Comment on "A Tuning-Free Robust and Efficient Approach to High-Dimensional Regression"

Jianqing Fan* Cong Ma† Kaizheng Wang‡

August 11, 2020

We congratulate the authors for their important and timely contributions to robust and tuning-free high dimensional linear regression. The proposed loss function originates from nonparametric rank-based estimation and enjoys certain pivotal properties that facilitate the selection of the tuning parameter. While tuning-free probably sounds over claimed, the proposed penalization parameter is indeed more interpretable, easier to select, and is independent of noise variance. We welcome the opportunity to make a few comments from various perspectives and discuss open questions that are worth studying.

1 Historical perspectives on Rank Lasso

Consider the linear model

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

where $\boldsymbol{y} = (y_1, \cdots, y_n)^\top \in \mathbb{R}^n$ is a response vector, $\boldsymbol{X} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)^\top \in \mathbb{R}^{n\times p}$ is a design matrix, $\boldsymbol{\beta}_0 \in \mathbb{R}^p$ is an unknown coefficient vector, and $\boldsymbol{\varepsilon} = (\varepsilon_1, \cdots, \varepsilon_n)^\top \in \mathbb{R}^n$ is a noise vector with i.i.d. entries. The Rank Lasso estimator [Wang et al., 2020+] for estimating $\boldsymbol{\beta}_0$ is given by

$$\hat{\boldsymbol{\beta}}(\lambda) = \operatorname*{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^p} \quad \{L_n(\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1\}, \tag{2}$$

where the empirical loss function $L_n(\boldsymbol{\beta})$ is defined as

$$L_n(\boldsymbol{\beta}) = \frac{1}{n(n-1)} \sum_{i\neq j} |(y_i - \boldsymbol{x}_i^\top\boldsymbol{\beta}) - (y_j - \boldsymbol{x}_j^\top\boldsymbol{\beta})|, \tag{3}$$

and $\lambda \geq 0$ is a regularization parameter to be chosen. This is equivalent to fitting the linear model

$$y_i - y_j = (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top\boldsymbol{\beta}_0 + (\varepsilon_i - \varepsilon_j) \tag{4}$$

using ℓ_1 regression, since the error distribution $\varepsilon_i - \varepsilon_j$ is symmetric.

The proposed Rank Lasso estimator (2) is an extension of the traditional nonparametric rank-based estimator for linear models to the high dimensional setting; see Hettmansperger and McKean [1978, 2010]. Without the ℓ_1 penalty, a general rank-based estimator takes the following form

$$\hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}\in\mathbb{R}^p} \left\{ \sum_{i=1}^n a\left(R\left(y_i - \boldsymbol{x}_i^\top\boldsymbol{\beta}\right)\right)\left(y_i - \boldsymbol{x}_i^\top\boldsymbol{\beta}\right) \right\}, \tag{5}$$

*Department of Operations Research and Financial Engineering, Princeton University, Princeton, NJ 08544, USA; Email: jqfan@princeton.edu. The research was supported by NSF grants DMS-1662139, DMS-1712591, NIH grant 2R01-GM072611-16 and ONR grant N00014-19-1-2120.

†Department of Electrical Engineering and Computer Sciences, UC Berkeley, Berkeley, CA 94720, USA; Email: congm@berkeley.edu.

‡Department of Industrial Engineering and Operations Research, Columbia University, New York, NY 10027, USA; Email: kw2934@columbia.edu.

where $R(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})$ denotes the rank of $y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}$ among the residuals $\{y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}\}_{1 \le i \le n}$ and $a(i) = \phi(i/(n+1))$ for some nondecreasing function $\phi(\cdot) : (0,1) \mapsto \mathbb{R}$ satisfying $\int \phi(u)du = 0$ and $\int \phi^2(u)du = 1$. An illustrate example of $\phi(\cdot)$ is the sign function, i.e. $\phi(u) = \mathrm{sgn}(u - 1/2)$. Under this circumstance, the general rank-based estimator (5) recovers the well-known least absolute deviation estimator.

In fact, Rank Lasso corresponds to the rank-based estimator with the Wilcoxon score function $\phi(u) = \sqrt{12}(u - 1/2)$, i.e.

$$L_n(\boldsymbol{\beta}) = \frac{2\,(n+1)}{\sqrt{3}n(n-1)} \sum_{i=1}^{n} a\left(R\left(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}\right)\right)\left(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}\right).$$

For future convenience, we denote $D(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = \sum_{i=1}^n a(R(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}))(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta})$ and rewrite the Rank Lasso estimator as

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\mathrm{argmin}} \left\{ \frac{2\,(n+1)}{\sqrt{3}n\,(n-1)} D(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) + \lambda \left\|\boldsymbol{\beta}\right\|_1 \right\}. \tag{6}$$

2 Computational efficiency and effect of subsampling

Denote by $f(\cdot)$ the density of the error distribution. The relative efficiency between the rank regression and the ℓ_1-regression for symmetric error distributions is $3[\int f^2(u)du]^2/f(0)^2$ [Hodges Jr and Lehmann, 1963], which is $0.75, 1.04, 1.25, 1.37, 1.43, 1.50$ for t_ν distribution with $\nu = 1, 2, 4, 8, 16, \infty$ (normal) and 3 for the uniform distribution. Namely, for a variety of distributions, the rank regression estimator has high efficiency in comparison with ℓ_1-regression. Why, then, the ℓ_1-regression is far more popular? One possible cause, as pointed out by the authors in Section 4.2 of Wang et al. [2020+], is the computational burden for obtaining the Rank Lasso estimator — the U-statistics structure in (3), which involves $n(n-1)$ pairs of samples. To alleviate this issue, the authors suggest a subsampling mechanism to reduce the computational cost of the Rank Lasso estimator. More specifically, $N = m \cdot n$ terms are *sampled with replacement* from $n(n-1)/2$ terms in the loss function (3), where m can vary from 1 to $(n-1)/2$. A smaller m brings more computational benefits, but larger bias as well as variance. In other words, it loses statistical efficiency when compared with using the full sample. In this section, we complement the paper by investigating empirically the effect of the sampling budget m. In addition, we propose another principled subsampling mechanism for Rank Lasso that demonstrates superiority over the above mentioned one.

To motivate the new subsampling mechanism, we recall from Section 1 that the loss function L_n is equal to the ℓ_1 loss (absolute deviation) over symmetrized samples $\{(\boldsymbol{x}_i - \boldsymbol{x}_j, y_i - y_j)\}_{i \neq j}$. When n is even and τ is a random permutation over $[n]$, a simplified loss with the same expectation is

$$\tilde{L}_n(\boldsymbol{\beta}; \tau) = \frac{2}{n} \sum_{i=1}^{n/2} |(y_{\tau(2i)} - \boldsymbol{x}_{\tau(2i)}^\top \boldsymbol{\beta}) + (y_{\tau(2i-1)} - \boldsymbol{x}_{\tau(2i-1)}^\top \boldsymbol{\beta})|, \tag{7}$$

which only concerns $n/2$ i.i.d. samples $\{(\boldsymbol{x}_{\tau(2i)} - \boldsymbol{x}_{\tau(2i-1)}, y_{\tau(2i)} - y_{\tau(2i-1)})\}_{i=1}^{n/2}$. Minimizing $\tilde{L}_n(\boldsymbol{\beta}; \tau) + \lambda\|\boldsymbol{\beta}\|_1$ amounts to ℓ_1-penalized least absolute deviations, which can be studied using general results for sparse quantile regression by Belloni and Chernozhukov [2011], Fan et al. [2014]. Similar to the current paper, they also propose to compute a pivotal quantity as the tuning parameter by simulation.

The loss in (7) based on non-overlapping pairwise differences greatly facilitates computation while enjoying the same rate of convergence. Yet it may suffer from loss of efficiency. Consider the classical asymptotics with fixed p, diverging n, and random designs. Ignore the ℓ_1 penalty for a moment. Suppose that ε_i has density f, and $\mathbb{E}(\boldsymbol{x}_i \boldsymbol{x}_i^\top) = \boldsymbol{\Sigma}$. Define $\tilde{\varepsilon}_i = \varepsilon_{\tau(2i)} - \varepsilon_{\tau(2i-1)}$ and $\tilde{\boldsymbol{x}}_i = \boldsymbol{x}_{\tau(2i)} - \boldsymbol{x}_{\tau(2i-1)}$. Then the density function of $\tilde{\varepsilon}_i$ is $\tilde{f}(x) = (f * f)(x)$, where $*$ denotes convolution, and $\mathbb{E}(\tilde{\boldsymbol{x}}_i \tilde{\boldsymbol{x}}_i^\top) = 2\boldsymbol{\Sigma}$. According to Bassett and Koenker [1978], the least absolute deviations estimator $\tilde{\boldsymbol{\beta}}_n = \mathrm{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^p} \tilde{L}_n(\boldsymbol{\beta}; \tau)$ satisfies

$$\sqrt{\frac{n}{2}}(\tilde{\boldsymbol{\beta}}_n - \boldsymbol{\beta}_0) \xrightarrow{\mathrm{d}} \mathcal{N}\left(\boldsymbol{0}, \frac{1}{[2\tilde{f}(0)]^2} \tilde{\boldsymbol{\Sigma}}^{-1}\right) = \mathcal{N}\left(\boldsymbol{0}, \frac{1}{8[\int_{\mathbb{R}} f^2(x)\mathrm{d}x]^2} \boldsymbol{\Sigma}^{-1}\right).$$

The asymptotic relative efficiency of $\tilde{\boldsymbol{\beta}}_n$ with respect to the least squares estimate is $4\sigma^2[\int_{\mathbb{R}} f^2(x)\mathrm{d}x]^2$, which is $1/3$ of that for the estimator based on Jaeckel's Wilcoxon-type dispersion function in Wang et al. [2020+]. A natural idea to bridge this gap is to average the loss (7) first among a few permutations and then optimize the

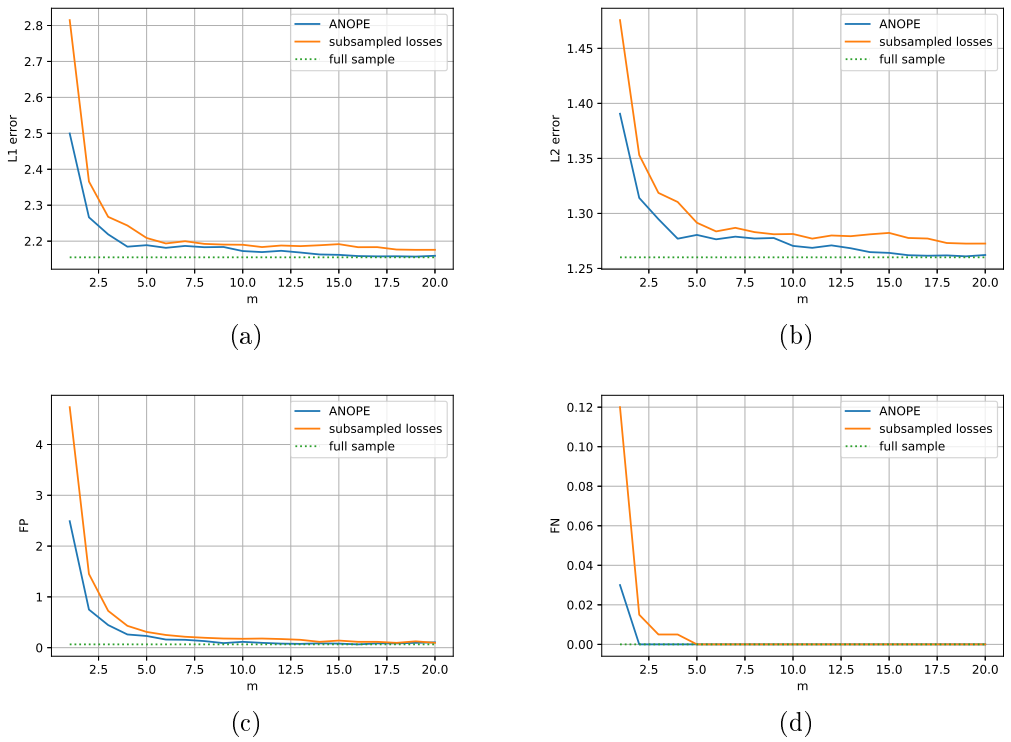

Figure 1: Different error metrics for estimating $\boldsymbol{\beta}_0$ vs. the sampling budget m. (a) ℓ_1 error, (b) ℓ_2 error, (c) False positives, and (d) False negatives. The results are based on the average of 200 simulations.

object. More specifically, for some positive integer m, one randomly draws $2m$ permutations $\{\tau_1, \tau_2, \cdots, \tau_{2m}\}$ over $[n]$ (so that we have mn pairs), and then seeks

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \frac{1}{2m} \sum_{i=1}^{2m} \tilde{L}_n \left(\boldsymbol{\beta}; \tau_i \right) + \lambda \left\| \boldsymbol{\beta} \right\|_1 \right\}.$$

It turns out that this average non-overlapping-pair estimator (ANOPE) bears deep connections to the Rank Lasso estimator. Indeed, the original symmetrized loss function $L_n(\boldsymbol{\beta})$ satisfies

$$L_n \left(\boldsymbol{\beta} \right) = \frac{1}{n!} \sum_{\tau} \tilde{L}_n \left(\boldsymbol{\beta}; \tau \right),$$

where the summation is over all permutations over $[n]$. In view of this connection, the ANOPE constitutes another subsampled version of the Rank Lasso estimator.

Both ANOPE and the subsampling strategy adopted by Wang et al. [2020+] are incomplete U-statistics that use mn out of $n(n-1)/2$ terms to approximate the complete version $L_n(\boldsymbol{\beta})$ in (3). All sampling mechnisms generating mn pairs of indices from $\{(i,j) : \ 1 \le i < j \le n\}$ independently of the data $\{\mathbf{x}_i, y_i\}_{i=1}^n$ produce unbiased estimates for $\mathbb{E}L_n(\boldsymbol{\beta})$. Among them, ANOPE has the minimum variance, according to Example 1 in [Lee, 2019, Section 4.3.2]. In addition, the first few m provide most variance reduction, as new pairs provide more independent information. When m is sufficiently large, additional pairs do not add much information. Let us illustrate these by a simulation study.

To compare the two different subsampling-based estimators, we generate $\boldsymbol{x}_i$ from $\mathcal{N}(\mathbf{0}, \boldsymbol{I}_p)$ and $\varepsilon_i \sim \mathcal{N}(0,1)$. We take $n = 100, p = 400$, and $\boldsymbol{\beta}_0 = (\sqrt{3}, \sqrt{3}, \sqrt{3}, 0, \cdots 0)^\top$. We vary m from 1 to 20, and in addition we take into account the full sample regime where no subsampling is used. Note that same m yields the same number of pairwise losses in both estimators. For the purpose of comparison, we use the same error metrics as in Wang et al. [2020+], namely the ℓ_1 error, ℓ_2 error, the number of false positives and the number of false negatives. Their averages over 200 trials are plotted in Figure 1. Two crucial observations are worth mentioning. First,

for a wide range of choices of m, the estimator based on subsampling permutations performs uniformly better. Second, $m = 5$ seems to be a good choice for practical implementations to balance computational and statistical efficiency. Third, even with $m = 1$, approximately the same computation as penalized LAD, the efficiency of ANOPE is already very high, approximately $(\frac{1.26}{1.39})^2 = 0.82$ in ℓ_2 error. Therefore, for the remaining experiments in this paper, we take $m = 5$, though we are aware m depends on various factors in the simulation, including the tails of the design and noise.

3 Choice of the regularization parameter λ

In this section, we focus on the choice of the regularization parameter λ in (6). Let $\boldsymbol{S}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})$ be the negative gradient of $D(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})$[1], i.e.

$$\boldsymbol{S}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = -\nabla D(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = \frac{\sqrt{3}}{n+1}\boldsymbol{X}^\top \big[2\boldsymbol{r}(\boldsymbol{\beta}) - (n+1)\big], \tag{8}$$

with $\boldsymbol{r}(\boldsymbol{\beta}) \in \mathbb{R}^n$ being the rank vector of the residuals $\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}$. The gradient at $\boldsymbol{\beta} = \boldsymbol{\beta}_0$ plays an important role in the choice of the regularization parameter λ. Note that $\boldsymbol{S}_n$ defined in the paper has the following equivalent representation

$$\boldsymbol{S}_n = -\frac{2}{\sqrt{3}}\frac{n+1}{\sqrt{n}(n-1)}\frac{1}{\sqrt{n}}\boldsymbol{S}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0).$$

The recommended choice of λ is then given by

$$\lambda^\star = c \cdot G^{-1}_{\|\boldsymbol{S}_n\|_\infty}(1 - \alpha_0)$$

with $c = 1.01$ and $\alpha_0 = 0.1$, where $G^{-1}_{\|\boldsymbol{S}_n\|_\infty}(1 - \alpha_0)$ denotes the $(1 - \alpha_0)$-quantile of the distribution of $\|\boldsymbol{S}_n\|_\infty$. We now develop a basic understanding of the distribution of $\boldsymbol{S}_n$. It has been shown in [Hettmansperger and McKean, 2010, Theorem 3.5.2] that

$$\frac{1}{\sqrt{n}}\boldsymbol{S}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}_0) \xrightarrow{\mathrm{d}} \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$$

in the fixed p, large n regime. Here $\boldsymbol{\Sigma} \coloneqq \lim_n \boldsymbol{X}'\boldsymbol{X}/n$ is the limit of the covariance matrix of $\boldsymbol{X}$. This motivates another way to select the regularization parameter

$$\lambda^{\mathrm{new}} = 1.01 \cdot \frac{2}{\sqrt{3}}\frac{n+1}{\sqrt{n}(n-1)}G^{-1}_{\|\boldsymbol{\nu}\|_\infty}(1 - \alpha_0),$$

where $\boldsymbol{\nu} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{X}'\boldsymbol{X}/n)$, which can be simulated via multiplier bootstrap, that is $\boldsymbol{\nu} = n^{-1/2}\sum_{i=1}^n \eta_i \mathbf{X}_i$ with $\eta_i \sim \mathcal{N}(0, 1)$. The consistency of the multiplier bootstrap has been established [Chernozhukov et al., 2014, Fan et al., 2018]. An advantage of this choice is it does not depends unknown $\boldsymbol{\beta}_0$ and can be computed before running Rank Lasso.

4 Bridging Rank Lasso and other pivotal procedures

Thanks to the pivotal property of L_n's subgradient function (see Section 2.2 in Wang et al. [2020+]), tuning can be easily done via simulation without any knowledge of the error distribution. In particular, the selection of λ is not affected by the size of noise. In stark contrast, the optimal λ for Lasso

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \quad \Big\{\frac{1}{2n}\|\boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_1\Big\} \tag{9}$$

is of order $n^{-1}\|\boldsymbol{X}^\top\boldsymbol{\varepsilon}\|_\infty$ that scales linearly in the standard deviation of ε_i's.

We now take a closer look at the loss function L_n in (3) under a Gaussian model and then draw links between the Rank Lasso and two other popular approaches, square-root Lasso [Belloni et al., 2011] and scaled

[1] Since $D(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})$ is piecewise linear in $\boldsymbol{\beta}$, the negative gradient $\boldsymbol{S}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})$ is defined everywhere excluding the linear boundaries.

Lasso [Sun and Zhang, 2012], with similar pivotal properties. Suppose that $\{\boldsymbol{x}_i\}_{i=1}^n$ are i.i.d. copies $N(\mathbf{0}, \boldsymbol{\Sigma})$ and $\boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2 \mathbf{I}_n)$ is independent of $\boldsymbol{X}$. By (1), the population version of L_n is

$$L(\boldsymbol{\beta}) = \mathbb{E} L_n(\boldsymbol{\beta}) = \mathbb{E}|(y_1 - \boldsymbol{x}_1^\top \boldsymbol{\beta}) - (y_2 - \boldsymbol{x}_2^\top \boldsymbol{\beta})|.$$

Since

$$\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} \sim N\Big(\mathbf{0}, [\sigma^2 + (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top \boldsymbol{\Sigma} (\boldsymbol{\beta} - \boldsymbol{\beta}_0)] \mathbf{I}_n\Big) \tag{10}$$

has i.i.d. centered Gaussian entries, we have

$$\begin{aligned} L(\boldsymbol{\beta}) &= \sqrt{\frac{2}{\pi}} \mathbb{E}^{1/2} |(y_1 - \boldsymbol{x}_1^\top \boldsymbol{\beta}) - (y_2 - \boldsymbol{x}_2^\top \boldsymbol{\beta})|^2 \\ &= \frac{2}{\sqrt{\pi}} \mathbb{E}^{1/2} |y_1 - \boldsymbol{x}_1^\top \boldsymbol{\beta}|^2 = \frac{2}{\sqrt{\pi n}} \mathbb{E}^{1/2} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2. \end{aligned} \tag{11}$$

Then

$$L(\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1 = \frac{2}{\sqrt{\pi}} \left(n^{-1} \mathbb{E} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \right)^{1/2} + \lambda \|\boldsymbol{\beta}\|_1.$$

The right-hand side can be viewed as a population version of the penalized loss in square-root Lasso [Belloni et al., 2011].

On the other hand, (10) and (11) lead to

$$L(\boldsymbol{\beta}) = \frac{2}{\sqrt{\pi}} \sqrt{\sigma^2 + (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top \boldsymbol{\Sigma} (\boldsymbol{\beta} - \boldsymbol{\beta}_0)} = \frac{2\sigma}{\sqrt{\pi}} [1 + \sigma^{-2} \mathbb{E} |\boldsymbol{x}_1^\top (\boldsymbol{\beta} - \boldsymbol{\beta}_0)|^2]^{1/2}.$$

The term

$$\mathbb{E}|\boldsymbol{x}_1^\top (\boldsymbol{\beta} - \boldsymbol{\beta}_0)|^2 = \mathbb{E}|y_1 - \boldsymbol{x}_1^\top \boldsymbol{\beta}|^2 - \mathbb{E}|y_1 - \boldsymbol{x}_1^\top \boldsymbol{\beta}_0|^2$$

is the excess risk of $\boldsymbol{\beta}$. When $\mathbb{E}|\boldsymbol{x}_1^\top (\boldsymbol{\beta} - \boldsymbol{\beta}_0)|^2 \ll \sigma^2$, we use Taylor expansion to derive

$$\begin{aligned} L(\boldsymbol{\beta}) &\approx \frac{2\sigma}{\sqrt{\pi}} \left(1 + \frac{1}{2\sigma^2} \mathbb{E} |\boldsymbol{x}_1^\top (\boldsymbol{\beta} - \boldsymbol{\beta}_0)|^2 \right) \\ &= \frac{2\sigma}{\sqrt{\pi}} \left(1 + \frac{\mathbb{E}|y_1 - \boldsymbol{x}_1^\top \boldsymbol{\beta}|^2 - \mathbb{E}|y_1 - \boldsymbol{x}_1^\top \boldsymbol{\beta}_0|^2}{2\sigma^2} \right) \\ &= \frac{\sigma}{\sqrt{\pi}} \left(1 + \frac{\mathbb{E}|y_1 - \boldsymbol{x}_1^\top \boldsymbol{\beta}|^2}{\sigma^2} \right). \end{aligned}$$

Therefore,

$$L(\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1 \approx \frac{1}{\sqrt{\pi}} \left(\sigma + \frac{1}{n\sigma} \mathbb{E} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 \right) + \lambda \|\boldsymbol{\beta}\|_1.$$

This time the right-hand side becomes a population version of the penalized loss in scaled Lasso [Sun and Zhang, 2012].

5 Inference based on Rank Lasso

Here we explore the possibility of conducting inference (e.g. constructing confidence intervals for coordinates in $\boldsymbol{\beta}_0$) using the Rank Lasso estimator. As in the standard Lasso estimator, the Rank Lasso estimator $\hat{\boldsymbol{\beta}}$ is no longer unbiased for $\boldsymbol{\beta}_0$ due to the ℓ_1 regularizer. Inspired by the de-biased Lasso estimator Zhang and Zhang [2014], Van de Geer et al. [2014], Javanmard and Montanari [2014], a natural approach to inference is to remove the bias in $\hat{\boldsymbol{\beta}}$ and construct an asymptotically unbiased estimator $\hat{\boldsymbol{\beta}}^{\mathrm{d}}$, whose distribution is easy to characterize.

In what follows, we sketch a heuristic argument that would ultimately lead us towards a de-biased estimator. In view of the definition (6), $\hat{\boldsymbol{\beta}}$ satisfies the first order optimality condition of (6)

$$-\frac{2\,(n+1)}{\sqrt{3}n\,(n-1)}\boldsymbol{S}(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})+\lambda\partial\|\hat{\boldsymbol{\beta}}\|_1=\boldsymbol{0}. \tag{12}$$

Here, $\partial\|\hat{\boldsymbol{\beta}}\|_1$ denotes a subgradient of $\|\boldsymbol{\beta}\|_1$ at $\boldsymbol{\beta}=\hat{\boldsymbol{\beta}}$, $\boldsymbol{S}\,(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})$ is the negative gradient of $D\,(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})$ defined in (8). In view of the asymptotic theory for rank based estimators in the low dimensional setting, $\boldsymbol{S}\,(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})$ admits a linear approximation [Hettmansperger and McKean, 2010, Theorem A.3.1], i.e.

$$\frac{1}{\sqrt{n}}\boldsymbol{S}\,(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})\approx\frac{1}{\sqrt{n}}\boldsymbol{S}\,(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)-\sqrt{12}\int f^2\,(u)\,du\boldsymbol{\Sigma}\cdot\sqrt{n}\,(\boldsymbol{\beta}-\boldsymbol{\beta}_0)\,, \tag{13}$$

where $\boldsymbol{\Sigma}$ is the sample covariance of $\boldsymbol{X}$ and $f(u)$ denotes the density function of the error $\boldsymbol{\varepsilon}$. Combine (12) and (13) to reach[2]

$$\sqrt{n}\left(\hat{\boldsymbol{\beta}}+\boldsymbol{\Sigma}^{-1}\frac{1}{\int f^2\,(u)\,du}\frac{n-1}{4\,(n+1)}\lambda\partial\|\hat{\boldsymbol{\beta}}\|_1-\boldsymbol{\beta}_0\right)\approx\frac{1}{\sqrt{12}\int f^2\,(u)\,du}\boldsymbol{\Sigma}^{-1}\frac{1}{\sqrt{n}}\boldsymbol{S}\,(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)\,.$$

Use the optimality condition (12) and the definition of $S(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta})$ to arrive at

$$\begin{aligned}
&\sqrt{n}\left(\hat{\boldsymbol{\beta}}+\frac{1}{\int f^2\,(u)\,du}\frac{1}{2n\,(n+1)}\boldsymbol{\Sigma}^{-1}\boldsymbol{X}^\top\left(2\boldsymbol{r}(\hat{\boldsymbol{\beta}})-(n+1)\right)-\boldsymbol{\beta}_0\right)\\
&\quad\approx\frac{1}{\sqrt{12}\int f^2\,(u)\,du}\boldsymbol{\Sigma}^{-1}\frac{1}{\sqrt{n}}\boldsymbol{S}\,(\boldsymbol{y}-\boldsymbol{X}\boldsymbol{\beta}_0)\\
&\quad=\frac{1}{\int f^2\,(u)\,du}\frac{1}{2\sqrt{n}\,(n+1)}\boldsymbol{\Sigma}^{-1}\boldsymbol{X}^\top\left(2\boldsymbol{r}\,(\boldsymbol{\beta}_0)-(n+1)\right)=:\boldsymbol{\Xi}.
\end{aligned}$$

Clearly, the right hand side $\boldsymbol{\Xi}$ has zero mean and is asymptotically normal. In fact, $\boldsymbol{\Xi}$ is closely related to $\boldsymbol{S}_n$ defined in the paper:

$$\frac{1}{\sqrt{n}}\boldsymbol{\Xi}=\frac{1}{4\int f^2\,(u)\,du}\boldsymbol{\Sigma}^{-1}\boldsymbol{S}_n. \tag{14}$$

This motivates the construction of the following de-biased estimator[3]

$$\hat{\boldsymbol{\beta}}^{\mathrm{d}}:=\hat{\boldsymbol{\beta}}+\frac{1}{\int f^2\,(u)\,du}\frac{1}{2n\,(n+1)}\boldsymbol{\Sigma}^{-1}\boldsymbol{X}^\top\left(2\boldsymbol{r}(\hat{\boldsymbol{\beta}})-(n+1)\right). \tag{15}$$

In addition, for each $1\le j\le p$, a valid $(1-\alpha)$-confidence interval of β_{0j} is given by

$$[\hat{\beta}_j^{\mathrm{d}}-G^{-1}_{\Xi_j/\sqrt{n}}(1-\alpha/2),\quad\hat{\beta}_j^{\mathrm{d}}-G^{-1}_{\Xi_j/\sqrt{n}}(1-\alpha/2)],$$

where $G^{-1}_{\Xi_j/\sqrt{n}}(1-\alpha/2)$ denotes the $(1-\alpha/2)$-quantile of the distribution of $\Xi_j/\sqrt{n}$.

Below, we empirically demonstrate the validity of the de-biased estimator $\hat{\boldsymbol{\beta}}^{\mathrm{d}}$. The experimental setup is similar to that in Section 2 except that we set $n=600$ and $p=400$. For any set $\mathcal{S}\subseteq[p]$, we define the coverage probability of $\hat{\boldsymbol{\beta}}^{\mathrm{d}}$ on $\mathcal{S}$ to be

$$\mathrm{Coverage}\,(\mathcal{S})=\frac{1}{|S|}\sum_{j\in\mathcal{S}}1\left\{\beta_{0j}\in[\hat{\beta}_j^{\mathrm{d}}-G^{-1}_{\Xi_j/\sqrt{n}}(1-\alpha/2),\quad\hat{\beta}_j^{\mathrm{d}}-G^{-1}_{\Xi_j/\sqrt{n}}(1-\alpha/2)]\right\}.$$

In particular, we focus on the coverage probability on important variables $\mathcal{S}_0:=\{j\in[p]\,|\,\beta_{0j}\neq0\}$, unimportant variables $\mathcal{S}_0^c$ and all the variables $\Omega=[p]$. Table 1 reports the averaged results over 200 Monte Carlo simulations for $1-\alpha=0.9$. As can be seen from Table 1, the average coverage probabilities are quite close to the nominal level 90%. While the de-biasing approach gives correct coverage, the confidence intervals are much wider than the oracle estimator, for both active and inactive components. For example, for inactive components, any intervals, however small, give 100% coverage, as long as they contain the origin. How to construct confidence intervals taking more into account of the lengths of the intervals, in addition to the correct coverage? This requires more effective use of sparsity.

[2]For simplicity, here we assume the sample covariance is invertible (which requires $n>p$), otherwise an estimator of the inverse covariance is needed as in Javanmard and Montanari [2014].

[3]A consistent estimator of $\int f^2\,(u)\,du$ is needed to make this fully practical.

Table 1: Average coverage probabilities for 90% confidence intervals

	Coverage ($\mathcal{S}_0$)	Coverage ($\mathcal{S}_0^c$)	Coverage ($[p]$)
Normal	89.67%	90.71%	90.70%
Cauchy	88.50%	90.43%	90.42%

6 Regularization under strongly dependent covariates

The main theorem is established under restricted eigenvalue condition in Wang et al. [2020+]. This requires the weakly dependent covariates, which usually do not hold in high-dimensional setting, as high-dimensional covariates often measures similar things (e.g. economic health, financial returns, gene expressions). The strongly dependent covariates are often modeled through common factors and the factor adjustments are needed in the regularization [Fan et al., 2020a].

Suppose that $\{\boldsymbol{x}_i\}_{i=1}^n$ are generated from the approximate factor model

$$\boldsymbol{x}_i = \boldsymbol{B}\boldsymbol{f}_i + \boldsymbol{u}_i, \qquad i \in [n], \tag{16}$$

where $\boldsymbol{B} \in \mathbb{R}^{p\times K}$ is a loading matrix, $\boldsymbol{f}_i \in \mathbb{R}^K$ gives the latent factors, and $\boldsymbol{u}_i \in \mathbb{R}^p$ records idiosyncratic components which are weak dependent. Factor-Adjusted Regularized Model Selection (FarmSelect) [Fan et al., 2020a] decorrelates the covariates as follows. Note that

$$y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta}_0 + \varepsilon_i = \boldsymbol{f}_i^\top \boldsymbol{B}^\top \boldsymbol{\beta}_0 - \boldsymbol{u}_i^\top \boldsymbol{\beta}_0 + \varepsilon_i. \tag{17}$$

If $\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^n$ are observable, then we can perform Rank Lasso using $(\boldsymbol{f}_i, \boldsymbol{u}_i)$ rather than $\boldsymbol{x}_i$:

$$(\hat{\boldsymbol{\gamma}}, \hat{\boldsymbol{\beta}}) \in \underset{\boldsymbol{\gamma}\in\mathbb{R}^K,\ \boldsymbol{\beta}\in\mathbb{R}^p}{\operatorname{argmin}} \left\{ \frac{1}{n(n-1)} \sum_{i\neq j} |(y_i - \boldsymbol{f}_i^\top \boldsymbol{\gamma} - \boldsymbol{u}_i^\top \boldsymbol{\beta}) - (y_j - \boldsymbol{f}_j^\top \boldsymbol{\gamma} - \boldsymbol{u}_j^\top \boldsymbol{\beta})| + \lambda \|\boldsymbol{\beta}\|_1 \right\}, \tag{18}$$

and then use $\hat{\boldsymbol{\beta}}$ as the estimate for $\boldsymbol{\beta}_0$. Here we only enforce the sparsity of $\boldsymbol{\beta}$. Regression using the factors and idiosyncratic components facilitates model selection as the new covariates have weaker correlations. When the factors are not observable, we replace $\{\boldsymbol{f}_i, \boldsymbol{u}_i\}_{i=1}^n$ in (18) by their empirical estimates from $\{\boldsymbol{x}_i\}_{i=1}^n$ by using the principal component analysis.

We now demonstrate the efficacy of the procedure above on synthetic data. Consider the linear model (1) with $n = 100$, $p = 400$, $\boldsymbol{\beta}_0 = (2, 2, 2, 0, \cdots, 0)^\top \in \mathbb{R}^p$ and $\varepsilon_i \sim N(0, 1)$. Let $\boldsymbol{x}_i \sim N(\boldsymbol{0}, \rho \boldsymbol{1}\boldsymbol{1}^\top + (1-\rho)\mathbf{I})$ with $\rho = 0.2$. This is equivalent to the factor model (16) with $K = 1$, $\mathbf{B} = \sqrt{\rho}\mathbf{1} \in \mathbb{R}^p$, $\boldsymbol{f}_i \sim N(0, 1)$, and $\boldsymbol{u}_i \sim N(\boldsymbol{0}, (1-\rho)\mathbf{I})$ independent of $\boldsymbol{f}_i$. We compare the Rank Lasso estimator (2) and its decorrelated version (18) with $\boldsymbol{f}_i$ and $\boldsymbol{u}_i$ estimated using principal component analysis, see Section 3.1 in Fan et al. [2020a]. Throughout the experiment, we choose the penalty parameters using the simulation method [Wang et al., 2020+] with 500 replicates, fix $\alpha_0 = 0.1$ and vary c from 1 to 2.2 by 0.01. To speedup computation, we subsample $5n$ pairs in $\{(i, j)\}_{1\le i<j\le n}$ with replacement to approximate the U-statistics. Figure 2 shows the model selection errors, which are averaged over 200 independent runs.

The selection error is the total number of false positives and false negatives, i.e. the symmetric difference between the selected set of variables and the true one. Since the average selection error of Rank Lasso is bounded away from zero, it cannot consistently identify the true set. Fortunately, decorrelation helps select the true model. When both methods achieve their minimum selection error ($c \approx 1.6$ for Rank Lasso and $c \approx 1.15$ for the decorrelated one), the ℓ_2 error of the Rank Lasso (1.84) is higher than that of its decorrelated version (1.47). The figure also shows that the regularization parameter c or equivalently λ is sensitive to the regularization.

If the covariates are correlated, ℓ_1-regularized sparse regression needs a large penalty parameter in order to return a set of variables close to the true one. That inevitably induces high bias and may jeopardize the estimation accuracy. While model selection and parameter estimation can be at odds with each other, the decorrelation step before regression provides a reconciliation.

7 Further comments

There is a large literature on robust high-dimension regression based on Huber type of loss with diverging tuning parameter τ. See, for example, Catoni [2012], Devroye et al. [2016], Fan et al. [2017], Sun et al. [2020], Fan

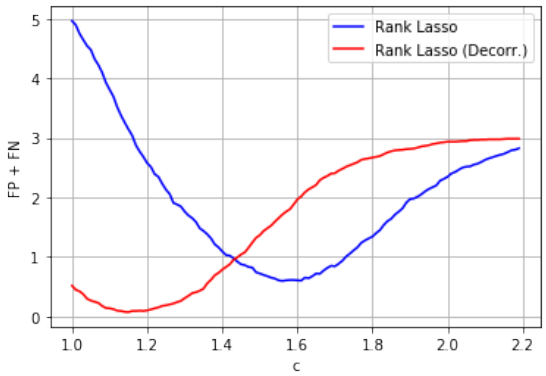

Figure 2: Rank Lasso and decorrelation: selection errors (false positives + false negatives).

et al. [2020b]. Wang et al. [2020] propose a data-driven method of the robustification parameter. The pros and cons of using Rank Lasso or adaptive Huber regression cycle back to those in the low-dimensional setting. Rank Lasso requires no moment conditions, no tuning parameters in the loss, but less efficient where error is normal and requires the error density bounded away from zero at original. This also explains that the minimax result of non-$\sqrt{n}$ -consistency of Sun et al. [2020] bears no contradiction with the result in Wang et al. [2020+].

The current paper [Wang et al., 2020+] analyze theoretical properties for the Rank Lasso under a homoscedastic model (1) where $\{\varepsilon_i\}_{i=1}^n$ are i.i.d. given $\{\boldsymbol{x}_i\}_{i=1}^n$. For such model, both the conditional median and the conditional mean of y_i given $\boldsymbol{x}_i$ are affine functions of $\boldsymbol{x}_i$, and they differ only by the intercept. Based on the observation, the ℓ_1-regularized least absolute deviations

$$\min_{\gamma\in\mathbb{R},\ \boldsymbol{\beta}\in\mathbb{R}^p} \left\{ \frac{1}{n}\sum_{i=1}^n |y_i - (\gamma + \boldsymbol{x}_i^\top \boldsymbol{\beta})| + \lambda\|\boldsymbol{\beta}\|_1 \right\} \tag{19}$$

should also serve the needs. This is computationally more attractive as we get rid of the U-statistic. The same intuition also holds for general quantile regression. Results on robustness and tuning are available, see Belloni and Chernozhukov [2011]. To better demonstrate the efficacy of Rank Lasso, one could consider, for example, a class of heteroscedastic model where $\{(\boldsymbol{x}_i, \varepsilon_i)\}_{i=1}^n$ are i.i.d. but the conditional distribution of ε_i given $\boldsymbol{x}_i = \boldsymbol{x}$ varies with $\boldsymbol{x}$. Quantile regression breaks down in this scenario. Thanks to the symmetrization step for constructing the loss (3), the Rank Lasso is still expected to enjoy nice theoretical guarantees.

In addition, it would be nice to see how to go beyond linear regression. While Rank Lasso is a convex optimization problem, things are complicated for generalized linear models such as the logistic model. It seems challenging to design a loss function, rank-based or not, that still possesses robustness and pivotal properties for nonlinear problems.

References

Lan Wang, Bo Peng, Jelena Bradic, Runze Li, and Yunan Wu. A tuning-free robust and efficient approach to high-dimensional regression. *Journal of the American Statistical Association*, 2020+.

Thomas P Hettmansperger and Joseph W McKean. Statistical inference based on ranks. *Psychometrika*, 43(1): 69–79, 1978.

Thomas P Hettmansperger and Joseph W McKean. *Robust nonparametric statistical methods*. CRC Press, 2010.

Joseph L Hodges Jr and Erich L Lehmann. Estimates of location based on rank tests. *Annals of Mathematical Statistics*, pages 598–611, 1963.

Alexandre Belloni and Victor Chernozhukov. ℓ_1-penalized quantile regression in high-dimensional sparse models. *The Annals of Statistics*, 39(1):82–130, 2011.

Jianqing Fan, Yingying Fan, and Emre Barut. Adaptive robust variable selection. *Annals of statistics*, 42(1): 324, 2014.

Gilbert Bassett and Roger Koenker. Asymptotic theory of least absolute error regression. *Journal of the American Statistical Association*, 73(363):618–622, 1978.

A J Lee. *U-statistics: Theory and Practice*. Routledge, 2019.

Victor Chernozhukov, Denis Chetverikov, Kengo Kato, et al. Gaussian approximation of suprema of empirical processes. *The Annals of Statistics*, 42(4):1564–1597, 2014.

Jianqing Fan, Qi-Man Shao, and Wen-Xin Zhou. Are discoveries spurious? distributions of maximum spurious correlations and their applications. *Annals of statistics*, 46(3):989, 2018.

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

Tingni Sun and Cun-Hui Zhang. Scaled sparse linear regression. *Biometrika*, 99(4):879–898, 2012.

Cun-Hui Zhang and Stephanie S Zhang. Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B: Statistical Methodology*, pages 217–242, 2014.

Sara Van de Geer, Peter Bühlmann, Ya'acov Ritov, and Ruben Dezeure. On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42(3):1166–1202, 2014.

Adel Javanmard and Andrea Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909, 2014.

Jianqing Fan, Yuan Ke, and Kaizheng Wang. Factor-adjusted regularized model selection. *Journal of Econometrics*, 2020a.

Olivier Catoni. Challenging the empirical mean and empirical variance: a deviation study. In *Annales de l'IHP Probabilités et statistiques*, volume 48, pages 1148–1185, 2012.

Luc Devroye, Matthieu Lerasle, Gabor Lugosi, Roberto I Oliveira, et al. Sub-gaussian mean estimators. *The Annals of Statistics*, 44(6):2695–2725, 2016.

Jianqing Fan, Quefeng Li, and Yuyan Wang. Estimation of high dimensional mean regression in the absence of symmetry and light tail assumptions. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 79(1):247, 2017.

Qiang Sun, Wen-Xin Zhou, and Jianqing Fan. Adaptive huber regression. *Journal of the American Statistical Association*, 115(529):254–265, 2020.

Jianqing Fan, Weichen Wang, and Ziwei Zhu. A shrinkage principle for heavy-tailed data: High-dimensional robust low-rank matrix recovery. *Annals of Statistics*, page to appear, 2020b.

Lili Wang, Chao Zheng, Wen Zhou, and Wen-Xin Zhou. A new principle for tuning-free huber regression. *Statistica Sinica*, 2020.