A Tuning-free Robust and Efficient Approach to High-dimensional Regression

Lan Wang, Bo Peng, Jelena Bradic, Runze Li and Yunan Wu

Abstract

We introduce a novel approach for high-dimensional regression with theoretical guarantees. The new procedure overcomes the challenge of tuning parameter selection of Lasso and possesses several appealing properties. It uses an easily simulated tuning parameter that automatically adapts to both the unknown random error distribution and the correlation structure of the design matrix. It is robust with substantial efficiency gain for heavy-tailed random errors while maintaining high efficiency for normal random errors. Comparing with other alternative robust regression procedures, it also enjoys the property of being equivariant when the response variable undergoes a scale transformation. Computationally, it can be efficiently solved via linear programming. Theoretically, under weak conditions on the random error distribution, we establish a finite-sample error bound with a near-oracle rate for the new estimator with the simulated tuning parameter. Our results make useful contributions to mending the gap between the practice and theory of Lasso and its variants. We also prove that further improvement in efficiency can be achieved by a second-stage enhancement with some light tuning. Our simulation results demonstrate that the proposed methods often outperform cross-validated Lasso in various settings.

KEY WORDS: Efficiency, Heavy-tailed error, High dimension, Linear regression, Tuning parameter, Robustness.

[1] Lan Wang is Professor and Yunan Wu is graduate student, School of Statistics, University of Minnesota. Emails: wangx346@umn.edu and wuxx1375@umn.edu. Bo Peng is a data scientist at Adobe. Jelena Bradic is Associate Professor, Department of Mathematics and Halicioglu Data Science Institute, University of California at San Diego. Email: jbradic@ucsd.edu. Runze Li is Eberly Family Chair Professor, Department of Statistics, Pennsylvania State University. Email: rzli@psu.edu. Wang and Wu's research was supported by NSF DMS-1512267 and DMS-1712706. Bradic's research was supported by NSF DMS-1712481. Li's research was supported by NSF DMS 1820702 and DMS 1953196. The authors are indebted to the referees, the associate editor and the Co-editor for their valuable comments, which have significantly improved the paper.

1 Introduction

Consider a linear regression model

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

where $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ is an $n \times 1$ vector of responses, $\mathbf{X}$ is an $n \times p$ centered matrix of covariates, $\boldsymbol{\beta}_0 = (\beta_{01}, \ldots, \beta_{0p})^T$ is a $p \times 1$ vector of unknown parameters, and $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^T$ is an $n \times 1$ vector of independent and identically distributed random errors. We are interested in estimating $\boldsymbol{\beta}_0$ in the ultrahigh-dimensional setting, where the number of covariates (features) p can grow exponentially fast with the sample size n.

In the last decade, substantial progress has been achieved in high-dimensional regression analysis. In particular, L_1 regularized least squares regression techniques as represented by Lasso (Tibshirani, 1996; Chen et al., 2001), Dantzig selector (Candes and Tao, 2007), and their variants such as SCAD (Fan and Li, 2001), MCP (Zhang, 2010a), Capped L_1 (Zhang, 2010b), among others, have become popular tools. The literature in this area is vast. We refer to Bühlmann and van de Geer (2011) and the reviews of Fan and Lv (2010) and Zhang and Zhang (2012) for a fuller list of references. Despite the significant advances in algorithm and theory development, at least two challenges remain.

The first challenge is to determine the right amount of regularization in a computationally efficient way with proper theoretical justification. Practical performance of regularized high-dimensional regression depends crucially on the choice of tuning parameter λ, which prescribes the level of penalty or shrinkage. For Lasso, it is well understood that the optimal choice of λ depends on both the random error distribution and the design; see for example Meinshausen and Bühlmann (2006), Zhao and Yu (2006), Bunea et al. (2007), Van de Geer et al. (2008), Zhang and Huang (2008), Bickel et al. (2009) and Wainwright (2009). However, theory is often derived while fixing λ at an ideal value $\tau\sigma\sqrt{\log p/n}$, where σ is the standard deviation of the random error distribution and τ is some positive constant. Estimation of σ in high dimensions is itself a very difficult problem (Fan et al., 2012; Sun and Zhang, 2012; Dicker, 2014; Yu and Bien, 2019). To circumvent this difficulty, practitioners often employ cross-validation to select λ. Several recent work shed light on the properties of cross-validated Lasso, for example, Homrighausen and McDonald (2013), Chatterjee and Jafarov (2015), Homrighausen and McDonald (2017) and Chetverikov et al. (2016). Comparing with the corresponding theory for Lasso with fixed theoretical choice of λ, these results, however, generally require stronger regularity conditions and have less sharp bounds. Hence, there still exits an important gap between the theory and practice of Lasso.

The second challenge is concerned with how to properly handle heavy-tailed error contamination in high dimensions so that one achieves robustness while maintaining efficiency at the normal error setting. Heavy-tailed error contamination is ubiquitous in high-dimensional microarray data, climate data, insurance claim data, e-commerce data and many other applications. For such data, Gaussian or sub-Gaussian error assumption is rarely justified. Direct application of standard procedures can result in biased estimation and misleading conclusions. Heavy-tailed errors also affect the choice of tuning parameter.

It is important to note that the above two challenges are usually intertwined. Solutions focusing on only one aspect of the two issues run the risk of making the other aspect more challenging. Several authors (Fan et al., 2017; Loh, 2017; Sun et al., 2020) recently made important contributions to high-dimensional M-estimation based on Huber's loss which possesses desirable robustness properties. Lozano et al. (2016) proposed a minimum distance estimator. Avella-Medina and Ronchetti (2018) studied robust penalized quasilikelihood. However, these work have not fully addressed the challenge of selecting λ and may even require some additional tuning parameter to achieve robustness. For example, Huber's loss function contains an additional tuning parameter and in general penalized Huber regression requires cross-validation for tuning parameter selection. An alternative robust loss function is the least absolute deviation loss (or more generally, quantile loss), see, e.g., Belloni et al. (2011), Bradic et al. (2011), Wang et al. (2012), Wang (2013), and Fan et al. (2014). Although being robust, this loss may incur significant efficiency loss for normal random errors. On the other hand, an interesting stream of research has investigated how to alleviate the difficulty of selecting λ for Lasso. The scaled Lasso of Sun and Zhang (2012) iteratively estimates the regression parameter and σ. The square-root Lasso (Belloni et al., 2011) eliminates the need to calibrate λ for σ but does not adjust for the design matrix. TREX (Lederer and Müller, 2015; Bien et al., 2016, 2018) automatically adjusts λ for both the tail of the error distribution and the design matrix but the modified loss function is no longer convex. Sabourin et al. (2015) adopts a permutation approach and Chichignoud et al. (2016) develops a novel testing procedure to select λ. Yu and Bien (2019) proposed the organic Lasso and derived a prediction error bound under weaker assumptions on the design matrix with a theoretical choice of the tuning parameter that does not depend on σ. These, however, have not addressed the robustness challenge and may have sub-optimal performance for heavy-tailed errors. For example, the theory of scaled Lasso requires the Gaussian error assumption. Estimating σ is particularly challenging for high-dimensional regression with heavy-tailed errors.

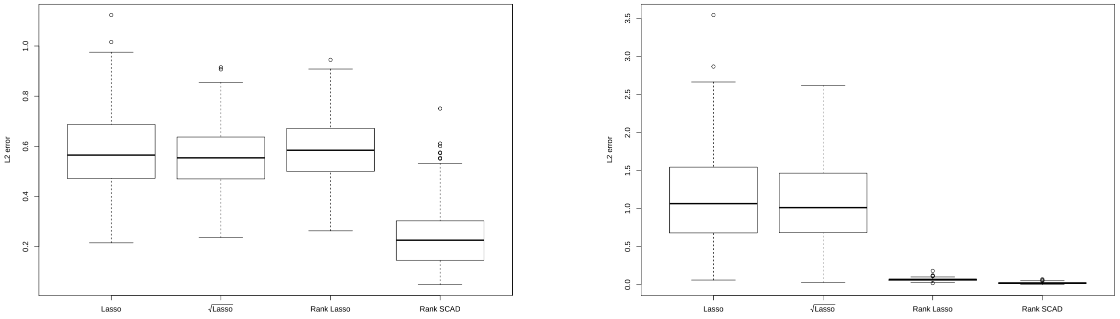
This paper makes a useful contribution to the high-dimensional regression literature by showing that a rather simple solution exists that addresses these two challenges simultaneously. Our new procedure is inspired by Jaeckel's dispersion function with Wilcoxon scores (Jaeckel, 1972), which plays an important role in classical nonparametric statistics due to its robustness and efficiency properties. In the low-dimensional setting ($p < n$), regression with Wilcoxon loss function was investigated by Wang and Li (2009) and Wang et al. (2009). However, they required computationally-intensive tuning and did not study the theory in high dimensions as is done in this paper. Here, we carefully develop a new L_1 regularized estimator based on this dispersion function with theoretical guarantees in the ultrahigh-dimensional setting. We demonstrate that the new estimator achieves several goals simultaneously.

- The new estimator is convenient to implement. It is the solution to a convex optimization problem and can be obtained by linear programming. Its tuning parameter λ can be easily simulated and automatically adjusts to both the random error distribution and the design matrix correlation structure without sacrificing the severity of vanilla assumptions.

- Theoretically, we derive a non-asymptotic L_2 estimation error bound for the L_1 regularized new estimator with simulated tuning parameter in ultrahigh dimensions under mild regularity conditions. Let q be the number of nonzero regression coefficients of the underlying model. We prove that with high probability the L_2 estimation error bound achieves the rate $O(\sqrt{q \log p/n})$, the same near-oracle bound for Lasso with the theoretical tuning parameter (see Theorem 1 in Section 2.3). However, we do not need the sub-Gaussian error assumption as Lasso does or the lower-order moment assumption as Huber-loss based procedures require.

- For random errors with distributions symmetric about zero, our modeling parameter $\mathbf{x}_i^T\boldsymbol{\beta}_0$ coincides with the conditional mean. However, we do not require the symmetric random error assumption. For i.i.d. random errors, since Jaeckel's dispersion function is invariant to a location change, $\mathbf{x}_i^T\boldsymbol{\beta}_0$ differs from the conditional mean only by a constant. In particular, the slopes in $\boldsymbol{\beta}_0$ still bear the same interpretation as the effects of the covariates on the conditional mean. This is different from some alternative robust methods which may imply an altered interpretation on what parameters are directly estimated due to the modified loss function.

The new estimator is very close to Lasso for normal random errors and is robust with substantial efficiency gain for heavy-tailed errors. Figure 1, corresponding to Example 1 in Section 4, provides an example of the robust and efficient behavior of the proposed new



(a) N(0,1) error

(b) MN error

Figure 1: Boxplots for L_2 estimation error for four different procedures in Example 1

estimators. The left panel of Figure 1 displays the boxplots of the L_2 estimation error of four different estimators for normal error distribution; while the plots in the right panel are for the heavy-tailed mixture normal error distribution. It is worth emphasizing that our conditions on the random error distribution are much weaker than the sub-Gaussian condition and permit heavy-tailed distributions such as Cauchy distribution. Due to the strong nonsmoothness of Jaeckel's dispersion function, advanced empirical processes techniques are needed to establish the theory for the new estimator comparing with that for the L_1 regularized least squares estimator.

As another contribution of the paper, we show that a second-stage enhancement can further improve the estimation efficiency due to the reduction of the bias induced by the L_1 penalty. This is motivated by the work on nonconvex penalized regression such as SCAD (Fan and Li, 2001), MCP (Zhang, 2010a), Capped L_1 (Zhang, 2010b), which recognize that Lasso tends to overshrink large coefficients and leads to biased estimate. This second step does require some tuning but fairly light. The tuning parameter for this step can be efficiently computed via a high-dimensional BIC procedure (Chen and Chen, 2008). Theoretically, we prove that with high probability the second-stage estimator possess the strong oracle property, that is, it is exactly equal to what one would obtain if the underlying data generative model is known in advance. With hight probability, the zero coefficients are estimated to be exactly zero. For estimating nonzero coefficients, we derive interesting efficiency results: the resulted estimator is almost as efficient as the oracle least squares estimator for normal random errors; and can be substantially more efficient for heavy-tailed random errors. Indeed, the asymptotic relative efficiency (ARE) is shown to be the same as that of the one-sample Wilcoxon test with respect to the t-test. This implies that the ARE is as high as 0.955 for normal error distribution, and can be significantly higher than one for many heavier-tailed distributions. In particular, the efficiency of using the Jaeckel's dispersion loss function with Wilcoxon score is about 1.5 times that of using the absolute deviation (or L_1 loss) function if the random error distribution is normal. In fact, the asymptotic efficiency of Jaeckel's dispersion loss function with Wilcoxon score amounts to what would be achieved when one combines quantile losses at infinitely many quantiles, see Section 3 for more discussions. We also rigorously establish the proposed high-dimensional BIC is consistent for variable selection.

The rest of the paper is organized as follows. In Section 2, we introduce the L_1 regularized new estimator based on Jaeckel's dispersion function with Wilcoxon scores and derive the nonasymptotic near-oracle L_2 estimation error bound. Section 3 introduces a second-stage enhancement for further bias reduction and efficiency improvement. It presents a strong oracle property and a consistency result for a high-dimensional BIC procedure. Monte Carlo simulations and a real data example are reported in Section 4 to demonstrate the superior performance of the proposed estimators. Section 5 concludes the paper with some discussions. The Appendix provides the proofs of the main theory. The supplement contains additional technical and numerical results.

2 The methodology

2.1 Background

In model (1), all parameters are allowed to depend on the sample size n, but the dependence is suppressed in the notation for simplicity. The regression parameter $\boldsymbol{\beta}_0$ is sparse in the sense that most of its components are zero.

The Lasso estimator is the solution to the regularized least squares minimization problem

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

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. The magnitude of λ controls the complexity of the model. A larger value of λ indicates heavier shrinkage.

The optimal choice of λ involves a careful trade-off. Motivated by the Karush-Kuhn-Tucker condition for convex optimization (Boyd and Vandenberghe, 2004), the general principal of tuning parameter selection for penalized regression (Bickel et al., 2009) suggests that λ should satisfy $\lambda \geq n^{-1}||\mathbf{X}^T\boldsymbol{\epsilon}||_\infty$, where $||\cdot||_\infty$ denotes the infinity norm. As an example, if the random errors are independent $N(0, \sigma^2)$ variables and the design matrix is normalized such that each column has L_2-norm equal to $\sqrt{n}$, then the above lower bound is satisfied with high probability by the choice $\lambda = \tau\sigma\sqrt{\log p/n}$ for some positive constant τ.

On the other hand, the theory of Lasso reveals that λ is an important factor appearing in its estimation error bound. Motivated by the subgradient condition of Lasso (Bickel et al., 2009; Bühlmann and van de Geer, 2011), it is known that on the event

$$\Big\{ 2n^{-1}||\mathbf{X}^T\boldsymbol{\epsilon}||_\infty \leq \lambda \Big\}, \tag{3}$$

Lasso enjoys the near-oracle error bound: $||\widehat{\boldsymbol{\beta}}^{\text{Lasso}}(\lambda) - \boldsymbol{\beta}_0||_2 \leq \gamma_0\sqrt{q}\lambda$, where $||\cdot||_2$ denotes the Euclidean norm of a vector, γ_0 is a constant depending on n and p only through the structure of the scaled Gram matrix $n^{-1}\mathbf{X}^T\mathbf{X}$, and q is the sparsity index or the number of nonzero coefficients in $\boldsymbol{\beta}_0$. This suggests it is desirable to choose a small λ such that the event (3) holds with high probability.

2.2 The new method

We will study the following L_1 regularized estimator of $\boldsymbol{\beta}_0$:

$$\widehat{\boldsymbol{\beta}}(\lambda) = \underset{\boldsymbol{\beta}\in\mathcal{R}^p}{\arg\min}\Big\{\big[n(n-1)\big]^{-1}\sum\sum_{i\neq j}\big|(Y_i - \mathbf{x}_i^T\boldsymbol{\beta}) - (Y_j - \mathbf{x}_j^T\boldsymbol{\beta})\big| + \lambda\sum_{k=1}^{p}|\beta_k|\Big\}. \qquad (4)$$

The loss function in (4) has its origin in classical nonparametric statistics, see, e.g., Hettmansperger and McKean (1998) for an introduction. Minimizing this loss is equivalent to minimizing Jaeckel's dispersion function (Jaeckel, 1972) with Wilcoxon scores:

$$\sqrt{12}\sum_{i=1}^{n}\left[\frac{R(Y_i - \mathbf{x}_i^T\boldsymbol{\beta})}{n+1} - \frac{1}{2}\right](Y_i - \mathbf{x}_i^T\boldsymbol{\beta}),$$

where $R(Y_i - \mathbf{x}_i^T\boldsymbol{\beta})$ denotes the rank of $Y_i - \mathbf{x}_i^T\boldsymbol{\beta}$ among $Y_1 - \mathbf{x}_1^T\boldsymbol{\beta}, \ldots, Y_n - \mathbf{x}_n^T\boldsymbol{\beta}$. This paper emphasizes other interesting and important features of Jaeckel's dispersion function in the high-dimensional regression setting. The new estimator $\widehat{\boldsymbol{\beta}}(\lambda)$ is the solution to a convex optimization problem and can be obtained by linear programming (see Section 4.2). In terms of statistical performance, we will show that the new estimator behaves very similarly as Lasso for normal random errors and remains robust with potential significant efficiency gains under heavy-tailed errors for which the cross-validated Lasso could break down.

First, the loss function in (4) is invariant to a location change. Under weak conditions, $\boldsymbol{\beta}_0$ is the minimizer of the population version of the loss function. This is different from most other robust methods which alter the least squares loss function by truncation or downweighting. The corresponding population parameter of the altered loss function may no longer be the regression parameter in the original model, see Fan et al. (2017). This property can be easily seen by noting that the population version of our loss function can be expressed as $\mathrm{E}\big|(\epsilon_i - \epsilon_j) - (\mathbf{x}_i - \mathbf{x}_j)^T(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\big|$, which is minimized at $\boldsymbol{\beta}_0$, as $\epsilon_i - \epsilon_j$ has a symmetric distribution about zero whenever the random errors are independent and identically distributed (i.i.d.). In model (1), the intercept term is absorbed into ϵ_i and can be identified with an additional location constraint on ϵ_i, however, what we estimate using this loss function does not depend on what location constraint is imposed on ϵ_i. In particular, for random errors with distributions symmetric about zero, $\mathbf{x}_i^T\boldsymbol{\beta}_0$ coincides with the conditional mean. However, symmetric random error distribution assumption is not required. For i.i.d. random errors, $\boldsymbol{\beta}_0$ still bears the interpretation as the effects of the covariates on the conditional mean. This is different from Huber's loss function which combines L_2 and L_1 loss with an additional tuning parameter. The minimizer of Huber loss approximates $\boldsymbol{\beta}_0$ when the additional tuning parameter is carefully tuned to diverge.

Second, it was noted in Parzen et al. (1994) in a different setting that the gradient function of our loss function is *completely pivotal*, which as we will show, leads to an appealing tuning-free property of the new estimator in the high-dimensional setting. This allows the procedure to circumvent the difficulty of tuning parameter selection. We write

$$Q_n(\boldsymbol{\gamma}) \quad = \quad \big[n(n-1)\big]^{-1}\sum\sum_{i\neq j}\big|(\epsilon_i - \epsilon_j) - (\mathbf{x}_i - \mathbf{x}_j)^T\boldsymbol{\gamma}\big|. \tag{5}$$

Then, the loss function in (4) is $Q_n(\boldsymbol{\beta} - \boldsymbol{\beta}_0)$. Denote the subgradient of $Q_n(\boldsymbol{\gamma})$ at $\boldsymbol{\gamma} = \mathbf{0}$ (or equivalently $\boldsymbol{\beta} = \boldsymbol{\beta}_0$) by $\mathbf{S}_n = \frac{\partial Q_n(\boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}}\big|_{\boldsymbol{\gamma}=\mathbf{0}}$. Motivated by the general principal of tuning parameter selection discussed in Section 2.1, we choose λ such that

$$P\big(\lambda > c||\mathbf{S}_n||_\infty\big) \geq 1 - \alpha_0, \tag{6}$$

for a given small $\alpha_0 > 0$ and a theoretical constant $c > 1$, where c is a theoretical constant that does not depend n or p.

Direct computation yields

$$\begin{aligned}\mathbf{S}_n &= \big[n(n-1)\big]^{-1}\sum\sum_{i\neq j}(\mathbf{x}_j-\mathbf{x}_i)\text{sign}(\epsilon_i-\epsilon_j)\\ &= -2\big[n(n-1)\big]^{-1}\sum_{j=1}^{n}\mathbf{x}_j\Big(\sum_{i=1,i\neq j}^{n}\text{sign}(\epsilon_j-\epsilon_i)\Big).\end{aligned}$$

where $\text{sign}(t)=1$ if $t>0$, $=-1$ if $t<0$, and $=0$ if $t=0$. Denote $\xi_j=\sum_{i=1,i\neq j}^{n}\text{sign}(\epsilon_j-\epsilon_i)$, $j=1,\ldots,n$. It is important to observe that ξ_j is closely related to the rank of ϵ_j among $\{\epsilon_1,\ldots,\epsilon_n\}$. Denote $\text{rank}(\epsilon_j)=r_j$, then

$$\xi_j=\sum_{i=1,i\neq j}^{n}\text{sign}(\epsilon_j-\epsilon_i)=(r_j-1)+(-1)(n-r_j)=2r_j-(n+1).$$

Lemma 1 below characterizes the distribution of the subgradient $\mathbf{S}_n$.

Lemma 1. *Assume model (1) holds, then* $\mathbf{S}_n=-2\big[n(n-1)\big]^{-1}\mathbf{X}^T\boldsymbol{\xi}$, *where the* n*-dimensional random vector* $\boldsymbol{\xi}=2\mathbf{r}-(n+1)$, *with* $\mathbf{r}$ *following the uniform distribution on the permutations of the integers* $\{1,2,\ldots,n\}$.

We refer to this property of the gradient function as the *completely pivotal* property, as it does not depend on the error distribution. It is straightforward to simulate the distribution of $||\mathbf{S}_n||_\infty$ since $\{r_1,\ldots,r_n\}$ can be simulated by generating a random permutation of the integers between 1 and n. Moreover, the simulations can be easily paralleled. For any given $c>1$ and α_0, we suggest to take λ equal to

$$\lambda^*=cG^{-1}_{||\mathbf{S}_n||_\infty}(1-\alpha_0) \tag{7}$$

where $G^{-1}_{||\mathbf{S}_n||_\infty}(1-\alpha_0)$ denotes the $(1-\alpha_0)$-quantile of the distribution of $||\mathbf{S}_n||_\infty$.

The simulated λ^* does not depend on the pre-estimation of any unknown population quantity and automatically adjusts to both the random error distribution and the structure of the design matrix $\mathbf{X}$. In the numerical studies in Section 4.1, we considered data generative models corresponding to a variety of error distributions and different covariance matrices for the distribution of $\mathbf{X}$. It is interesting to compare the above *completely pivotal* property of $\mathbf{S}_n$ with the *partial pivotal* property of square-root Lasso. The gradient function of the loss function of square-root Lasso, evaluated at $\boldsymbol{\beta}_0$, has the form $(\sum_{i=1}^n \epsilon_i^2)^{1/2} \sum_{i=1}^n \mathbf{x}_i \epsilon_i$, which does not depend on σ but its distribution still depends on the distribution of ϵ_i. As a result, square-root Lasso circumvents the difficulty of tuning λ for σ but does not adjust to other aspects of the error distribution nor the design matrix. Moreover, Hebiri and Lederer (2013) reveals that the standard tuning parameters that do not depend on the design matrix $\mathbf{X}$ may lead to sub-optimal performance of Lasso.

Remark 1. The work of Parzen et al. (1994) also reveals that the *completely pivotal* property also holds for the L_1 loss function. This was also recognized in Belloni et al. (2011) and Wang (2013) for penalized quantile regression. However, direct use of quantile loss may potentially result in significant efficiency loss for normal random errors. Indeed, the asymptotic efficiency analysis in Section 3 reveals that when the second-stage enhancement is implemented, the efficiency of using the Wilcoxon loss function is about 1.5 times that of using the absolute deviation (or L_1 loss) function if the random error distribution is normal for estimating the nonzero coefficients. Alternative robust loss functions such as Huber loss usually do not possess the pivotal property.

Remark 2. Let $\widehat{\boldsymbol{\beta}}(\lambda^*, \mathbf{Y}, \mathbf{X})$ be the estimator in (4) with the simulated tuning parameter λ^* in (7), given the response vector $\mathbf{Y}$ and the design matrix $\mathbf{X}$. It is easy to see $\widehat{\boldsymbol{\beta}}(\lambda^*, b\mathbf{Y}, \mathbf{X}) = b\widehat{\boldsymbol{\beta}}(\lambda^*, \mathbf{Y}, \mathbf{X})$ for any nonzero constant b. This equivariance property is convenient for coherent interpretation of results from regularized regression. This property is not shared by robust high-dimensional regression procedure based on Huber's loss function. Note that when $\mathbf{Y}$ has a scale change, the simulated tuning parameter λ^* remains the same when it is computed using the transformed data.

2.3 Near-oracle rate of the L_2 error bound

We consider the estimator $\widehat{\boldsymbol{\beta}}(\lambda^*)$, which is obtained by setting $\lambda = \lambda^*$ in (4). The main result of this subsection establishes that $\widehat{\boldsymbol{\beta}}(\lambda^*)$ enjoys the same near-oracle rate for the L_2 error bound as Lasso does when its λ is fixed at a theoretical value.

Let $A = \{j : \beta_{0j} \neq 0,\ j = 1, \ldots, p\}$ be the index set of nonzero coefficients in $\boldsymbol{\beta}_0$. The cardinality $||A||_0 = q$ is the sparsity size of the underlying data generative model, where $||\cdot||_0$ denotes the L_0 norm and q is allowed to depend on the sample size n. For a given index set $B \in \{1, 2, \ldots, p\}$, let $\mathbf{x}_B$ denote the p-dimensional vector that has the same coordinates as $\mathbf{x}$ on the index set B and zero coordinates on B^c. For a matrix D, we use $\xi_{\min}(D)$ and $\xi_{\max}(D)$ to denote the smallest eigenvalue and the largest eigenvalue of D, respectively.

For the constant c in (6), define $\bar{c} = \frac{c+1}{c-1}$ and consider the following cone set

$$\Gamma = \Big\{\boldsymbol{\gamma} \in \mathbf{R}^p : ||\boldsymbol{\gamma}_{B^c}||_1 \leq \bar{c}||\boldsymbol{\gamma}_B||_1,\ B \subset \{1, 2, \ldots, p\} \text{ and } ||B||_0 \leq q\Big\}.$$

We impose the following regularity conditions to facilitate our technical derivation.

(C1) (Conditions on the design) There exists a positive constant b_1 such that $|x_{ij}| \leq b_1$ for all $1 \leq i \leq n$, $1 \leq j \leq p$. The covariates are empirically centered in the sense that $n^{-1}\sum_{i=1}^n x_{ij} = 0$, for $1 \leq j \leq p$.

(C2) (Lower restricted eigenvalue condition) There exist some positive constant b_2 such that

$$\inf_{\boldsymbol{\gamma}\in\Gamma} \frac{n^{-1}\boldsymbol{\gamma}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\gamma}}{||\boldsymbol{\gamma}||_2^2} \geq b_2.$$

(C3) (Conditions on the random error) The random errors ϵ_i are independent and identically distributed with density function $f(\cdot)$. Let $\zeta_{ij} = \epsilon_i - \epsilon_j$, $1 \leq i \neq j \leq n$. Let $F^*(\cdot)$ denote the distribution function of ζ_{ij} and let $f^*(\cdot)$ denote the corresponding probability density function. There exists a positive constants b_3 such that $f^*(t) \geq b_3$ uniformly in $\{t : |t| \leq q\sqrt{\log p/n}\}$.

The above conditions are mild for theoretical analysis of high-dimensional regression. Condition (C1) is common for fixed design regression and can be relaxed under additional technicality. For example, we can allow b_1 to diverge to ∞ at an appropriate rate and the error bound derived in this paper still holds with probability approaching one. Alternatively, we can consider random designs and the results of the paper hold for sub-Gaussian design matrix. The lower restricted eigenvalue condition is also standard to studying the error bound for Lasso-type estimators. The lower bound only needs to hold for $\boldsymbol{\gamma}$ in the cone set Γ.

Remark 3. It is worth emphasizing that our assumptions on the random error distribution (condition (C3)) are considerably weaker than what are usually imposed in the literature for high-dimensional regression. Existing theoretical work on Lasso often requires ϵ_i to have a sub-Gaussian distribution. Although the class of sub-Gaussian distribution is large, it excludes many commonly encountered heavy-tailed distributions. For example, the χ^2-distribution is not sub-Gaussian. Square-root Lasso requires ϵ_i to have finite variance. Existing work on high-dimensional robust regression based on Huber loss also imposes moment conditions on ϵ_i, for example, Fan et al. (2017) assumes $E(|\epsilon_i|^k)$ is bounded for some $k \geq 2$, and Sun et al. (2020) assumes $E(|\epsilon_i|^{(1+\delta)})$ is bounded for some $\delta > 0$. These assumptions exclude heavy-tailed error distributions such as Cauchy distribution, which is not sub-Gaussian and does not have a finite mean.

Before presenting the main theorem, we first state a useful lemma.

Lemma 2. *(i) Let* $\widehat{\boldsymbol{\gamma}}(\lambda) = \widehat{\boldsymbol{\beta}}(\lambda) - \boldsymbol{\beta}_0$. *For any* $\lambda \geq c||\mathbf{S}_n||_\infty$, *we have* $\widehat{\boldsymbol{\gamma}}(\lambda) \in \Gamma$.

(ii) Assume condition (C1) is satisfied. There exists a universal constant $c_0 = 4\sqrt{2}b_1 c$*, where* c *is the constant in (7), such that for any positive constant* $l > 1$,

$$P\big(c||\mathbf{S}_n||_\infty < lc_0\sqrt{\log p/n}\big) \geq 1 - 2\exp\big(-(l^2-1)\log p\big). \tag{8}$$

(iii) If $p > (2/\alpha_0)^{1/3}$ *where* α_0 *is the positive constant in (7), then* $\lambda^* < 2c_0\sqrt{\log p/n}$.

Remark 4. Part (i) of Lemma 2 states that $\widehat{\boldsymbol{\gamma}}(\lambda)$ belongs to the cone set Γ with an appropriate choice of λ. Part (ii) implies that with high probability $c||\mathbf{S}_n||_\infty$ has an upper bound $lc_0\sqrt{\log p/n}$. Part (iii) implies that under mild conditions on p, the simulated tuning parameter λ^* has an upper bound $2c_0\sqrt{\log p/n}$. It is worth emphasizing that the statement in (iii) is deterministic, which follows by combining (ii) with the observation that λ^* is defined as the $(1-\alpha_0)$-quantile of the distribution of $c||\mathbf{S}_n||_\infty$. The proof of Lemma 2 is given in the Appendix. Although the upper bound in (iii) has a simple form and is of the same order of the theoretical value of λ, it tends to be larger than the simulated tuning parameter value, which we recommend for real data applications as the latter automatically adjusts for the error distribution or the design matrix.

In the following theorem, we provide a nonasymptotic bound for the L_2 estimation error of $\widehat{\boldsymbol{\beta}}(\lambda^*)$.

Theorem 1. *Suppose conditions (C1)–(C3) hold. If* $p > (2/\alpha_0)^{1/3}$*, then the estimated* L_1*-penalized Wilcoxson rank regression estimator* $\widehat{\boldsymbol{\beta}}(\lambda^*)$*, where* λ^* *is defined in (7), satisfies*

$$||\widehat{\boldsymbol{\beta}}(\lambda^*) - \boldsymbol{\beta}_0||_2 \leq \frac{8(1+\bar{c})c_0}{b_2 b_3}\sqrt{\frac{q\log p}{n}} \tag{9}$$

with probability at least $1 - \alpha_0 - \exp(-2\log p)$.

Theorem 1 proves that the L_2 error bound of $\widehat{\boldsymbol{\beta}}(\lambda^*)$ achieves the same near-oracle rate as Lasso has with theoretical choice of λ. The requirement $p > (2/\alpha_0)^{1/3}$ is very weak. For

$\alpha_0 = 0.01$, this amounts to requiring $p \geq 6$. The proof of Theorem 1 is given in the Appendix, and uses advanced empirical process theory techniques to overcome the challenge of the nonsmoothness of Jaeckel's dispersion function. The results are of independent interest as the techniques can be applied to handle other nonsmooth loss functions.

3 Bias reduction and efficiency improvement for high-dimensional heavy-tailed data

3.1 A second-stage enhancement

We next consider a second-stage enhancement by using $\widehat{\boldsymbol{\beta}}(\lambda^*)$ as an initial estimator. The major goal is to further reduce the mean-squared error in the high-dimensional setting at the presence of heavy-tailed errors, comparing with standard least-squares based penalized regression procedures. This is achieved by a combination of an appropriate loss function (rank loss introduced earlier) and the use of a nonconvex penalty function. It is now widely recognized that L_1 penalty tends to over-penalize large coefficients, since the magnitude of L_1 penalty increases linearly with the magnitude of the coefficient. We prove that with high probability the second-stage estimator possesses the strong oracle property, that is, it is equal to the estimator one would obtain if the underlying model is known in advance. This implies that: (1) one can recover the support of the generative model with high probability; (2) one can estimate the nonzero coefficients more efficiently.

Let the initial estimator $\widetilde{\boldsymbol{\beta}}^{(0)} = (\widetilde{\beta}_1^{(0)}, \ldots, \widetilde{\beta}_p^{(0)})^T$ be $\widehat{\boldsymbol{\beta}}(\lambda^*)$, the L_1 regularized estimator defined in (4) with simulated tuning parameter λ^*. The second-stage estimator is defined as

$$
\begin{aligned}
\widetilde{\boldsymbol{\beta}}^{(1)} \quad &= \quad \arg\min_{\boldsymbol{\beta}} \Big\{ \big[n(n-1)\big]^{-1} \sum\sum_{i \neq j} \big|(Y_i - \mathbf{x}_i^T\boldsymbol{\beta}) - (Y_j - \mathbf{x}_j^T\boldsymbol{\beta})\big| \\
&\qquad + \sum_{k=1}^{p} p'_\eta(|\widetilde{\beta}_k^{(0)}|)|\beta_j| \Big\}, \qquad (10)
\end{aligned}
$$

where $p'_\eta(\cdot)$ denotes the derivative of some nonconvex penalty function $p_\eta(\cdot)$, where $\eta > 0$ is a tuning parameter. The second stage estimator is motivated by the local linear approximation algorithm of Zou and Li (2008) in the lower-dimensional case for penalized likelihood setting. The penalty function is only assumed to satisfy some general conditions. More specifically, $p_\eta(t)$ is increasing and concave for $t \in [0, +\infty)$ with a continuous derivative $p'_\eta(t)$ on $(0, +\infty)$. It has a singularity at the origin, i.e., $p'_\eta(0+) > 0$. Without loss of generality, the penalty function can be standardized such that $p'_\eta(0+) = \eta$. Furthermore, there exist constants $a_1 > 0$ and $a_2 > 1$ such that $p'_\eta(t) \geq a_1\eta$, $\forall\ 0 < t < a_2\eta$; and $p'_\eta(t) = 0$, $\forall\ t > a_2\eta$. Two popular choices of nonconvex penalty functions satisfying these conditions are the SCAD penalty function (Fan and Li, 2001) and the MCP penalty function (Zhang, 2010a). The SCAD penalty function is given by

$$\begin{aligned} p_\eta(|\beta|) &= \eta|\beta| I(0 \leq |\beta| < \eta) + \frac{a\eta|\beta| - (\beta^2 + \eta^2)/2}{a-1} I(\eta \leq |\beta| \leq a\eta) \\ &\quad + \frac{(a+1)\eta^2}{2} I(|\beta| > a\eta), \end{aligned}$$

for some $a > 2$. The MCP function has the form

$$p_\eta(|\beta|) = \eta\Big(|\beta| - \frac{\beta^2}{2a\eta}\Big) I(0 \leq |\beta| < a\eta) + \frac{a\eta^2}{2} I(|\beta| \geq a\eta),$$

for some $a > 1$. In practice, these two popular choices lead to similar performance.

3.2 Statistical properties of second-stage estimation

We first consider the property of the oracle estimator while allowing the underlying model dimension to diverge. Without loss of generality, we assume that the first q components of $\boldsymbol{\beta}_0$ are nonzero and the remaining $p - q$ components are zero. Hence, we can write $\boldsymbol{\beta}_0 = \big(\boldsymbol{\beta}_{01}^T, \mathbf{0}_{p-q}^T\big)^T$, where $\mathbf{0}_{p-q}$ denotes a $(p-q)$-vector of zeros. Let $\mathbf{x}_{1i}$ be the subvector of $\mathbf{x}_i$ that consists its first q components, $i = 1, \ldots, n$. It is assumed that $(\mathbf{x}_{1i}, Y_i)$ are in general positions (Koenker, 2005) and that there is at least one continuous covariate in the true underlying model. Condition (C2) implies that $\xi_{\min}\big(n^{-1}\mathbf{X}_A^T\mathbf{X}_A\big) \geq b_2$, where $\mathbf{X}_A$ denotes the $n \times q$ matrix that consists of the columns of $\mathbf{X}$ whose indices are in A.

Let

$$L_n(\boldsymbol{\beta}_1) = \sum\sum_{i\neq j} \big|(Y_i - \mathbf{x}_{1i}^T\boldsymbol{\beta}_1) - (Y_j - \mathbf{x}_{1j}^T\boldsymbol{\beta}_1)\big|$$

and $\widehat{\boldsymbol{\beta}}_1^{(o)} = \arg\min_{\boldsymbol{\beta}_1} L_n(\boldsymbol{\beta}_1)$. The oracle estimator for $\boldsymbol{\beta}_0$ is $\widehat{\boldsymbol{\beta}}^o = \Big(\widehat{\boldsymbol{\beta}}_1^{(o)T}, \mathbf{0}_{p-q}^T\Big)^T$. In other words, this is the estimator we would obtain if the support of the generative model is known. Lemma 3 below provides the convergence rate of the oracle estimator when the number of nonzero coefficients $q = |A|$ is diverging with the sample size n.

Lemma 3. *Assume conditions (C1) and (C3) hold and that $q = o(n)$. Then,*

$$||\widehat{\boldsymbol{\beta}}_1^{(o)} - \boldsymbol{\beta}_{01}||_2 = O_p(q^{1/2}n^{-1/2}).$$

Note that if q is fixed, then this reduces to the classical rate of convergence. The proof of Lemma 3 is given in the Appendix. The kth subgradient of the unpenalized loss function $\big[n(n-1)\big]^{-1}\sum\sum_{i\neq j}\big|(Y_i - \mathbf{x}_i^T\boldsymbol{\beta}) - (Y_j - \mathbf{x}_j^T\boldsymbol{\beta})\big|$ is given by

$$\begin{aligned}\delta_k(\boldsymbol{\beta}) &= \big[n(n-1)\big]^{-1}\sum\sum_{i\neq j}(x_{jk} - x_{ik})\text{sign}\big(Y_i - Y_j - (\mathbf{x}_i - \mathbf{x}_j)^T\boldsymbol{\beta}\big) \\ &- \big[n(n-1)\big]^{-1}\sum\sum_{i\neq j}(x_{jk} - x_{ik})v_{ij}I\big(Y_i - Y_j = (\mathbf{x}_i - \mathbf{x}_j)^T\boldsymbol{\beta}\big),\end{aligned}$$

where $v_{ij} \in [-1, 1]$, $k = 1, 2, \ldots, p$. Lemma B in the supplementary material characterizes the important properties of the gradient functions when being evaluated at the oracle estimator.

Theorem 2 below states that the second-stage estimator $\widetilde{\boldsymbol{\beta}}^{(1)}$ possesses the strong oracle property with high probability.

Theorem 2. *Assume the conditions of Theorem 1 are satisfied. Suppose $q = O(n^{c_1})$, $\eta = O(n^{-(1-c_2)/2})$, $\min_{1\le j\le q}|\beta_{0j}| \ge bn^{-(1-c_3)/2}$, $p = \exp(n^{c_4})$ for some positive constants b and c_i $(i = 1, \ldots, 4)$ such that $2c_1 < c_2 < c_3 \le 1$ and $c_1 + c_4 < c_2$. We have*

$$P\big(\widetilde{\boldsymbol{\beta}}^{(1)} = \widehat{\boldsymbol{\beta}}^{(o)}\big) \ge 1 - \alpha_0 - h_n,$$

where $h_n \to 0$ as $n \to \infty$.

Remark 5. Let $\widetilde{\boldsymbol{\beta}}_1^{(1)}$ be the subvector containing the first q elements of $\widetilde{\boldsymbol{\beta}}^{(1)}$. Theorem 2 indicates that $\sqrt{n}(\widetilde{\boldsymbol{\beta}}_1^{(1)} - \boldsymbol{\beta}_{01})$ follows an asymptotic normal distribution in the case q is fixed. It follows from the theory of the classical nonparametric estimator based on Jaeckel's Wilcoxon-type dispersion function (Hettmansperger and McKean, 1998) that the relative efficiency (ARE) of $\widetilde{\boldsymbol{\beta}}_1^{(1)}$ with respect to the least-squares oracle for estimating $\boldsymbol{\beta}_{01}$ has the form ARE $= 12\sigma^2\big(\int f^2(u)du\big)^2$, where σ^2 is the random error variance. It is worth noting that this asymptotic relative efficiency is the same as that of the one-sample Wilcoxon rank test with respect to the t-test. The ARE is as high as 0.955 for normal error distribution, and can be significantly higher than one for many heavier-tailed error distributions. For instance, ARE is 1.5 for the double exponential distribution, and is 1.9 for the t distribution with 3 degrees of freedom. For symmetric error distributions with finite Fisher information, the ARE is known to have a lower bound equal to 0.864, see for example Theorem 1.7.6 of Hettmansperger and McKean (1998). Although theoretically it is possible to introduce nonconvex penalized rank-based loss with adaptive optimal weights to obtain an efficient estimator (first-order equivalent to MLE), the weights nonetheless depend on the unknown error density function, see the discussions in Naranjo and McKean (1997) for the classical low-dimensional setting. In the high-dimensional setting, it is usually challenging to estimate the random error density function.

Remark 6. It is interesting to note that the above asymptotic relative efficiency is equivalent to that of composite quantile regression (Zou and Yuan, 2007) when K, the number of quantiles, goes to infinity. The objective function of composite quantile regression involves a mixture of quantile objective functions at different quantiles (the suggested value of K for practical use is 19). As a result, besides the regression parameters one also needs to estimate K additional parameters corresponding to K different quantiles of the error distribution.

3.3 High-dimensional BIC for tuning parameter selection

As the main objective of the second step is to alleviate the bias due to the over-fitting of L_1 penalty, it is intuitive to not tune η by cross-validation which aims for prediction optimality. Motivated by (Chen and Chen, 2008), we propose a modified high-dimensional BIC criterion for selecting η. Let $\widetilde{\boldsymbol{\beta}}^{(1)}_n(\eta) = (\widetilde{\beta}^{(1)}_{n1}(\eta), \ldots, \widetilde{\beta}^{(1)}_{n1}(\eta))^T$ denote the estimator defined in (10) with the tuning parameter value η. Let $A_\eta = \{j : \widetilde{\beta}^{(1)}_{nj}(\eta) \neq 0, j = 1, \cdots, p\}$ be the index set of the selected model and A^{C}_η be its complement; and let

$$\widehat{\boldsymbol{\beta}}_\eta = \underset{\boldsymbol{\beta}\in\mathbb{R}^p,\ \boldsymbol{\beta}_{A^{\mathrm{C}}_\eta}=\mathbf{0}}{\arg\min} \Big\{ \big[n(n-1)\big]^{-1} \sum\sum_{i\neq j} \big|(Y_i - \mathbf{x}_i^T\boldsymbol{\beta}) - (Y_j - \mathbf{x}_j^T\boldsymbol{\beta})\big| \Big\}.$$

Define the high-dimensional BIC (HBIC) for selecting η as

$$\mathrm{HBIC}(\eta) = \log\big\{ \sum\sum_{i\neq j} \big|(Y_i - \mathbf{x}_i^T\widehat{\boldsymbol{\beta}}_\eta) - (Y_j - \mathbf{x}_j^T\widehat{\boldsymbol{\beta}}_\eta)\big| \big\} + |A_\eta| \frac{\log\log n}{n} \log p, \tag{11}$$

where $|A_\eta|$ denotes the cardinality of set A_η. We select the value of η that minimizes $\mathrm{HBIC}(\eta)$.

As we observe in Section 4.2, HBIC can be computationally fast. High-dimensional BIC-type criterion for nonconvex penalized regression has been recently investigated by Chen and Chen (2008), Wang and Li (2009), Wang et al. (2013), Lee et al. (2014), Peng and Wang (2015), among others. For nonconvex penalized least-squares regression with fixed p, Wang et al. (2007) proved that cross-validation leads to a tuning parameter that would yield an over-fitted model with a positive probability. Wang et al. (2013) established the consistency of high-dimensional BIC for penalized least squares regression. Lee et al. (2014) established the consistency of high-dimensional BIC for penalized quantile regression. However, the results in these earlier work do not apply to our setting. To rigorously prove the consistency of the HBIC in (11), the main challenge is to establish a uniform approximation of a U-process that involves nonsmooth functions over a class of models.

Let $\Lambda_n = \{\eta : |A_\eta| \leq k_n\}$, where $k_n > q$ represents a rough estimate of an upper bound of the sparsity size of the underlying model and is allowed to diverge to ∞. We select the tuning parameter $\widehat{\eta} = \arg\min_{\eta\in\Lambda_n} \text{HBIC}(\eta)$. Theorem 3 shows that under some general regularity conditions, HBIC achieves model selection consistency. The proof of Theorem 3 is given in the supplementary material.

Theorem 3 (Consistency of HBIC)**.** *Assume the conditions of Theorem 2 are satisfied, and that* $k_n \log(p \vee n) = o(\sqrt{n})$*. Assume* $\beta^*_{\min} \gg \max\left\{\sqrt{\frac{\log(\log n)}{n}}\log p, \sqrt{\frac{q\log q}{n}}\right\}$*, where* $\beta^*_{\min} = \min\{|\beta_{0j}| : j \in A\}$*. Then*

$$P(A_{\widehat{\eta}} = A) \to 1, \quad \textit{as } n \to \infty.$$

where $A = \{j : \beta_{0j} \neq 0, j = 1, \cdots, p\}$*.*

4 Numerical studies

4.1 Monte Carlo examples

This subsection summarizes the simulation results from three experiments. Additional simulation results are reported in the supplementary material. We compare the performance of the cross-validated Lasso (standard Lasso with cross-validated choice of tuning parameter), the square root lasso (denoted by $\sqrt{\text{Lasso}}$), SCAD (Fan and Li, 2001), Rank Lasso (the L_1 regularized estimator in (4)), and Rank SCAD (the two-stage estimator in (10) with the SCAD penalty). The cross-validated Lasso is computed using the R package `glmnet` (Friedman et al., 2010). The square-root Lasso is computed using the R package `flare` (Li et al., 2018). For Lasso or square root Lasso, the tuning parameter is chosen based on a five-fold cross-validation as usually recommended in the literature. We compute the SCAD estimator using the function "glmnet" in R package `glmnet`. The initial estimator is selected as the Lasso estimator, by "cv.glmnet" function. For Rank Lasso, the tuning parameter λ^* is obtained by simulation based on 500 repetitions using (7) with $\alpha_0 = 0.10$ and $c = 1.01$.

Example 1 (comparison under different random error distributions). We simulate random data from the regression model $Y_i = \mathbf{X}_i^T \boldsymbol{\beta}_0 + \epsilon_i$, $i = 1, \ldots, n$, where $\mathbf{X}_i$ is generated from a p-dimensional multivariate normal distribution $N_p(0, \Sigma)$ and is independent of ϵ_i. In this example, we take $n = 100$, $p = 400$, and $\boldsymbol{\beta}_0 = (\sqrt{3}, \sqrt{3}, \sqrt{3}, 0, ..., 0)^T$. The correlation matrix Σ has a compound symmetry structure: $\Sigma_{(i,j)} = 0.5$ for $i \neq j$; and $\Sigma_{(i,j)} = 1$ for $i = j$. We consider six different distributions for ϵ_i: (1) normal distribution with mean 0 and variance 0.25 (denoted by $N(0, 0.25)$); (2) normal distribution with mean 0 and variance 1 (denoted by $N(0, 1)$); (3) normal distribution with mean 0 and variance 2 (denoted by $N(0, 2)$); (4) mixture normal distribution $\epsilon \sim 0.95N(0, 1) + 0.05N(0, 100)$ (denoted by MN); (5) $\epsilon \sim \sqrt{2}t(4)$, where $t(4)$ denotes the t distribution with 4 degree of freedom; and (6) $\epsilon \sim$ Cauchy$(0, 1)$, where Cauchy$(0, 1)$ denotes the standard Cauchy distribution.

Table 1 summarizes the average (and standard error) of the L_1 estimation error, the L_2 estimation error, the model error (denoted by ME), number of false positive variables (FP) and number of false negative variables (FN) for the six methods based on 200 simulation runs. More specifically, for an estimator $\widehat{\boldsymbol{\beta}}$ from a given method, its L_1 error is $||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0||_1$; its L_2 error is $||\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0||_2$; its model error is $(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)^T \Sigma_{\mathbf{X}} (\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0)$ where $\Sigma_{\mathbf{X}}$ is the population covariance matrix of $\mathbf{X}$. FP is the number of noise covariates that are selected in the model; and FN is the number of active variables that are not selected in the model.

Table 1: Simulation results for Example 1

Error	Method	L1 error	L2 error	ME	FP	FN
$N(0, 0.25)$	Lasso	0.83 (0.03)	0.28 (0.00)	0.05 (0.00)	13.48 (0.46)	0 (0)
	$\sqrt{\text{Lasso}}$	0.70 (0.01)	0.26 (0.00)	0.05 (0.00)	11.58 (0.29)	0 (0)
	SCAD	0.24 (0.01)	0.14 (0.00)	0.02 (0.00)	0 (0)	0 (0)
	Rank Lasso	0.59 (0.01)	0.28 (0.00)	0.10 (0.00)	6.23 (0.23)	0 (0)
	Rank SCAD	0.16 (0.00)	0.11 (0.00)	0.01 (0.00)	0 (0)	0 (0)
$N(0, 1)$	Lasso	1.54 (0.04)	0.57 (0.01)	0.20 (0.01)	13.08 (0.43)	0 (0)
	$\sqrt{\text{Lasso}}$	1.41 (0.03)	0.54 (0.01)	0.21 (0.01)	11.46 (0.33)	0 (0)
	SCAD	0.46 (0.02)	0.28 (0.01)	0.06 (0.00)	0 (0)	0 (0)
	Rank Lasso	1.24 (0.02)	0.59 (0.01)	0.37 (0.00)	6.32 (0.23)	0 (0)
	Rank SCAD	0.41 (0.02)	0.28 (0.01)	0.04 (0.00)	0 (0)	0 (0)
$N(0, 2)$	Lasso	2.16 (0.05)	0.80 (0.01)	0.41 (0.01)	12.32 (0.31)	0 (0)
	$\sqrt{\text{Lasso}}$	2.07 (0.04)	0.78 (0.01)	0.42 (0.01)	11.52 (0.27)	0 (0)
	SCAD	0.66 (0.03)	0.38 (0.01)	0.11 (0.01)	0 (0)	0 (0)
	Rank Lasso	1.79 (0.04)	0.82 (0.01)	0.76 (0.02)	6.65 (0.23)	0 (0)
	Rank SCAD	0.81 (0.03)	0.52 (0.02)	0.10 (0.00)	0 (0)	0 (0)
MN	Lasso	3.02 (0.12)	1.12 (0.04)	0.81 (0.04)	12.00 (0.35)	0 (0)
	$\sqrt{\text{Lasso}}$	3.10 (0.11)	1.09 (0.04)	0.93 (0.05)	14.23 (0.27)	0 (0)
	SCAD	0.91 (0.05)	0.54 (0.03)	0.21 (0.02)	0.38 (0.04)	0 (0)
	Rank Lasso	0.14 (0.00)	0.07 (0.00)	0.04 (0.00)	6.85 (0.22)	0 (0)
	Rank SCAD	0.03 (0.00)	0.02 (0.00)	0.03 (0.00)	0 (0)	0 (0)
$\sqrt{2}t_4$	Lasso	3.42 (1.21)	1.18 (0.02)	0.77 (0.02)	15.52 (0.54)	0 (0)
	$\sqrt{\text{Lasso}}$	3.01 (0.06)	1.10 (0.02)	0.79 (0.02)	12.58 (0.30)	0 (0)
	SCAD	0.86 (0.03)	0.52 (0.02)	0.21 (0.01)	0 (0)	0 (0)
	Rank Lasso	2.20 (0.04)	1.02 (0.02)	1.44 (0.03)	7.39 (0.25)	0 (0)
	Rank SCAD	1.21 (0.04)	0.78 (0.03)	0.18 (0.01)	0 (0)	0 (0)
Cauchy	Lasso	7.32 (0.16)	3.16 (0.03)	10.92 (0.52)	5.84 (0.38)	2.17 (0.08)
	$\sqrt{\text{Lasso}}$	9.31 (0.20)	3.45 (0.06)	10.83 (0.52)	6.93 (0.28)	2.16 (0.08)
	SCAD	6.73 (0.11)	3.45 (0.05)	20.39 (0.39)	0.00 (0.00)	3.00 (0.00)
	Rank Lasso	2.60 (0.05)	1.27 (0.02)	3.73 (0.20)	6.00 (0.20)	0 (0)
	Rank SCAD	1.89 (0.07)	1.21 (0.04)	2.32 (0.19)	0 (0)	0 (0)

We have the following important observations. (1) Even for normal errors, Rank Lasso performs slightly better (particularly with respect to the false positives) compared with the cross-validated Lasso and square-root Lasso. This is probably due to the fact its tuning parameter is optimally chosen. It is worth pointing out that Rank Lasso uses the same tuning parameter (around 0.39) for all six error distributions. This choice is observed to have uniform good performance across different error distributions. SCAD and Rank SCAD perform similarly, and both outperform other methods for normal errors. (2) For heavy-tailed errors, the robustness and efficiency gain of Rank Lasso is substantial. Rank SCAD is observed to further improve the performance of Rank Lasso with respect to estimation error and variable selection performance. For example, for normal mixture error distribution, the average L_1 estimator error for cross-validated Lasso or square-root Lasso is above 3, while that of Rank Lasso is 0.14 and that of Rank SCAD is 0.03. The new procedures also yield smaller false positives and false negatives rates for heavy-tailed errors.

Example 2 (more challenging setting with a denser model and weaker signals). Here we consider the same data generative model as in Example 1 except that $\boldsymbol{\beta}_0 = (2, 2, 2, 2, 1.75, 1.75, 1.75, 1.5, 1.5, 1.5, 1.25, 1.25, 1.25, 1, 1, 1, 0.75, 0.75, 0.75, 0.5, 0.5, 0.5, 0.25, 0.25, 0.25, \mathbf{0}_{p-25})^T$, where $\mathbf{0}_{p-25}$ is a $(p-25)$-dimensional vector of zeros. Comparing with Example 1, this is a considerably more challenging scenario with 25 active variables and a number of weak signals. Table 2 summarizes the simulation results. We observe similar results as in Example 1. Rank Lasso improve both the estimation and prediction accuracy in all cases. Rank SCAD further improves the model selection performance.

Example 3 (comparisons under different design matrices). We consider the same data generative model as in Example 1 but with the following three different choices of Σ: (1) the compound symmetry correlated correlation matrix with correlation coefficient 0.8 (Σ_1); (2) the compound symmetry correlation matrix with correlation coefficient 0.2 (Σ_2); and (3) the AR(1) correlation matrix with auto-correlation coefficient 0.5 (Σ_3). For each choice of Σ, we consider three different error distributions: $N(0, 1)$, the mixture normal distribution in Example 1, and Cauchy distribution. The simulation results are summarized in Table 3. Note that Table 1 contains the simulation results on compound symmetry correlation matrix with correlation coefficient 0.5. The tuning parameter selection of L_1 regularized new estimator automatically adapts to the design matrix. In all cases considered in this example, the new procedures display superior performance with notable improvement even for normal error distribution and substantial improvements for the heavy-tailed error distributions.

Table 2: Simulation results for Example 2

Error	Method	L1 error	L2 error	ME	FP	FN
$N(0,0.25)$	Lasso	12.91 (0.08)	2.06 (0.01)	2.69 (0.03)	46.78 (0.27)	0.73 (0.05)
	$\sqrt{\text{Lasso}}$	9.27 (0.10)	1.40 (0.01)	1.01 (0.02)	53.56 (0.29)	0.52 (0.04)
	SCAD	6.66 (0.12)	1.47 (0.02)	1.12 (0.03)	7.37 (0.27)	3.87 (0.12)
	Rank Lasso	5.81 (0.05)	1.08 (0.01)	0.71 (0.01)	33.19 (0.40)	0 (0)
	Rank SCAD	4.40 (0.05)	1.05 (0.01)	0.57 (0.01)	1.70 (0.11)	2.34 (0.05)
$N(0,1)$	Lasso	16.48 (0.15)	2.56 (0.02)	3.65 (0.06)	50.06 (0.31)	2.48 (0.07)
	$\sqrt{\text{Lasso}}$	16.52 (0.16)	2.50 (0.02)	3.23 (0.06)	53.45 (0.36)	2.65 (0.08)
	SCAD	9.94 (0.16)	2.11 (0.03)	2.27 (0.05)	5.87 (0.27)	6.23 (0.10)
	Rank Lasso	9.15 (0.09)	1.68 (0.02)	1.78 (0.03)	32.95 (0.34)	0.27 (0.03)
	Rank SCAD	6.95 (0.09)	1.63 (0.02)	1.40 (0.03)	3.99 (0.17)	2.64 (0.06)
$N(0,2)$	Lasso	20.31 (0.15)	3.14 (0.02)	5.11 (0.07)	50.78 (0.28)	4.01 (0.10)
	$\sqrt{\text{Lasso}}$	20.64 (0.16)	3.16 (0.02)	5.12 (0.07)	50.9 (0.33)	4.25 (0.1)
	SCAD	15.31 (0.25)	3.12 (0.04)	5.08 (0.13)	8.30 (0.30)	8.32 (0.12)
	Rank Lasso	12.61 (0.12)	2.3 (0.02)	3.37 (0.06)	33.92 (0.35)	0.59 (0.05)
	Rank SCAD	10.11 (0.13)	2.33 (0.03)	2.89 (0.07)	5.74 (0.21)	3.25 (0.08)
MN	Lasso	25.12 (0.47)	3.87 (0.07)	8.11 (0.26)	49.51 (0.37)	6.18 (0.19)
	$\sqrt{\text{Lasso}}$	25.02 (0.44)	3.88 (0.07)	7.93 (0.27)	48.19 (0.49)	6.12 (0.20)
	SCAD	20.40 (0.57)	4.03 (0.10)	8.90 (0.39)	9.05 (0.30)	10.57 (0.23)
	Rank Lasso	5.36 (0.10)	0.97 (0.02)	0.61 (0.02)	36.7 (0.48)	0 (0)
	Rank SCAD	4.62 (0.09)	1.10 (0.02)	0.64 (0.02)	1.15 (0.10)	2.53 (0.05)
$\sqrt{2}t_4$	Lasso	24.76 (0.20)	3.78 (0.03)	7.38 (0.11)	50.82 (0.32)	6.13 (0.11)
	$\sqrt{\text{Lasso}}$	24.56 (0.20)	3.77 (0.03)	7.42 (0.11)	48.68 (0.34)	6.14 (0.11)
	SCAD	20.46 (0.31)	4.07 (0.05)	8.66 (0.21)	8.81 (0.24)	10.76 (0.13)
	Rank Lasso	16.33 (0.17)	2.96 (0.03)	5.58 (0.12)	35.06 (0.36)	1.17 (0.07)
	Rank SCAD	14.38 (0.23)	3.21 (0.05)	5.61 (0.16)	8.17 (0.23)	5.30 (0.15)
Cauchy	Lasso	48.07 (0.53)	8.46 (0.14)	42.38 (1.65)	25.73 (0.98)	19.35 (0.31)
	$\sqrt{\text{Lasso}}$	45.66 (0.46)	8.15 (0.13)	44.69 (1.90)	23.12 (0.87)	19.57 (0.30)
	SCAD	36.53 (0.30)	7.40 (0.08)	249.49 (13.23)	11.19 (0.75)	21.20 (0.31)
	Rank Lasso	30.59 (0.51)	5.33 (0.08)	19.97 (0.65)	35.48 (0.35)	6.81 (0.26)
	Rank SCAD	32.92 (0.57)	6.78 (0.12)	25.7 (0.82)	8.68 (0.27)	13.81 (0.25)

4.2 Computational aspects

The new estimator in (4) is the minimizer of a convex objective function and can be conveniently solved via linear programming. With the aid of slack variables ξ_{ij}^+, ξ_{ij}^-, and ζ_k, the convex optimization problem in (4) can be equivalently rewritten as

$$\min_{\boldsymbol{\beta},\boldsymbol{\xi},\boldsymbol{\zeta}} \Big\{ \big[n(n-1)\big]^{-1} \sum\sum_{i \neq j} (\xi_{ij}^+ + \xi_{ij}^-) + \lambda \sum_{k=1}^{p} \zeta_k \Big\}$$

$$\text{subject to} \quad \xi_{ij}^+ - \xi_{ij}^- = (Y_i - Y_j) - (\mathbf{x}_i - \mathbf{x}_j)^T \boldsymbol{\beta}, \ i,j = 1,2,\cdots,n;$$

$$\xi_{ij}^+ \geq 0, \xi_{ij}^- \geq 0, \ i,j = 1,2,\cdots,n;$$

$$\zeta_k \geq \beta_k, \zeta_k \geq -\beta_k, \ k = 1,2,\cdots,p.$$

Table 3: Simulation results for Example 3

Σ	Error	Method	L1 error	L2 error	ME	FP	FN
Σ_1	$N(0,1)$	Lasso	2.42 (0.06)	0.88 (0.02)	0.17 (0.00)	14.42 (0.46)	0 (0)
		$\sqrt{\text{Lasso}}$	2.22 (0.04)	0.83 (0.01)	0.15 (0.00)	13.09 (0.30)	0 (0)
		SCAD	0.68 (0.03)	0.40 (0.02)	0.06 (0.00)	0 (0)	0 (0)
		Rank Lasso	2.23 (0.04)	0.91 (0.01)	0.25 (0.01)	11.48 (0.28)	0 (0)
		Rank SCAD	0.55 (0.02)	0.34 (0.01)	0.04 (0.00)	0 (0)	0 (0)
	MN	Lasso	4.87 (0.18)	1.75 (0.06)	0.76 (0.04)	12.30 (0.25)	0 (0)
		$\sqrt{\text{Lasso}}$	4.79 (0.17)	1.72 (0.06)	0.75 (0.04)	12.23 (0.26)	0 (0)
		SCAD	2.87 (0.17)	1.63 (0.09)	0.78 (0.06)	0.61 (0.05)	0.56 (0.05)
		Rank Lasso	0.27 (0.01)	0.11 (0.00)	0.00 (0.00)	12.29 (0.27)	0 (0)
		Rank SCAD	0.05 (0.00)	0.04 (0.00)	0.00 (0.00)	0 (0)	0 (0)
	Cauchy	Lasso	8.70 (0.17)	3.65 (0.05)	5.33 (0.25)	5.54 (0.30)	2.73 (0.04)
		$\sqrt{\text{Lasso}}$	11.14 (0.17)	4.31 (0.09)	4.56 (0.18)	7.85 (0.23)	2.73 (0.54)
		SCAD	6.48 (0.08)	3.25 (0.03)	25.31 (0.23)	0.00 (0.00)	3.00 (0.00)
		Rank Lasso	5.11 (0.11)	2.03 (0.04)	1.27 (0.04)	11.52 (0.26)	0 (0)
		Rank SCAD	4.07 (0.17)	2.16 (0.08)	1.19 (0.01)	1.12 (0.08)	0.91 (0.05)
Σ_2	$N(0,1)$	Lasso	1.21 (0.04)	0.45 (0.01)	0.18 (0.01)	13.08 (0.62)	0 (0)
		$\sqrt{\text{Lasso}}$	0.99 (0.02)	0.44 (0.01)	0.17 (0.00)	6.50 (0.23)	0 (0)
		SCAD	0.42 (0.01)	0.26 (0.01)	0.07 (0.00)	0 (0)	0 (0)
		Rank Lasso	0.94 (0.01)	0.55 (0.01)	0.39 (0.01)	1.34 (0.09)	0 (0)
		Rank SCAD	0.32 (0.01)	0.20 (0.01)	0.04 (0.00)	0.56 (0.07)	0 (0)
	MN	Lasso	2.49 (0.10)	0.92 (0.03)	0.82 (0.05)	11.37 (0.35)	0 (0)
		$\sqrt{\text{Lasso}}$	2.18 (0.08)	0.93 (0.03)	0.81 (0.05)	7.65 (0.23)	0 (0)
		SCAD	0.87 (0.04)	0.48 (0.02)	0.25 (0.02)	0.41 (0.04)	0 (0)
		Rank Lasso	0.11 (0.00)	0.07 (0.00)	0.01 (0.00)	1.62 (0.11)	0 (0)
		Rank SCAD	0.03 (0.00)	0.02 (0.00)	0.00 (0.00)	0.40 (0.04)	0 (0)
	Cauchy	Lasso	5.90 (0.10)	2.97 (0.01)	10.00 (0.24)	3.78 (0.32)	2.15 (0.08)
		$\sqrt{\text{Lasso}}$	12.00 (0.37)	3.75 (0.11)	11.19 (0.52)	18.63 (0.33)	1.86 (0.08)
		SCAD	6.33 (0.10)	3.21 (0.03)	13.87 (0.24)	0.00 (0.00)	3.00 (0.00)
		Rank Lasso	2.36 (0.05)	1.35 (0.03)	2.31 (0.08)	1.45 (0.09)	0 (0)
		Rank SCAD	1.15 (0.05)	0.76 (0.03)	0.56 (0.04)	0.72 (0.06)	0 (0)
Σ_3	$N(0,1)$	Lasso	0.80 (0.03)	0.36 (0.01)	0.14 (0.00)	9.38 (0.60)	0 (0)
		$\sqrt{\text{Lasso}}$	0.71 (0.01)	0.35 (0.01)	0.12 (0.00)	4.47 (0.15)	0 (0)
		SCAD	0.48 (0.02)	0.29 (0.01)	0.08 (0.00)	0.39 (0.04)	0 (0)
		Rank Lasso	0.64 (0.01)	0.43 (0.01)	0.25 (0.01)	0 (0)	0 (0)
		Rank SCAD	0.41 (0.02)	0.25 (0.01)	0.04 (0.00)	1.11 (0.13)	0 (0)
	MN	Lasso	1.53 (0.06)	0.67 (0.02)	0.59 (0.04)	7.12 (0.33)	0 (0)
		$\sqrt{\text{Lasso}}$	1.55 (0.06)	0.68 (0.02)	0.57 (0.04)	6.32 (0.19)	0 (0)
		SCAD	1.07 (0.05)	0.60 (0.03)	0.34 (0.03)	0.45 (0.05)	0 (0)
		Rank Lasso	0.08 (0.00)	0.05 (0.00)	0.00 (0.00)	0 (0)	0 (0)
		Rank SCAD	0.04 (0.00)	0.02 (0.00)	0.00 (0.00)	0.62 (0.05)	0 (0)
	Cauchy	Lasso	4.96 (0.04)	2.69 (0.04)	11.44 (0.40)	2.57 (0.24)	1.75 (0.09)
		$\sqrt{\text{Lasso}}$	8.99 (0.37)	3.33 (0.12)	11.91 (0.67)	9.78 (0.23)	1.49 (1.17)
		SCAD	6.63 (0.13)	3.41 (0.06)	18.61 (0.45)	0.00 (0.00)	3.00 (0.00)
		Rank Lasso	1.51 (0.03)	0.99 (0.02)	1.37 (0.05)	0 (0)	0 (0)
		Rank SCAD	1.27 (0.06)	0.82 (0.04)	0.38 (0.03)	0.63 (0.05)	0 (0)

This is a linear programming problem and can be solved using existing optimization software packages. The second-stage estimator in (10) can be computed similarly by incorporating weights. The major computational barrier is due to the U-statistics structure of the loss function in (4), where the sum consists of $O(n^2)$ terms. An effective approach to alleviate this challenge is to adopt a resampling technique called incomplete U-statistic (Clémençon et al. (2016)), which reduces the computational complexity of the loss function to $O(n)$. Our numerical results below demonstrate that this approximation scheme substantially improves the computation time without deteriorating the quality of the final estimators.

Table 4: Comparison of computation time

Error	Method	L1 error	L2 error	ME	FP	FN	time/s
$N(0,1)$	Lasso	0.83 (0.03)	0.28 (0.00)	0.05 (0.00)	13.48 (0.46)	0 (0)	0.44
	$\sqrt{\text{Lasso}}$	0.70 (0.01)	0.26 (0.00)	0.05 (0.00)	11.58 (0.29)	0 (0)	8.08
	SCAD	0.24 (0.01)	0.14 (0.00)	0.02 (0.00)	0 (0)	0 (0)	0.93
	Rank Lasso	1.07 (0.02)	0.55 (0.01)	0.35 (0.01)	4.31 (0.19)	0 (0)	0.54
	Rank SCAD	0.47 (0.01)	0.26 (0.01)	0.05 (0.00)	0 (0)	0 (0)	3.72
Cauchy	Lasso	7.32 (0.16)	3.16 (0.03)	10.92 (0.52)	5.84 (0.38)	2.17 (0.08)	0.87
	$\sqrt{\text{Lasso}}$	9.31 (0.20)	3.45 (0.06)	10.83 (0.52)	6.93 (0.28)	2.16 (0.08)	10.20
	SCAD	6.73 (0.11)	3.45 (0.05)	20.39 (0.39)	0.00 (0.00)	3.00 (0.00)	0.79
	Rank Lasso	3.84 (0.11)	1.82 (0.05)	3.45 (0.19)	7.05 (0.25)	0 (0)	0.49
	Rank SCAD	2.86 (0.12)	1.64 (0.07)	2.41 (0.17)	0.38 (0.05)	0 (0)	3.72

We consider the simulation setup in Example 1 with $N(0,1)$ error and Cauchy random error. The results are summarized in Table 4. The last column of the table reports the computational time per simulation run (measured by seconds) for each procedure with tuning parameter computation time included. We observe that Rank Lasso has computational time comparable to cross-validated Lasso (implemented by the R package `glmnet` with default options) for normal error distribution, and can be slightly faster than cross-validated Lasso for heavy-tailed Cauchy error distribution. Furthermore, Rank SCAD can be implemented quite efficiently.

4.3 A real data example

We use a genetic data set to illustrate the performance of the new Wilcoxon rank based procedures. Scheetz et al. (2006) investigated gene regulation in the mammalian eye to identify genetic variation relevant to human eye disease based on expression quantitative trait locus (eQTL) mapping data. The data set we analyze contains expression values on 300 probes (after preprocessing) from 120 twelve-week-old male offspring of rats. The response variable is the expression of gene TRIM32, a gene identified to be associated with human hereditary diseases of the retina, corresponding to probe 1389163_at. The sample standard deviation of TRIM32 is 0.14.

We conducted 100 random partitions of the data set. For each partition, we randomly select 60 rats as the training data and the other 60 as the testing data. Regularized regression is fitted using the training data with the performance being evaluated on the testing data set. Table 3 summarizes the average (with the standard error in the parenthesis) of the L_1 prediction error, L_2 prediction error and model size, respectively, across 100 partitions. We observe that the new rank based procedures tends to select a sparser model with similar predictive performance comparing with Lasso and square-root Lasso.

Table 5: Analysis of eQTL data: results based on 100 random partitions

Method	L_1 error	L_2 error	model size
Lasso	0.075 (0.001)	0.011 (0.000)	19.50 (1.09)
$\sqrt{\text{Lasso}}$	0.074 (0.001)	0.011 (0.000)	19.09 (0.88)
SCAD	0.083 (0.001)	0.015 (0.000)	5.04 (0.27)
Rank Lasso	0.080 (0.001)	0.014 (0.001)	6.72 (0.27)
Rank SCAD	0.077 (0.001)	0.012 (0.001)	8.17 (0.39)

5 Conclusions and discussions

The paper proposes a new approach for high-dimensional regression. Comparing to Lasso, the proposed L_1 regularized new estimator achieves several goals simultaneously: it keeps the convex structure for convenient computation, has a tuning parameter that can be easily simulated and automatically adapts to both the error distribution and the design matrix, and is equivariant to scale transformation of the response variable. Moreover, the L_2 estimation error bound of the new estimator achieves the same near-oracle rate as Lasso does. It has similar performance as Lasso does with normal random error distribution and can be substantially more efficient with heavy-tailed error distribution. Its efficiency can be further improved via a second-stage enhancement with some light tuning.

Rather than choosing λ to control the L_∞ bound of the noises (Bickel et al., 2009), Sun and Zhang (2013) recently investigated an interesting alternative that chooses λ to control a sparse L_2 measure of the noises. For scaled Lasso, they showed that a penalty level with order smaller than $\sqrt{\log p/n}$ can still be valid and that it can lead to faster convergence rate in some settings. It will be interesting to investigate this alternative in our proposed setting in the future.

References

Avella-Medina, M. and Ronchetti, E. (2018). Robust and consistent variable selection in high-dimensional generalized linear models. *Biometrika*, 105:31–44.

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

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

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

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

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Bradic, J., Fan, J., and Wang, W. (2011). Penalized composite quasi-likelihood for ultra-high dimensional variable selection. *Journal of the Royal Statistical Society: Series B*, 73(3):325–349.

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

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

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

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

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

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

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

Chichignoud, M., Lederer, J., and Wainwright, M. J. (2016). A practical scheme and fast algorithm to tune the lasso with optimality guarantees. *Journal of Machine Learning Research*, 17(231):1–20.

Clémençon, S., Colin, I., and Bellet, A. (2016). Scaling-up empirical risk minimization: optimization of incomplete u-statistics. *The Journal of Machine Learning Research*, 17(1):2682–2717.

Clémençon, S., Lugosi, G., and Vayatis, N. (2008). Ranking and empirical minimization of u-statistics. *The Annals of Statistics*, 36(2):844–874.

Dicker, L. H. (2014). Variance estimation in high-dimensional linear models. *Biometrika*, 101(2):269–284.

Fan, J., Fan, Y., and Barut, E. (2014). Adaptive robust variable selection. *Annals of Statistics*, 42(1):324.

Fan, J., Guo, S., and Hao, N. (2012). Variance estimation using refitted cross-validation in ultrahigh dimensional regression. *Journal of the Royal Statistical Society: Series B*, 74(1):37–65.

Fan, J., Li, Q., and Wang, Y. (2017). Estimation of high dimensional mean regression in the absence of symmetry and light tail assumptions. *Journal of the Royal Statistical Society: Series B*, 79(1):247–265.

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

Fan, J. and Lv, J. (2010). A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101–148.

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

Hebiri, M. and Lederer, J. (2013). How correlations influence lasso prediction. *IEEE Transactions on Information Theory*, 59(3):1846–1854.

Hettmansperger, T. P. and McKean, J. W. (1998). *Robust Nonparametric Statistical Methods.* London: Arnold.

Hjort, N. L. and Pollard, D. (2011). Asymptotics for minimisers of convex processes. *arXiv preprint arXiv:1107.3806.*

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

Homrighausen, D. and McDonald, D. J. (2017). Risk consistency of cross-validation with lasso-type procedures. *Statistica Sinica*, 27(3):1017–1036.

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

Koenker, R. (2005). *Quantile Regression.* Cambridge University Press, New York.

Lederer, J. and Müller, C. (2015). Don't fall for tuning parameters: tuning-free variable selection in high dimensions with the trex. *AAAI Conference on Artificial Intelligence, 2729–2735.*

Ledoux, M. and Talagrand, M. (2013). *Probability in Banach Spaces: Isoperimetry and Processes.* Springer Science & Business Media.

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

Li, X., Zhao, T., Wang, L., Yuan, X., and Liu, H. (2018). *Flare: family of Lasso regression.* R package version 1.6.0.

Loh, P.-L. (2017). Statistical consistency and asymptotic normality for high-dimensional robust m-estimators. *The Annals of Statistics*, 45(2):866–896.

Lozano, A. C., Meinshausen, N., Yang, E., et al. (2016). Minimum distance lasso for robust high-dimensional regression. *Electronic Journal of Statistics*, 10(1):1296–1340.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462.

Naranjo, J. D. and McKean, J. W. (1997). Rank regression with estimated scores. *Statistics & probability letters*, 33(2):209–216.

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

Peng, B. and Wang, L. (2015). An iterative coordinate descent algorithm for high-dimensional nonconvex penalized quantile regression. *Journal of Computational and Graphical Statistics*, 24(3):676–694.

Sabourin, J. A., Valdar, W., and Nobel, A. B. (2015). A permutation approach for selecting the penalty parameter in penalized model selection. *Biometrics*, 71(4):1185–1194.

Sun, Q., Zhou, W.-X., and Fan, J. (2020). Adaptive huber regression. *Journal of the American Statistical Association*, 115:254–265.

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

Sun, T. and Zhang, C.-H. (2013). Sparse matrix inversion with scaled lasso. *The Journal of Machine Learning Research*, 14(1):3385–3418.

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

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

van der Vaart, A. and Wellner, J. (1996). *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer Science & Business Media.

Wainwright, M. J. (2009). Sharp thresholds for high-dimensional and noisy sparsity recovery using l_1-constrained quadratic programming (lasso). *IEEE transactions on information theory*, 55(5):2183–2202.

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

Wang, L. (2013). The l_1 penalized lad estimator for high dimensional linear regression. *Journal of Multivariate Analysis*, 120:135–151.

Wang, L., Kai, B., and Li, R. (2009). Local rank inference for varying coefficient models. *Journal of the American Statistical Association*, 104(488):1631–1645.

Wang, L., Kim, Y., and Li, R. (2013). Calibrating non-convex penalized regression in ultra-high dimension. *Annals of statistics*, 41(5):2505–2536.

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

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

Yu, G. and Bien, J. (2019). Estimating the error variance in a high-dimensional linear model. *Biometrika*, 106(3):533–546.

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

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

Zhang, C.-H. and Zhang, T. (2012). A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, 27(4):576–593.

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

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

Zou, H. and Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *Annals of Statistics*, 36:1509–1566.

Appendix: Proofs of Theorem 1 and Theorem 2

The results in Lemma 1 are straightforward. The proofs of Lemma 2 and Lemma 3 are given in the supplemental material.

Proof of Theorem 1. Write $L_n(\boldsymbol{\gamma}) = Q_n(\boldsymbol{\gamma}) + \lambda^* ||\boldsymbol{\beta}_0 + \boldsymbol{\gamma}||_1$, where $Q_n(\boldsymbol{\gamma})$ is defined in (5). Then

$$\widehat{\boldsymbol{\gamma}}(\lambda^*) = \widehat{\boldsymbol{\beta}}(\lambda^*) - \boldsymbol{\beta}_0 = \arg min_{\boldsymbol{\gamma}} L_n(\boldsymbol{\gamma}).$$

It follows from Lemma 2 that $P\big(\widehat{\boldsymbol{\gamma}}(\lambda^*) \in \Gamma\big) \geq 1 - \alpha_0$ and $\lambda^* < 2c_0\sqrt{\log p/n}$. Let $h_n = \sqrt{n^{-1} q \log p}$. Denote

$$\Gamma^* = \big\{\boldsymbol{\gamma} \in R^p : \boldsymbol{\gamma} \in \Gamma, ||\boldsymbol{\gamma}||_2 = \Delta h_n\big\}, \tag{12}$$

where $\Delta = \frac{8(1+\bar{c})c_0}{b_2 b_3}$ with c_0 being the universal positive constant in Lemma 2. We have

$$\begin{aligned} P\big(||\widehat{\boldsymbol{\gamma}}(\lambda^*)||_2 \le \Delta h_n\big) &\ge 1 - P\big(||\widehat{\boldsymbol{\gamma}}(\lambda^*)||_2 \le \Delta h_n \big| \widehat{\boldsymbol{\gamma}}(\lambda^*) \in \Gamma\big) - P\big(\widehat{\boldsymbol{\gamma}}(\lambda^*) \notin \Gamma\big) \\ &\ge P\left(\inf_{||\boldsymbol{\gamma}||_2 \ge \Delta h_n, \boldsymbol{\gamma} \in \Gamma} L_n(\boldsymbol{\gamma}) > L_n(\mathbf{0})\right) - \alpha_0 \\ &\ge P\big(\inf_{\boldsymbol{\gamma} \in \Gamma^*} L_n(\boldsymbol{\gamma}) > L_n(\mathbf{0})\big) - \alpha_0, \qquad (13) \end{aligned}$$

where the second inequality is due to the convexity of $L_n(\boldsymbol{\gamma})$ (e.g., Hjort and Pollard (2011)). To see this, for an arbitrary point $\mathbf{s}$ outside the L_2 ball centered at $\mathbf{0}$ with radius Δh_n, we can write it as $\mathbf{s} = l\mathbf{u}$, where $\mathbf{u}$ is a unit vector and $l > \Delta h_n$ is a positive constant. The convexity of $L_n(\boldsymbol{\gamma})$ implies that $\big(1 - l^{-1}\Delta h_n\big) L_n(\mathbf{0}) + l^{-1}\Delta h_n L_n(\mathbf{s}) \ge L_n(\Delta h_n \mathbf{u})$. Hence $l^{-1}\Delta h_n\big(L_n(\mathbf{s}) - L_n(\mathbf{0})\big) \ge L_n(\Delta h_n \mathbf{u}) - L_n(\mathbf{0}) \ge \inf_{||\boldsymbol{\gamma}||_2 = \Delta h_n}(L_n(\boldsymbol{\gamma}) - L_n(\mathbf{0}))$. This implies (13).

We first note that $\forall\ \boldsymbol{\gamma} \in \Gamma^*$,

$$\begin{aligned} \lambda^*\big|||\boldsymbol{\beta}_0 + \boldsymbol{\gamma}||_1 - ||\boldsymbol{\beta}_0||_1\big| &\le \lambda^*||\boldsymbol{\gamma}||_1 = \lambda^*\big(||\boldsymbol{\gamma}_A||_1 + ||\boldsymbol{\gamma}_{A^c}||_1\big) \le \lambda^*(1+\bar{c})||\boldsymbol{\gamma}_A||_1 \\ &\le \lambda^*(1+\bar{c})\sqrt{q}||\boldsymbol{\gamma}_A||_2 \le \lambda^*(1+\bar{c})\sqrt{q}\Delta h_n \\ &\le 2c_0(1+\bar{c})\Delta h_n^2, \qquad (14) \end{aligned}$$

where the second inequality follows because $\boldsymbol{\gamma} \in \Gamma^*$, and the third inequality follows by applying the Cauchy-Schwarz inequality. Let $Q(\boldsymbol{\gamma}) = \mathrm{E}\{Q_n(\boldsymbol{\gamma})\}$. We have

$$\begin{aligned} \inf_{\boldsymbol{\gamma} \in \Gamma^*} \big\{L_n(\boldsymbol{\gamma}) - L_n(\mathbf{0})\big\} &\ge \inf_{\boldsymbol{\gamma} \in \Gamma^*} \big\{Q(\boldsymbol{\gamma}) - Q(\mathbf{0})\big\} - \sup_{\boldsymbol{\gamma} \in \Gamma^*} \big|Q_n(\boldsymbol{\gamma}) - Q_n(\mathbf{0}) - Q(\boldsymbol{\gamma}) + Q(\mathbf{0})\big| \\ &\quad -\lambda^*(1+\bar{c})\sqrt{q}\Delta h_n. \qquad (15) \end{aligned}$$

By Knight's identity (Koenker, 2005),

$$\begin{aligned} Q_n(\boldsymbol{\gamma}) - Q_n(\mathbf{0}) &= [n(n-1)]^{-1}\sum\sum_{i\neq j}(\mathbf{x}_i - \mathbf{x}_j)^T\boldsymbol{\gamma}\big[I(\zeta_{ij} < 0) - 1/2\big] \\ &\quad +[n(n-1)]^{-1}\sum\sum_{i\neq j}\int_0^{(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}}\big[I(\zeta_{ij}\leq s) - I(\zeta_{ij}\leq 0)\big]ds. \end{aligned}$$

In the above expression, the first term has mean 0 and the second term is always nonnegative. Hence,

$$\begin{aligned} Q(\boldsymbol{\gamma}) - Q(\mathbf{0}) &= [n(n-1)]^{-1}\sum\sum_{i\neq j}\int_0^{(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}}\big[F^*(s) - F^*(0)\big]ds \\ &= [n(n-1)]^{-1}\sum\sum_{i\neq j}\int_0^{(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}}\big[F^*(s) - F^*(0)\big]ds\mathrm{I}\big\{(\mathbf{x}_i - \mathbf{x}_j)^T\boldsymbol{\gamma} > 0\big\} \\ &\quad +[n(n-1)]^{-1}\sum\sum_{i\neq j}\int_0^{(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}}\big[F^*(s) - F^*(0)\big]ds\mathrm{I}\big\{(\mathbf{x}_i - \mathbf{x}_j)^T\boldsymbol{\gamma} \leq 0\big\} \\ &= I_1 + I_2, \end{aligned}$$

where the definition of I_i, $i = 1, 2$, is clear from the context. By the mean value theorem, for some ξ_{ij} between 0 and $(\mathbf{x}_i - \mathbf{x}_j)^T\boldsymbol{\gamma}$, we have

$$\begin{aligned} I_1 &= [n(n-1)]^{-1}\sum\sum_{i\neq j}\int_0^{(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}} f^*(\xi_{ij})sds\mathrm{I}\big\{(\mathbf{x}_i - \mathbf{x}_j)^T\boldsymbol{\gamma} > 0\big\} \\ &\geq 0.5b_3[n(n-1)]^{-1}\sum\sum_{i\neq j}\big[(\mathbf{x}_i - \mathbf{x}_j)^T\boldsymbol{\gamma}\big]^2\mathrm{I}\big\{(\mathbf{x}_i - \mathbf{x}_j)^T\boldsymbol{\gamma} > 0\big\}, \end{aligned}$$

by condition (C3). To evaluate I_2, we apply the transformation of variable $t = -s$ and obtain for some ξ_{ij} between 0 and $|(\mathbf{x}_i - \mathbf{x}_j)^T\boldsymbol{\gamma}|$,

$$\begin{aligned} I_2 &= [n(n-1)]^{-1}\sum\sum_{i\neq j}\int_0^{-(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}}\big[F^*(0)-F^*(-t)\big]dt\mathrm{I}\big\{(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}\leq 0\big\} \\ &\geq [n(n-1)]^{-1}\sum\sum_{i\neq j}\int_0^{|(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}|} f^*(\xi_{ij})tdt\mathrm{I}\big\{(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}\leq 0\big\} \\ &\geq 0.5b_3[n(n-1)]^{-1}\sum\sum_{i\neq j}\big[(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}\big]^2\mathrm{I}\big\{(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}\leq 0\big\}. \end{aligned}$$

Hence, by condition (C2),

$$\begin{aligned} Q(\boldsymbol{\gamma})-Q(\mathbf{0}) &\geq 0.5b_3[n(n-1)]^{-1}\sum\sum_{i\neq j}\big[(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}\big]^2 \\ &= b_3n^{-1}\sum_{i=1}^n\boldsymbol{\gamma}^T\mathbf{x}_i\mathbf{x}_i^T\boldsymbol{\gamma}\geq b_2b_3||\boldsymbol{\gamma}||_2^2 = b_2b_3\Delta^2h_n^2. \end{aligned} \tag{16}$$

Write $W_n(\boldsymbol{\gamma}) = \sup_{\boldsymbol{\gamma}\in\Gamma^*}\big|Q_n(\boldsymbol{\gamma})-Q_n(\mathbf{0})-Q(\boldsymbol{\gamma})+Q(\mathbf{0})\big|$ and $h(\epsilon_i,\epsilon_j) = \big|(\epsilon_i-\epsilon_j)-(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}\big| - \big|\epsilon_i-\epsilon_j\big|$. Note that

$$\begin{aligned} |h(\epsilon_i,\epsilon_j)| &\leq \big|(\mathbf{x}_i-\mathbf{x}_j)^T\boldsymbol{\gamma}\big| \leq 2b_1||\boldsymbol{\gamma}||_1 \leq 2b_1(1+\bar{c})||\boldsymbol{\gamma}_A||_1 \\ &\leq 2b_1(1+\bar{c})\sqrt{q}||\boldsymbol{\gamma}_A||_2 \leq 2b_1(1+\bar{c})\sqrt{q}\Delta h_n. \end{aligned}$$

Hence if we perturb one observation of the data set, the value of $W_n(\boldsymbol{\gamma})$ changes at most $c_0(1+\bar{c})n^{-1}\sqrt{q}\Delta h_n$, where $c_0 = 4\sqrt{2}b_1c$. By the bounded difference inequality, ∀ $t>0$,

$$P\big(W_n(\boldsymbol{\gamma})-\mathrm{E}\{W_n(\boldsymbol{\gamma})\}>t\big)\leq\exp\Big(-\frac{2nt^2}{c_0^2(1+\bar{c})^2q\Delta^2h_n^2}\Big). \tag{17}$$

Let $M_n = \lfloor n/2\rfloor$, the smallest integer greater than or equal to $n/2$. Let σ_i, $i=1,\ldots,n$, denote a Rademacher sequence independent of $\epsilon_1,\ldots,\epsilon_n$, such that $P(\sigma_i = 1) = P(\sigma_i =$

$-1) = 1/2$. We have

$$\begin{aligned}
\mathrm{E}\big\{W_n(\boldsymbol{\gamma})\big\} &= \mathrm{E}\Big(\sup_{\boldsymbol{\gamma}\in\Gamma^*}\frac{1}{n!}\Big|\sum_{\pi} M_n^{-1}\sum_{i=1}^{M_n}\big(h(\epsilon_{\pi(i)},\epsilon_{\pi(M_n+i)} - \mathrm{E}\{h(\epsilon_{\pi(i)},\epsilon_{\pi(M_n+i)})\}\Big|\Big) \\
&\le \frac{2}{n!}\sum_{\pi}\mathrm{E}\Big(\sup_{\boldsymbol{\gamma}\in\Gamma^*}\Big|M_n^{-1}\sum_{i=1}^{M_n}\sigma_i h(\epsilon_{\pi(i)},\epsilon_{\pi(M_n+i)})\Big|\Big) \\
&\le \frac{4}{n!}\sum_{\pi}\mathrm{E}\Big(\sup_{\boldsymbol{\gamma}\in\Gamma^*}\Big|M_n^{-1}\sum_{i=1}^{M_n}\sigma_i(\mathbf{x}_{\pi(i)} - \mathbf{x}_{\pi(M_n+i)})^T\boldsymbol{\gamma}\Big|\Big) \\
&\le \frac{4}{n!}\sum_{\pi}\sup_{\boldsymbol{\gamma}\in\Gamma^*}||\boldsymbol{\gamma}||_1\mathrm{E}\Big(\Big|\Big|M_n^{-1}\sum_{i=1}^{M_n}\sigma_i(\mathbf{x}_{\pi(i)} - \mathbf{x}_{\pi(M_n+i)})\Big|\Big|_\infty\Big) \\
&\le 4(1+\bar{c})\sqrt{q}\Delta h_n\frac{1}{n!}\sum_{\pi}\mathrm{E}\Big(\max_{1\le j\le p}\Big|M_n^{-1}\sum_{i=1}^{M_n}\sigma_i(x_{\pi(i)j} - x_{\pi(M_n+i)j})\Big|\Big)
\end{aligned}$$

where the equality is a result of Lemma A.1 of Clémençon et al. (2008) on U-statistic with the first sum taking over all permutations π of $\{1,\ldots,n\}$; the second inequality follows from the symmetrization theorem (van der Vaart and Wellner, 1996); the third inequality is due to the contraction theorem (Ledoux and Talagrand, 2013); while the last inequality follows because $\sup_{\boldsymbol{\gamma}\in\Gamma^*}||\boldsymbol{\gamma}||_1 \le (1+\bar{c})\sqrt{q}||\boldsymbol{\gamma}_A||_2$. Applying Lemma 14.12 of Bühlmann and van de Geer (2011), we have

$$\begin{aligned}
\mathrm{E}\Big(\max_{1\le j\le p}\Big|M_n^{-1}\sum_{i=1}^{M_n}\sigma_i(x_{\pi(i)j} - x_{\pi(M_n+i)j})\Big|\Big) &\le 4b_1\big\{n^{-1}\log(p+1) + \sqrt{2n^{-1}\log(p+1)}\big\} \\
&\le 8b_1\sqrt{n^{-1}\log p},
\end{aligned}$$

for all n sufficiently large. This implies $\mathrm{E}\big\{W_n(\boldsymbol{\gamma})\big\} \le 6(1+\bar{c})c_0\Delta h_n^2$. Now we take $t = (1+\bar{c})c_0\Delta h_n^2$ in (17). This gives

$$P\Big(W_n(\boldsymbol{\gamma}) - \mathrm{E}\{W_n(\boldsymbol{\gamma})\} > (1+\bar{c})c_0\Delta h_n^2\sqrt{n^{-1}q\log p}\Big) \le \exp(-2\log p).$$

Hence, $W_n(\boldsymbol{\gamma}) \le 7(1+\bar{c})c_0\Delta h_n^2$ with probability at least $1-\exp(-2\log p)$. Combining this result with (15) and (16), we have with probability at least $1-\exp(-2\log p)$,

$$\begin{aligned}\inf_{\boldsymbol{\gamma}\in\Gamma^*}\big(L_n(\boldsymbol{\gamma})-L_n(\mathbf{0})\big) &\geq b_2b_3\Delta^2h_n^2-7(1+\bar{c})c_0\Delta h_n^2-2c_0(1+\bar{c})\Delta h_n^2\\ &= (b_2b_3\Delta-7(1+\bar{c})c_0)\Delta h_n^2>0.\end{aligned}$$

Hence $P\big(||\widehat{\boldsymbol{\gamma}}(\lambda^*)||_2\leq\Delta h_n\big)>1-\alpha_0-\exp(-2\log p)$. □

Proof of Theorem 2. Let $\widetilde{\boldsymbol{\beta}}^{(0)}=(\widetilde{\beta}_1^{(0)},\ldots,\widetilde{\beta}_p^{(0)})^T$ be the initial estimator obtained from the L_1 penalized Wilcoxon rank regression. By Theorem 1, $\sup_{1\leq j\leq p}|\widetilde{\beta}_j^{(0)}-\beta_{0j}|\leq\tilde{b}\sqrt{q\log p/n}$ for some positive constant $\tilde{b}$ with probability at least $1-\alpha_0-\exp(-2\log p)$. By the conditions of Theorem 2, we have $\sqrt{q\log p/n}=O(n^{-(1-c_1-c_4)/2})$. By Lemma B in the supplementary material, for the oracle estimator $\widehat{\boldsymbol{\beta}}^{(o)}$ there exist v_{ij}^* which satisfies $v_{ij}^*=0$ if $Y_i-Y_j\neq(\mathbf{x}_i-\mathbf{x}_j)^T\widehat{\boldsymbol{\beta}}^{(o)}$, and $v_{ij}^*\in[-1,1]$ if $Y_i-Y_j=(\mathbf{x}_i-\mathbf{x}_j)^T\widehat{\boldsymbol{\beta}}^{(o)}$, such that for $\delta_k(\widehat{\boldsymbol{\beta}}^{(o)})$ with $v_{ij}=v_{ij}^*$, we have with probability approaching one, $\delta_k(\widehat{\boldsymbol{\beta}}^{(o)})=0$, $k=1,\ldots,q$; and $|\delta_k(\widehat{\boldsymbol{\beta}}^{(o)})|<a_1\eta$, $k=q+1,\ldots,p$. Consider $\delta_k(\widehat{\boldsymbol{\beta}}^{(o)})$ with $v_{ij}=v_{ij}^*$. Define the following two events,

$$\begin{aligned}F_{n1} &= \big\{|\widetilde{\beta}_j^{(0)}-\beta_{0j}|>\eta,\ \text{for some } 1\leq j\leq p\big\},\\ F_{n2} &= \big\{|\delta_k(\widehat{\boldsymbol{\beta}}^{(o)})|\geq a_1\eta, \text{for some } q+1\leq k\leq p;\ \text{or } \delta_k(\widehat{\boldsymbol{\beta}}^{(o)})\neq 0 \text{ for some } 0\leq k\leq q\big\}\end{aligned}$$

where $\widehat{\boldsymbol{\beta}}^{(o)}=(\widehat{\beta}_1^{(o)},\ldots,\widehat{\beta}_p^{(o)})^T$ is the oracle estimator. Define $G_n=F_{n1}^c\cap F_{n2}^c$. By Theorem 1 and Lemma C in the supplementary material, for all n sufficiently large, we have $P\big(G_n\big)\geq 1-\alpha_0-h_n$, where $h_n=o(1)$.

We observe, for all n sufficiently large on the event G_n, $|\widetilde{\beta}_j^{(0)}|<a_2\eta$ for $q+1\leq j\leq p$, and $|\widetilde{\beta}_j^{(0)}|\geq|\beta_{0j}|-|\widetilde{\beta}_j^{(0)}-\beta_{0j}|\geq a_2\eta$, for $1\leq j\leq q$. By the assumptions on the penalty function, we have $p_\eta'(|\widetilde{\beta}_j^{(0)}|)=0$ for $1\leq j\leq q$; and $p_\eta'(|\widetilde{\beta}_j^{(0)}|)\geq a_1\eta$ for $q+1\leq j\leq p$. Therefore, on the event G_n, the second-stage estimator $\widetilde{\boldsymbol{\beta}}^{(1)}$ can be expressed as the solution

to the following convex optimization problem:

$$\widetilde{\boldsymbol{\beta}}^{(1)} = \arg\min_{\boldsymbol{\beta}} \big[n(n-1)\big]^{-1} \sum\sum_{i \neq j} \big|(Y_i - \mathbf{x}_i^T \boldsymbol{\beta}) - (Y_j - \mathbf{x}_j^T \boldsymbol{\beta})\big| + \sum_{k=q+1}^{p} p_\eta'(|\widetilde{\beta}_k|)|\beta_k|. \quad (18)$$

By the property of the subgradient of a convex function, we have, $\forall\ \boldsymbol{\beta} \in R^p$,

$$\begin{aligned} & \big[n(n-1)\big]^{-1} \sum\sum_{i \neq j} \big|(Y_i - \mathbf{x}_i^T \boldsymbol{\beta}) - (Y_j - \mathbf{x}_j^T \boldsymbol{\beta})\big| \\ \geq\ & \big[n(n-1)\big]^{-1} \sum\sum_{i \neq j} \big|(Y_i - \mathbf{x}_i^T \widehat{\boldsymbol{\beta}}^{(o)}) - (Y_j - \mathbf{x}_j^T \widehat{\boldsymbol{\beta}}^{(o)})\big| + \sum_{k=1}^{p} \delta_k(\widehat{\boldsymbol{\beta}}^{(o)})(\beta_k - \widehat{\beta}_k^{(o)}) \\ =\ & \big[n(n-1)\big]^{-1} \sum\sum_{i \neq j} \big|(Y_i - \mathbf{x}_i^T \widehat{\boldsymbol{\beta}}^{(o)}) - (Y_j - \mathbf{x}_j^T \widehat{\boldsymbol{\beta}}^{(o)})\big| + \sum_{k=q+1}^{p} \delta_k(\widehat{\boldsymbol{\beta}}^{(o)})(\beta_k - \widehat{\beta}_k^{(o)}). \end{aligned}$$

Hence, $\forall\ \boldsymbol{\beta} \in R^p$,

$$\begin{aligned} & \Big\{ \big[n(n-1)\big]^{-1} \sum\sum_{i \neq j} \big|(Y_i - \mathbf{x}_i^T \boldsymbol{\beta}) - (Y_j - \mathbf{x}_j^T \boldsymbol{\beta})\big| + \sum_{k=q+1}^{p} p_\eta'(|\widetilde{\beta}_k^{(0)}|)|\beta_k| \Big\} \\ & - \Big\{ \big[n(n-1)\big]^{-1} \sum\sum_{i \neq j} \big|(Y_i - \mathbf{x}_i^T \widehat{\boldsymbol{\beta}}^{(o)}) - (Y_j - \mathbf{x}_j^T \widehat{\boldsymbol{\beta}}^{(o)})\big| + \sum_{k=q+1}^{p} p_\eta'(|\widetilde{\beta}_k^{(0)}|)|\widehat{\beta}_k^{(o)}| \Big\} \\ \geq\ & \sum_{k=q+1}^{p} \Big\{ p_\eta'(|\widetilde{\beta}_k^{(0)}|) + \delta_k(\widehat{\boldsymbol{\beta}}^{(o)}) \mathrm{sign}(\beta_k) \Big\} |\beta_k| \\ \geq\ & \sum_{k=q+1}^{p} \Big\{ a_1 \eta - |\delta_k(\widehat{\boldsymbol{\beta}}^{(o)})| \Big\} |\beta_k| \geq 0. \end{aligned}$$

since $\widehat{\beta}_k^{(o)} = 0$ for $k = q+1, \ldots, p$. The inequality is strict unless $\beta_k = 0$ for $k = q+1, \ldots, p$. This implies on G_n, $\widetilde{\boldsymbol{\beta}}^{(1)} = (\widetilde{\boldsymbol{\beta}}_1^{(1)T}, \mathbf{0}_{p-q}^T)^T$ with $\widetilde{\boldsymbol{\beta}}_1^{(1)} = \arg\min_{\boldsymbol{\beta}_1} \big[n(n-1)\big]^{-1} \sum\sum_{i \neq j} \big|(Y_i - \mathbf{x}_{1i}^T \boldsymbol{\beta}_1) - (Y_j - \mathbf{x}_{1j}^T \boldsymbol{\beta}_1)\big|$. Hence, $\widetilde{\boldsymbol{\beta}}^{(1)}$ is the oracle estimator. □