}\beta_*\right\|_2^2+\rho(\lambda)+\lambda\left\|\beta_*\right\|_1.\end{aligned}$$

Here we have used the fact that $\widehat{\lambda}$ minimized $n^{-1}\left\|Y-\mathbb{X}\widehat{\beta}(\lambda)\right\|_2^2+\rho(\lambda)$ and $\widehat{\beta}(\lambda)$ minimized $n^{-1}\left\|Y-\mathbb{X}\beta\right\|_2^2+\lambda\left\|\beta\right\|_1$. Using $Y=\mathbb{X}\beta_*+\epsilon$ gives

$$\begin{aligned}\left\|Y-\mathbb{X}\widehat{\beta}(\widehat{\lambda})\right\|_2^2 &= \left\|\mathbb{X}\beta_*+\epsilon-\mathbb{X}\widehat{\beta}(\widehat{\lambda})\right\|_2^2\\ &= \left\|\epsilon\right\|_2^2+\left\|\mathbb{X}(\beta_*-\widehat{\beta}(\widehat{\lambda}))\right\|_2^2+2\epsilon^\top\mathbb{X}(\beta_*-\widehat{\beta}(\widehat{\lambda}))\end{aligned}$$

while $\left\|Y-\mathbb{X}\beta_*\right\|_2^2=\left\|\epsilon\right\|_2^2$. Therefore,

$$\frac{1}{n}\left\|\mathbb{X}\beta_*-\mathbb{X}\widehat{\beta}(\widehat{\lambda})\right\|_2^2+\lambda\left\|\widehat{\beta}(\lambda)\right\|_1\leq\frac{2}{n}\epsilon^\top\mathbb{X}(\beta_*-\widehat{\beta}(\widehat{\lambda}))+\rho(\lambda)+\lambda\left\|\beta_*\right\|_1.$$

□

Lemma 4 (Generalization of [13], Theorem 6.1)**.** *Suppose Condition 1 and Condition 4 hold. Then on $\mathcal{G}$, for any $\lambda>M$,*

$$\left\|\widehat{\beta}(\lambda)-\beta_*\right\|_1\leq\frac{s_*(3\lambda+M)^2}{4(\lambda-M)\phi^2}.$$

Proof. Note that $\beta_*=0$ on $\mathcal{S}_*^c$. Then, by the triangle inequality, we have,

$$\left\|\widehat{\beta}(\lambda)\right\|_1\geq\left\|\widehat{\beta}_{\mathcal{S}_*^c}(\lambda)\right\|_1-\left\|\widehat{\beta}_{\mathcal{S}_*}(\lambda)-\beta_*\right\|_1+\left\|\beta_*\right\|_1. \tag{A1}$$

Therefore, on $\mathcal{G}$ for any $\lambda\geq 0$,

$$\begin{aligned}&\frac{1}{n}\left\|\mathbb{X}(\widehat{\beta}(\lambda)-\beta_*)\right\|_2^2+\lambda\left(\left\|\widehat{\beta}_{\mathcal{S}_*^c}(\lambda)\right\|_1-\left\|\widehat{\beta}_{\mathcal{S}_*}(\lambda)-\beta_*\right\|_1+\left\|\beta_*\right\|_1\right)\\ &\leq\frac{1}{n}\left\|\mathbb{X}(\widehat{\beta}(\lambda)-\beta_*)\right\|_2^2+\lambda\left\|\widehat{\beta}(\lambda)\right\|_1,\\ &\leq\frac{2}{n}\epsilon^\top\mathbb{X}(\widehat{\beta}(\lambda)-\beta_*)+\lambda\left\|\beta_*\right\|_1\\ &\leq M\left\|\widehat{\beta}(\lambda)-\beta_*\right\|_1+\lambda\left\|\beta_*\right\|_1\\ &= M\left\|\widehat{\beta}_{\mathcal{S}_*}(\lambda)-\beta_*\right\|_1+M\left\|\widehat{\beta}_{\mathcal{S}_*^c}(\lambda)\right\|_1+\lambda\left\|\beta_*\right\|_1,\end{aligned}$$

where the first inequality is due to equation (A1) and the second and third follow from

Lemma 3. The final equality follows by noting that

$$\left\|\widehat{\beta}(\lambda)-\beta_*\right\|_1=\left\|\widehat{\beta}_{\mathcal{S}_*}(\lambda)-\beta_*\right\|_1+\left\|\widehat{\beta}_{\mathcal{S}_*^c}(\lambda)\right\|_1.$$

Collecting terms shows that

$$\frac{1}{n}\left\|\mathbb{X}(\widehat{\beta}(\lambda)-\beta_*)\right\|_2^2+(\lambda-M)\left\|\widehat{\beta}_{\mathcal{S}_*^c}\right\|_1\leq(\lambda+M)\left\|\widehat{\beta}_{\mathcal{S}_*}-\beta_*\right\|_1. \tag{A2}$$

By using the above inequality twice, we see that

$$\begin{aligned}
&\frac{1}{n}\left\|\mathbb{X}(\widehat{\beta}(\lambda)-\beta_*)\right\|_2^2+(\lambda-M)\left\|\widehat{\beta}(\lambda)-\beta_*\right\|_1\\
&\leq\frac{1}{n}\left\|\mathbb{X}(\widehat{\beta}(\lambda)-\beta_*)\right\|_2^2+(\lambda-M)\left\|\widehat{\beta}_{\mathcal{S}_*}(\lambda)-\beta_*\right\|_1+(\lambda+M)\left\|\widehat{\beta}_{\mathcal{S}_*}-\beta_*\right\|_1\\
&=\frac{1}{n}\left\|\mathbb{X}(\widehat{\beta}(\lambda)-\beta_*)\right\|_2^2+2\lambda\left\|\widehat{\beta}_{\mathcal{S}_*}(\lambda)-\beta_*\right\|_1\\
&\leq(\lambda+M)\left\|\widehat{\beta}_{\mathcal{S}_*}(\lambda)-\beta_*\right\|_1+2\lambda\left\|\widehat{\beta}_{\mathcal{S}_*}(\lambda)-\beta_*\right\|_1\\
&=(3\lambda+M)\left\|\widehat{\beta}_{\mathcal{S}_*}(\lambda)-\beta_*\right\|_1
\end{aligned}$$

By equation (A2), $\left\|\widehat{\beta}_{\mathcal{S}_*^c}\right\|\leq(\lambda+M)(\lambda-M)^{-1}\left\|\widehat{\beta}_{\mathcal{S}_*}-\beta_*\right\|_1$ and hence Condition 4 with $L=(\lambda+M)(\lambda-M)^{-1}$ applies. Also, observe that $uv\leq u^2/4+v^2$. Therefore,

$$\begin{aligned}
(3\lambda+M)\left\|\widehat{\beta}_{\mathcal{S}_*}(\lambda)-\beta_*\right\|_1&\leq(3\lambda+M)\left(\frac{\sqrt{s_*}}{\phi\sqrt{n}}\right)\left\|\mathbb{X}(\widehat{\beta}(\lambda)-\beta_*)\right\|_2\\
&\leq\left(\frac{(3\lambda+M)^2s_*}{4\phi^2}\right)+\frac{1}{n}\left\|\mathbb{X}(\widehat{\beta}(\lambda)-\beta_*)\right\|_2^2.
\end{aligned}$$

Rearranging produces the desired result as long as $\lambda>M$.

□

Proof of Theorem 1. On the set $\mathcal{G}$,

$$\frac{2}{n}\epsilon^\top\mathbb{X}(\beta_*-\widehat{\beta}(\widehat{\lambda}))<M\left\|\widehat{\beta}(\widehat{\lambda})-\beta_*\right\|_1.$$

By Lemma 3 and Lemma 4 for any $\lambda > M$

$$\begin{aligned}
\frac{1}{n}\left\|\mathbb{X}\beta_* - \mathbb{X}\widehat{\beta}(\widehat{\lambda})\right\|_2^2 &< M\left\|\widehat{\beta}(\widehat{\lambda}) - \beta_*\right\|_1 + \lambda\left\|\beta_*\right\|_1 - \lambda\left\|\widehat{\beta}(\lambda)\right\|_1 + \rho(\lambda) \\
&\leq M \sup_{\lambda' \in \Lambda}\left\|\widehat{\beta}(\lambda') - \beta_*\right\|_1 + \lambda\left\|\beta_* - \widehat{\beta}(\lambda)\right\|_1 + \rho(\lambda) \\
&\leq (M+\lambda) \sup_{\lambda' \in \Lambda}\left\|\widehat{\beta}(\lambda') - \beta_*\right\|_1 + \rho(\lambda) \\
&\leq 2\lambda \sup_{\lambda' \in \Lambda}\left\|\widehat{\beta}(\lambda') - \beta_*\right\|_1 + \rho(\lambda) \\
&\leq 2\lambda \sup_{\lambda' \in \Lambda}\frac{s_*(3\lambda' + M)^2}{4(\lambda' - M)\phi^2} + \rho(\lambda) \\
&\leq \left(\frac{s_*}{2\phi^2}\right)\left(\frac{\lambda(3\lambda_{\max} + M)^2}{M}\right) + \rho(\lambda) \\
&\leq \left(\frac{s_*}{\phi^2}\right)\left(\frac{\lambda}{M}\right)\left(9\lambda_{\max}^2 + M^2\right) + \rho(\lambda).
\end{aligned}$$

Where for this last inequality we use that $\lambda_{\min} = 2M$. Finally, since this inequality holds for all $\lambda > M$ and $\rho(\lambda)$ is decreasing in λ, we take $\lambda = 2M$. □