A Survey of Tuning Parameter Selection for High-dimensional Regression

Yunan Wu and Lan Wang

Abstract

Penalized (or regularized) regression, as represented by Lasso and its variants, has become a standard technique for analyzing high-dimensional data when the number of variables substantially exceeds the sample size. The performance of penalized regression relies crucially on the choice of the tuning parameter, which determines the amount of regularization and hence the sparsity level of the fitted model. The optimal choice of tuning parameter depends on both the structure of the design matrix and the unknown random error distribution (variance, tail behavior, etc). This article reviews the current literature of tuning parameter selection for high-dimensional regression from both theoretical and practical perspectives. We discuss various strategies that choose the tuning parameter to achieve prediction accuracy or support recovery. We also review several recently proposed methods for tuning-free high-dimensional regression.

1 Introduction

High-dimensional data, where the number of covariates/features (e.g., genes) may be of the same order or substantially exceed the sample size (e.g., number of patients), have become common in many fields due to the advancement in science and technology. Statistical methods for analyzing high-dimensional data have been the focus of an enormous amount of research in the past

[1]Yunan Wu is a Ph.D. candidate and Lan Wang is Professor, School of Statistics, University of Minnesota, Minneapolis, MN 55455. Email: wangx346@umn.edu. The authors thank the editor's interest in this topic and an anonymous referee's constructive comments. They acknowledge the support of VA IIR 16-253 and NSF DMS-1712706.

decade or so, see the books of Hastie and Friedman [2009], Bühlmann and van de Geer [2011], Hastie et al. [2015] and Wainwright [2019], among others, for extensive discussions.

In this article, we consider a linear regression model of the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta}_0 + \boldsymbol{\epsilon}, \tag{1}$$

where $\mathbf{y} = (y_1, \ldots, y_n)^T$ is the vector of responses, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)^T$ is an $n \times p$ matrix of covariates, $\boldsymbol{\beta}_0 = (\beta_{01}, \ldots, \beta_{0p})^T$ is the vector of unknown regression coefficients, $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^T$ is a random noise vector with each entry having mean zero and variance σ^2. We are interested in the problem of estimating $\boldsymbol{\beta}_0$ when $p \gg n$. The parameter $\boldsymbol{\beta}_0$ is usually not identifiable in high dimension without imposing additional structural assumption, as there may exist $\boldsymbol{\beta}_0' \neq \boldsymbol{\beta}_0$ but $\mathbf{X}\boldsymbol{\beta}_0' = \mathbf{X}\boldsymbol{\beta}_0$. One intuitive and popular structural assumption underlying a large body of the past work on high-dimensional regression is the assumption of strong (or hard) sparsity. Loosely speaking, it means only a relatively small number—usually much less than the sample size n— of the p covariates are active in the regression model.

To overcome the issue of over-fitting, central to high-dimensional data analysis are penalized or regularized regression techniques represented by Lasso [Tibshirani, 1996, Chen et al., 2001] and its variants such as Dantzig selector [Candes and Tao, 2007], SCAD [Fan and Li, 2001], MCP [Zhang, 2010a] and Capped L_1 [Zhang, 2010b]. In a nutshell, a high-dimensional penalized regression estimator solves

$$\min_{\boldsymbol{\beta}} \Big\{ (2n)^{-1} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \sum_{j=1}^{p} p_\lambda(|\beta_j|) \Big\}, \tag{2}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T$, $||\cdot||$ denotes the L_2 vector norm, and $p_\lambda(\cdot)$ is a penalty function which depends on a tuning parameter $\lambda > 0$. Customarily, the intercept β_0 is not penalized.

Regardless of the penalty function, the choice of the tuning parameter λ plays a crucial role in the performance of the penalized high-dimensional regression estimator. The tuning parameter λ determines the level of the sparsity of the solution. Generally speaking, a larger value of λ indicates heavier penalty and tends to produce a sparser model.

The paper aims to provide a broad review of the current literature on tuning parameter selection for high-dimensional penalized regression from

both theoretical and practical perspectives. We discuss different strategies for tuning parameter selection to achieve accurate prediction performance or to identify active variables in the model, where the later goal is often referred to as support recovery. We also review several recently proposed tuning-free high-dimensional regression procedures, which circumvent the difficulty of tuning parameter selection.

2 Tuning parameter selection for Lasso

2.1 Background

A simple yet successful approach for avoiding over-fitting and enforcing sparsity is to regularize the classical least-squares regression with the L_1 penalty, corresponding to adopting $p_\lambda(|\beta_j|) = \lambda|\beta_j|$ in (2). This choice leads to the well known Least Absolute Shrinkage and Selection Operator (Lasso, Tibshirani [1996]), which simultaneously performs estimation and variable selection. In the field of signal processing, the Lasso is also known as basis pursuit [Chen et al., 2001].

Formally, the Lasso estimator $\widehat{\boldsymbol{\beta}}^{\text{Lasso}}(\lambda)$ is obtained by minimizing the regularized least squares loss function, that is,

$$\widehat{\boldsymbol{\beta}}^{\text{Lasso}}(\lambda) = \arg\min_{\boldsymbol{\beta}} \Big\{ (2n)^{-1} \sum_{i=1}^{n} (Y_i - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda ||\boldsymbol{\beta}||_1 \Big\}, \tag{3}$$

where $\mathbf{x}_i^T = (x_{i1}, \ldots, x_{ip})$ is the ith row of $\mathbf{X}$, $||\boldsymbol{\beta}||_1$ denotes the L_1-norm of $\boldsymbol{\beta}$ and λ denotes the tuning parameter. By varying the value of λ and solving the above minimization problem for each λ, we obtain a solution path for Lasso.

In the literature, a great deal of work has been devoted to understanding the theoretical properties of Lasso, including the theoretical guarantee on the nonasymptotic estimation error bound $||\widehat{\boldsymbol{\beta}}^{\text{Lasso}}(\lambda) - \boldsymbol{\beta}_0||_2$, the prediction error bound $||\mathbf{X}(\widehat{\boldsymbol{\beta}}^{\text{Lasso}}(\lambda) - \boldsymbol{\beta}_0)||_2$, and the ability of recovering the support set or the active set of the model $\{j : \beta_{0j} \neq 0, j = 1, \ldots, p\}$, see [Greenshtein and Ritov, 2004, Meinshausen and Bühlmann, 2006, Zhao and Yu, 2006a, Bunea et al., 2007, Van de Geer et al., 2008, Zhang et al., 2008, Bickel et al., 2009, Candès et al., 2009], among others. The tremendous success of L_1-regularized regression technique is partly due to its computational convenience. Efficient

algorithms such as the exact path-following LARs algorithm [Efron et al., 2004] and the fast coordinate descent algorithm [Friedman et al., 2007, Wu et al., 2008] have greatly facilitated the use of Lasso.

2.2 A theoretical perspective for tuning parameter selection

Motivated by the Karush-Kuhn-Tucker condition for convex optimization [Boyd and Vandenberghe, 2004], Bickel et al. [2009] proposed a general principal for selecting λ for Lasso. More specifically, it is suggested that λ should be chosen such that

$$P\Big\{||n^{-1}\mathbf{X}^T\boldsymbol{\epsilon}||_\infty \le \lambda\Big\} \ge 1-\alpha, \tag{4}$$

for some small $\alpha > 0$, where $||\cdot||_\infty$ denotes the infinity (or supremum) norm.

Consider the important example where the random errors ϵ_i, $i = 1, \ldots, n$, are independent $N(0, \sigma^2)$ random variables and the design matrix is normalized such that each column has L_2-norm equal to $\sqrt{n}$. One can show that an upper bound of λ satisfying (5) is given by $\tau\sigma\sqrt{\log p/n}$ for some positive constant τ. To see this, we observe that by the property of the tail probability of Gaussian distribution and the union bound,

$$P\Big\{||n^{-1}\mathbf{X}^T\boldsymbol{\epsilon}||_\infty \le \tau\sigma\sqrt{\log p/n}\Big\} \ge 1 - 2\exp\big(-(\tau^2-2)\log p/2\big),$$

for some $\tau > \sqrt{2}$. Similar probability bound holds if the random errors have sub-Gaussian distributions (e.g., Section 4.2 of [Negahban et al., 2012]).

Most of the existing theoretical properties of Lasso were derived while fixing λ at an oracle value satisfying (5) or within a range of oracle values whose bounds satisfying similar constraints. For example, the near-oracle error bound of Lasso given in Bickel et al. [2009] was derived assuming $\lambda = \tau\sigma\sqrt{\log p/n}$ for some $\tau > 2\sqrt{2}$ when $\mathbf{X}$ satisfies a restricted eigenvalue condition. See Bühlmann and van de Geer [2011] for further discussions on the restricted eigenvalue condition and other similar conditions on $\mathbf{X}$ to guarantee that the design matrix is well behaved in high dimension.

The theory of Lasso suggests that λ is an important factor appearing in its estimation error bound. To achieve optimal estimation error bound, it is desirable to choose the smallest λ such that (5) holds. This choice, however, depends on both the unknown random error distribution and the structure of

the design matrix $\mathbf{X}$. As discussed above, a reasonable upper bound for such a theoretical choice of λ requires the knowledge of σ, the standard deviation of the random error. Estimation of σ in high dimension is itself a difficult problem. As a result, it is often infeasible to apply the theoretical choice of λ in real data problems.

2.3 Tuning parameter selection via cross-validation

In practice, a popular approach to selecting the tuning parameter λ for Lasso is a data-driven scheme called cross-validation, which aims for optimal prediction accuracy. Its basic idea is to randomly split the data into a training data set and a testing (or validation) data set such that one may evaluate the prediction error on the testing data while fitting the model using the training data set. There exist several different versions of cross-validation, such as leave-k-out cross-validation, repeated random sub-sampling validation (also known as Monte Carlo cross-validation), and K-fold cross-validation. Among these options, K-fold cross-validation is most widely applied in real-data analysis.

The steps displayed in Algorithm 1 illustrate the implementation of K-fold cross-validation for Lasso. The same idea broadly applies to more general problems such as penalized likelihood estimation with different penalty functions. First, the data is randomly partitioned into K roughly equal-sized subsets (or folds), where typical choice of K is 5 or 10. Given a value of λ, one of the K folds is retained as the validation data set to evaluate the prediction error, and the remaining data are used as the training data set to obtain $\widehat{\boldsymbol{\beta}}^{\text{Lasso}}(\lambda)$. This cross-validation process is then repeated, with each of the K folds being used as the validation data set exactly once. For example, in carrying out a 5-fold cross-validation for Lasso, we randomly split the data into five roughly equal-sized parts $\mathcal{V}_1, \cdots, \mathcal{V}_5$. Given a tuning parameter λ, we first train the model and estimate $\widehat{\boldsymbol{\beta}}^{\text{Lasso}}(\lambda)$ on $\{\mathcal{V}_2, \cdots, \mathcal{V}_5\}$ and then compute the total prediction error on $\mathcal{V}_1$. Repeat this process by training on $\{\mathcal{V}_1, \mathcal{V}_3, \mathcal{V}_4, \mathcal{V}_5\}$ and validating on $\mathcal{V}_2$, and so on. The cross-validation error $\text{CV}(\lambda)$ is obtained as the average of the prediction errors over the K validation data sets from this iterative process.

Given a set Λ of candidate tuning parameter values, say a grid $\{\lambda_1, \ldots, \lambda_m\}$, one would compute $\text{CV}(\lambda)$ according to Algorithm 1 for each $\lambda \in \Lambda$. This yields the cross-validation error curve $\{\text{CV}(\lambda) : \lambda \in \Lambda\}$. To select the optimal λ for Lasso, two useful strategies are usually recommended. A simple

Algorithm 1 K-fold cross-validation for Lasso

1: Randomly divide the data of sample size n into K folds, $\mathcal{V}_1, ...\mathcal{V}_K$, of roughly equal sizes.
2: Set $\text{Err}(\lambda) = 0$.
3: **for** $k = 1, \cdots, K$ **do**
4: Training dataset $(\mathbf{y}_T, \mathbf{X}_T) = \{(y_i, \mathbf{x}_i) : i \notin \mathcal{V}_k\}$.
5: Validation dataset $(\mathbf{y}_V, \mathbf{X}_V) = \{(y_i, \mathbf{x}_i) : i \in \mathcal{V}_k\}$.
6: $\widehat{\boldsymbol{\beta}}^{\text{Lasso}}(\lambda) \leftarrow \arg\min_{\boldsymbol{\beta}} \left\{(2|\mathcal{V}_k|)^{-1}||\mathbf{y}_T - \mathbf{X}_T\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_1\right\}$.
7: $\text{Err}(\lambda) \leftarrow \text{Err}(\lambda) + ||\mathbf{y}_V - \mathbf{X}_V\widehat{\boldsymbol{\beta}}^{\text{Lasso}}(\lambda)||^2$.
return $\text{CV}(\lambda) = n^{-1}\text{Err}(\lambda)$.

and intuitive approach is to select the λ that minimizes the cross-validation error, i.e.,

$$\widehat{\lambda} = \arg\min_{\lambda} \text{CV}(\lambda). \tag{5}$$

An alternative strategy is based on the so-called "one-standard-error rule", which chooses the most parsimonious model (here corresponding to larger λ and more regulation) such that its cross-validation error is within one standard-error of $\text{CV}(\widehat{\lambda})$. This is feasible as the K-fold cross-validation allows one to estimate the standard error of the cross-validation error. The second strategy acknowledges that the cross-validation error curve is estimated with error and is motivated by the principle of parsimony (e.g., Section 2.3 of Hastie et al. [2015]).

Several R functions are available to implement K-fold cross-validation for Lasso, such as the "cv.glmnet" function in the R package `glmnet` [Friedman et al., 2010] and the"cv.lars"function in the R package `lars` [Hastie and Efron, 2013]. Below are the sample R codes for performing the 5-fold cross-validation for Lasso using the "cv.glmnet" function.

```
library(glmnet)
data(SparseExample)
cvob1=cv.glmnet(x, y, nfolds=5)
plot(cvob1)
```

The plot produced by the above commands is given in Figure 1, which depicts the cross-validation error curve (based on the mean-squared prediction error in this example) as well as the one-standard-error band. In the plot, λ_{min} is

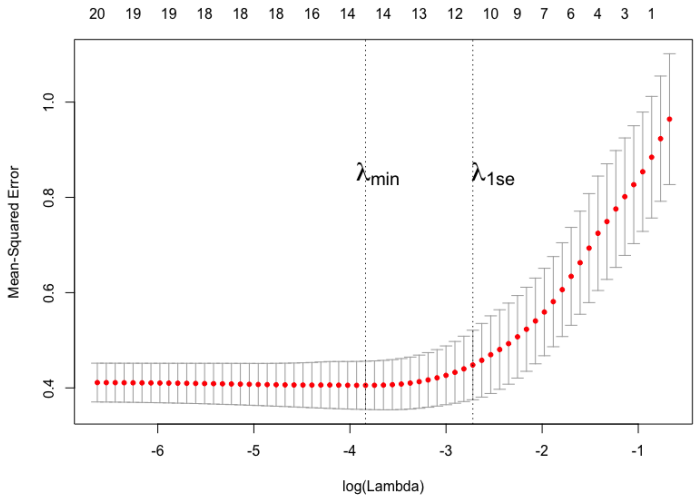

Figure 1: Cross-validation for Lasso

the tuning parameter obtained by (5), i.e., the value of the tuning parameter that minimizes the cross-validation prediction error. And λ_{1se} denotes the tuning parameter selected via the one-standard-error rule. The numbers at the top of the plot correspond to the numbers of non-zero coefficients (or model sizes) for models fitted with different λ values. For this data example, the prediction errors based on $\lambda_{\min}$ and λ_{1se} are close, while the model based on λ_{1se} is notably sparser than the one based on $\lambda_{\min}$.

In the existing work on the theory of Lasso, the tuning parameter is usually considered to be deterministic, or fixed at a pre-specified theoretical value. Despite the promising empirical performance of cross-validation, much less is known about the theoretical properties of the cross-validated Lasso, where the tuning parameter is selected in a data-driven manner. Some important progress has been made recently in understanding the properties of cross-validated Lasso. Homrighausen and McDonald [2013], Chatterjee and Jafarov [2015] and Homrighausen and McDonald [2017] investigated the risk consistency of cross-validated Lasso under different regularity conditions. Chetverikov et al. [2016] derived a nonasymptotic error bound for cross-validated Lasso and showed that it can achieve the optimal estimation rate up to a factor of order $\sqrt{\log(pn)}$.

2.4 Scaled Lasso: adapting to unknown noise level

As discussed earlier, the optimal choice of λ for Lasso requires the knowledge of the noise level σ, which is usually unknown in real data analysis. Motivated by Städler et al. [2010] and the discussions on this paper by Antoniadis [2010] and Sun and Zhang [2010], Sun and Zhang [2012] thoroughly investigated the performance of an iterative algorithm named scaled Lasso, which jointly estimates the regression coefficients $\boldsymbol{\beta}_0$ and the noise level σ in a sparse linear regression model.

Denote the loss function for Lasso regression by

$$L_\lambda(\boldsymbol{\beta}) = (2n)^{-1}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_1. \tag{6}$$

The iterative algorithm for scaled Lasso is described in Algorithm 2, where $\boldsymbol{\beta}^0$ and λ_0 are initial values independent of $\boldsymbol{\beta}_0$ and σ. In this algorithm, the tuning parameter is rescaled iteratively. In the output of the algorithm, $\widehat{\boldsymbol{\beta}}(\mathbf{X}, \mathbf{y})$ is referred to as the scaled Lasso estimator.

Algorithm 2 Scaled Lasso algorithm (Sun and Zhang, 2012)

Input $(\mathbf{X}, \mathbf{y})$, $\boldsymbol{\beta}^0$, and λ_0.
$\boldsymbol{\beta} \leftarrow \boldsymbol{\beta}^0$.
while $L_\lambda(\boldsymbol{\beta}) \leq L_\lambda(\boldsymbol{\beta}^0)$ **do**
    $\boldsymbol{\beta}^0 \leftarrow \boldsymbol{\beta}$
    $\widehat{\sigma} \leftarrow n^{-1/2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^0||$.
    $\lambda \leftarrow \widehat{\sigma}\lambda_0$.
    $\boldsymbol{\beta} \leftarrow \arg\min_{\boldsymbol{\beta}} L_\lambda(\boldsymbol{\beta})$.
return $\widehat{\sigma}(\mathbf{X}, \mathbf{y}) \leftarrow \widehat{\sigma}$ and $\widehat{\boldsymbol{\beta}}(\mathbf{X}, \mathbf{y}) \leftarrow \boldsymbol{\beta}^0$.

Sun and Zhang [2012] showed that the outputs of Algorithm 2 converge to the solutions of a joint minimization problem, specifically,

$$(\widehat{\boldsymbol{\beta}}, \widehat{\sigma}) = \arg\min_{\boldsymbol{\beta},\sigma} \Big\{(2n\sigma)^{-1}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \sigma/2 + \lambda_0||\boldsymbol{\beta}||_1\Big\}. \tag{7}$$

This is equivalent to jointly minimizing Huber's concomitant loss function with the L_1 penalty [Owen, 2007, Antoniadis, 2010]. This loss function possesses the nice property of being jointly convex in $(\boldsymbol{\beta}, \sigma)$. It can also be shown that the solutions are scale-equivariant in $\mathbf{y}$, i.e., $\widehat{\boldsymbol{\beta}}(\mathbf{X}, c\mathbf{y}) = c\widehat{\boldsymbol{\beta}}(\mathbf{X}, \mathbf{y})$ and

$\widehat{\sigma}(\mathbf{X}, c\mathbf{y}) = |c|\widehat{\sigma}(\mathbf{X}, \mathbf{y})$ for any constant c. This property is practically important in data analysis. Under the Gaussian assumption and other mild regularity conditions, Sun and Zhang [2012] derived oracle inequalities for prediction and joint estimation of σ and $\boldsymbol{\beta}_0$ for the scaled lasso, which in particular imply the consistency and asymptotic normality of $\widehat{\sigma}(\mathbf{X}, \mathbf{y})$ as an estimator for σ.

The function "scalreg" in the R package `scalreg` implements Algorithm 2 for the scaled Lasso. The sample codes below provide an example on how to analyze the "sp500" data in that package with scaled Lasso.

```
library(scalreg)
data(sp500)
attach(sp500)
x = sp500.percent[,3: (dim(sp500.percent)[2])]
y = sp500.percent[,1]
scaleob <- scalreg(x, y)
```

3 Alternative L_1-penalty based methods: from tuning selection to tuning free

This section provides a brief review of several recently proposed L_1-penalty based tuning-free procedures for high-dimensional sparse linear regression. These procedures tackle the challenge of tuning parameter selection for Lasso from different angles. As suggested by (5), the theoretically optimal tuning parameter for Lasso depends on both the design matrix $\mathbf{X}$ and the unknown error distribution (standard deviation σ, tail behavior, etc). The three procedures we review here (square-root Lasso, TREX and Rank Lasso) aim to automatically adapt to one or more aspects of these factors.

3.1 Scale-free square-root Lasso

Square-root Lasso is a variant of Lasso proposed by Belloni et al. [2011] which enjoys the advantage to avoid calibrating the tuning parameter with respect to the noise level σ. Square-root Lasso replaces least squares loss (or L_2 loss) function in Lasso with its positive square root. Assuming the ϵ_i are independently distributed with mean zero and variance σ^2, the square-root

Lasso estimator is defined as

$$\widehat{\boldsymbol{\beta}}_{\sqrt{\text{Lasso}}}(\lambda) = \arg\min_{\boldsymbol{\beta}} \Big\{ n^{-1/2} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}|| + \lambda ||\boldsymbol{\beta}||_1 \Big\}. \tag{8}$$

Let $L_{\text{SR}}(\boldsymbol{\beta}) = n^{-1/2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||$ denote the loss function of square-root Lasso and let S_{SR} denote its subgradient evaluated at $\boldsymbol{\beta} = \boldsymbol{\beta}_0$. The general principal of tuning parameter selection (e.g., Bickel et al. [2009]) suggests to choose λ such that $P(\lambda > c||S_{\text{SR}}||_\infty) \geq 1 - \alpha_0$, for some constant $c > 1$ and a given small $\alpha_0 > 0$. An important observation that underlies the advantage of square-root Lasso is that

$$S_{\text{SR}} = \frac{n^{-1} \sum_{i=1}^n \mathbf{x}_i \epsilon_i}{(n^{-1} \sum_{i=1}^n \epsilon_i^2)^{1/2}}$$

does not depend on σ.

Computationally, the square-root lasso can be formulated as a solution to a convex conic programming problem. The function "slim" in the R package `flare` [Li et al., 2018] implements a family of Lasso variants for high-dimensional regression, including the square-root Lasso. The sample codes below demonstrate how to implement the square-root Lasso using this function to analyze the "sp500" data in the `scalreg` package. The arguments `method="lq", q = 2` yield square-root Lasso, which are also the default options in the "slim" function.

```
library(flare)
data(sp500)
attach(sp500)
x = sp500.percent[,3: (dim(sp500.percent)[2])]
y = sp500.percent[,1]
sqrtob <- slim(x, y, method="lq", q = 2)
```

Belloni et al. [2011] recommended the choice $\lambda = cn^{-1/2}\Phi^{-1}\Big(1 - \frac{\alpha}{2p}\Big)$, for some constant $c > 1$ and $\alpha > 0$. Note that this choice of λ does not depend on σ, and it is valid asymptotically without requiring the random errors to be Gaussian. Under general regularity conditions, Belloni et al. [2011] showed that there exists some positive constant C_n such that

$$P\Big(||\widehat{\boldsymbol{\beta}}_{\sqrt{\text{Lasso}}}(\lambda) - \boldsymbol{\beta}_0|| \leq C_n \sigma \big\{ n^{-1} s \log(2p/\alpha) \big\}^{1/2} \Big) \geq 1 - \alpha,$$

where $s = ||\boldsymbol{\beta}_0||_0$ is the sparsity size of the true model. Hence, square root Lasso achieves the near-oracle rate of Lasso even when σ is unknown.

The square-root Lasso and Lasso are equivalent families of estimators. There exists a one-to-one mapping between the tuning parameter paths of square-root Lasso and Lasso [Tian et al., 2018]. It is also worth pointing out that the square-root Lasso is related to but should not be confused with the scaled Lasso [Sun and Zhang, 2012]. The current literature contain some confusion (particularly in the use of names) about these two methods. The connection and distinction between them are nicely discussed in Section 3.7 of Van de Geer [2016].

3.2 TREX

The scaled Lasso and square-root Lasso both address the need to calibrate λ for σ. However, the tail behavior of the noise vector, as well as the structure of the design matrix, could also have significant effects on the optimal selection of λ. To alleviate these additional difficulties, Lederer and Müller [2015] proposed a new approach for high-dimensional variable selection. The authors named the new approach TREX to emphasize that it aims at Tuning-free Regression that adapts to the Entire noise and the design matrix $\mathbf{X}$. Indeed, the most attractive property of TREX is that it automatically adjusts λ for the unknown noise standard deviation σ, the tail of the error distribution and the design matrix.

In contrast to the square-root Lasso, the TREX estimator modifies the Lasso loss function in a different way. The TREX estimator is defined as

$$\widehat{\boldsymbol{\beta}}_{\text{TREX}} = \arg\min_{\boldsymbol{\beta}} \Big\{ L_{\text{TREX}}(\boldsymbol{\beta}) + ||\boldsymbol{\beta}||_1 \Big\}, \tag{9}$$

where

$$L_{\text{TREX}}(\boldsymbol{\beta}) = \frac{2||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2}{||\mathbf{X}^T(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})||_\infty}.$$

TREX does not require a tuning parameter. In this sense, it is a completely tuning-free procedure. Lederer and Müller [2015] proved that the TREX estimator is close to a Lasso solution with tuning parameter of the same order as the theoretically optimal λ. They presented examples where TREX has promising performance comparing with Lasso.

The modified loss function for the TREX estimator, however, is no longer convex. Bien et al. [2016] showed the remarkable result that despite the non-convexity, there exists a polynomial-time algorithm that is guaranteed to find

the global minimum of the TREX problem. Bien et al. [2018] recently established a prediction error bound for TREX, which deepens the understanding of the theoretical properties of TREX.

3.3 Rank Lasso: a tuning free and efficient procedure

Recently, Wang et al. [2018] proposed an alternative approach to overcoming the challenges of tuning parameter selection for Lasso. The new method, named Rank Lasso, has an optimal tuning parameter that can be easily simulated and automatically adapts to both the unknown random error distribution and the structure of the design matrix. Moreover, it enjoys several other appealing properties: it is a solution to a convex optimization problem and can be conveniently computed via linear programming; it has similar performance as Lasso does when the random errors are normally distributed and is robust with substantial efficiency gain for heavy-tailed random errors; it leads to a scale-equivariant estimator which permits coherent interpretation when the response variable undergoes a scale transformation.

Specifically, the new estimator is defined as

$$\widehat{\boldsymbol{\beta}}_{\text{rank}}(\lambda) = \arg\min_{\boldsymbol{\beta}} \Big\{ Q_n(\boldsymbol{\beta}) + \lambda ||\boldsymbol{\beta}||_1 \Big\}, \tag{10}$$

where the loss function

$$Q_n(\boldsymbol{\beta}) = [n(n-1)]^{-1} \sum \sum_{i \neq j} \big| (y_i - \mathbf{x}_i^T \boldsymbol{\beta}) - (y_j - \mathbf{x}_j^T \boldsymbol{\beta}) \big|. \tag{11}$$

The loss function $Q_n(\boldsymbol{\beta})$ is related to Jaeckel's dispersion function with Wilcoxon scores [Jaeckel, 1972] in the classical nonparametric statistics literature. For this reason, the estimator in (10) is referred to as the rank Lasso estimator. In the classical low-dimensional setting, regression with Wilcoxon loss function was investigated by Wang [2009] and Wang and Li [2009].

To appreciate its tuning free property, we observe that the gradient function of $Q_n(\boldsymbol{\beta})$ evaluated at $\boldsymbol{\beta}_0$ is

$$\mathbf{S}_n := \frac{\partial Q_n(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}\Big|_{\boldsymbol{\beta}=\boldsymbol{\beta}_0} = -2[n(n-1)]^{-1} \mathbf{X}^T \boldsymbol{\xi},$$

where $\boldsymbol{\xi} = (\xi_1, \cdots, \xi_n)^T$ with $\xi_i = 2r_i - (n+1)$ and $r_i = \text{rank}(\epsilon_i)$ among $\epsilon_1, \cdots, \epsilon_n$. Note that the random vector $\{r_1, \cdots, r_n\}$ follows the uniform

distribution on the permutations of integers $\{1, \cdots, n\}$. Consequently, $\boldsymbol{\xi}$ has a completely known distribution that is independent of the random error distribution. Hence, the gradient function has the complete pivotal property [Parzen et al., 1994], which implies the tuning-free property of rank-Lasso. To see this, recall that the general principal of tuning parameter selection [Bickel et al., 2009] suggests choosing λ such that $P(\lambda > c||\mathbf{S}_n||_\infty) \geq 1 - \alpha_0$ for some constant $c > 1$ and a given small $\alpha_0 > 0$. With the design matrix $\mathbf{X}$ and a completely known distribution of $\boldsymbol{\xi}$, we can easily simulate the distribution of $\mathbf{S}_n$ and hence compute the theoretically optimal λ.

Wang et al. [2018] established a finite-sample estimation error bound for the Rank Lasso estimator with the aforementioned simulated tuning parameter and showed that it achieves the same optimal near-oracle estimation error rate as Lasso does. In contrast to Lasso, the conditions required by rank Lasso for the error distribution are much weaker and allow for heavy-tailed distributions such as Cauchy distribution. Moreover, they proved that further improvement in efficiency can be achieved by a second-stage enhancement with some light tuning.

4 Other alternative tuning parameter selection methods for Lasso

4.1 Bootstrap-based approach

Hall et al. [2009] developed an m-out-of-n bootstrap algorithm to select the tuning parameter for Lasso, pointing out that standard bootstrap methods would fail. Their algorithm employs a wild bootstrap procedure (see Algorithm 3), which allows one to estimate the mean squared error of the parameter estimators for different tuning parameters. For each candidate λ, this algorithm computes the bootstrapped mean-square error estimate $\mathrm{Err}(\lambda)$. The optimal tuning parameter is chosen as $\widehat{\lambda}_{\text{boots}} = \big(n/m\big)^{1/2} \arg\min_{\lambda} \mathrm{Err}(\lambda)$. The final estimator for $\boldsymbol{\beta}_0$ is given by

$$\widehat{\boldsymbol{\beta}}_{\text{boots}} = \arg\min_{\boldsymbol{\beta}} \Big\{ \sum_{i=1}^{n} (y_i - \bar{y} - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \widehat{\lambda}_{\text{boots}} ||\boldsymbol{\beta}||_1 \Big\}.$$

Their method and theory were mostly developed for the $p < n$ case. The algorithm requires that the covariates are centered at their empirical means

Algorithm 3 Bootstrap algorithm

Input $(\mathbf{y}, \mathbf{X})$, a $\sqrt{n}$-consistent "pilot estimator" $\widetilde{\boldsymbol{\beta}}$, and λ.
$\widehat{\epsilon}_i \leftarrow y_i - \bar{y} - \mathbf{x}_i^T \widetilde{\boldsymbol{\beta}}$.
$\widetilde{\epsilon}_i \leftarrow \widehat{\epsilon}_i - n^{-1} \sum_{j=1}^n \widehat{\epsilon}_j$.
Set $\mathrm{Err}(\boldsymbol{\lambda}) \leftarrow 0$.
for k = 1,...,N **do**
    Obtain $\epsilon_1^*, ..., \epsilon_m^*$ by sampling randomly from $\widetilde{\epsilon}_1, ..., \widetilde{\epsilon}_n$ with replacement.
    $y_i^* \leftarrow \bar{y} + \mathbf{x}_i^T \widetilde{\boldsymbol{\beta}} + \epsilon_i^*$, $i = 1, ..., m$.
    $\widehat{\boldsymbol{\beta}}^*(\lambda) \leftarrow \arg\min_{\boldsymbol{\beta}} \big\{ \sum_{i=1}^m (y_i^* - \bar{y}^* - \mathbf{x}_i^T \boldsymbol{\beta})^2 + \lambda ||\boldsymbol{\beta}||_1 \big\}$.
    $\mathrm{Err}(\lambda) \leftarrow \mathrm{Err}(\lambda) + ||\widehat{\boldsymbol{\beta}}^*(\lambda) - \widetilde{\boldsymbol{\beta}}||^2$.
return $\mathrm{Err}(\lambda)$.

and that a $\sqrt{n}$-consistent "pilot estimator" $\widetilde{\boldsymbol{\beta}}$ is available. Hall et al. [2009] proved that if $m = O(n/(\log n)^{1+\eta})$ for some $\eta > 0$, then the estimator $\widehat{\boldsymbol{\beta}}_{\text{boots}}$ can identify the true model with probability approaching one as $n \to \infty$. They also suggested that the theory can be generalized to the high dimensional case with fixed sparsity $||\boldsymbol{\beta}_0||_0$, however, the order of p would depend on the "generalized parameters" of the model such as the tail behaviors of the random noise.

Chatterjee and Lahiri [2011] proposed a modified bootstrap method for Lasso. This method first computes a thresholded version of the Lasso estimator and then applies the residual bootstrap. In the classical $p \ll n$ setting, Chatterjee and Lahiri [2011] proved that the modified bootstrap method provides valid approximation to the distribution of the Lasso estimator. They further recommended to choose λ to minimize the bootstrapped approximation to the mean squared error of the Lasso estimator.

4.2 Adaptive calibration for l_∞

Motivated by Lepski's method for non-parametric regression [Lepski, 1991, Lepski and Spokoiny, 1997], Chichignoud et al. [2016] proposed a novel adaptive validation method for tuning parameter selection for Lasso. The method, named Adaptive Calibration for l_∞ (AV_∞), performs simple tests along a single Lasso path to select the optimal tuning parameter. The method is equipped with a fast computational routine and theoretical guarantees on its

finite-sample performance with respect to the super-norm loss.

Let $\Lambda = \{\lambda_1, ..., \lambda_N\}$ be a set of candidate values for λ, where $0 < \lambda_1 < \cdots < \lambda_N = \lambda_{\max} = 2n^{-1}||\mathbf{X}^T\mathbf{y}||_\infty$. Denote $\widehat{\boldsymbol{\beta}}^{\text{Lasso}}(\lambda_j)$ as the Lasso estimator in (3) with tuning parameter set as $\lambda = \lambda_j$, $j = 1, \cdots, N$. The proposed AV_∞ selects λ based on the tests for sup-norm differences of Lasso estimates with different tuning parameters. It is defined as

$$\widehat{\lambda}_{\text{AC}} = \min\Big\{\lambda \in \Lambda : \max_{\substack{\lambda', \lambda'' \in \Lambda \\ \lambda', \lambda'' \geq \lambda}} \Big[\frac{||\widehat{\boldsymbol{\beta}}^{\text{Lasso}}(\lambda') - \widehat{\boldsymbol{\beta}}^{\text{Lasso}}(\lambda'')||_\infty}{\lambda' + \lambda''} - \bar{C}\Big] \leq 0\Big\}, \quad (12)$$

where $\bar{C}$ is a constant with respect to the L_∞ error bound of Lasso estimator. Chichignoud et al. [2016] recommended the universal choice $\bar{C} = 0.75$ for all practical purposes.

Chichignoud et al. [2016] proposed a simple and fast implementation for the tuning parameter selection via AV_∞, see the description in Algorithm 4, where in the algorithm the binary random variable $\widehat{t}_{\lambda_j}$ is defined as

$$\widehat{t}_{\lambda_j} = \prod_{k=j}^{N} \mathbb{1}\Big\{\frac{||\widehat{\boldsymbol{\beta}}(\lambda_j) - \widehat{\boldsymbol{\beta}}(\lambda_k)||_\infty}{\lambda_j + \lambda_k} \leq \bar{C}\Big\}, \quad j = 1, \ldots, N,$$

with $\mathbb{1}$ being the indicator function. The final estimator for the AV_∞ method is the Lasso estimator with the tuning parameter $\widehat{\lambda}_{\text{AC}}$, denoted as $\widehat{\boldsymbol{\beta}}(\widehat{\lambda}_{\text{AC}})$. As shown in Algorithm 4, AV_∞ only needs to compute one solution path, in contrast to the K paths in the K-fold cross-validation for Lasso. The new method is usually faster than cross-validation. Chichignoud et al. [2016] proved that $||\widehat{\boldsymbol{\beta}}(\widehat{\lambda}_{\text{AC}}) - \boldsymbol{\beta}_0||_\infty$ achieves the optimal sup-norm error bound of Lasso up to a constant pre-factor with high probability under some regularity conditions.

Algorithm 4 AV_∞ algorithm

1: Input $\widehat{\boldsymbol{\beta}}(\lambda_1), ..., \widehat{\boldsymbol{\beta}}(\lambda_N), \bar{C}$.
2: Set $j \leftarrow N$.
3: **while** $\widehat{t}_{\lambda_{j-1}} \neq 0$ and $j > 1$ **do**
4: Update index $j \leftarrow j - 1$.
return $\widehat{\lambda} \leftarrow \lambda_j$.

5 Nonconvex penalized high-dimensional regression and tuning for support recovery

5.1 Background

Lasso is known to achieve accurate prediction under rather weak conditions [Greenshtein and Ritov, 2004]. However, it is also widely recognized that Lasso requires stringent conditions on the design matrix $\mathbf{X}$ to achieve variable selection consistency [Zou, 2006, Zhao and Yu, 2006b]. In many scientific problems, it is of importance to identify relevant or active variables. For example, biologists are often interested in identifying the genes associated with certain disease. This problem is often referred to as *support recovery*, with the goal to identify $\mathcal{S}_0 = \{j : \beta_{0j} \neq 0, j = 1, \ldots, p\}$.

To alleviate the bias of Lasso due to the over-penalization of L_1 penalty, nonconvex penalized regression has been studied in the literature as an alternative to Lasso [Fan and Lv, 2010, Zhang and Zhang, 2012]. Two popular choices of nonconvex penalty functions are SCAD [Fan and Li, 2001] and MCP [Zhang, 2010a]. The SCAD penalty function is given by

$$p_\lambda(|\beta_j|) = \begin{cases} \lambda|\beta_j|, & \text{if } |\beta_j| \leq \lambda, \\ \frac{2a\lambda|\beta_j| - \beta_j^2 - \lambda^2}{2(a-1)}, & \text{if } \lambda < |\beta_j| < a\lambda, \\ \frac{(a+1)\lambda^2}{2}, & \text{if } |\beta_j| \geq a\lambda, \end{cases} \tag{13}$$

where $a > 2$ is a constant and Fan and Li [2001] recommended the choice $a = 3.7$. The MCP penalty function is given by

$$p_\lambda(|\beta_j|) = \begin{cases} \lambda|\beta_j| - \frac{\beta_j^2}{2a}, & \text{if } |\beta_j| \leq a\lambda, \\ \frac{a\lambda^2}{2}, & \text{if } |\beta_j| > a\lambda. \end{cases} \tag{14}$$

where $a > 1$ is a constant. Figure 2 depicts the two penalty functions.

As cross-validation for Lasso aims for prediction accuracy, it tends to select a somewhat smaller tuning parameter (i.e., less regulation). The resulted model size hence is usually larger than the true model size. In the fixed p setting, Wang et al. [2007] proved that with a positive probability cross-validation leads to a tuning parameter that would yield an over-fitted model.

Recent research has shown that when nonconvex penalized regression is combined with some modified BIC-type criterion, the underlying model can

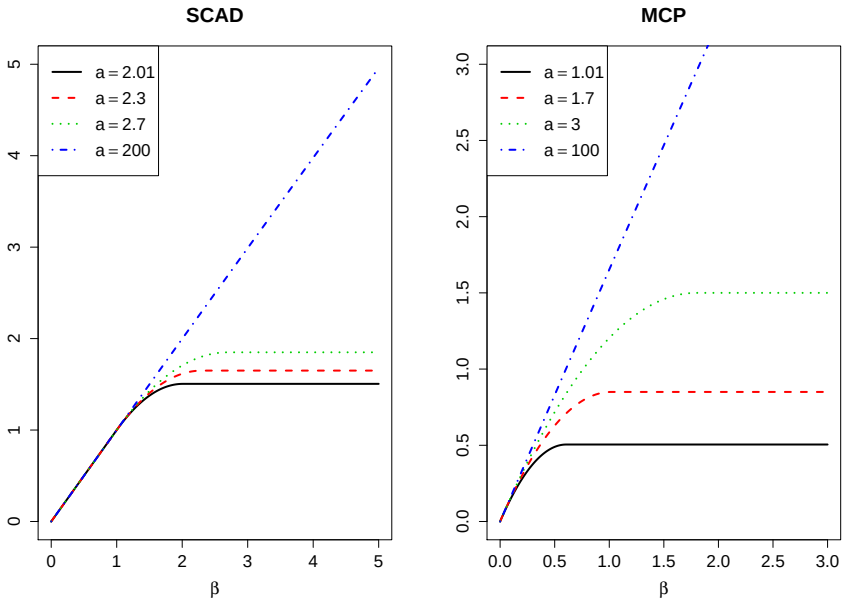

Figure 2: SCAD and MCP penalty functions ($\lambda = 1$)

be identified with probability approaching one under appropriate regularity conditions. Several useful results were obtained in the low-dimensional setting. For example, effective Bayesian information criterion (BIC) type criterion for tuning parameter selection for nonconvex penalized regression was investigated in Wang, Li and Tsai (2007) for fixed p and Wang, Li and Leng (2009) for diverging p (but $p < n$). Zou et al. [2007] considered Akaike information criterion (AIC) and BIC type criterion based on the degrees of freedom for Lasso. Also in the fixed p setting, Zhang et al. [2010] studied generalized information criterion, encompassing AIC and BIC. They revealed that BIC-type selector enables identification of the true model consistently and that AIC-type selector is asymptotically loss efficient.

In the rest of this section, we review several modified BIC-type criteria in the high-dimensional setup ($p \gg n$) for tuning parameter selection with the goal of support recovery.

5.2 Extended BIC for comparing models when $p \gg n$

Let $\mathcal{S}$ be an arbitrary subset of $\{1, \cdots, p\}$. Hence, each $\mathcal{S}$ indexes a candidate model. Given the data $(\mathbf{X}, \mathbf{y})$, the classical BIC, proposed by Schwarz [1978], is defined as follows

$$\text{BIC}(\mathcal{S}) = -2 \log L_n\{\widehat{\boldsymbol{\beta}}(\mathcal{S})\} + ||\mathcal{S}||_0 \log n,$$

where $L_n(\cdot)$ is the likelihood function, $\widehat{\boldsymbol{\beta}}(\mathcal{S})$ is the maximum likelihood estimator for the model with support $\mathcal{S}$, and $||\mathcal{S}||_0$ is the cardinality of the set $\mathcal{S}$. Given different candidate models, BIC selects the model with support $\mathcal{S}$ such that $\text{BIC}(\mathcal{S})$ is minimized.

In the classical framework where p is small and fixed, it is known [Rao and Wu, 1989] that under standard conditions BIC is variable selection consistent, i.e., $\mathcal{S}_0$ is identified with probability approaching one as $n \to \infty$ if the true model is in the set of candidate models. However, in the large p setting, the number of candidate models grows exponentially fast in p. The classical BIC is no longer computationally feasible.

Chen and Chen [2008] was the first to rigorously study the extension of BIC to high-dimensional regression where $p \gg n$. They proposed an extended family of BIC of the form

$$\text{BIC}_\gamma(\mathcal{S}) = -2 \log L_n\{\widehat{\boldsymbol{\beta}}(\mathcal{S})\} + ||\mathcal{S}||_0 \log n + 2\gamma \log \binom{p}{||\mathcal{S}||_0}, \qquad (15)$$

where $\gamma \in [0, 1]$. Comparing with the classical BIC, the above modification incorporates the model size in the penalty term. It was proved that if $p = O(n^\kappa)$ for some constant κ, and $\gamma > 1 - (2\kappa)^{-1}$, then this extended BIC is variable selection consistent under some regularity conditions. Kim et al. [2012] also investigated variants of extended BIC for comparing models for high-dimensional least-squares regression.

5.3 HBIC for tuning parameter selection and support recovery

The extended BIC is most useful if a candidate set of models is provided and if the true model is contained in such a candidate set with high probability. One practical choice is to construct such a set of candidate models from a Lasso solution path. As Lasso requires stringent conditions on the design

matrix $\mathbf{X}$ to be variable selection consistent. It is usually not guaranteed that the Lasso solution path contains the oracle estimator, the estimator corresponding to support set $\mathcal{S}_0$. Alternatively, one may construct a set of candidate models from the solution path of SCAD or MCP. However, as the objective function of SCAD or MCP is nonconvex, multiple minima may be present. The solution path of SCAD or MCP hence may be nonunique and do not necessarily contain the oracle estimator. Even if a solution path is known to contain the oracle estimator, to find the optimal tuning parameter which yields the oracle estimator with theoretical guarantee is challenging in high dimension.

To overcome these difficulties, Wang et al. [2013] thoroughly studied how to calibrate non-convex penalized least squares regression to find the optimal tuning parameter for support recovery when $p \gg n$. Define a consistent solution path to be a path that contains the oracle estimator with probability approaching one. Wang et al. [2013] first proved that an easy-to-calculate calibrated CCCP (CCCP stands for ConCave Convex procedure) algorithm produces a consistent solution path. Furthermore, they proposed HBIC, a high-dimensional BIC criterion, and proved that it can be applied to the solution path to select the optimal tuning parameter which asymptotically identifies the oracle estimator. Let $\widetilde{\boldsymbol{\beta}}(\lambda)$ be the solution corresponding to λ on a consistent solution path, for example, the one obtained by the aforementioned calibrated nonconvex-penalized regression with SCAD or MCP penalty. HBIC selects the optimal tuning parameter λ in $\Lambda_n = \{\lambda : ||\widetilde{\boldsymbol{\beta}}(\lambda)||_0 \leq K_n\}$, where K_n is allowed to diverge to infinity, by minimizing

$$\text{HBIC}(\lambda) = \log\left\{\frac{1}{n}||\mathbf{y} - \mathbf{X}\widetilde{\boldsymbol{\beta}}(\lambda)||^2\right\} + ||\widetilde{\boldsymbol{\beta}}(\lambda)||_0 \frac{C_n \log p}{n}, \tag{16}$$

where C_n diverges to infinity. Wang et al. [2013] proves that if $C_n||\boldsymbol{\beta}_0||_0 \log p = o(n)$ and $K_n^2 \log p \log n = o(n)$, then under mild conditions, HBIC identifies the true model with probability approaching one. For example, one can take $C_n = \log(\log n)$. Note that the consistency is valid in the ultra-high dimensional setting, where p is allowed to grow exponentially fast in n.

In addition, Wang and Zhu [2011] studied a variant of HBIC in combination of a sure screening procedure. Fan and Tang [2013] investigated proxy generalized information criterion, a proxy of the generalized information criterion [Zhang et al., 2010] when $p \gg n$. They identified a range of complexity penalty levels such that the tuning parameter that is selected by optimizing the proxy generalized information criterion can achieve model

selection consistency.

6 A real data example

We consider the data set sp500 in the R package `scalreg`, which contains a year's worth of close-of-day data for most of the Standard and Poors 500 stocks. The response variable `sp500.percent` is the daily percentage change. The data set has 252 observations of 492 variables.

We demonstrate the performance of Lasso with K-fold cross validation, scaled Lasso and $\sqrt{\text{Lasso}}$ methods on this example. Other methods reviewed in this paper which do not yet have publicly available software packages are not implemented. We evaluate the performance of different methods based on 100 random splits. For each split, we randomly select half of the data to train the model, and then compute the L_1 and L_2-prediction errors and estimated model sizes on the other half of the data. For Lasso, we select the tuning parameter by 10-fold cross validation, using the R function "cv.glmnet" and the one-standard-error rule. For scaled Lasso, we apply the default tuning parameter selection method in the R function "scalreg", which is the quantile-based penalty level (lam0="quantile") introduced and studied in Sun and Zhang [2013]. For $\sqrt{\text{Lasso}}$ method, we use R function "slim" to train the model. However, the package does not have a build-in tuning parameter selection method. As the optimal tuning parameter depends on the tail behavior of the random error, it is also chosen by 10-fold cross validation.

Table 1 summarizes the averages and standard deviations of the L_1 and L_2- prediction errors and estimated model sizes for the three methods with 100 random splits. Lasso and $\sqrt{\text{Lasso}}$ have similar performance, though Lasso method tends to yield sparser models. Scaled Lasso has slightly larger prediction errors and model sizes. The difference may be due to the non-normality of th data, which would affect the performance of the default tuning parameter selection method in the "scalreg" function.

7 Discussions

Developing computationally efficient tuning parameter selection methods with theoretical guarantees is important for many high-dimensional statis-

Table 1: Analysis of sp500 data

	Lasso	Scaled Lasso	$\sqrt{\text{Lasso}}$
L_1 error	0.17 (0.02)	0.21 (0.02)	0.17 (0.02)
L_2 error	0.05 (0.01)	0.08 (0.03)	0.05 (0.01)
Sparsity	60.03 (5.39)	120.82 (4.70)	76.63 (8.27)

tical problems but has so far only received limited attention in the current literature. This paper reviews several commonly used tuning parameter selection approaches for high-dimensional linear regression and provides some insights on how they work. The aim is to bring more attention to this important topic to help stimulate future fruitful research in this direction.

The review article focused on regularized least squares types of estimation procedures for sparse linear regression. The specific choice of tuning parameter necessarily depends on the user's own research objectives: Is prediction the main research goal? Or is identifying relevant variables of more importance? How much computational time is the researcher willing to allocate? Is robustness of any concern for the data set under consideration?

The problem of tuning parameter selection is ubiquitous and has been investigated in settings beyond sparse linear least squares regression. Lee et al. [2014] extended the idea of extended BIC to high-dimensional quantile regression. They recommended to select the model that minimizes

$$\text{BIC}_{\text{Q}}(\mathcal{S}) = \log\Big\{\sum_{i=1}^{n}\rho_\tau\big(y_i - \mathbf{x}_i^T\widehat{\boldsymbol{\beta}}(\mathcal{S})\big)\Big\} + (2n)^{-1}C_n||\mathcal{S}||_0\log n, \qquad (17)$$

where $\rho_\tau(u) = 2u\big(\tau - I(u<0)\big)$ is the quantile loss function, and C_n is some positive constant that diverges to infinity as n increases. They also proved variable selection consistency property when $C_n\log n/n \to 0$ under some regularity conditions. Belloni and Chernozhukov [2011] and Koenker [2011] considered tuning parameter selection for penalized quantile regression based on the pivotal property of the quantile score function.Wang et al. [2012] considered tuning parameter selection using cross-validation with the quantile loss function. For support vector machines (SVM), a widely used approach for classification, Zhang et al. [2016] recently established the consistency of extended BIC type criterion for tuning parameter selection in the high-dimensional setting. For semiparametric regression models, Xie et al. [2009] explored cross-validation for high-dimensional partially linear mean

regression; Sherwood et al. [2016] applied an extended BIC type criterion for high-dimensional partially linear additive quantile regression. Datta et al. [2017] derived a corrected cross-validation procedure for high-dimensional linear regression with error in variables. In Guo et al. [2016], an extended BIC type criterion was used for high-dimensional and banded vector autoregressions. In studying high-dimensional panel data, Kock [2013] empirically investigated both cross validation and BIC for tuning parameter selection.

Although the basic ideas of cross validation and BIC can be intuitively generalized to more complex modeling settings, their theoretical justifications are often still lacking despite the promising numerical evidence. It is worth emphasizing that intuition is not always straightforward and theoretical insights can be valuable. For instance, when investigating high-dimensional graphs and variable selection with the lasso, Meinshausen and Bühlmann [2006] observed that the consistency of neighborhood selection hinges on the choice of the penalty parameter. The oracle value for optimal prediction does not lead to a consistent neighborhood estimate.

References

A. Antoniadis. Comments on: l_1-penalization for mixture regression models. *Test*, 19(2):257–258, 2010.

A. Belloni and V. Chernozhukov. L1-penalized quantile regression in high-dimensional sparse models. *The Annals of Statistics*, 39:82–130, 2011.

A. Belloni, V. Chernozhukov, and L. Wang. Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics*, 37(4):1705–1732, 2009.

J. Bien, I. Gaynanova, J. Lederer, and C. Müller. Non-convex global minimization and false discovery rate control for the trex. *arXiv preprint arXiv:1604.06815*, 2016.

J. Bien, I. Gaynanova, J. Lederer, and C. L. Müller. Prediction error bounds for linear regression with the trex. *TEST*, pages 1–24, 2018.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

P. Bühlmann and S. van de Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Science & Business Media, 2011.

F. Bunea, A. Tsybakov, M. Wegkamp, et al. Sparsity oracle inequalities for the lasso. *Electronic Journal of Statistics*, 1:169–194, 2007.

E. Candes and T. Tao. The dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

E. J. Candès, Y. Plan, et al. Near-ideal model selection by l1 minimization. *The Annals of Statistics*, 37(5A):2145–2177, 2009.

A. Chatterjee and S. N. Lahiri. Bootstrapping lasso estimators. *Journal of the American Statistical Association*, 106(494):608–625, 2011.

S. Chatterjee and J. Jafarov. Prediction error of cross-validated lasso. *arXiv preprint arXiv:1502.06291*, 2015.

J. Chen and Z. Chen. Extended Bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771, 2008.

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM review*, 43(1):129–159, 2001.

D. Chetverikov, Z. Liao, and V. Chernozhukov. On cross-validated lasso. *arXiv preprint arXiv:1605.02214*, 2016.

M. Chichignoud, J. Lederer, and M. Wainwright. A practical scheme and fast algorithm to tune the lasso with optimality guarantees. *Journal of Machine Learning Research*, 17:1–20, 12 2016.

A. Datta, H. Zou, et al. Cocolasso for high-dimensional error-in-variables regression. *The Annals of Statistics*, 45(6):2400–2426, 2017.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle property. *Journal of the American Statistical Association*, 96: 1348–1360, 2001.

J. Fan and J. Lv. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101, 2010.

Y. Fan and C. Y. Tang. Tuning parameter selection in high dimensional penalized likelihood. *Journal of the Royal Statistical Society: Series B*, 75 (3):531–552, 2013.

J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

E. Greenshtein and Y. Ritov. Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli*, 10(6):971–988, 2004.

S. Guo, Y. Wang, and Q. Yao. High-dimensional and banded vector autoregressions. *Biometrika*, page asw046, 2016.

P. Hall, E. R. Lee, and B. U. Park. Bootstrap-based penalty choice for the lasso, achieving oracle performance. *Statistica Sinica*, 19(2):449–471, 2009.

T. Hastie and B. Efron. *lars: Least Angle Regression, Lasso and Forward Stagewise*, 2013. URL `https://CRAN.R-project.org/package=lars`. R package version 1.2.

T. Hastie, R. Tibshirani, and M. Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. Chapman and Hall/CRC, 2015.

T. R. Hastie, T. and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.)*. New York: Springer., 2009.

D. Homrighausen and D. McDonald. The lasso, persistence, and cross-validation. In *International Conference on Machine Learning*, pages 1031–1039, 2013.

D. Homrighausen and D. J. McDonald. Risk consistency of cross-validation with lasso-type procedures. *Statistica Sinica*, pages 1017–1036, 2017.

L. A. Jaeckel. Estimating regression coefficients by minimizing the dispersion of the residuals. *The Annals of Mathematical Statistics*, 43(5):1449–1458, 1972.

Y. Kim, S. Kwon, and H. Choi. Consistent model selection criteria on high dimensions. *Journal of Machine Learning Research*, 13(Apr):1037–1057, 2012.

A. B. Kock. Oracle efficient variable selection in random and fixed effects panel data models. *Econometric Theory*, 29(1):115–152, 2013.

R. Koenker. Additive models for quantile regression: Model selection and confidence bandaids. *Brazilian Journal of Probability and Statistics*, 25(3): 239–262, 2011.

J. Lederer and C. Müller. Don't fall for tuning parameters: Tuning-free variable selection in high dimensions with the trex, 2015.

E. R. Lee, H. Noh, and B. U. Park. Model selection via bayesian information criterion for quantile regression models. *Journal of the American Statistical Association*, 109(505):216–229, 2014.

O. Lepski. On a problem of adaptive estimation in gaussian white noise. *Theory of Probability & Its Applications*, 35(3):454–466, 1991.

O. V. Lepski and V. G. Spokoiny. Optimal pointwise adaptive methods in nonparametric estimation. *The Annals of Statistics*, 25(6):2512–2546, 1997.

X. Li, T. Zhao, L. Wang, X. Yuan, and H. Liu. *flare: Family of Lasso Regression*, 2018. URL `https://CRAN.R-project.org/package=flare`. R package version 1.6.0.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, pages 1436–1462, 2006.

S. N. Negahban, P. Ravikumar, M. J. Wainwright, B. Yu, et al. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

A. B. Owen. A robust hybrid of lasso and ridge regression. 2007.

M. Parzen, L. Wei, and Z. Ying. A resampling method based on pivotal estimating functions. *Biometrika*, 81(2):341–350, 1994.

R. Rao and Y. Wu. A strongly consistent procedure for model selection in a regression problem. *Biometrika*, 76(2):369–374, 1989.

G. Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 03 1978.

B. Sherwood, L. Wang, et al. Partially linear additive quantile regression in ultra-high dimension. *The Annals of Statistics*, 44(1):288–317, 2016.

N. Städler, P. Bühlmann, and S. Van De Geer. l_1-penalization for mixture regression models. *Test*, 19(2):209–256, 2010.

T. Sun and C.-H. Zhang. Comments on: l_1-penalization for mixture regression models. *Test*, 19(2):270–275, 2010.

T. Sun and C.-H. Zhang. Scaled sparse linear regression. *Biometrika*, 99(4): 879–898, 2012.

T. Sun and C.-H. Zhang. Sparse matrix inversion with scaled lasso. *J. Mach. Learn. Res.*, 14(1):3385–3418, Jan. 2013. ISSN 1532-4435. URL `http://dl.acm.org/citation.cfm?id=2567709.2567771`.

X. Tian, J. R. Loftus, and J. E. Taylor. Selective inference with unknown variance via the square-root lasso. *Biometrika*, 105(4):755–768, 2018.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

S. Van de Geer. Estimation and testing under sparsity. 2016.

S. A. Van de Geer et al. High-dimensional generalized linear models and the lasso. *The Annals of Statistics*, 36(2):614–645, 2008.

M. Wainwright. *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics, 2019.

H. Wang, R. Li, and C.-L. Tsai. Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika*, 94(3):553–568, 2007.

L. Wang. Wilcoxon-type generalized bayesian information criterion. *Biometrika*, 96(1):163–173, 2009.

L. Wang and R. Li. Weighted wilcoxon-type smoothly clipped absolute deviation method. *Biometrics*, 65(2):564–571, 2009.

L. Wang, Y. Wu, and R. Li. Quantile regression for analyzing heterogeneity in ultra-high dimension. *Journal of the American Statistical Association*, 107(497):214–222, 2012.

L. Wang, Y. Kim, and R. Li. Calibrating nonconvex penalized regression in ultra-high dimension. *Ann. Statist.*, 41:2505–2536, 2013.

L. Wang, B. Peng, J. Bradic, R. Li, and Y. Wu. A tuning-free robust and efficient approach to high-dimensional regression. Technical report, School of Statistics, University of Minnesota, 2018.

T. Wang and L. Zhu. Consistent tuning parameter selection in high dimensional sparse linear regression. *Journal of Multivariate Analysis*, 102(7): 1141–1151, 2011.

T. T. Wu, K. Lange, et al. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, 2(1):224–244, 2008.

H. Xie, J. Huang, et al. Scad-penalized regression in high-dimensional partially linear models. *The Annals of Statistics*, 37(2):673–696, 2009.

C. H. Zhang. Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics*, 38:894–942, 2010a.

C.-H. Zhang and T. Zhang. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, pages 576–593, 2012.

C.-H. Zhang, J. Huang, et al. The sparsity and bias of the lasso selection in high-dimensional linear regression. *The Annals of Statistics*, 36(4):1567–1594, 2008.

T. Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11(Mar):1081–1107, 2010b.

X. Zhang, Y. Wu, L. Wang, and R. Li. A consistent information criterion for support vector machines in diverging model spaces. *The Journal of Machine Learning Research*, 17(1):466–491, 2016.

Y. Zhang, R. Li, and C.-L. Tsai. Regularization parameter selections via generalized information criterion. *Journal of the American Statistical Association*, 105(489):312–323, 2010.

P. Zhao and B. Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006a.

P. Zhao and B. Yu. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7:2541–2563, 2006b. ISSN 1532-4435.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

H. Zou, T. Hastie, and R. Tibshirani. On the "degrees of freedom" of the lasso. *The Annals of Statistics*, 35(5):2173–2192, 2007.