# Sparse Convoluted Rank Regression in High Dimensions

LE ZHOU*, BOXIANG WANG†AND HUI ZOU*

**Abstract**

Wang et al. (2020, JASA) studied the high-dimensional sparse penalized rank regression and established its nice theoretical properties. Compared with the least squares, rank regression can have a substantial gain in estimation efficiency while maintaining a minimal relative efficiency of 86.4%. However, the computation of penalized rank regression can be very challenging for high-dimensional data, due to the highly nonsmooth rank regression loss. In this work we view the rank regression loss as a non-smooth empirical counterpart of a population level quantity, and a smooth empirical counterpart is derived by substituting a kernel density estimator for the true distribution in the expectation calculation. This view leads to the convoluted rank regression loss and consequently the sparse penalized convoluted rank regression (CRR) for high-dimensional data. Under the same key assumptions for sparse rank regression, we establish the rate of convergence of the $\ell_1$-penalized CRR for a tuning free penalization parameter and prove the strong oracle property of the folded concave penalized CRR. We further propose a high-dimensional Bayesian information criterion for selecting the penalization parameter in folded concave penalized CRR and prove its selection consistency. We derive an efficient algorithm for solving sparse convoluted rank regression that scales well with high dimensions. Numerical examples demonstrate the promising performance of the sparse convoluted rank

*School of Statistics, University of Minnesota, Minneapolis, MN 55455, USA. Hui Zou is the corresponding author. Email: zouxx019@umn.edu. Zou's research is supported in part by NSF grants 1915842 and 2015120.

†Department of Statistics and Actuarial Science, University of Iowa, Iowa City, IA 52242, USA.

regression over the sparse rank regression. Our theoretical and numerical results suggest that sparse convoluted rank regression enjoys the best of both sparse least squares regression and sparse rank regression.

**Keywords:** Convolution, Efficiency, High dimensions, Information criterion, Rank regression

# 1 Introduction

Over the past two decades, there has been a surge of literature on high dimensional statistics. We refer to Bühlmann and Van De Geer (2011) and Fan et al. (2020) for a comprehensive review of the existing work on this topic. In particular, many penalization methods have been proposed for high-dimensional regression, including $\ell_1$-penalized regression (Tibshirani, 1996), the Dantzig selector (Candes and Tao, 2007), concave-penalized regression (Fan and Li, 2001), among others. These techniques are also applicable in other statistical models. The penalized least squares method is at the center of the stage in terms of theoretical and computational developments in high-dimensional regression. The theoretical setup typically assumes that the true model is a linear regression model with homoscedastic variance. As long as the error is sub-Gaussian, the penalized least squares estimator enjoys nice theoretical guarantees even if the number of covariates grows at a nearly exponential rate with sample size.

An approach for achieving a higher efficiency is the penalized Wilcoxon rank regression (or rank regression for short). Wilcoxon rank regression is well studied in the classical robust nonparametric statistics (Hettmansperger and McKean, 2010). The penalized rank regression was studied by several authors (Wang and Li, 2009) for the low dimension setting. Recently, penalized rank regression in high dimensional setting was fully investigated in Wang et al. (2020). The penalized rank regression solves the estimator of regression coefficient through minimizing

$$\frac{1}{n(n-1)}\sum\sum_{i\neq j}|(y_i-\mathbf{x}_i^{\mathrm{T}}\beta)-(y_j-\mathbf{x}_j^{\mathrm{T}}\beta)|+p_\lambda(|\beta_j|) \tag{1.1}$$

over $\beta\in\mathbb{R}^p$, where $p_\lambda(\cdot)$ is some penalty function. The penalized rank regression has several

advantages compared with the penalized least squares regression. First, penalized rank regression is shown to possess better efficiency than the least squares approach when error has a heavy-tailed distribution, while maintaining a good relative efficiency when error is normally distributed. Second, penalized rank regression enjoys tuning free property, which means the theoretical correct tuning parameter can be easily estimated from the dataset without any cross validation. Although tuning free property can be also obtained through other methodologies such as the square-root Lasso (Belloni, Chernozhukov and Li, 2012) and penalized quantile regression (Wang, Wu and Li, 2012), these methods do not necessarily have the first aforementioned efficiency property.

Although penalized rank regression has the aforementioned nice theoretical advantages, it can be difficult to use in practice due to computational challenges, especially when the number of covariates in the dataset is very large. It is known that high dimensional penalized regression with a smooth loss function can be efficiently computed by cyclical coordinate descent algorithm (Friedman, Hastie and Tibshirani, 2010). However, the loss function in penalized rank regression is highly non-smooth. In principle, coordinate descent may fail to deliver the right solution due to the non-smoothness of the objective function. A similar problem is quantile regression in which the check loss is nonsmooth. The computation of quantile regression is done by using interior point algorithms. One way of computing the penalized rank regression is to transform it into linear programming and then apply the interior point algorithm. However, the interior point algorithm does not scale well with high dimensions. Gu et al. (2018) developed an alternating directional method of multipliers for computing the high-dimensional quantile regression. Computationally speaking, sparse penalized rank regression is more challenging than penalized quantile regression. The interior point algorithm is not a suitable choice for solving high-dimensional sparse rank regression.

It is natural to ask whether the aforementioned good theoretical properties possessed by rank regression can be shared with good computational efficiency for practical applications. If one focuses on (1.1), then the only solution is to develop an efficient algorithm for solving (1.1) exactly for large $p$ problems. Recently, Fernades, Guerre and Horta (2021) proposed an interesting smoothing technique for solving quantile regression with statistical guarantees. They showed

that the smoothing quantile regression can even have a smaller mean squared error than the exact quantile regression for estimating the same conditional quantile function. Their work is more interesting from a statistical perspective, because fast computation for the quantile regression has already been solved in Gu et al. (2018). Their work motivated us to develop a smooth version of sparse rank regression from the statistical perspective, as opposed to trying to solve it exactly. For easy discussion, we call the first term in (1.1) the rank regression loss, although it is not like the empirical average of a loss function in empirical risk minimization. If we could replace the rank regression loss in (1.1) with a smooth loss such that the resulting estimator still has the nice theoretical properties of sparse rank regression, then we should focus on solving the smooth problem instead of (1.1). This is what Fernades, Guerre and Horta (2021) did for quantile regression. To this end, we consider the expectation of the rank regression loss with respect to the true distribution. The rank regression loss is viewed as the expectation of a random variable with respect to some empirical distribution assigning uniform discrete probability to each observed realization. If we estimate the true distribution by using a smoothed kernel density estimator, then we can take the expectation of the same random variable with respect to the smoothed kernel density estimator. The resulting quantity is shown to be smooth, convex and has a Lipschitz continuous derivative. We name it *convoluted rank loss* because the kernel density estimator has a convolution interpretation. We then replace the rank regression loss in (1.1) with the convoluted rank loss and the resulting estimator is called sparse convoluted rank regression. By its convexity and smoothness, the sparse convoluted rank regression can be efficiently solved by using the generalized coordinate descent algorithm (Yang and Zou, 2013).

We systematically study the theoretical properties of the sparse convoluted rank regression. The goal is to show that it maintains all the essential theoretical properties of rank regression. Specifically, we first establish the rate of convergence of the $\ell_1$-penalized convoluted rank regression in ultra-high dimensions without assuming a strong moment condition on the error and the $\ell_1$-penalized convoluted regression is also shown to enjoy the asymptotic tuning free property. Second, we analyze the folded concave penalized convoluted rank regression and establish its strong oracle

property without imposing strong moment conditions on the error. The folded concave penalized regression involves a tuning parameter. We thus further propose a high dimensional Bayesian information criterion (HBIC) and establish its consistency for the selection of the theoretically optimal tuning parameter.

The rest of this paper is organized as follows. In section 2, we introduce convoluted rank regression loss and the sparse convoluted rank regression estimator. In section 3, we present the theoretical justifications for the proposed estimators. We also present the HBIC criterion and its theoretical results. In section 4, we derive an efficient algorithm for solving sparse convoluted rank regression for high-dimensional data. In section 5, we use simulations and a real data example to compare sparse convoluted rank regression and sparse rank regression. The technical proofs are given in the supplement file.

# 2 Convoluted Rank Regression

In this section we present the main idea that leads to the convoluted rank regression loss and the sparse convoluted rank regression.

## 2.1 Notation and definitions

We begin with some necessary definitions. For an arbitrary index set $\mathbf{A} \subset \{1,\ldots,p\}$, any vector $\mathbf{c} = (c_1,\ldots,c_p)$ and any $n \times p$ matrix $\mathbf{U}$, let $\mathbf{c}_\mathbf{A} = (c_i, i \in \mathbf{A})$, and let $\mathbf{U}_\mathbf{A}$ be the submatrix with columns of $\mathbf{U}$ whose indices are in $\mathbf{A}$. The complement of an index set $\mathbf{A}$ is denoted as $\mathbf{A}^{\mathrm{c}} = \{1,\ldots,p\} \setminus \mathbf{A}$. For any finite set $\mathbf{B}$, let $|\mathbf{B}|$ be the number of elements in $\mathbf{B}$. For a vector $\mathbf{c} = (c_1,\ldots,c_p)^{\mathrm{T}}$ and $q \in [1,\infty)$, let $\|\mathbf{c}\|_q = (\sum_{j=1}^p |c_j|^q)^{\frac{1}{q}}$ be its $\ell_q$ norm, let $\|\mathbf{c}\|_\infty$ (or $\|\mathbf{c}\|_{\max}$) $= \max_j |c_j|$ be its $\ell_\infty$ norm, let $\|\mathbf{c}\|_0 = |\{j : c_j \neq 0\}|$ be its $\ell_0$ norm, and let $\|\mathbf{c}\|_{\min} = \min_j |c_j|$ be its minimum absolute value. For a matrix $\mathbf{M}$, let $\lambda_{\min}(\mathbf{M})$ and $\lambda_{\max}(\mathbf{M})$ be its eigenvalue with smallest absolute value and largest absolute value, respectively. This is the common notation for eigenvalues of a matrix, and $\lambda_{\min}$, $\lambda_{\max}$ should not be confused with the penalization parameter used in a penalty function. For any matrix $\mathbf{G}$,

let $\|\mathbf{G}\| = \sqrt{\lambda_{\max}(\mathbf{G}^{\mathrm{T}}\mathbf{G})}$ be its spectral norm. In particular, for a vector $\mathbf{c}$, $\|\mathbf{c}\| = \|\mathbf{c}\|_2$. For $a, b \in \mathbb{R}$, let $a \wedge b = \min\{a, b\}$ and $a \vee b = \max\{a, b\}$. For a sequence $\{a_n\}$ and another nonnegative sequence $\{b_n\}$, we write $a_n = O(b_n)$ if there exists a constant $c > 0$ such that $|a_n| \leq c b_n$ for all $n \geq 1$. And we write $a_n \asymp b_n$ if $a_n = O(b_n)$ and $b_n = O(a_n)$. Also, we use $a_n = o(b_n)$, or $a_n \ll b_n$, to represent $\lim_{n\to\infty} \frac{a_n}{b_n} = 0$. We write $b_n \gg a_n$ if $a_n \ll b_n$. Let $(\Omega, \mathscr{G}, \mathrm{P})$ be a probability space on which all the random variables that appear in this paper are defined. Let $\mathbb{E}[\cdot]$ be the expectation with respect to the probability measure P. For a sequence of random variables $\{Z_n\}_{n\geq 1}$, we write $Z_n = O_p(1)$ if $\lim_{M\to\infty} \limsup_{n\to\infty} \mathrm{P}(|Z_n| > M) = 0$, and we write $Z_n = o_p(1)$ if $\lim_{n\to\infty} \mathrm{P}(|Z_n| > \varepsilon) = 0, \forall \varepsilon > 0$. For two sequences of random variables $Z_n$ and $Z_n'$, we write $Z_n = O_p(Z_n')$ if $\frac{Z_n}{Z_n'} = O_p(1)$, and we write $Z_n = o_p(Z_n')$ if $\frac{Z_n}{Z_n'} = o_p(1)$.

## 2.2 Canonical Convoluted Rank Regression

Suppose we have the observed data $\{(y_i, \mathbf{x}_i)\}_{i=1}^n$ where $y_i \in \mathbb{R}$ is the response value and $\mathbf{x}_i \in \mathbb{R}^p$ is the $p$-dimensional covariate vector for the $i$th subject. Let $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_p) \in \mathbb{R}^{n\times p}$ be the design matrix, with $\mathbf{X}_j = (x_{1j}, \ldots, x_{nj})^{\mathrm{T}}$ containing observations for the $j$th variable, $j = 1, \ldots, p$. The $i$th row of $\mathbf{X}$ can be written as $\mathbf{x}_i^{\mathrm{T}}$, where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^{\mathrm{T}}$. Let $\mathbf{y} = (y_1, \ldots, y_n)^{\mathrm{T}}$ be the $n$-dimensional response vector. For the sake of brevity, we adopt the fixed design setting in the sequel, although our methodology can also be justified under the random design setting. Assume that the data are generated from the following linear regression model $\{(y_i, \mathbf{x}_i)\}_{i=1}^n$,

$$y_i = \mathbf{x}_i^{\mathrm{T}}\beta^* + \varepsilon_i, \tag{2.1}$$

where $\{\varepsilon_i\}_{i=1}^n$ are i.i.d. random errors, $\beta^* \in \mathbb{R}^p$ is the unknown vector to be estimated. Note that we do not assume the errors in (2.1) have mean zero. Consequently, the intercept can be absorbed into the error term.

The canonical rank regression (Jaeckel, 1972; Hettmansperger and McKean, 2010) in the fixed

dimension setting proposes to estimate $\beta^*$ through

$$\min_{\beta\in\mathbb{R}^p} \frac{1}{n(n-1)} \sum\sum_{i\neq j} |(y_i - \mathbf{x}_i^{\mathrm{T}}\beta) - (y_j - \mathbf{x}_j^{\mathrm{T}}\beta)|. \tag{2.2}$$

Compared with the standard least squares method, the rank regression estimator of $\beta^*$ can have arbitrarily high relative efficiency when error distribution is heavy-tailed, while having at least 86.4% asymptotic relative efficiency under arbitrary symmetric error distribution with finite Fisher information (Hettmansperger and McKean, 2010).

For each $(i,j)$ pair, define $\{\zeta_{ij}\}_{i\neq j}$ with $\zeta_{ij} = (y_i - \mathbf{x}_i^{\mathrm{T}}\beta) - (y_j - \mathbf{x}_j^{\mathrm{T}}\beta)$. For the discussion in this part, we treat $(y_i, \mathbf{x}_i)_{i=1}^n$ as independent and identically distributed. Although $\zeta_{ij}$s are not independent, they still follow an identical distribution. For any $\beta$, let $F(t,\beta)$ denote its cumulative distribution function. After taking the expectation of the objective function in (2.2) with respect to the true distribution of $\zeta_{ij}$, the population level objective function is $\int_{-\infty}^{\infty} |t| \mathrm{d}F(t,\beta)$. Then, we can view the objective function in (2.2) as $\int_{-\infty}^{\infty} |t| \mathrm{d}\hat{F}(t,\beta)$, where $\hat{F}(t,\beta) = \frac{1}{n(n-1)} \sum\sum_{i\neq j} 1_{\{\zeta_{ij}\leq t\}}$ is the estimated cumulative distribution function for $\{\zeta_{ij}\}_{i\neq j}$. Since the estimated CDF is discontinuous, it causes the objective function in (2.2) to have the same degree of smoothness as the absolute value function. This statistical view of the objective function in rank regression suggests us to use an alternative estimator for the distribution of $\zeta_{ij}$. If we use a smooth estiamtor $\tilde{F}(t,\beta)$, then $\int_{-\infty}^{\infty} |t| \mathrm{d}\tilde{F}(t,\beta)$ can be the new objective function and become smooth.

Specifically, we consider using the kernel density estimator

$$\tilde{F}(t,\beta) = \int_{-\infty}^{t} \frac{1}{n(n-1)} \sum\sum_{i\neq j} \frac{1}{h} K\Big(\frac{v-\zeta_{ij}}{h}\Big) \mathrm{d}v$$

with some kernel function $K : \mathbb{R} \to [0,\infty)$ satisfying $K(-t) = K(t)$, $\int_{-\infty}^{\infty} K(t)\mathrm{d}t = 1$, and $h > 0$. Replacing $\hat{F}$ with $\tilde{F}$, we obtain a new objective function

$$\int_{-\infty}^{\infty} |t| \mathrm{d}\tilde{F}(t,\beta) = \frac{1}{n(n-1)} \sum\sum_{i\neq j} \int_{-\infty}^{\infty} \frac{1}{h} K\Big(\frac{\zeta_{ij}-t}{h}\Big) |t| \mathrm{d}t \triangleq \frac{1}{n(n-1)} \sum\sum_{i\neq j} L_h\big((y_i - \mathbf{x}_i^{\mathrm{T}}\beta) - (y_j - \mathbf{x}_j^{\mathrm{T}}\beta)\big),$$

where $L_h(u) = \int_{-\infty}^{\infty} |u-v| \frac{1}{h} K(\frac{v}{h}) \mathrm{d}v$. It is worth noting that $L_h(\cdot)$ is a smooth convex function. The function $L_h$ satisfies the relation $L_h = L * K_h$, where $L(u) = |u|$, $K_h(u) = \frac{1}{h} K(\frac{u}{h})$ and "$*$" stands for convolution.

Thus, in the fixed dimension setting, we propose the canonical convoluted rank regression

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n(n-1)} \sum \sum_{i \neq j} L_h\big(y_i - \mathbf{x}_i^{\mathrm{T}}\beta) - (y_j - \mathbf{x}_j^{\mathrm{T}}\beta)\big). \tag{2.3}$$

It turns out that the rank regression (2.2) and the convoluted rank regression (2.3) shares interesting connection in the population sense. In fact, let $(y, \mathbf{x})$ and $(y', \mathbf{x}')$ be i.i.d. random vectors with continuous distribution in $\mathbb{R}^{p+1}$ satisfying $y = \mathbf{x}^{\mathrm{T}}\beta^* + \varepsilon$ and $y' = \mathbf{x}'^{\mathrm{T}}\beta^* + \varepsilon'$, where $\varepsilon$ is independent from $\mathbf{x}$, and $\varepsilon'$ is independent from $\mathbf{x}'$. For rank regression, it is well known that the minimizer of the population version of its loss function, i.e. $\arg\min_{\beta \in \mathbb{R}^p} \mathbb{E}[|\varepsilon - \varepsilon' - (\mathbf{x} - \mathbf{x}')^{\mathrm{T}}(\beta - \beta^*)|]$, is exactly the same as $\beta^*$, the true regression coefficients. This simple fact justifies that rank regression is valid in the population sense, which is necessary in order for its sample version to aim at estimating the true regression coefficients. One may naturally ask if the population version of (2.3) also has such property. Let $\beta_h^* = \arg\min_{\beta \in \mathbb{R}^p} \mathbb{E}[L_h(y - y' - (\mathbf{x} - \mathbf{x}')^{\mathrm{T}}\beta)]$. We have the following theorem, stating that smoothing via convolution does not incur any bias at all in the population sense.

**Theorem 1.** *For any $h > 0$ and any kernel $K(\cdot)$ satisfying $\int_{-\infty}^{\infty} K(u)\mathrm{d}u = 1$ and $K(u) = K(-u), \forall u \in \mathbb{R}$, we have $\beta_h^* = \beta^*$.*

**Remark 1.** *Note that the smoothing quantile regression (Fernades, Guerre and Horta , 2021) does not have the good property of zero smoothing bias as shown in Theorem 1. In fact, the proof of Theorem 1 crucially relies on the fact that the distributions of $\varepsilon - \varepsilon'$ and $\mathbf{x} - \mathbf{x}'$ are symmetric about zero, which can only be taken advantage of given the special form of rank regression.*

## 2.3 Sparse Convoluted Rank Regression

When $p$ is large, we consider designing the estimator under a sparsity assumption that $\beta^*$ in the data generating model has many zero components. Let $\mathbb{A} = \{j : \beta_j^* \neq 0\}$ be the support set of $\beta^*$,

i.e., the set of indices of the important covariates. Let $s = |\mathbb{A}|$. Throughout this paper, we allow $p = p_n$ and $s = s_n$ to diverge with $n$, and we assume $s_n \geq 1$ and $p_n$ goes to infinity as $n$ goes to infinity. For convenience, we still use $p$ and $s$ to represent these quantities since no confusion is caused. In ultra-high dimensions, the dimension $p$ is allowed to increase exponentially with the sample size $n$, and we assume that $s$ is relatively of smaller order compared to $n$. Otherwise, no consistent estimator is possible.

To estimate $\beta^*$, we propose the sparse Convoluted Rank Regression (CRR) by solving

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} L_h(y_i - y_j - (\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}} \beta) + \sum_{j=1}^{p} p_\lambda(|\beta_j|).$$

Here $p_\lambda(\cdot)$ is some sparsity-inducing penalty function with a tuning parameter $\lambda > 0$, $L_h(u) = \int_{-\infty}^{\infty} |u - v| \frac{1}{h} K(\frac{v}{h}) \mathrm{d}v$, where $K : \mathbb{R} \to [0, \infty)$ is a kernel function satisfying $\int_{-\infty}^{\infty} K(u) \mathrm{d}u = 1$ and $K(u) = K(-u), \forall u$, and $h > 0$ is a constant.

**Remark 2.** *There can be a lot of choices for the kernel function $K(\cdot)$ satisfying the conditions in our theory presented in section 3. In the numerical studies of this work, we focus on the Epanechnikov kernel $K(u) = \frac{3}{4}(1-u^2) I(-1 \leq u \leq 1)$ for illustration purposes, where $I(\cdot)$ is the indicator function.*

*Intuitively, $h$ should be small such that the sparse convoluted rank regression is very close to the sparse rank regression. As suggested by the theoretical results in Section 3, $h = O(1)$ is sufficient for our method to achieve optimal rate and oracle property. According to density estimator, the optimal rate for $h$ is $O(n^{-1/5})$. So, we use $h = 2.5n^{-1/5}$ as the default value in our implementation.*

# 3 Theoretical Justifications for Sparse CRR

In this section we study the theoretical properties of the $\ell_1$-penalized convoluted rank regression (CRR) and the folded concave penalized CRR under the same key regularity conditions for the rank regression in Wang et al. (2020).

### 3.1 $\ell_1$-penalized CRR

For a tuning parameter $\lambda_0 > 0$, we define the $\ell_1$-penalized CRR estimator as

$$\tilde{\beta}^{\lambda_0} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} L_h(y_i - y_j - (\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}} \beta) + \lambda_0 \sum_{j=1}^{p} |\beta_j|.$$

We now state the assumptions needed throughout this paper. We make the following assumptions for the kernel function $K(\cdot)$.

**Assumption 1.** *$K : \mathbb{R} \to [0, \infty)$ is a function satisfying the following properties: (i), $K(-t) = K(t)$, $\forall t \in \mathbb{R}$; (ii), $\exists \delta_0 > 0$ s.t. $\kappa_l := \inf_{t \in [-\delta_0, \delta_0]} K(t) > 0$; (iii), $\int_{-\infty}^{\infty} K(t)\,\mathrm{d}t = 1$; (iv), $\kappa_1 := \int_{-\infty}^{\infty} |t| K(t)\,\mathrm{d}t < \infty$.*

For the error distribution, we impose the following assumption.

**Assumption 2.** *The errors $\{\varepsilon_i\}_{i=1}^n$ are independent and identically distributed with density function $f(\cdot)$ with respect to the Lebesgue measure on $\mathbb{R}$. Besides, let $\varsigma_{ij} = \varepsilon_i - \varepsilon_j, 1 \leq i \neq j \leq n$. Let $g(\cdot)$ denote the probability density function of $\varsigma_{ij}$, we assume $\sup_{t \in \mathbb{R}} g(t) = \mu_0 < \infty$. Meanwhile, there exist positive constants $\delta_1, \mu_1$ such that $g(t) \geq \mu_1, \forall t \in [-\delta_1, \delta_1]$.*

For any index set $\mathbf{A} \subset \{1, \ldots, p\}$, let $\mathscr{S}_{\mathbf{A}} := \{\mathbf{u} \in \mathbb{R}^p : \|\mathbf{u}_{\mathbf{A}^c}\|_1 \leq 3\|\mathbf{u}_{\mathbf{A}}\|_1 \neq 0\}$. We also impose the following conditions on the design matrix.

**Assumption 3.** *There exists a constant $M > 0$ such that $\max_{1 \leq i \leq n, 1 \leq j \leq p} |x_{ij}| \leq M$. Also, the covariates are centered, i.e. $\sum_{i=1}^n x_{ij} = 0, \forall j = 1, \ldots, p$.*

**Assumption 4.** *There exists a constant $\rho > 0$ such that $\min_{\mathbf{u} \in \mathscr{S}_{\mathbb{A}}} \frac{\|\mathbf{X}\mathbf{u}\|_2^2}{n\|\mathbf{u}\|_2^2} \geq \rho$. In particular, this implies $\lambda_{\min}(\frac{\mathbf{X}_{\mathbb{A}}^T \mathbf{X}_{\mathbb{A}}}{n}) \geq \rho$.*

Assumption 3 is common in the fixed design case. It can be relaxed with $M$ increasing with $n$ at a suitable rate, without much difficulty. We keep it here for the sake of brevity. We can center the design matrix when estimating the $\beta^*$ vector because centering only affects the intercept part which is a nuisance parameter in our method as well as in rank regression. Assumption 4, which is

known as the restricted eigenvalue condition (RE), is needed to establish $\ell_2$-type error bound for $\ell_1$-penalized estimator. It is a commonly used assumption in the literature (Bühlmann and Van De Geer, 2011; Fan et al., 2020).

**Theorem 2.** *Assume assumptions 1-4 hold, and* $s = o(\sqrt{\frac{n}{\log p}})$. *Let* $0 < \lambda_0 = c_0\sqrt{\frac{\log p}{n}}$ *with* $8\sqrt{2}M < c_0 = O(1)$, *and let* $0 < h = O(1)$. *Then the* $\ell_1$*-penalized CRR estimator* $\tilde{\beta}^{\lambda_0}$ *satisfies*

$$\|\tilde{\beta}^{\lambda_0} - \beta^*\|_2 \leq \frac{96M + 4c_0}{\mu_2 \rho}\sqrt{\frac{s\log p}{n}}$$

*with probability at least* $1 - 2p^{-\left(\frac{c_0^2}{128M^2} - 1\right)} - 2p^{-2}$, *where* $\mu_2 := \kappa_l \mu_1 (2\delta_0 \wedge \frac{\delta_1}{h})$.

Notice that the probabilistic bound in Theorem 2 does not depend on unknown quantities, since with the design matrix at hand, $M$ and $p$ are both available. This means that in principle, the $\lambda_0$ in $\ell_1$-penalized CRR estimator is tuning-free. As long as $c_0$ is a constant which is larger than $8\sqrt{2}M$, the probabilistic lower bound in Theorem 2 converges to 1, and as a result we have $\|\tilde{\beta}^{\lambda_0} - \beta^*\|_2 = O_p(\sqrt{\frac{s\log p}{n}})$, which means the $\ell_1$-penalized CRR estimator achieves the near-optimal rate.

## 3.2 Folded concave penalized CRR

It has been well established in the literature that folded concave penalized estimators can enjoy strong oracle property. We prove the same is true for convoluted rank regression. Define

$$\hat{\beta}^{\text{ora}} := \underset{\beta \in \mathbb{R}^p : \beta_{\mathbb{A}^c} = \mathbf{0}}{\arg\min} \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} L_h(y_i - y_j - (\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}\beta) \tag{3.1}$$

as the CRR oracle estimator. It can be directly verified that $\hat{\beta}^{\text{ora}}$ exists due to the convexity of $L_h(\cdot)$, assumption 3 and assumption 4. We establish the following property for the oracle estimator.

**Theorem 3.** *Assume assumptions 1-4 hold,* $s = o(\sqrt{n})$ *and* $h = O(1)$. *Then we have* $\|\hat{\beta}^{ora} - \beta^*\|_2 = O_p\left(\sqrt{\frac{s}{n}}\right)$.

**Remark 3.** *In the case where $\hat{\beta}^{ora}$ is not unique, one may take any solution to (3.1), and our theory about CRR oracle estimator still holds.*

We now propose the concave penalized convoluted rank regression. It solves the following problem:

$$\min_{\beta\in\mathbb{R}^p} \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j\neq i} L_h(y_i - y_j - (\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}\beta) + \sum_{j=1}^{p} p_\lambda(|\beta_j|). \tag{3.2}$$

For the choice of $p_\lambda(\cdot)$, we adopt the folded concave penalty (Fan et al., 2014b), i.e. $p_\lambda(\cdot)$ is a function defined on $(-\infty,\infty)$ satisfying: (i), $p_\lambda(-z) = p_\lambda(z)$; (ii), $p_\lambda(z)$ is increasing and concave in $z\in[0,\infty)$, and $p_\lambda(0) = 0$; (iii), $p_\lambda(z)$ is differentiable in $z\in(0,\infty)$, and $p'_\lambda(0) := p'_\lambda(0+) \geq a_1\lambda$; (iv), $p'_\lambda(z) \geq a_1\lambda$ for $z\in(0,a_2\lambda]$; (v) $p'_\lambda(z) = 0$ for $z\in[a\lambda,\infty)$ with some pre-specified constant $a > a_2$. Here $a_1$ and $a_2$ are two fixed positive constants. Special cases of folded concave penalty are SCAD (Fan and Li, 2001) and MCP (Zhang, 2010). The SCAD penalty has the form

$$\begin{aligned} p_\lambda(|t|) =& \lambda|t| I(0\leq|t|<\lambda) + \frac{a\lambda|t| - \left(t^2+\lambda^2\right)/2}{a-1} I(\lambda\leq|t|\leq a\lambda) \\ &+ \frac{(a+1)\lambda^2}{2} I(|t|>a\lambda), \text{ for some } a>2, \end{aligned}$$

which corresponds to $a_1 = a_2 = 1$. The MCP penalty function is defined as

$$p_\lambda(|t|) = \lambda\left(|t| - \frac{t^2}{2a\lambda}\right) I(0\leq|t|<a\lambda) + \frac{a\lambda^2}{2} I(|t|\geq a\lambda), \text{ for some } a>1,$$

which corresponds to $a_1 = 1-\frac{1}{a}, a_2 = 1$.

We adopt the local linear approximation (LLA) (Zou and Li, 2008) algorithm to solve (3.2). The LLA algorithm iteratively solves

$$\hat{\beta}^{(k+1)} = \arg\min_{\beta\in\mathbb{R}^p} \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j\neq i} L_h(y_i - y_j - (\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}\beta) + \sum_{j=1}^{p} p'_\lambda\left(|\beta_j^{(k)}|\right)|\beta_j|, \ \ k = 0,1,2,\ldots, \tag{3.3}$$

where $\hat{\beta}^{(0)}$ is some initial estimator. We use $\hat{\beta}^{\lambda}$ to denote the folded concave penalized CRR estimator computed by the LLA algorithm, with tuning parameter $\lambda$. Below we establish theory for the folded concave penalized CRR estimator.

**Theorem 4.** *Let the conditions of Theorem 2 and Theorem 3 hold. Assume that $h = O(1)$. Let $a_0 = \min\{1, a_2\}$ where $a_2$ is the constant associated with the folded-concave penalty function. Choose the tuning parameter so that $\min_{j\in\mathbb{A}} |\beta_j^*| > (a+1)\lambda$.*

*(i) Suppose $s = o(\log p)$. Let the tuning parameter be chosen as $\lambda = c_1\sqrt{\frac{s\log p}{n}}$ such that $\frac{96M+4c_0}{a_0\mu_2\rho} \vee \frac{32\sqrt{2}M}{a_1} < c_1 = O(1)$, where $c_0$ is defined in Theorem 2. Then the LLA algorithm in (3.3) initialized by $\hat{\beta}^{(0)} = \tilde{\beta}^{\lambda_0}$, with $\lambda_0$ being defined in Theorem 2, converges to $\hat{\beta}^{ora}$ in two iterations with probability converging to 1 as $n \to \infty$.*

*(ii) Consider the SCAD or MCP as the penalty function. Suppose $s = o(\sqrt{\log p})$. Let the tuning parameter be chosen as $\lambda = c_1\sqrt{\frac{\log p}{n}}$ such that $\frac{(96M+4c_1)\sqrt{s}}{a_0\mu_2\rho} \vee \frac{32\sqrt{2}M}{a_1} \vee 8\sqrt{2}M < c_1 = O(1)$. Then the LLA algorithm in (3.3) initialized by $\hat{\beta}^{(0)} = \mathbf{0}$ converges to $\hat{\beta}^{ora}$ in three iterations with probability converging to 1 as $n \to \infty$.*

Theorem 4 shows that the folded concave penalized CRR estimator equals to the oracle estimator with overwhelming probability, which is typically referred to as strong oracle property. It means that our estimator can perform as well as if the true set of important covariates was given.

**Remark 4.** *Throughout our theory, we only need $h = O(1)$, which is a weaker condition on the smoothing bandwidth h than that is required for smoothing quantile regression (Fernades, Guerre and Horta , 2021) in which h should be $O((n/\log n)^{-1/3})$ and $h = o(1)$. Again, this is a consequence of the delicate form of rank regression which makes important first order terms vanish, as can be seen from our theoretical proofs.*

## 3.3 Consistent tuning parameter selection

For the folded concave penalization, Theorem 4 guarantees that there exists a good tuning parameter in principle. Since the tuning parameter depends on unknown quantities, a data-driven approach is

needed to specify the tuning parameter in practice. Motivated by Wang et al. (2013), we propose a modified high dimensional Bayesian information criteria, defined as

$$\mathrm{HBIC}(\lambda) = \log\left(\frac{1}{n(n-1)}\sum_{i=1}^{n}\sum_{j\neq i} L_h\big(y_i - y_j - (\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}\hat{\beta}^{\lambda}\big)\right) + |M_\lambda|\frac{C_n \log p}{n},$$

where $M_\lambda := \{j : \hat{\beta}_j^{\lambda} \neq 0\}$, and the choice of $C_n$ is discussed in Theorem 5. The corresponding tuning parameter for the folded concave penalty is chosen by minimizing the proposed HBIC.

**Theorem 5.** *Let* $\hat{\lambda} = \arg\min_{\lambda\in\Lambda} HBIC(\lambda)$, *where* $\Lambda = \{\lambda > 0 : |M_\lambda| \leq K_n\}$, *and* $K_n > s$ *is allowed to diverge to infinity. Under the conditions of Theorem 4, assume that* $\mathbb{E}[|\varsigma_{ij}|] < \infty$, $\phi := \min_{|\mathbb{S}|\leq 2K_n} \lambda_{\min}(\frac{\mathbf{X}_{\mathbb{S}}^T\mathbf{X}_{\mathbb{S}}}{n}) > 0$. *If* $\sqrt{\frac{C_n\sqrt{s}\log p}{n}} \vee \frac{C_n \log p\sqrt{sK_n}}{n} = o(\|\beta_{\mathbb{A}}^*\|_{\min})$, $\frac{C_n s\log p}{n} = o(1)$ *and* $K_n = o\big(\sqrt{n} \wedge \sqrt{C_n \log p}\big)$, *then we have* $\mathrm{P}(M_{\hat{\lambda}} = \mathbb{A}) \to 1$ *as* $n \to \infty$.

**Remark 5.** *The condition* $\min_{|\mathbb{S}|\leq 2K_n} \lambda_{\min}(\frac{\mathbf{X}_{\mathbb{S}}^T\mathbf{X}_{\mathbb{S}}}{n}) > 0$ *in Theorem 5 is known as the sparse Riesz condition and is widely used in literature on high dimensional statistics (Zhang and Huang, 2008). In our implementation, the sequence* $C_n$ *is chosen such that* $C_n \asymp \log\log n$. *This is the same choice as in the HBIC for the penalized rank regression (Wang et al., 2020).*

Theorem 5 shows that with proposed HBIC, our method can exactly identify the important variables with probability approaching to 1. Unlike cross validation, the HBIC criterion does not require sample splitting or repeated evaluation of the test error on each sub-dataset. As a result, our method requires no extra computation for tuning.

# 4 Computation

We have shown that we need to solve the folded concave penalized CRR by running the LLA iteration 2-3 times. In each LLA iteration, we need to solve a weighted $\ell_1$-penalized CRR problem. In this section, we develop an efficient algorithm for computing the solution path of a weighted $\ell_1$-penalized CRR.

Consider the following "weighted" $\ell_1$-penalized CRR problem:

$$\arg\min_{\beta\in\mathbb{R}^p} \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j\neq i} L_h(y_i - y_j - (\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}\beta) + \sum_{k=1}^{p} w_k|\beta_k|, \tag{4.1}$$

where each $w_k \geq 0$. In contrast to the sparse rank regression, the density convolution gives a smooth loss function $L_h$. To see this, recall $L_h(u) = \int_{-\infty}^{\infty} |u-v| \frac{1}{h}K(\frac{v}{h})\mathrm{d}v, u \in \mathbb{R}$, and a direct calculation gives $L_h'(u) = 2\int_{-\infty}^{u} \frac{1}{h}K(\frac{v}{h})\mathrm{d}v - 1$ and $L_h''(u) = \frac{2}{h}K(\frac{u}{h})$, $\forall u \in \mathbb{R}$. We thus establish some basic properties of $L_h(\cdot)$.

**Lemma 1.** *Under assumption 1, for any $t_1, t_2, t \in \mathbb{R}$, we have $L_h'(-t) = -L_h'(t)$ and $|L_h(t_1) - L_h(t_2)| \leq |t_1 - t_2|$. If we use a kernel such that $\sup_{t\in\mathbb{R}} K(t) = \kappa_u < \infty$, then $|L_h'(t_1) - L_h'(t_2)| \leq \frac{2}{h}\kappa_u|t_1 - t_2|$.*

Therefore, the objective function in problem (4.1) is the summation of a convex and smooth loss function and a convex and separable penalty term. It turns out that a coordinate descent-type algorithm usually works well in this situation (Tseng, 2001).

In a coordinate-wise manner, suppose we have updated the coordinates $\beta_1, \beta_2, \ldots, \beta_{k-1}$ and we now need to update $\beta_k$. Denote by $\tilde{\beta}$ the current solution and let $v_{ij} = y_i - y_j - (\mathbf{x}_i - \mathbf{x}_j)^{\mathrm{T}}\tilde{\beta}$. The standard coordinate descent algorithm cyclically updates $\beta_k$ by minimizing

$$F(\beta_k|\tilde{\beta}) = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j\neq i} L_h(v_{ij} - (x_{ik} - x_{jk})(\beta_k - \tilde{\beta}_k)) + w_k|\beta_k|.$$

We observe that minimizing the above function does not have a close-form solution, so we consider a generalized coordinate descent algorithm (Yang and Zou, 2013). The idea is to perform a majorization-minimization update rather than directly minimize $F(\beta_k|\tilde{\beta})$. Specifically, we need to find a quadratic function $G$ such that $F(\beta_k|\tilde{\beta}) = G(\beta_k|\tilde{\beta})$ and $F(\gamma|\tilde{\beta}) < G(\gamma|\tilde{\beta})$ for any $\gamma \neq \beta_k$.

From the last inequality of Lemma 1, we can obtain a quadratic majorization condition for CRR:

$$L_h(t_1) < L_h(t_2) + L_h'(t_2)(t_1 - t_2) + \frac{\kappa_u}{h}(t_1 - t_2)^2,$$

for $t_1 \neq t_2$. For each pair of $i \neq j$, by letting $t_1 = v_{ij} - (x_{ik} - x_{jk})(\beta_k - \tilde{\beta}_k)$ and $t_2 = v_{ij}$, we have the quadratic majorization function for $F(\beta_k|\tilde{\beta})$:

$$G(\beta_k|\tilde{\beta}) = \frac{\sum_{i=1}^n \sum_{j\neq i} L_h(v_{ij})}{n(n-1)} + a_k(\beta_k - \tilde{\beta}_k) + \frac{c_k \kappa_u}{h}(\beta_k - \tilde{\beta}_k)^2 + w_k|\beta_k|,$$

where $a_k = -\frac{1}{n(n-1)} \sum_{i=1}^n \sum_{j\neq i} L_h'(v_{ij})(x_{ik} - x_{jk})$ and $c_k = \frac{1}{n(n-1)} \sum_{i=1}^n \sum_{j\neq i} (x_{ik} - x_{jk})^2$. Hence, we update $\beta_k$ using the minimizer of $G\left(\beta_k|\tilde{\beta}\right)$:

$$\hat{\beta}_k = \operatorname{sgn}\left(\tilde{\beta}_k - \frac{ha_k}{2c_k\kappa_u}\right)\left(\left|\tilde{\beta}_k - \frac{ha_k}{2c_k\kappa_u}\right| - \frac{hw_k}{c_k\kappa_u}\right)_+.$$

Therefore, we solve problem (4.1) by cyclically performing the above update for each $k = 1, 2, \ldots, p$.

In our implementation, we directly compute the solution path problem (4.1) at a sequence of tuning parameters, $\lambda^{[1]}, \lambda^{[2]}, \ldots, \lambda^{[L]}$ instead of calling the algorithm $L$ times for each individual parameter. We let

$$\lambda^{[1]} = \left\| \frac{1}{n(n-1)} \sum_{i\neq j} L_h'\left(y_i - y_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right) \right\|_\infty,$$

which is the smallest penalization parameter to make all $\hat{\beta}_k = 0$. We then choose other $\lambda$-values such that they are uniformly distributed on a logarithm scale. In addition, we also employ the warm start and active set strategies to further accelerate the GCD algorithm; see details of these two strategies in Friedman, Hastie and Tibshirani (2010).

# 5 Numerical Examples

## 5.1 Simulation Study

In this section, we demonstrate the performance of the sparse convoluted rank regression in terms of estimation accuracy and variable selection using simulations. Because the most attractive property of rank regression is its efficiency argument, we focus on estimators with strong oracle properties such as the SCAD-penalized convoluted rank regression (denoted by CRR-SCAD) and SCAD-

penalized rank regression (denoted by RR-SCAD). We use zero vector as the initial value in the LLA algorithm for computing CRR-SCAD, so that we do not have to compute the $\ell_1$-penalized CRR in order to compute CRR-SCAD. We used the code from Wang et al. (2020) to compute RR-SCAD. In our numerical studies, we used Epanechnikov kernel as the density convolution kernel, $K(u) = \frac{3}{4}(1-u^2)I(-1 \le u \le 1)$, where $I(\cdot)$ is the indicator function, and the loss function is

$$L_h(u) = \begin{cases} u, & u \ge h, \\ \frac{3u^2}{4h} - \frac{u^4}{8h^3} + \frac{3h}{8}, & -h < u < h, \\ -u, & u \le -h. \end{cases}$$

Both the RR-SCAD and CRR-SCAD are tuned based on HBIC. For comparison, we also include the SCAD-penalized least squares (denoted by LS-SCAD) and tune it by its corresponding HBIC (Wang et al., 2013).

We consider a model $y = \mathbf{x}^{\mathrm{T}}\beta^\star + \varepsilon$, where $\beta^\star = (\sqrt{3}, \sqrt{3}, \sqrt{3}, 0, 0, \ldots, 0) \in \mathbb{R}^p$, $x$ is independently generated from $\mathrm{N}(0, \Sigma)$, and $\varepsilon$ is independently generated from some certain distributions. We fix the sample size $n = 100$ and use the dimensions $p = 400$ and 3000. We consider four situations for the correlation structure of $x$: CS (0.2), CS (0.5), CS (0.8), and AR (0.5), where each CS $(\rho)$ represents the compound symmetry correlation, i.e., $\Sigma_{i,j} = \rho$ if $i \ne j$ or 1 otherwise, and AR $(\rho)$ indicates the autoregressive correlation, that is, $\Sigma = (\rho^{|i-j|})_{p\times p}$.

We compare these methods based on five criteria: $\ell_1$ error $(\mathbb{E}\|\hat\beta - \beta^\star\|_1)$, $\ell_2$ error $(\mathbb{E}\|\hat\beta - \beta^\star\|_2)$, model error, $(\mathbb{E}(\hat\beta - \beta^\star)^{\mathrm{T}}\Sigma(\hat\beta - \beta^\star))$, the number of false positive variables, and the number of false negative variables. All the quantities are averaged over 200 independent runs and the standard errors are provided.

Table 1 exhibits the simulation results when $\varepsilon$ is from $\mathrm{N}(0,1)$. In each situation, we use boldface to indicate the best performance that is evaluated based on each of the five criteria. When $p = 400$, we observe that the estimation accuracy of LS-SCAD and CRR-SCAD is similar and better than that of RR-SCAD; when $p = 3000$, the estimation accuracy of CRR-SCAD is the best. In addition, both LS-SCAD and CRR-SCAD have perfect performance in variable selection and RR-SCAD is

Table 1: Comparison of least-square regression with SCAD (LS-SCAD), rank regression with SCAD (RR-SCAD) and convoluted rank regression with SCAD (CRR-SCAD). The comparison criteria are $\ell_1$ error, $\ell_2$ error, model error (ME), number of false positive variables (FP) and number of false negative variables (FN). In each example, the best method evaluated based on each criterion is in boldface. All the quantities are averaged over 200 independent runs and standard errors are given in parentheses. In all the examples shown in this table, the error term in the data generating model is drawn from the standard normal distribution.

| | | $p = 400$ | | | | | | $p = 3000$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\Sigma$ | criterion | LS-SCAD | | RR-SCAD | | CRR-SCAD | | LS-SCAD | | RR-SCAD | | CRR-SCAD | |
| CS (0.2) | $\ell_1$ | **0.31** | (0.01) | 0.37 | (0.01) | 0.32 | (0.01) | 0.36 | (0.01) | 0.53 | (0.01) | **0.33** | (0.01) |
| | $\ell_2$ | **0.18** | (0.00) | 0.21 | (0.01) | **0.18** | (0.00) | 0.22 | (0.01) | 0.29 | (0.01) | **0.19** | (0.00) |
| | ME | **0.03** | (0.00) | 0.04 | (0.00) | **0.03** | (0.00) | 0.05 | (0.00) | 0.09 | (0.00) | **0.04** | (0.00) |
| | FP | **0** | (0) | 0 | (0) | **0** | (0) | **0** | (0) | 1 | (0) | **0** | (0) |
| | FN | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) |
| CS (0.5) | $\ell_1$ | **0.36** | (0.01) | 0.38 | (0.01) | **0.36** | (0.01) | 0.39 | (0.01) | 0.46 | (0.01) | **0.37** | (0.01) |
| | $\ell_2$ | **0.21** | (0.01) | 0.23 | (0.01) | **0.21** | (0.01) | 0.23 | (0.01) | 0.27 | (0.01) | **0.22** | (0.01) |
| | ME | **0.03** | (0.00) | 0.04 | (0.00) | 0.04 | (0.00) | 0.04 | (0.00) | 0.05 | (0.00) | **0.03** | (0.00) |
| | FP | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) |
| | FN | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) |
| AR (0.5) | $\ell_1$ | **0.35** | (0.01) | 0.45 | (0.01) | **0.35** | (0.01) | 0.39 | (0.01) | 0.62 | (0.02) | **0.37** | (0.01) |
| | $\ell_2$ | **0.20** | (0.01) | 0.23 | (0.01) | 0.21 | (0.01) | 0.23 | (0.01) | 0.34 | (0.01) | **0.22** | (0.01) |
| | ME | **0.03** | (0.00) | 0.05 | (0.00) | **0.03** | (0.00) | **0.04** | (0.00) | 0.09 | (0.00) | **0.04** | (0.00) |
| | FP | **0** | (0) | 1 | (0) | **0** | (0) | **0** | (0) | 0 | (0) | **0** | (0) |
| | FN | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) |

the only method that makes mistakes. By comparing the performance of CRR-SCAD when $p = 400$ and 3000, we see the performance of CRR-SCAD is less prone to the increase in $p$.

Table 2 summarizes the simulation results when $\varepsilon$ is from a mixture normal distribution: $\varepsilon \sim 0.95\mathrm{N}(0,1) + 0.05\mathrm{N}(0,100)$. From Table 2, we find that LS-SCAD fails to work well in this situation. For both $p = 400$ and $p = 3000$, RR-SCAD and CRR-SCAD perform similarly. Table 3 shows the results when $\varepsilon/\sqrt{2}$ follows the $t$-distribution with four degrees of freedom. In all situations, CRR-SCAD performs better than the other two methods, in terms of both estimation accuracy and variable selection. When $p$ is increased from 400 to 3000, CRR-SCAD suffers minimal impact, while RR-SCAD shows a significant loss in estimation accuracy.

Table 2: Comparison of least-square regression with SCAD (LS-SCAD), rank regression with SCAD (RR-SCAD) and convoluted rank regression with SCAD (CRR-SCAD). The comparison criteria are $\ell_1$ error, $\ell_2$ error, model error (ME), number of false positive variables (FP) and number of false negative variables (FN). In each example, the best method evaluated based on each criterion is in boldface. All the quantities are averaged over 200 independent runs and standard errors are given in parentheses. In all the examples shown in this table, the error term in the data generating model follows a mixture normal distribution: $\varepsilon \sim 0.95\mathrm{N}(0,1) + 0.05\mathrm{N}(0,100)$.

| | | $p = 400$ | | | | | | $p = 3000$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\Sigma$ | criterion | LS-SCAD | | RR-SCAD | | CRR-SCAD | | LS-SCAD | | RR-SCAD | | CRR-SCAD | |
| CS (0.2) | $\ell_1$ | 1.51 | (0.08) | **0.18** | (0.01) | 0.22 | (0.01) | 3.18 | (0.14) | **0.19** | (0.01) | 0.21 | (0.01) |
| | $\ell_2$ | 0.79 | (0.04) | **0.16** | (0.01) | 0.17 | (0.01) | 1.68 | (0.07) | **0.16** | (0.01) | **0.16** | (0.01) |
| | ME | 0.67 | (0.06) | **0.03** | (0.00) | **0.03** | (0.00) | 5.18 | (0.33) | **0.03** | (0.00) | **0.03** | (0.00) |
| | FP | 1 | (0) | **0** | (0) | **0** | (0) | 1 | (0) | **0** | (0) | **0** | (0) |
| | FN | **0** | (0) | **0** | (0) | **0** | (0) | 1 | (0) | **0** | (0) | **0** | (0) |
| CS (0.5) | $\ell_1$ | 1.86 | (0.11) | **0.21** | (0.01) | 0.24 | (0.01) | 3.72 | (0.15) | 0.25 | (0.01) | **0.21** | (0.01) |
| | $\ell_2$ | 0.90 | (0.05) | **0.16** | (0.01) | 0.18 | (0.01) | 1.84 | (0.08) | 0.18 | (0.01) | **0.17** | (0.01) |
| | ME | 0.47 | (0.04) | **0.03** | (0.00) | **0.03** | (0.00) | 7.82 | (0.51) | **0.03** | (0.00) | **0.03** | (0.00) |
| | FP | 2 | (0) | **0** | (0) | **0** | (0) | 2 | (0) | **0** | (0) | **0** | (0) |
| | FN | **0** | (0) | **0** | (0) | **0** | (0) | 1 | (0) | **0** | (0) | **0** | (0) |
| AR (0.5) | $\ell_1$ | 1.22 | (0.05) | **0.19** | (0.01) | 0.26 | (0.01) | 1.72 | (0.07) | **0.20** | (0.01) | 0.22 | (0.01) |
| | $\ell_2$ | 0.73 | (0.03) | **0.16** | (0.01) | 0.18 | (0.01) | 1.03 | (0.04) | **0.16** | (0.01) | **0.16** | (0.01) |
| | ME | 0.50 | (0.04) | **0.03** | (0.00) | **0.03** | (0.00) | 1.44 | (0.10) | **0.03** | (0.00) | **0.03** | (0.00) |
| | FP | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) |
| | FN | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) |

## 5.2 A real data application

We illustrate our proposed method on a microarray gene expression data reported in (Scheetz et al., 2006). The dataset contains RNA expression levels of more than 31,000 gene probes from 120 twelve-week-old laboratory rats. Following Scheetz et al. (2006), we include 18,976 genes that have sufficient variation and are considered expressed in mammalian eyes. Among these genes, TRIM32 has genetic influences on a rare genetic disorder, the Bardet-Biedl syndrome (Chiang et al., 2006). Thus TRIM32 is chosen as the target variable and our goal is to identify the genes that are associated with TRIM32.

In our experiments, we randomly split the original data into a training set and a test set in the ratio 1:1. On the training set, we apply the fused Kolmogorov filter (Mai and Zou, 2015) to obtain a reduced set of 300 probes and retained the same 300 probes on the test set. We then

Table 3: Comparison of least-square regression with SCAD (LS-SCAD), rank regression with SCAD (RR-SCAD), and convoluted rank regression with SCAD (CRR-SCAD). The comparison criteria are $\ell_1$ error, $\ell_2$ error, model error (ME), number of false positive variables (FP) and number of false negative variables (FN). In each example, the best method evaluated based on each criterion is in boldface. All the quantities are averaged over 200 independent runs and standard errors are given in parentheses. In all the examples shown in this table, the error term in the data generating model $\varepsilon \sim \sqrt{2}t(4)$.

| | | $p = 400$ | | | | | | $p = 3000$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\Sigma$ | criterion | LS-SCAD | | RR-SCAD | | CRR-SCAD | | LS-SCAD | | RR-SCAD | | CRR-SCAD | |
| CS (0.2) | $\ell_1$ | 1.13 | (0.03) | 0.79 | (0.02) | **0.58** | (0.02) | 3.33 | (0.10) | 1.69 | (0.06) | **0.63** | (0.02) |
| | $\ell_2$ | 0.63 | (0.02) | 0.43 | (0.01) | **0.34** | (0.01) | 1.74 | (0.05) | 0.82 | (0.02) | **0.37** | (0.01) |
| | ME | 0.42 | (0.02) | 0.19 | (0.01) | **0.12** | (0.01) | 4.70 | (0.26) | 0.64 | (0.03) | **0.14** | (0.01) |
| | FP | 1 | (0) | 0 | (0) | **0** | (0) | 2 | (0) | 5 | (0) | **0** | (0) |
| | FN | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) |
| CS (0.5) | $\ell_1$ | 1.33 | (0.05) | 0.72 | (0.02) | **0.70** | (0.02) | 4.01 | (0.12) | 1.10 | (0.03) | **0.72** | (0.02) |
| | $\ell_2$ | 0.69 | (0.02) | 0.41 | (0.01) | **0.40** | (0.01) | 1.95 | (0.06) | 0.63 | (0.02) | **0.41** | (0.01) |
| | ME | 0.34 | (0.02) | 0.14 | (0.01) | **0.13** | (0.01) | 7.53 | (0.47) | 0.26 | (0.01) | **0.14** | (0.01) |
| | FP | 2 | (0) | **0** | (0) | **0** | (0) | 4 | (0) | **0** | (0) | **0** | (0) |
| | FN | **0** | (0) | **0** | (0) | **0** | (0) | 1 | (0) | **0** | (0) | **0** | (0) |
| AR (0.5) | $\ell_1$ | 1.12 | (0.03) | 0.89 | (0.03) | **0.62** | (0.02) | 1.56 | (0.04) | 1.50 | (0.04) | **0.71** | (0.02) |
| | $\ell_2$ | 0.66 | (0.02) | 0.46 | (0.01) | **0.37** | (0.01) | 0.93 | (0.02) | 0.86 | (0.03) | **0.41** | (0.01) |
| | ME | 0.37 | (0.02) | 0.18 | (0.01) | **0.12** | (0.01) | 1.00 | (0.05) | 0.64 | (0.03) | **0.15** | (0.01) |
| | FP | **0** | (0) | 1 | (0) | **0** | (0) | **0** | (0) | 1 | (0) | **0** | (0) |
| | FN | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) | **0** | (0) |

fit SCAD-penalized least squares (SCAD), rank regression (RR-SCAD) and our convoluted rank regression (CRR-SCAD) on the training set and compute the prediction error on the test set. To illustrate the performance in higher dimensions, we repeat the same above procedure except that the reduced set from the fused Kolmogorov filter has 5,000 probes.

Based on 200 random partitions, we report the prediction error and run time in Table 4. We observe CRR-SCAD has the lowest prediction error whereas LS-SCAD has the highest error. When $p$ grows from 300 to 5000, both RR-SCAD and CRR-SCAD become more accurate; this may be because some important variables are discarded in the screening step. In terms of speed, we see the smoothed rank loss offers some obvious benefits in the computational efficiency: CRR-SCAD is as fast as LS-SCAD and it is about two orders of magnitude faster than RR-SCAD. LS-SCAD is implemented in a standard way by using the LLA algorithm with the `glmnet` package. When

Table 4: Real data analysis. Comparison of prediction error and run time using least-square regression with SCAD (LS-SCAD), rank regression with SCAD (RR-SCAD), and convoluted rank regression with SCAD (CRR-SCAD). The data is split into a training and a test set in the ratio of 1:1 and the fused Kolmogorov filter is applied to reduced the dimension to 300 and 5000. All the quantities are averaged over 200 random partitions. The lowest prediction errors are in boldface, and standard errors are given in parentheses.

| | $p = 300$ | | | $p = 5000$ | | |
|---|---|---|---|---|---|---|
| method | prediction error | | time (sec) | prediction error | | time (sec) |
| LS-SCAD | 1.027 | (0.018) | 2.52 | 1.061 | (0.017) | 8.76 |
| RR-SCAD | 0.942 | (0.015) | 20.86 | 0.865 | (0.012) | 487.91 |
| CRR-SCAD | **0.898** | (0.010) | 1.86 | **0.825** | (0.009) | 7.81 |

we implemented CRR-SCAD, we made some efforts to integrate the GCD and LLA algorithms by avoiding some repeated computation, thus our CRR-SCAD is even faster than LS-SCAD when $p = 5000$. Without such implementation efforts, our CRR-SCAD would be slower than LS-SCAD.

# References


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

BÜHLMANN, P. and VAN DE GEER, S. (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications.* Springer Science & Business Media.

CANDES, E. and TAO, T. (2007). The Dantzig selector: statistical estimation when $p$ is much larger than $n$. *Annals of Statistics* **35**, 2313–2351.

CHIANG, A. P., BECK, J. S. ,YEN, H.-J., TAYEH, M. K., SCHEETZ, T. E., SWIDERSKI, R., NISHIMURA, D., BRAUN, T. A., KIM, K.-Y., HUANG, J., ELBEDOUR, K., CARMI, R., SLUSARSKI, D. C., CASAVANT, T. L., STONE, E. M. and SHEFFIELD, V. C. (2006). Homozygosity mapping with SNP arrays identifies TRIM32, an E3 ubiquitin ligase, as a Bardet-Biedl syndrome gene (BBS11). *Proceedings of the National Academy of Sciences* **103**, 6287–6292.

FAN, J. and LI, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96**, 1348–1360.

FAN, J., LI, R., ZHANG, C.-H. and ZOU, H. (2020). *Statistical Foundations of Data Science.* Chapman & Hall, CRC.

FAN, J., XUE, L. and ZOU, H. (2014b). Strong oracle optimality of folded concave penalized estimation. *Annals of Statistics* **42**, 819–849.

FERNADES, M., GUERRE, E. and HORTA, E. Smoothing Quantile Regressions. *Journal of Business and Economic Statistics* **39**, 338–357.

FRIEDMAN, J., HASTIE, T. and TIBSHIRANI, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software* **33**, 1.

GU, Y., FAN,J., KONG,L., MA, S. and ZOU, H. (2018). ADMM for high-dimensional sparse penalized quantile regression. *Technometrics* **60**, 319–331.

HETTMANSPERGER, T. P. and MCKEAN, J. W. (2010). *Robust Nonparametric Statistical Methods.* CRC Press.

JAECKEL, L. A. (1972). Estimating regression coefficients by minimizing the dispersion of the residuals. *The Annals of Mathematical Statistics* **58**, 1449–1458.

MAI, Q. and ZOU, H. (2015). The fused Kolmogorov filter: A nonparametric model-free screening method. *Annals of Statistics* **43**, 1471–1497.

SCHEETZ, T.E., KIM, K.-Y. A. ,SWIDERSKI, R.E., PHILP, A.R., BRAUN, T.A., KNUDTSON, K.L., DORRANCE, A.M., DIBONA, G.F., HUANG, J., CASAVANT, T.L., SHEFFIELD, V.C. and STONE, E.M. (2006). Regulation of gene expression in the mammalian eye and its relevance to eye disease. *Proceedings of the National Academy of Sciences* **103**, 14429–14434.

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

TSENG, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications* **109**, 475–494.

WANG, L., KIM, Y. and LI, R. (2013). Calibrating non-convex penalized regression in ultra-high dimension. *Annals of Statistics* **41**, 2505–2536.

WANG, L. and LI, R. (2009). Weighted Wilcoxon-type smoothly clipped absolute deviation method. *Biometrics* **65**, 564–571.

WANG, L., PENG, B., BRADIC, J., LI, R. and WU, Y. (2020). A tuning-free robust and efficient approach to high-dimensional regression (with discussion). *Journal of the American Statistical Association* **115**, 1700–1714.

WANG, L., WU, Y. and LI, R. (2012). Quantile regression for analyzing heterogeneity in ultra-high dimension. *Journal of the American Statistical Association* **107**, 214–222.

YANG, Y. and ZOU, H. (2013). An efficient algorithm for computing the HHSVM and its generalizations. *Journal of Computational and Graphical Statistics* **22**, 396–415.

ZHANG, C.-H. (2010). 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. *Annals of Statistics* **36**, 1567–1594.

ZOU, H. and LI, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *Annals of Statistics* **36**, 1509–1533.