High-Dimensional Regression Under Correlated Design: An Extensive Simulation Study

S. Ejaz Ahmed, Hwanwoo Kim, Gökhan Yıldırım, and Bahadır Yüzbaşı

Abstract Regression problems where the number of predictors, p, exceeds the number of responses, n, have become increasingly important in many diverse fields in the last couple of decades. In the classical case of "small p and large n," the least squares estimator is a practical and effective tool for estimating the model parameters. However, in this so-called Big Data era, models have the characteristic that p is much larger than n. Statisticians have developed a number of regression techniques for dealing with such problems, such as the Lasso by Tibshirani (J R Stat Soc Ser B Stat Methodol 58:267–288, 1996), the SCAD by Fan and Li (J Am Stat Assoc 96(456):1348–1360, 2001), the LARS algorithm by Efron et al. (Ann Stat 32(2):407–499, 2004), the MCP estimator by Zhang (Ann Stat. 38:894–942, 2010), and a tuning-free regression algorithm by Chatterjee (High dimensional regression and matrix estimation without tuning parameters, 2015, https://arxiv.org/abs/1510.07294). In this paper, we investigate the relative performances of some of these methods for parameter estimation and variable selection through analyzing real and synthetic data sets. By an extensive Monte Carlo simulation study, we also compare the relative performance of proposed methods under correlated design matrix.

Keywords Correlated design · Penalized and non-penalized methods · High-dimensional data · Monte Carlo

S. E. Ahmed (✉)
Department of Mathematics and Statistics, Brock University, St. Catharines, ON, Canada
e-mail: sahmed5@brocku.ca

H. Kim
Department of Statistics, University of Michigan, Ann Arbor, MI, USA
e-mail: moloque@umich.edu

G. Yıldırım
Department of Mathematics, Bilkent University, Çankaya, Ankara, Turkey
e-mail: gokhan.yildirim@bilkent.edu.tr

B. Yüzbaşı
Department of Econometrics, Inonu University, Malatya, Turkey
e-mail: bahadir.yuzbasi@inonu.edu.tr

S. E. Ahmed et al. (eds.), *Matrices, Statistics and Big Data*,
Contributions to Statistics, https://doi.org/10.1007/978-3-030-17519-1_11

1 Introduction

There are a host of buzzwords in today's data-centric world. We encounter data in all walks of life, and for analytically- and objectively-minded people, data is crucial to their goals. However, making sense of the data and extracting meaningful information from it may not be an easy task. The rapid growth in the size and scope of data sets in a variety of disciplines have naturally led to the usage of the term, Big Data. The word Big Data is nebulously defined. Generally speaking, it is often used to denote a dataset containing a large number of sample observations with factors that could induce significant problems when analyzing it. Due to these barriers when analyzing data, statisticians could play a vital role in the data world. A variety of statistical and computational tools are needed to reveal the story that is contained in the data and statisticians should fulfill expectations on a need for innovative statistical strategies for understanding and analyzing them.

Among many problems arisen from Big Data, in the realm of statistics, many people worked on the so-called high dimensional data (HDD), which are data sets containing larger number of predictors than the number of observations. The analysis of HDD is important in multiple research fields such as engineering, social media networks, bioinformatics and medical, environmental, and financial studies among others. There is an increasing demand for efficient prediction strategies and variable selection procedures for analyzing HDD. Some examples of HDD that have prompted demand are gene expression arrays, social network modeling and clinical, genetic, and phenotypic data. Developing innovative statistical learning algorithms and data analytic techniques play a fundamental role for the future of research in these fields. More public and private sectors are now acknowledging the importance of statistical tools and its critical role in analyzing HDD.

The challenges are to find novel statistical methods to extract meaningful conclusions and interpretable results from HDD. The classical statistical strategies do not provide solutions to such problems. Traditionally, statisticians used best-subset selection or other variable selection procedures to choose predictors that are highly correlated with the response variable. Based on the selected predictors, statisticians employed classical statistical methods to analyze HDD. However, with a huge number of predictors, implementing a best-subset selection is already computationally burdensome. On top of that, these variable selection techniques suffer from high variability due to their nature. To resolve such issues, a class of penalized estimation methods have been proposed. They are referred to as penalized estimation methods since they share the idea to estimate parameters in a model using classical least squares approach with an additional penalty term. Some of these methods not only perform variable selection and parameter estimation simultaneously, but also are extremely computationally efficient.

There are two main objectives of this paper. First is to give an idea how existing HDD methods perform on datasets when the correlations among response variables are present. Many statisticians studied HDD methods under the assumption that response variables are independent. This assumption certainly allowed statisticians to develop more complex estimation methods and provided practitioners with cautionary aspects when dealing with datasets containing non-independent variables. However, the assumption is not realistic, which necessitates statistical methods not relying on independence assumption. By examining how existing HDD method works on non-independent setting, we expect to gain some insights on developing novel estimation strategies and variable selection procedures for the datasets with non-independent variables. In addition to that many existing HDD analysis methods rely on a tuning parameter, which is burdensome to calibrate. It makes harder for non-technical scientists to analyze HDD and further, even for technical scientists, tuning parameter brings difficulty in reproducing research outcomes. Recently original methods that don't require tuning parameter when analyzing HDD were introduced. In this paper, we also compare the performance of these new methods with the ones using tuning parameters in a variety of different settings.

The rest of the paper is organized as follows: In Sect. 2, we review the definitions and basic properties of the regression methods which we will mainly focus on. In Sect. 3, we explain our simulation set-up for synthetic data and present the simulation results. Sect. 4 shows an application of the methods to two real data sets, prostate data and riboflavin production in *Bacillus subtilis* data. We finish the paper by Sect. 5 with some concluding remarks and future research directions.

2 Penalized Regression Methods

Consider the following linear model

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

where $\mathbf{Y} = (y_1, y_2, \ldots, y_n)'$ is a vector of responses, $\mathbf{X} = [x_{ij}]$ is an $n \times p$ fixed design matrix, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$ is an unknown vector of parameters, and $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)'$ is the vector of unobserved random errors. We assume that coordinates of the error vector $\boldsymbol{\varepsilon}$ are i.i.d. normally distributed with mean zero and variance σ^2. For the rest of the paper, without loss of generality, we assume that the predictors and responses in (1) are standardized so that $\sum_{i=1}^{n} y_i = 0$ and $\sum_{i=1}^{n} x_{ij} = 0$, $\sum_{i=1}^{n} x_{ij}^2 = 1$, for all j.

For a given model as in (1), there are three main tasks that need to be performed by a practitioner:

1. Parameter estimation: Finding an estimator $\widehat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$.
2. Variable selection or model selection: Selecting the non-zero entries of $\boldsymbol{\beta}$ accurately.
3. Prediction: Estimating $\mathbf{X}\boldsymbol{\beta}$.

For the case $n > p$, the classical estimator of $\boldsymbol{\beta}$ is the ordinary least square estimator (OLS), which is obtained by minimizing the residual sum of squares. It is given by

$$\widehat{\boldsymbol{\beta}}^{\text{OLS}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}.$$

However, in the high-dimensional setting, where $p > n$, the inverse of the Gram matrix, $(\mathbf{X}'\mathbf{X})^{-1}$, does not exist. More precisely, there will be infinitely many solutions for the least squares minimization, hence there is no well-defined solution. In fact, even in the case $p \leq n$ and p close to n, the OLS estimator is not considered very useful because standard deviations of estimators are usually very high. In many regression models, in particular for the high dimensional case, only some of the predictors have a direct significant effect on the response variables. Therefore it is convenient to assume that the underlying true model is sparse; that is, the true model has only a relatively small number of non-zero predictors. The sparsity induced methods also play an important role in high dimensional statistics because they induce interpretable models. It is well-known that the least squares estimation procedure is unlikely to yield zero estimates for many of the model coefficients. There are many alternatives to the least square estimation such as subset selection, dimension reduction, and penalization methods. Each of them has its own advantages and disadvantages. For a thorough exposition, see [1, 14].

In this paper, we consider the penalized least square regression methods to obtain estimators for the model parameters in (1). The key idea in penalized regression methods is minimizing an objection function $L_{\rho,\lambda}$ in the form of

$$L_{\rho,\lambda}(\boldsymbol{\beta}) = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\rho(\boldsymbol{\beta}) \tag{2}$$

to obtain an estimate for the model parameter $\boldsymbol{\beta}$. The first term in the objective function is the sum of the squared error loss, the second term ρ is a penalty function, and λ is a tuning parameter which controls the trade-off between two components of $L_{\rho,\lambda}$.

The penalty function is usually chosen as a norm on $\mathbb{R}^p$, in most cases an l_q-norm, which can be written as

$$\rho_q(\boldsymbol{\beta}) = \sum_{j=1}^{p} |\beta_j|^q, \ q > 0. \tag{3}$$

The class of estimators employing the above type of penalties are called the bridge estimators, proposed by Frank and Friedman [11].

The ridge regression [11, 17] minimizes the residual sum of squares subject to an l_2-penalty, that is,

$$\widehat{\boldsymbol{\beta}}^{\text{Ridge}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}, \tag{4}$$

where λ is a tuning parameter. Although the ridge estimator is a continuous shrinkage method and has a better prediction performance than OLS through bias-variance trade-off, it does not set any OLS estimates to zero, so obtaining a sparse model is not possible. However, in the case of l_q-penalty with $q \leq 1$, some coefficients are set exactly to zero. And the optimization problem for (2) becomes a convex optimization problem, which can be easily solved, for the case $q \geq 1$. Therefore, l_1-penalty is special for both reasons.

There are other penalized regression methods with more sophisticated penalty functions which not only shrink all the coefficients toward zero, but also set some of them exactly to zero. As a result, this class of estimators usually produce biased estimates for the parameters due to the shrinkage, but have some advantages such as producing more interpretable submodels and reducing the variance of the estimator.

Several penalty estimators have been proposed in the literature for linear and generalized linear models. In this paper, we only consider the *least absolute shrinkage and selection operator* (Lasso) [26], the *smoothly clipped absolute deviation* (SCAD) [9], *the adaptive Lasso* (aLasso) [30], and the *minimax concave penalty* (MCP) method [29]. These methods perform parameter estimation and model selection simultaneously. In addition to these penalty estimators, we also consider the *tuning-free regression method* (CTFR) which has been recently proposed by Chatterjee in [6].

It is known that as the prediction performance of Ridge, Bridge, and Lasso are compared, none of them uniformly dominates others [12]. But the Lasso has a significant advantage over ridge and bridge estimators in terms of variable selection performance, see [12] and [26]. Another important advantage of penalized regression techniques is that they can be used when the number of predictors, p, is much larger than the number of observations, n. However, in an effort to achieve meaningful estimation and selection properties, most penalized regression methods make some important assumptions on both the true model and the designed matrix. We refer to [5] and [14] for more insights.

2.1 Lasso

The Lasso was proposed by Tibshirani [26], which performs variable selection and parameter estimation simultaneously, thanks to the l_1-penalty. The Lasso estimator is defined by

$$\widehat{\boldsymbol{\beta}}_n^{\text{Lasso}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{5}$$

Note that for the high-dimensional case, $p > n$, there might be multiple solutions of (5); nevertheless for any two solutions $\widehat{\boldsymbol{\beta}}_n^1, \widehat{\boldsymbol{\beta}}_n^2$, we have $\mathbf{X}\widehat{\boldsymbol{\beta}}_n^1 = \mathbf{X}\widehat{\boldsymbol{\beta}}_n^2$ which implies that all solutions have the same prediction performance [27].

In order to understand the role of the penalty function in shrinkage, it is instructive to consider the orthogonal design, that is, $n = p$ and the design matrix $\mathbf{X} = \boldsymbol{I}_n$ is the identity matrix [19]. In this case, the OLS solution minimizes

$$\sum_{i=1}^{p}(y_i - \beta_i)^2$$

and the estimates are given by

$$\widehat{\beta}_i^{\mathrm{OLS}} = y_i \quad \text{for all} \quad 1 \le i \le p.$$

The ridge regression solution minimizes

$$\sum_{i=1}^{p}(y_i - \beta_i)^2 + \lambda \sum_{i=1}^{p} \beta_i^2$$

and hence the estimates are given by

$$\widehat{\beta}_i^{\mathrm{Ridge}} = y_i/(1+\lambda) \quad \text{for all} \quad 1 \le i \le p.$$

Similarly, the Lasso solution minimizes

$$\sum_{i=1}^{p}(y_i - \beta_i)^2 + \lambda \sum_{i=1}^{p} |\beta_i|$$

and the estimates take the form

$$\widehat{\beta}_i^{\mathrm{Lasso}} = \begin{cases} y_i - \lambda/2 & \text{if} \quad y_i > \lambda/2; \\ y_i + \lambda/2 & \text{if} \quad y_i < -\lambda/2; \\ 0 & \text{if} \quad |y_i| \le \lambda/2. \end{cases}$$

We see that the shrinkage applied by Ridge and Lasso affect the estimated parameters differently. In the Lasso solution, the least square coefficients with absolute value less than $\lambda/2$ are set exactly equal to zero, and other least squares coefficients are shrunken towards zero by a constant amount, $\lambda/2$. As a result, sufficiently small coefficients are all estimated as zero. On the other hand, the ridge regression shrinks each least squares estimate towards zero by multiplying each one by a constant proportional to $1/\lambda$. For more general design matrix, we do not have explicit solutions for the estimates, but the effect of the shrinkage is similar as in orthogonal design case, see [14, 15] and [19] for more on this topic.

Originally, the Lasso solutions were obtained via quadratic programming by Tibshirani [26]. Later, Efron et al. proposed Least Angle Regression (LARS) algorithm, which is a homotopy method that constructs a piece-wise linear solution path in an effective way [8] . Coordinate descent algorithms, which use the sparsity assumption, are also simple and very fast to compute for the Lasso estimator [10]. The popular *glmnet* package of R language implements coordinate descent for the Lasso solution [10]. Further, the Lasso estimator remains numerically feasible for dimensions of p that are much higher than the sample size n.

Tuning parameter plays a very crucial role for the performance of the Lasso as well. Meinshausen and Bühlmann [23] showed that if the penalty parameter λ is tuned to obtain optimal prediction, then consistent variable selection cannot hold: the Lasso solution includes many noise variables besides the true signals. Leng et al. [22] proved this fact in a short argument by considering a model with orthogonal design. Thus, we can say that variable selection and parameter estimation are closely related but different problems.

There has been significant progress on the theoretical properties of Lasso's performance for parameter estimation and prediction in the last two decades. It was first proved by Knight and Fu [20] that the estimator $\widehat{\boldsymbol{\beta}}$ is consistent when p is fixed and n tends to infinity provided that the tuning parameter satisfies a growth condition. Under the assumption that

$$\frac{1}{n}\mathbf{X}'\mathbf{X} \to \boldsymbol{C}$$

where $\boldsymbol{C}$ is a positive definite matrix, Knight and Fu also [20] proved that the Lasso solution has the following properties depending on how the tuning parameter is chosen. Consider

$$\widehat{\boldsymbol{\beta}}_n = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda_n \sum_{j=1}^{p} |\beta_j| \right\}. \tag{6}$$

Then

(a) If $\lambda_n/n \to \lambda_0 \geq 0$, then $\hat{\boldsymbol{\beta}}_n \xrightarrow{p} \arg\min V_1$ where

$$V_1(u) = (u - \boldsymbol{\beta})'\boldsymbol{C}(u - \boldsymbol{\beta}) + \lambda_0 \sum_{i=1}^{p} |u_i|.$$

(b) If $\lambda_n/\sqrt{n} \to \lambda_0 \geq 0$, then $\sqrt{n}(\widehat{\boldsymbol{\beta}}_n - \boldsymbol{\beta}) \xrightarrow{d} \arg\min V_2$ where

$$V_2(u) = -2u^T\boldsymbol{W} + u^T\boldsymbol{C}u + \lambda_0 \sum_{i=1}^{p}[u_i \operatorname{sgn}(\beta_i) I_{\beta_i \neq 0} + |u_i| I_{\beta_i = 0}].$$

The results above imply that when $\lambda_n = O(\sqrt{n})$, the Lasso estimator cannot recover the true signals with a positive probability, and $\lambda_n = o(n)$ is sufficient for consistency.

It is well-known that with non-singular $\boldsymbol{C}$, the OLS estimator is consistent when p is fixed and n tends to infinity, and

$$\sqrt{n}(\widehat{\boldsymbol{\beta}}_n^{\text{OLS}} - \boldsymbol{\beta}) \to_d N(\mathbf{0}, \sigma^2 \boldsymbol{C}^{-1}).$$

In the regime where both n and p tend to infinity, the standard consistency definition is not valid for any estimator $\widehat{\boldsymbol{\beta}}$. Greenshtein and Ritov [13] introduced the "persistency" concept in this setting as an analogue of consistency of an estimator, and proved that the Lasso estimator is persistent under some assumptions on the design matrix. Since the Lasso automatically sets some components of $\widehat{\boldsymbol{\beta}}$ to zero, model selection consistency for the Lasso estimator is a crucial problem. In the case that the tuning parameter is chosen by a deterministic rule, under a couple of assumptions on the design matrix and sparsity of $\boldsymbol{\beta}$, it is known that the Lasso estimator recovers the true parameter set, see [3, 5]. Even though these theoretical results are satisfactory, the assumption that the tuning parameter is to be chosen deterministically does not shed light on practical applications of the Lasso, because in most applications the tuning parameter is chosen using some data-driven methods such as cross-validation. There are only few theoretical results on the Lasso when the tuning parameter is chosen in a data-dependent way, see the recent result of [7] and the references therein. For an in-depth study of Lasso, see two recent excellent books [5] and [15].

In many diverse applications of regression, it is not realistic to assume that the predictors are independent. Therefore the influence of correlations among predictors on parameter estimation and prediction is an important problem. In general, it was believed that there would be large prediction errors when the predictors are correlated. However, recent results in [16] show that this is not necessarily true, and they argue that for correlated designs, small tuning parameters can be chosen so that some satisfactory error bounds on the prediction error can be achieved. Their theoretical arguments and simulation results show that Lasso performs well under any degree of correlations if the tuning parameter is chosen suitably. Besides this fact, they show that choosing λ proportional to $\sqrt{n \log p}$ and ignoring the correlations in the design is not favorable. In the next section, we present our simulation results on the cross-validated Lasso's prediction performance under correlated design.

2.2 aLasso

Zhou [30] introduced the adaptive Lasso (aLasso) by modifying the Lasso penalty by using adaptive weights on the l_1-penalty. In the same paper, it has been shown theoretically that the adaptive Lasso estimator is able to identify the true model consistently, and has the so-called oracle property. An estimator is said to have oracle property if asymptotically the method performs as well as if the statistician had known which coefficients were non-zero and which were zero in advance.

The aLasso $\widehat{\beta}^{\text{aLasso}}$ is obtained by

$$\widehat{\boldsymbol{\beta}}^{\text{aLasso}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \widehat{w}_j |\beta_j| \right\}, \tag{7}$$

where the weight function is

$$\widehat{w}_j = \frac{1}{|\widehat{\beta}_j^*|^{\gamma}}; \quad \gamma > 0,$$

and $\widehat{\beta}_j^*$ is a root-n-consistent estimator of β. The minimization procedure for the aLasso solution does not induce any computational difficulty and can be solved very efficiently, for the details see section 3.5 in [30], see and [18]. Zhou [30] also proved that if $\lambda_n/\sqrt{n} \to 0$ and $\lambda_n n^{(\gamma-1)/2} \to \infty$, the aLasso estimates have the following properties:

1. aLasso has variable selection consistency with probability one as n tends to infinity.
2. $$\sqrt{n}(\widehat{\boldsymbol{\beta}}_n^{\text{aLasso}} - \boldsymbol{\beta}) \to_d N(\mathbf{0}, \sigma^2 \boldsymbol{C}_{11}^{-1})$$

   where $\boldsymbol{C}_{11}^{-1}$ is the submatrix of $\boldsymbol{C}$ which corresponds to the non-zero entries of $\boldsymbol{\beta}$.

2.3 SCAD

Although the Lasso method does both shrinkage and variable selection due to the nature of the l_1-penalty by setting many coefficients identically to zero, it does not possess oracle properties, as discussed in [9]. To overcome the inefficiency of traditional variable selection procedures, Fan and Li [9] proposed SCAD to select variables and estimate the coefficients of variables automatically and simultaneously. In the same paper, they proved that SCAD has oracle property as well. This method not only retains the good features of both subset selection and ridge regression, but also produces sparse solutions, ensures continuity of the selected models (for the stability of model selection), and has unbiased estimates for large coefficients. The estimator is obtained by

$$\widehat{\boldsymbol{\beta}}^{\mathrm{SCAD}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \sum_{j=1}^{p} p_{\gamma,\lambda}(|\beta_j|) \right\}. \tag{8}$$

Here $p_{\gamma,\lambda}(\cdot)$ is the smoothly clipped absolute deviation penalty which is defined on $[0, \infty)$ by

$$p_{\gamma,\lambda}(x) = \begin{cases} \lambda x, & \text{if } x \leq \lambda, \\ \frac{\gamma\lambda x - 0.5(x^2+\lambda^2)}{\gamma-1} & \text{if } \lambda < x \leq \gamma\lambda, \\ \frac{\lambda^2(\gamma^2-1)}{2(\gamma-1)} & \text{if } x > \gamma\lambda \end{cases}$$

where $\lambda \geq 0$ and $\gamma > 2$. Note that SCAD is identical with the Lasso for $|x| \leq \lambda$, then continuously changes to a quadratic function until $|x| = \gamma\lambda$, and then it remains constant for all $|x| > \gamma\lambda$. The lower values of γ produce more variable but less biased estimates. For $\gamma = \infty$, the SCAD penalty is equivalent to the l_1-penalty.

2.4 MCP

Zhang [28] introduced a new penalization method for variable selection, which is given by

$$\widehat{\boldsymbol{\beta}}^{\mathrm{MCP}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \sum_{j=1}^{p} p_{\gamma,\lambda}(|\beta_j|) \right\},$$

where the MCP penalty $p_{\gamma,\lambda}(\cdot)$ is given by

$$p_{\gamma,\lambda}(x) = \begin{cases} \lambda|x| - \frac{x^2}{2\gamma}, & \text{if } |x| \leq \gamma\lambda, \\ \frac{1}{2}\gamma\lambda^2 & \text{if } |x| > \gamma\lambda, \end{cases}$$

where $\gamma > 1$ and λ are regularization parameters. The MCP has the threshold value $\gamma\lambda$. The penalty is a quadratic function for values less than the threshold and is constant for values greater than it. The parameter $\gamma > 0$ controls the convexity and therefore the bias of the estimators. The lower values of γ give us more variable but less biased estimates. By controlling the parameter γ, under less restricted assumptions than those required by the Lasso, one can reduce almost all the bias of the estimators and obtain consistent variables. The MCP solution path converges to Lasso path as $\gamma \to \infty$. Zhang [29] proves that the estimator possesses selection consistency at the universal penalty level $\lambda = \sigma\sqrt{2/n \log p}$ under the sparse Riesz condition on the design matrix $\mathbf{X}$. It has been proven that the MCP has oracle property, for more on the properties of the estimator, see [28, 29].

2.5 CTFR

Chatterjee [6] introduced a general theory for Gaussian mean estimation that, when applied to the linear regression problem, only requires the design matrix and the response vector as input and hence no tuning parameter is required. In [6], Chatterjee showed that the proposed estimator is adaptively minimax rate-optimal in high-dimensional regression case. We call this estimator Chatterjee's tuning-free regression, CTFR in short, in this paper. The estimator $\widehat{\boldsymbol{\beta}}$ is obtained by

$$\widehat{\boldsymbol{\beta}}^{\text{CTFR}} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\{|\boldsymbol{\beta}|_1 : ||\mathbf{Y}' - \mathbf{X}\boldsymbol{\beta}||_2^2 \leq k\hat{\sigma}^2\}$$

where k is the rank of $\mathbf{X}$, $\mathbf{Y}'$ is the projection of $\mathbf{Y}$ onto the column space of $\mathbf{X}$, and $\hat{\sigma}$ is a randomized estimator of σ introduced in the same paper. For the details, see the paper [6]. The following result from the same paper gives an upper bound on the expected mean squared prediction error of $\hat{\boldsymbol{\beta}}$:

$$\frac{\mathbb{E}||\mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta}||_2^2}{n\sigma^2} \leq C\left(r + r^2 + \sqrt{\frac{\log(p+n)}{n}} + \frac{\log(p+n)}{n}\right)$$

where $r = \frac{|\boldsymbol{\beta}|_1\gamma}{\sigma}\sqrt{\frac{\log(p+n)}{n}}$, $\gamma = \max_{1\leq j\leq p}||\mathbf{X}_j||/\sqrt{n}$, and C is a universal constant.

Chatterjee [6] compared his proposed estimator's performance with the 10-fold cross validated Lasso. The proposed estimator has generally higher prediction error than the 10-fold cross-validated Lasso. On the other hand, the proposed estimator has better performance at model selection: the number of false positives returned by the proposed estimator is significantly less than that of the 10-fold cross-validated Lasso. See the next section, for more simulation results.

2.6 TREX

Lederer and Muller [21] introduced another tuning-free regression method, TREX, which is obtained by a careful analysis of Square-Root Lasso [2]. They showed that TREX can outperform a cross-validated Lasso in terms of variable selection and computational efficiency through a detailed numeric study. The estimator is defined as

$$\widehat{\boldsymbol{\beta}}^{\text{TREX}} \in \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p}\{\frac{||\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}||_2^2}{\frac{1}{2}||\mathbf{X}'(\mathbf{Y}-\mathbf{X}\boldsymbol{\beta})||_\infty}+|\boldsymbol{\beta}|_1\}.$$

For some numerical results on the variable selection performance of the estimator, see [21].

3 Experimental Study

3.1 Simulation Setup

We consider the model

$$\mathbf{Y}=\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\epsilon} \tag{9}$$

where $\mathbf{X}$ is $n \times p$ dimensional predictor matrix, $\mathbf{Y}$ is the n-dimensional response vector, and $\boldsymbol{\epsilon}$ is the n-dimensional unobserved error vector. $\boldsymbol{\beta}$ is a p-dimensional vector of coefficients. Each component of $\boldsymbol{\epsilon}$ is generated from standard normal distribution.

In our simulation study, we basically follow the simulation set-up of [24]. All simulations were based on a sample size of $n = 100$. We considered two different values for the number of predictor variables: $p = 500$ and $p = 1000$. Entries of the predictor matrix $\mathbf{X}$ were randomly sampled from the standard normal distribution. Correlation between columns of $\mathbf{X}$ is set to ρ, where $\rho \in \{0, 0.2, 0.4, 0.6, 0.8\}$.

The number of non-zero elements of $\boldsymbol{\beta}$ was set to $\lceil n^\alpha \rceil$, where α controls the sparsity of $\boldsymbol{\beta}$. We chose $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. We picked $\lceil n^\alpha \rceil$ number of indices randomly. For these indices, the coordinates of $\boldsymbol{\beta}$ were sampled from a $Laplace(1)$ distribution while the rest of the coordinates of $\boldsymbol{\beta}$ were set to zero. The resulting $\boldsymbol{\beta}$ vector was rescaled according to the pre-specified signal-to-noise ratio, snr, defined as $\frac{\boldsymbol{\beta}'\boldsymbol{\Sigma}\boldsymbol{\beta}}{\sigma^2}$, where $snr \in \{0.5, 1, 2, 5, 10, 20\}$. Here $\boldsymbol{\Sigma}$ is the covariance matrix of the predictors. We assumed homoscedasticity condition where σ^2 is the error variance of each predictor variables. We assume that $\sigma = 1$ in all our simulations.

We investigate the relative performance of the following five estimators:

1. Lasso
2. aLasso
3. CTFR
4. SCAD
5. MCP

Tuning parameters for the above methods, except for CTFR, were chosen to minimize 10-fold cross validation error. For aLasso, we used the obtained Lasso estimate as the weight for the penalty term. In the simulation tables, we summarize our simulation results by comparing the performance of estimator CTFR, SCAD, MCP, and aLasso with relative to Lasso. The performance was measured based on the following metrics, respectively:

(a) TP = the number of true positives
(b) FP = the number of false positives
(c) PE = the prediction error

We define the relative number of true positives of listed methods to Lasso as:

$$RTP = \frac{\text{number of true positives of any method in the list}}{\text{number of true positives of Lasso}}.$$

Clearly the value greater than one, which is shown as bold in Tables 1–10, will indicate the superiority of the suggested method over Lasso in selecting true positives, otherwise Lasso is relatively performing well. For instance, in Table 1, for $p = 500$ and parameters $(\rho, \alpha, \text{snr}) = (0, 0.1, 0.5)$, the value 0.712 in RTP of CTFR is computed by

$$\frac{\text{number of true positives of CTFR}}{\text{number of true positives of Lasso}} = 0.712.$$

Similarly two other relative measures RFP and RPE are, respectively, defined as:

$$RFP = \frac{\text{number of false positives of any method in the list}}{\text{number of false positives of Lasso}}.$$

$$RPE = \frac{\text{prediction error of any method in the list}}{\text{prediction error of Lasso}}.$$

For our simulations, we used a *cv.glmnet* function in the *glmnet* package in R language for Lasso and aLasso, and a *cv.ncvreg* function in the *ncvreg* package for SCAD and MCP methods. The implementation of CTFR is done in R language, and it can be provided upon request.

3.2 Simulation Results

In this section, we present our simulation results (Tables 2, 3, 4, 5, 6, 7, 8, 9, 10).

Table 1 Simulation results for $p = 500$ and $\rho = 0$

	α	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
		snr = 0.5					snr = 1				
RTP	CTFR	0.712	0.227	0.679	0.475	0.907	0.926	0.220	0.193	**1.155**	**1.089**
	MCP	0.619	**1.258**	0.450	0.530	0.608	0.473	**1.981**	0.197	0.572	**1.680**
	SCAD	0.443	1.000	**2.364**	0.434	**1.291**	0.577	**1.045**	**2.056**	0.658	0.971
	aLasso	0.500	0.608	**1.711**	**1.970**	**1.079**	0.176	**1.155**	0.911	**5.630**	**1.023**
RFP	CTFR	0.105	0.108	0.225	0.130	0.170	0.027	0.038	0.029	0.164	0.158
	MCP	0.233	0.320	0.268	0.205	0.135	0.155	0.414	0.142	0.232	0.357
	SCAD	0.500	0.872	0.903	0.500	0.738	0.572	0.820	0.849	0.633	0.498
	aLasso	**1.200**	0.625	**1.416**	0.742	**1.198**	0.497	**1.731**	0.551	**1.666**	**1.012**
RPE	CTFR	**1.587**	0.837	0.936	**1.283**	**1.979**	**2.542**	0.843	0.902	**3.439**	**3.212**
	MCP	0.496	0.900	0.795	**1.017**	**1.020**	0.312	**1.003**	0.302	1.149	**2.813**
	SCAD	0.309	0.812	**1.596**	**1.117**	0.856	0.288	0.630	**1.276**	0.925	**1.166**
	aLasso	**1.114**	**1.323**	**2.868**	**3.250**	**1.619**	0.759	**2.234**	**1.623**	**4.947**	**1.313**
		snr = 2					snr = 5				
RTP	CTFR	0.970	0.171	0.312	0.628	**1.109**	0.820	0.199	0.320	0.717	**1.115**
	MCP	0.781	**1.140**	0.235	0.673	**1.500**	0.500	**2.000**	0.367	0.744	**1.280**
	SCAD	0.556	0.945	**1.750**	0.486	**1.394**	0.301	**1.000**	**3.260**	0.706	0.954
	aLasso	0.306	0.711	**1.391**	**3.150**	0.986	0.149	0.602	**1.640**	**6.450**	0.936
RFP	CTFR	0.001	0.002	0.006	0.031	0.102	0.002	0.001	0.008	0.043	0.059
	MCP	0.156	0.165	0.109	0.170	0.342	0.084	0.189	0.157	0.163	0.171
	SCAD	0.620	0.451	0.577	0.541	0.547	0.229	0.423	**1.042**	0.575	0.378
	aLasso	0.604	**1.047**	0.921	**1.411**	0.934	0.430	0.849	0.967	**1.967**	0.842
RPE	CTFR	**3.712**	**1.040**	**1.954**	**3.231**	**6.045**	**5.945**	**1.745**	**3.441**	**7.927**	**12.286**
	MCP	0.284	0.469	0.295	0.811	**2.743**	0.185	0.418	0.161	**1.179**	4.221
	SCAD	0.289	0.437	0.984	0.501	**1.611**	0.139	0.351	0.977	0.597	**2.011**
	aLasso	**1.108**	**1.606**	**2.169**	**3.538**	**1.148**	0.731	**1.551**	**2.164**	**3.674**	0.831
		snr = 10					snr = 20				
RTP	CTFR	0.520	0.102	0.222	0.429	**1.140**	**1.000**	0.084	0.307	0.574	**1.437**
	MCP	0.667	**1.470**	0.399	0.713	**1.135**	0.508	**1.930**	0.347	0.857	**1.157**
	SCAD	0.241	**1.007**	**4.130**	0.662	0.957	0.228	0.995	**4.450**	0.635	0.930
	aLasso	0.140	0.361	**2.767**	**6.930**	0.946	0.127	0.449	**2.213**	**7.74**	0.955
RFP	CTFR	0.000	0.000	0.002	0.013	0.029	0.000	0.000	0.001	0.011	0.071
	MCP	0.098	0.274	0.139	0.150	0.176	0.067	0.279	0.090	0.113	0.160
	SCAD	0.305	0.487	**1.389**	0.402	0.403	0.176	0.637	0.989	0.412	0.303
	aLasso	0.323	0.639	**1.316**	**2.645**	0.862	0.256	0.641	**1.374**	**3.304**	0.881
RPE	CTFR	**8.631**	**2.160**	**3.318**	**10.859**	**22.52**	**17.507**	**2.128**	**9.489**	**19.077**	**48.814**
	MCP	0.208	0.521	0.289	**1.231**	**3.826**	0.127	0.726	0.164	0.775	**10.836**
	SCAD	0.168	0.497	**1.665**	**1.009**	**1.454**	0.118	0.529	**1.701**	0.268	**6.561**
	aLasso	0.411	**1.078**	**2.678**	**4.806**	0.874	0.631	**1.122**	**2.446**	**4.640**	0.598

Table 2 Simulation results for $p = 500$ and $\rho = 0.2$

	α	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
		snr = 0.5					snr = 1				
RTP	CTFR	0.658	**0.190**	0.425	0.441	0.873	0.697	0.389	0.210	0.846	0.589
	MCP	0.944	0.816	0.463	0.438	0.294	0.678	**1.414**	0.341	0.476	0.669
	SCAD	0.574	**1.042**	**1.737**	0.546	0.781	0.728	**1.027**	**1.212**	0.845	0.72
	aLasso	0.500	0.544	**2.028**	**2.237**	**1.083**	0.350	**1.081**	0.993	3.01	**1.21**
RFP	CTFR	0.141	0.064	0.145	0.158	0.309	0.024	0.123	0.045	0.173	0.097
	MCP	0.281	0.16	0.287	0.179	0.143	0.124	0.262	0.177	0.161	0.112
	SCAD	0.696	0.951	0.621	0.539	0.511	0.615	0.625	0.497	0.838	0.353
	aLasso	**1.511**	0.758	**1.974**	0.985	**1.068**	0.857	**1.286**	0.812	**1.376**	**1.387**
RPE	CTFR	**1.436**	0.653	**1.016**	**1.553**	**2.837**	**2.496**	0.793	0.918	**2.585**	**2.74**
	MCP	0.666	0.460	0.571	**1.029**	**1.656**	0.292	0.84	0.316	**1.127**	**2.565**
	SCAD	0.453	0.666	**1.314**	0.804	**1.136**	0.371	0.632	**1.070**	0.757	**1.314**
	aLasso	**1.232**	**1.420**	**3.455**	**2.746**	**1.397**	**1.026**	**1.981**	**1.780**	**3.287**	**1.086**
		snr = 2					snr = 5				
RTP	CTFR	0.970	0.232	0.469	0.720	0.725	0.670	0.062	0.280	0.569	**2.367**
	MCP	0.529	**1.730**	0.389	0.650	**1.081**	1.000	0.740	0.336	0.723	0.963
	SCAD	0.474	0.995	**2.100**	0.635	**1.071**	0.283	0.796	**3.230**	0.624	0.958
	aLasso	0.282	0.886	**1.095**	**3.470**	0.967	0.181	0.283	3.643	**5.410**	0.968
RFP	CTFR	0.036	0.036	0.122	0.121	0.087	0.002	0.000	0.056	0.029	0.213
	MCP	0.052	0.316	0.209	0.224	0.139	0.169	0.120	0.141	0.126	0.160
	SCAD	0.248	0.592	**1.213**	0.840	0.385	0.291	0.386	0.947	0.413	0.375
	aLasso	0.673	0.985	**1.185**	**2.15**	0.933	0.510	0.608	**2.307**	**2.117**	0.903
RPE	CTFR	**4.165**	**1.074**	**1.611**	**3.687**	**5.036**	**5.894**	**1.105**	**2.275**	**6.683**	**13.925**
	MCP	0.165	**1.107**	0.226	0.972	**3.298**	0.410	0.851	0.444	0.891	**3.595**
	SCAD	0.176	0.631	**1.290**	0.486	**1.943**	0.252	0.731	**2.233**	0.513	**2.038**
	aLasso	0.833	**1.993**	**1.657**	**4.283**	**1.069**	0.599	0.944	**2.748**	**4.194**	0.926
		snr = 10					snr = 20				
RTP	CTFR	0.980	0.128	0.266	0.603	**2.257**	1.000	0.153	0.206	0.759	**1.420**
	MCP	0.329	**2.941**	0.303	0.677	**1.256**	0.500	**2.000**	0.266	0.762	**1.405**
	SCAD	0.159	0.993	**6.118**	0.779	0.756	0.316	1.000	**3.040**	0.610	0.895
	aLasso	0.064	0.475	**2.105**	**15.412**	0.949	0.130	0.633	**1.575**	**7.540**	0.963
RFP	CTFR	0.000	0.001	0.046	0.032	0.371	0.000	0.001	0.001	0.068	0.103
	MCP	0.073	0.144	0.096	0.094	0.279	0.067	0.064	0.070	0.100	0.140
	SCAD	0.147	0.190	1.220	0.390	0.322	0.147	0.154	0.582	0.344	0.271
	aLasso	0.259	0.780	**1.326**	**3.922**	0.893	0.300	0.794	**1.204**	**2.866**	0.898
RPE	CTFR	**11.352**	**2.611**	**2.308**	**18.033**	**15.039**	**21.037**	**2.078**	**6.812**	**18.013**	**46.238**
	MCP	0.125	0.442	0.310	**1.403**	**4.361**	0.134	0.249	0.069	**1.176**	**13.914**
	SCAD	0.091	0.192	**2.381**	**1.139**	**1.655**	0.100	0.175	0.830	0.471	**4.997**
	aLasso	0.398	1.509	**2.044**	**7.357**	0.994	0.506	**1.363**	**2.264**	**5.146**	0.619

Table 3 Simulation results for $p = 500$ and $\rho = 0.4$

	α	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
		snr = 0.5					snr = 1				
RTP	CTFR	0.838	0.268	0.421	0.598	0.761	0.904	0.357	0.323	0.736	0.673
	MCP	0.540	**1.368**	0.369	0.443	0.472	0.354	**2.654**	0.333	0.423	0.557
	SCAD	0.504	0.973	**1.574**	0.556	0.793	0.393	0.98	**2.365**	0.813	0.517
	aLasso	0.479	0.882	**1.310**	**2.559**	**1.293**	0.187	**1.050**	0.98	**5.846**	**1.365**
RFP	CTFR	0.184	0.155	0.201	0.263	0.224	0.041	0.104	0.250	0.099	0.153
	MCP	0.145	0.189	0.204	0.176	0.098	0.072	0.225	0.341	0.094	0.091
	SCAD	0.466	0.573	0.606	0.735	0.371	0.245	0.528	**1.140**	0.734	0.252
	aLasso	**1.531**	**1.126**	**1.468**	**1.550**	**1.294**	0.575	0.955	**1.696**	**2.014**	**1.533**
RPE	CTFR	**1.678**	0.992	**1.657**	**1.120**	**1.686**	**2.367**	**1.006**	0.764	**3.500**	**3.079**
	MCP	0.358	**1.442**	0.729	**1.095**	**1.522**	0.291	0.930	0.363	**1.097**	**3.059**
	SCAD	0.346	0.846	**1.801**	0.559	**1.793**	0.334	0.576	**1.269**	0.860	**1.197**
	aLasso	**1.769**	**1.837**	**2.115**	**3.028**	**1.460**	0.676	**2.440**	**2.037**	**4.562**	**1.091**
		snr = 2					snr = 5				
RTP	CTFR	1.000	0.187	0.351	0.769	0.900	0.554	0.206	0.255	0.621	**1.188**
	MCP	0.550	**1.560**	0.283	0.488	0.726	0.466	**1.913**	0.417	0.472	0.592
	SCAD	0.427	0.978	**1.930**	0.422	0.997	0.277	0.984	**3.685**	0.558	0.689
	aLasso	0.298	0.786	**1.267**	**3.490**	0.987	0.173	0.551	**1.806**	**5.511**	**1.049**
RFP	CTFR	0.098	0.045	0.024	0.211	0.153	0.000	0.022	0.002	0.255	0.151
	MCP	0.080	0.136	0.059	0.146	0.101	0.054	0.106	0.088	0.129	0.086
	SCAD	0.247	0.470	0.426	0.415	0.364	0.216	0.254	0.700	0.611	0.240
	aLasso	**1.152**	0.999	**1.097**	**1.570**	0.970	0.380	0.854	**1.045**	**2.307**	**1.157**
RPE	CTFR	**2.591**	**1.006**	**2.454**	**2.395**	**5.005**	**5.673**	**1.737**	**4.061**	**4.630**	**8.596**
	MCP	0.212	**1.201**	0.335	**1.243**	**2.896**	0.296	0.626	0.360	**1.363**	**3.855**
	SCAD	0.153	0.759	**1.379**	0.666	**2.156**	0.222	0.353	**1.882**	0.636	**2.792**
	aLasso	**1.072**	**1.460**	**2.278**	**3.410**	**1.045**	0.735	**1.223**	**2.203**	**4.195**	0.798
		snr = 10					snr = 20				
RTP	CTFR	1.000	0.130	0.249	0.586	**1.240**	1.000	0.118	0.310	0.338	**1.096**
	MCP	0.649	**1.480**	0.282	0.633	0.703	0.532	**1.930**	0.297	0.637	0.960
	SCAD	0.283	0.961	**3.390**	0.584	0.569	0.266	**1.043**	**3.620**	0.516	0.720
	aLasso	0.127	0.433	**2.273**	**7.650**	0.967	0.131	0.519	**1.989**	**7.380**	0.918
RFP	CTFR	0.006	0.019	0.012	0.018	0.070	0.001	0.002	0.049	0.007	0.065
	MCP	0.038	0.093	0.044	0.076	0.088	0.053	0.053	0.038	0.066	0.082
	SCAD	0.109	0.316	0.478	0.242	0.150	0.103	0.296	0.386	0.223	0.167
	aLasso	0.508	0.957	1.311	**2.270**	0.904	0.559	0.657	**1.365**	**2.000**	0.830
RPE	CTFR	**7.373**	0.999	**4.014**	**14.454**	**30.866**	**11.127**	**1.383**	**6.855**	**22.267**	**50.654**
	MCP	0.145	0.533	0.175	**1.623**	**10.358**	0.158	0.562	0.106	**1.583**	**15.923**
	SCAD	0.114	0.429	**1.305**	0.746	**5.527**	0.109	0.396	**1.064**	0.681	**7.130**
	aLasso	0.566	**1.092**	**2.844**	**4.668**	0.718	0.546	**1.199**	**2.298**	**4.481**	0.652

Table 4 Simulation results for $p = 500$ and $\rho = 0.6$

	α	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
		snr = 0.5					snr = 1				
RTP	CTFR	0.627	0.287	0.296	0.537	0.841	0.897	0.256	0.348	0.521	0.816
	MCP	**1.111**	0.72	0.336	0.211	0.317	0.816	0.586	0.189	0.247	0.326
	SCAD	0.988	0.952	0.773	0.510	0.634	0.549	0.855	**1.161**	0.469	0.511
	aLasso	0.620	0.805	**1.524**	**2.227**	**1.65**	0.374	0.514	**1.974**	**2.874**	**1.142**
RFP	CTFR	0.077	0.099	0.151	0.166	0.346	0.211	0.330	0.302	0.178	0.101
	MCP	0.058	0.069	0.058	0.087	0.119	0.085	0.075	0.144	0.071	0.048
	SCAD	0.358	0.272	0.310	0.342	0.620	0.304	0.367	0.452	0.384	0.177
	aLasso	**1.356**	**1.254**	**1.647**	**1.336**	**1.766**	**1.531**	**1.354**	**1.774**	**1.136**	**1.202**
RPE	CTFR	**1.587**	**1.649**	0.908	**2.012**	**1.636**	**1.729**	0.497	0.879	**2.911**	**5.476**
	MCP	0.767	0.517	0.993	**1.208**	**1.172**	0.999	0.820	0.524	**1.247**	**3.192**
	SCAD	0.528	0.601	0.824	**2.327**	0.476	0.562	0.868	**1.435**	0.854	**1.517**
	aLasso	**1.190**	**1.901**	**2.819**	**2.828**	**2.869**	**1.028**	**1.669**	**3.480**	**3.521**	**1.190**
		snr = 2					snr = 5				
RTP	CTFR	0.788	0.238	0.260	0.602	**1.648**	0.682	0.080	0.324	0.748	0.993
	MCP	0.615	0.750	0.289	0.402	0.388	0.449	**1.261**	0.176	0.453	0.458
	SCAD	0.347	0.780	**1.700**	0.810	0.443	0.307	0.913	**2.080**	0.367	0.481
	aLasso	0.188	0.454	**2.308**	**5.012**	**1.283**	0.165	0.605	**1.681**	**5.852**	0.947
RFP	CTFR	0.192	0.174	0.144	0.064	0.316	0.033	0.003	0.195	0.054	0.103
	MCP	0.040	0.073	0.054	0.035	0.060	0.033	0.035	0.032	0.031	0.031
	SCAD	0.127	0.265	0.394	0.291	0.271	0.126	0.277	0.273	0.115	0.118
	aLasso	0.866	**1.121**	**1.753**	**1.703**	**1.599**	0.959	0.837	**1.701**	**1.407**	0.906
RPE	CTFR	**3.057**	**1.029**	**1.198**	**4.154**	**4.341**	**3.621**	0.957	**1.399**	**6.32**	**15.521**
	MCP	0.609	0.998	0.952	**1.473**	**1.617**	0.504	0.851	0.337	**1.413**	**7.398**
	SCAD	0.278	0.642	**2.473**	**1.827**	0.839	0.278	0.746	**1.738**	0.895	**3.342**
	aLasso	0.731	**1.362**	**3.509**	**5.013**	**1.509**	0.753	**1.445**	**2.473**	**3.838**	0.888
		snr = 10					snr = 20				
RTP	CTFR	0.515	0.196	0.233	0.578	**1.686**	1.000	0.139	0.252	0.409	0.930
	MCP	0.635	**1.515**	0.37	0.669	0.537	0.500	**2.000**	0.398	0.715	0.631
	SCAD	0.311	0.962	**2.949**	0.710	0.436	0.225	1.000	**4.290**	0.665	0.555
	aLasso	0.140	0.478	**2.051**	**7.000**	0.957	0.120	0.449	**2.215**	**8.080**	0.958
RFP	CTFR	0.000	0.120	0.009	0.039	0.328	0.017	0.011	0.011	0.014	0.070
	MCP	0.029	0.025	0.015	0.039	0.075	0.016	0.019	0.013	0.027	0.038
	SCAD	0.089	0.094	0.149	0.156	0.175	0.037	0.057	0.172	0.147	0.098
	aLasso	0.359	0.908	**1.576**	**2.354**	0.919	0.533	0.734	**1.401**	**2.641**	0.890
RPE	CTFR	**8.243**	**1.327**	**6.267**	**13.243**	**14.622**	**9.918**	**2.532**	**12.086**	**27.137**	**39.189**
	MCP	0.267	0.269	0.193	**1.420**	**3.765**	0.160	0.287	0.122	**1.539**	**9.805**
	SCAD	0.118	0.297	0.930	**1.110**	**2.468**	0.090	0.152	**1.187**	0.430	**4.786**
	aLasso	0.732	**1.070**	**3.361**	**4.406**	**1.028**	0.518	**1.208**	**2.533**	**5.022**	0.655

Table 5 Simulation results for $p = 500$ and $\rho = 0.8$

	α	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
		snr = 0.5					snr = 1				
RTP	CTFR	0.727	0.364	0.580	0.276	0.926	0.891	0.191	0.218	0.752	1.737
	MCP	0.185	0.818	0.227	0.140	0.161	0.860	0.855	0.093	0.183	0.381
	SCAD	0.230	0.796	**1.212**	0.227	**1.040**	0.476	0.842	**1.055**	0.350	0.341
	aLasso	0.600	0.678	**1.852**	**2.424**	**2.173**	0.234	0.590	**2.035**	**5.509**	**1.191**
RFP	CTFR	0.390	0.322	0.268	0.340	0.483	0.195	0.155	0.275	0.124	0.290
	MCP	0.110	0.068	0.092	0.067	0.068	0.024	0.036	0.065	0.042	0.021
	SCAD	0.265	0.306	0.396	0.412	0.416	0.047	0.146	0.483	0.171	0.123
	aLasso	**1.896**	**1.330**	**2.135**	**2.038**	**2.113**	**1.181**	0.976	**2.789**	**2.391**	**1.333**
RPE	CTFR	**1.826**	**1.741**	**3.041**	0.686	**1.304**	**2.472**	**1.015**	0.809	**2.897**	**3.735**
	MCP	**1.401**	**1.317**	**2.374**	**1.323**	0.844	0.409	0.613	0.994	**1.269**	**1.407**
	SCAD	0.674	**1.177**	**2.083**	0.889	**1.224**	0.253	0.572	**1.896**	**2.012**	0.639
	aLasso	**4.137**	**1.879**	**4.102**	**3.505**	**4.485**	0.843	**1.471**	**4.418**	**5.314**	**1.850**
		snr = 2					snr = 5				
RTP	CTFR	0.850	0.249	0.163	0.213	0.688	0.515	0.289	0.185	0.629	**1.619**
	MCP	0.351	**1.633**	0.588	0.332	0.130	0.730	0.598	0.263	0.402	0.404
	SCAD	0.217	0.786	3.033	0.763	0.215	0.636	0.659	**1.134**	0.735	0.290
	aLasso	0.148	0.542	**1.799**	**7.500**	**1.163**	0.155	0.768	**1.452**	**6.216**	**1.237**
RFP	CTFR	0.057	0.022	0.007	0.013	0.314	0.000	0.456	0.025	0.033	0.460
	MCP	0.015	0.088	0.052	0.035	0.046	0.035	0.098	0.028	0.028	0.120
	SCAD	0.048	0.141	0.185	0.138	0.184	0.085	0.278	0.081	0.107	0.165
	aLasso	0.516	0.763	**1.174**	**2.276**	**1.425**	0.336	**2.441**	**1.227**	**2.419**	**1.512**
RPE	CTFR	**2.954**	**3.133**	**1.966**	**4.271**	**2.185**	**5.646**	**1.478**	**1.187**	**13.083**	**5.675**
	MCP	0.388	**1.844**	**1.661**	**1.701**	**1.042**	0.621	**1.530**	0.627	**1.834**	**2.417**
	SCAD	0.224	0.983	**3.186**	**3.348**	0.668	0.272	**1.467**	**1.351**	**3.054**	0.942
	aLasso	0.574	**1.201**	**2.521**	**5.198**	**1.854**	0.579	**1.684**	**2.702**	**4.246**	**1.425**
		snr = 10					snr = 20				
RTP	CTFR	0.825	0.151	0.160	0.323	0.986	0.725	0.092	0.146	0.367	0.699
	MCP	0.392	**1.587**	0.247	0.554	0.695	0.355	**1.912**	0.184	0.547	**1.097**
	SCAD	0.176	0.937	**3.062**	0.538	0.386	0.227	0.928	**2.712**	0.478	0.442
	aLasso	0.100	0.416	**2.350**	**9.600**	0.938	0.094	0.601	**1.633**	**10.325**	0.965
RFP	CTFR	0.108	0.167	0.001	0.005	0.121	0.052	0.001	0.037	0.003	0.037
	MCP	0.018	0.068	0.017	0.064	0.067	0.024	0.023	0.029	0.030	0.056
	SCAD	0.028	0.094	0.118	0.078	0.070	0.050	0.029	0.138	0.060	0.053
	aLasso	0.581	0.843	**1.314**	**3.154**	0.866	0.586	0.594	**1.272**	**2.431**	0.857
RPE	CTFR	**7.220**	0.955	**6.125**	**19.082**	**27.456**	**8.939**	**1.910**	**3.453**	**38.43**	**62.878**
	MCP	0.515	**1.482**	0.389	**2.041**	**4.982**	0.474	0.630	0.158	**1.997**	**17.944**
	SCAD	0.233	0.795	**3.028**	**1.335**	**2.709**	0.312	0.347	**1.947**	0.684	**7.063**
	aLasso	0.550	**1.183**	**2.885**	**6.099**	0.971	0.574	**1.406**	**2.095**	**5.125**	0.607

Table 6 Simulation results for $p = 1000$ and $\rho = 0$

	α	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
		snr = 0.5					snr = 1				
RTP	CTFR	0.759	0.626	0.556	0.459	0.368	0.735	0.331	0.299	0.770	**1.039**
	MCP	0.481	**1.379**	0.929	0.563	0.215	**1.197**	0.704	0.669	0.692	0.587
	SCAD	0.452	**1.009**	**2.086**	**1.091**	0.476	0.778	**1.013**	**1.276**	**1.364**	0.564
	aLasso	0.468	0.807	**1.292**	**2.103**	**1.081**	0.419	0.595	**1.645**	**2.378**	**1.214**
RFP	CTFR	0.057	0.204	0.244	0.118	0.196	0.033	0.022	0.030	0.128	0.192
	MCP	0.174	0.342	0.446	0.304	0.058	0.206	0.192	0.361	0.233	0.188
	SCAD	0.453	0.896	0.935	0.945	0.439	0.687	0.734	0.836	**1.327**	0.337
	aLasso	**1.175**	0.675	**1.416**	0.806	0.998	0.724	0.843	**1.166**	**1.262**	**1.229**
RPE	CTFR	**1.491**	0.888	0.900	**1.231**	**1.387**	**2.105**	0.648	**1.109**	**2.353**	**4.058**
	MCP	0.335	**1.591**	0.660	**1.008**	**1.225**	0.480	0.518	0.316	0.983	**2.523**
	SCAD	0.337	0.975	**1.514**	0.938	**1.018**	0.415	0.630	**1.048**	0.642	**1.475**
	aLasso	**1.035**	**1.555**	**1.870**	**3.003**	**1.320**	**1.022**	**1.755**	**2.943**	**3.191**	**1.094**
		snr = 2					snr = 5				
RTP	CTFR	0.735	0.213	0.326	0.582	0.662	0.530	0.085	0.257	0.503	**2.944**
	MCP	0.444	**1.809**	0.671	0.620	0.470	**1.796**	0.510	0.566	0.772	0.746
	SCAD	0.259	0.985	**3.912**	0.914	0.685	0.287	0.944	**3.590**	0.776	0.866
	aLasso	0.183	0.526	**1.902**	**5.603**	0.931	0.196	0.158	**6.278**	**4.970**	0.952
RFP	CTFR	0	0.001	0.011	0.130	0.120	0.000	0.000	0.005	0.030	0.239
	MCP	0.053	0.392	0.242	0.162	0.177	0.351	0.122	0.213	0.157	0.117
	SCAD	0.121	0.692	**2.271**	0.580	0.438	0.248	0.585	**1.423**	0.636	0.383
	aLasso	0.326	0.837	**1.078**	**2.730**	0.933	0.418	0.319	**3.034**	**2.428**	0.914
RPE	CTFR	**4.262**	0.837	**1.353**	**3.693**	**5.560**	**5.893**	0.768	**2.886**	**5.751**	**28.943**
	MCP	0.226	**1.057**	0.269	**1.239**	**3.184**	0.763	0.292	0.263	**1.152**	**4.738**
	SCAD	0.146	0.627	**1.982**	0.951	**1.324**	0.226	0.55	**1.904**	0.619	**2.354**
	aLasso	0.431	**1.336**	**2.230**	**5.961**	**1.046**	0.546	0.748	**5.573**	**4.379**	0.809
		snr = 10					snr = 20				
RTP	CTFR	0.980	0.143	0.243	0.625	**1.079**	1.000	0.200	0.375	0.289	0.965
	MCP	0.329	**2.922**	0.441	0.886	**1.262**	0.505	**1.990**	0.599	0.802	0.923
	SCAD	0.156	0.987	**6.314**	0.914	0.956	0.220	**1.005**	**4.640**	0.768	**1.045**
	aLasso	0.083	0.472	**2.099**	**11.941**	0.964	0.168	0.436	**2.268**	**5.840**	0.947
RFP	CTFR	0.000	0.000	0.000	0.032	0.051	0.000	0.000	0.002	0.002	0.044
	MCP	0.091	0.166	0.170	0.129	0.170	0.108	0.176	0.116	0.134	0.141
	SCAD	0.098	0.484	**1.683**	0.571	0.375	0.109	0.509	0.947	0.446	0.416
	aLasso	0.253	0.615	**1.487**	**3.27**	0.873	0.432	0.576	**1.400**	**2.026**	0.890
RPE	CTFR	**9.020**	**1.403**	**3.391**	**13.554**	**28.543**	**15.614**	**2.076**	**9.076**	**13.473**	**41.154**
	MCP	0.164	0.438	0.132	**1.117**	**6.171**	0.143	0.443	0.115	**1.044**	**7.410**
	SCAD	0.073	0.440	**1.498**	0.519	**2.170**	0.067	0.404	**1.362**	0.308	**3.708**
	aLasso	0.361	**1.159**	**2.822**	**5.423**	0.691	0.653	**1.039**	**2.358**	**3.825**	0.540

Table 7 Simulation results for $p = 1000$ and $\rho = 0.2$

	α	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
		snr = 0.5					snr = 1				
RTP	CTFR	0.885	0.447	0.604	0.566	0.581	0.926	0.447	0.411	0.462	0.579
	MCP	0.581	**1.500**	0.582	0.414	0.475	0.376	**2.370**	0.941	0.598	0.401
	SCAD	0.515	0.965	**2.038**	0.589	**1.072**	0.274	**1.023**	**3.796**	**1.082**	0.548
	aLasso	0.468	0.970	**1.337**	**2.404**	**1.326**	0.247	0.675	**1.534**	**4.593**	**1.165**
RFP	CTFR	0.213	0.106	0.268	0.267	0.330	0.028	0.075	0.085	0.158	0.170
	MCP	0.213	0.428	0.186	0.143	0.339	0.079	0.327	0.322	0.214	0.122
	SCAD	0.641	0.804	**1.324**	0.416	**1.010**	0.407	0.670	**1.114**	**1.070**	0.291
	aLasso	0.940	**2.095**	**1.501**	**1.963**	**1.524**	0.784	**1.124**	0.986	**2.347**	**1.296**
RPE	CTFR	**1.983**	**1.033**	0.914	**1.327**	**1.822**	**2.179**	**1.011**	**1.564**	**1.797**	**2.534**
	MCP	0.348	**1.478**	0.939	0.988	**1.043**	0.171	**1.007**	0.564	**1.090**	**1.741**
	SCAD	0.291	0.903	**2.507**	**1.283**	0.693	0.201	0.671	**1.998**	0.700	**1.506**
	aLasso	0.897	**1.536**	**2.567**	**4.958**	**1.983**	0.984	**1.510**	**1.950**	**3.868**	**1.115**
		snr = 2					snr = 5				
RTP	CTFR	0.820	0.299	0.474	0.612	0.944	0.505	0.095	0.210	0.879	0.955
	MCP	0.794	**1.06**	0.761	0.678	0.616	0.752	**1.232**	0.283	0.730	**1.226**
	SCAD	0.380	0.937	**2.710**	0.855	0.941	0.503	0.962	**1.899**	0.842	0.759
	aLasso	0.329	0.456	**2.063**	**3.240**	**1.186**	0.191	0.693	**1.617**	**5.152**	**1.005**
RFP	CTFR	0.008	0.066	0.049	0.128	0.143	0.000	0.000	0.006	0.106	0.112
	MCP	0.122	0.191	0.305	0.182	0.151	0.157	0.245	0.086	0.143	0.207
	SCAD	0.358	0.552	**1.109**	0.757	0.589	0.405	0.624	0.680	0.548	0.358
	aLasso	0.825	**1.207**	**1.252**	**1.763**	**1.450**	0.496	**1.002**	**1.860**	**2.350**	0.959
RPE	CTFR	**3.771**	**1.030**	**2.273**	**2.335**	**4.520**	**5.604**	0.933	**1.905**	**8.132**	**14.42**
	MCP	0.239	0.653	0.537	**1.138**	**1.636**	0.276	0.841	0.216	**1.069**	**5.701**
	SCAD	0.174	0.545	**1.648**	0.870	**1.190**	0.268	0.765	**1.376**	0.586	**2.383**
	aLasso	**1.034**	**1.239**	**2.743**	**3.355**	**1.140**	0.657	**1.357**	**2.453**	**3.867**	0.762
		snr = 10					snr = 20				
RTP	CTFR	1.000	0.165	0.403	0.579	**1.224**	0.500	0.133	0.289	0.525	0.780
	MCP	0.641	**1.460**	0.569	0.803	0.449	0.524	**1.870**	0.480	0.829	0.796
	SCAD	0.255	0.949	**3.950**	0.707	0.771	0.268	**1.010**	**3.780**	0.767	0.836
	aLasso	0.212	0.408	**2.487**	**4.770**	0.994	0.169	0.512	**1.932**	**5.88**	0.969
RFP	CTFR	0.001	0.001	0.027	0.061	0.156	0.000	0.000	0.004	0.042	0.067
	MCP	0.055	0.180	0.115	0.147	0.072	0.110	0.176	0.097	0.116	0.101
	SCAD	0.129	0.429	0.890	0.453	0.368	0.258	0.587	0.786	0.404	0.333
	aLasso	0.783	**1.119**	**1.452**	**2.005**	0.966	0.842	0.631	**1.492**	**1.922**	0.904
RPE	CTFR	**8.882**	0.78	**4.895**	**6.526**	**24.055**	**10.489**	**1.118**	**6.953**	**14.263**	**49.102**
	MCP	0.115	0.729	0.127	**1.010**	**11.007**	0.285	0.612	0.05	**1.025**	**17.04**
	SCAD	0.089	0.628	**1.562**	0.265	**5.789**	0.195	0.483	0.843	0.227	**6.902**
	aLasso	0.618	**1.139**	**2.416**	**3.732**	0.591	0.559	**1.243**	**2.278**	**3.743**	0.524

Table 8 Simulation results for $p = 1000$ and $\rho = 0.4$

	α	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
		snr = 0.5					snr = 1				
RTP	CTFR	0.804	0.600	0.413	0.693	0.288	0.839	0.377	0.534	0.662	**1.049**
	MCP	0.538	**1.625**	0.678	0.433	0.307	0.505	**1.500**	0.503	0.437	0.531
	SCAD	0.853	0.952	**1.393**	**1.033**	0.673	0.438	**1.039**	**1.887**	0.814	0.908
	aLasso	0.606	**1.413**	0.740	**2.339**	**1.678**	0.362	0.908	**1.379**	**3.161**	**1.344**
RFP	CTFR	0.096	0.184	0.115	0.168	0.175	0.084	0.141	0.138	0.355	0.340
	MCP	0.122	0.226	0.257	0.169	0.196	0.056	0.133	0.137	0.209	0.225
	SCAD	0.631	0.562	0.833	0.668	0.711	0.308	0.468	0.647	0.748	0.724
	aLasso	**1.351**	**1.388**	**1.011**	**1.380**	**2.037**	**1.612**	**1.971**	**1.744**	**2.607**	**1.898**
RPE	CTFR	**1.721**	**1.083**	0.922	**1.685**	**1.478**	**1.923**	**1.727**	**1.795**	**1.785**	**1.637**
	MCP	0.335	**1.201**	0.644	0.992	**1.436**	0.193	**1.078**	0.935	**1.193**	**1.268**
	SCAD	0.389	0.712	**1.309**	**1.115**	0.846	0.198	0.661	**1.539**	0.931	**1.092**
	aLasso	**1.256**	**2.169**	**1.840**	**3.910**	**1.692**	**1.965**	**1.958**	**2.004**	**4.032**	**2.062**
		snr = 2					snr = 5				
RTP	CTFR	0.610	0.369	0.235	0.600	0.538	0.797	0.179	0.431	0.627	0.948
	MCP	0.503	**1.86**	0.439	0.477	0.432	0.273	**2.922**	0.487	0.636	0.448
	SCAD	0.526	0.959	**1.810**	0.833	0.514	0.204	0.995	**4.031**	0.571	0.911
	aLasso	0.310	**1.037**	0.954	**3.290**	**1.167**	0.208	0.699	**1.418**	**5.328**	**1.157**
RFP	CTFR	0.012	0.019	0.010	0.111	0.065	0.004	0.003	0.009	0.083	0.212
	MCP	0.045	0.126	0.117	0.104	0.063	0.019	0.169	0.073	0.09	0.085
	SCAD	0.386	0.315	0.488	0.624	0.252	0.064	0.418	0.791	0.336	0.434
	aLasso	0.826	**1.576**	0.613	**1.155**	**1.224**	**1.234**	0.776	**1.235**	**2.666**	**1.446**
RPE	CTFR	**3.331**	0.968	**1.354**	**4.099**	**3.961**	**5.762**	**1.190**	**4.086**	**3.541**	**8.023**
	MCP	0.164	0.737	0.300	**1.300**	**4.057**	0.177	**1.029**	0.212	**1.137**	**4.446**
	SCAD	0.200	0.448	**1.241**	0.795	**1.746**	0.124	0.457	**2.027**	0.449	**3.143**
	aLasso	0.766	**2.062**	**1.619**	**3.412**	0.879	0.784	**1.297**	**1.976**	**5.084**	0.659
		snr = 10					snr = 20				
RTP	CTFR	0.760	0.266	0.295	0.449	**1.234**	1.000	0.218	0.199	0.819	0.625
	MCP	0.543	**1.670**	0.668	0.776	0.598	0.500	**2.000**	0.403	0.699	0.734
	SCAD	0.245	0.946	**4.070**	0.963	0.669	0.292	1.000	**3.190**	0.544	0.984
	aLasso	0.169	0.453	**2.196**	**5.750**	**1.105**	0.223	0.585	**1.695**	**4.740**	0.964
RFP	CTFR	0.000	0.001	0.011	0.013	0.136	0.000	0.000	0.004	0.219	0.042
	MCP	0.027	0.063	0.080	0.053	0.052	0.043	0.032	0.073	0.069	0.065
	SCAD	0.067	0.147	0.638	0.337	0.238	0.094	0.150	0.567	0.282	0.261
	aLasso	0.358	0.630	**1.369**	**2.511**	**1.330**	0.534	0.739	**1.377**	**2.538**	0.929
RPE	CTFR	**9.362**	**2.832**	**5.703**	**10.875**	**10.578**	**17.463**	**1.954**	**5.972**	**6.616**	**43.547**
	MCP	0.089	0.599	0.287	0.996	**3.081**	0.142	0.211	0.127	**1.025**	**14.857**
	SCAD	0.084	0.348	**1.510**	0.552	**1.855**	0.080	0.190	**1.500**	0.212	**8.766**
	aLasso	0.550	**1.100**	**2.085**	**3.886**	0.702	0.779	**1.208**	**2.117**	**3.327**	0.533

Table 9 Simulation results for $p = 1000$ and $\rho = 0.6$

	α	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
		snr = 0.5					snr = 1				
RTP	CTFR	0.640	0.234	0.606	0.333	**1.111**	0.942	0.295	0.560	0.489	**1.172**
	MCP	**2.167**	0.135	0.449	0.212	0.158	0.781	**1.058**	0.337	0.213	0.259
	SCAD	0.754	0.861	0.978	0.505	0.485	0.370	0.984	**1.788**	0.554	0.716
	aLasso	0.929	0.316	**3.556**	**1.697**	**1.140**	0.369	0.526	**2.438**	**4.000**	1.488
RFP	CTFR	0.084	0.651	0.113	0.274	0.186	0.071	0.030	0.322	0.231	0.289
	MCP	0.063	0.116	0.098	0.079	0.039	0.037	0.033	0.117	0.093	0.052
	SCAD	0.353	0.668	0.278	0.745	0.150	0.141	0.285	0.471	0.613	0.345
	aLasso	0.935	**2.771**	**1.444**	**2.399**	**1.363**	**1.779**	**1.265**	**3.143**	**2.200**	**1.985**
RPE	CTFR	**1.491**	0.556	**1.456**	0.984	**4.102**	**1.850**	0.752	**1.252**	**2.103**	**4.051**
	MCP	**1.19**	0.815	0.841	**1.145**	**1.405**	0.253	0.546	0.661	**1.382**	**2.295**
	SCAD	0.344	**1.148**	**1.430**	0.555	**1.706**	0.185	0.484	**1.821**	0.815	**1.574**
	aLasso	**1.847**	**1.598**	**4.677**	**2.539**	**1.432**	**1.353**	**1.716**	**3.678**	**3.777**	**1.204**
		snr = 2					snr = 5				
RTP	CTFR	0.840	0.158	0.320	0.610	**1.821**	1.000	0.300	0.315	0.403	0.540
	MCP	**1.071**	0.667	0.441	0.270	0.300	0.495	**1.970**	0.612	0.525	0.332
	SCAD	0.340	0.893	**2.240**	0.450	0.504	0.270	0.995	**3.510**	0.614	0.483
	aLasso	0.273	0.280	**3.750**	**3.840**	**1.196**	0.202	0.541	**1.830**	**4.910**	0.958
RFP	CTFR	0.181	0.062	0.024	0.280	0.223	0.093	0.066	0.014	0.020	0.053
	MCP	0.026	0.029	0.047	0.067	0.058	0.024	0.010	0.031	0.033	0.028
	SCAD	0.197	0.138	0.343	0.393	0.215	0.084	0.073	0.245	0.209	0.145
	aLasso	**1.687**	**1.710**	**1.550**	**2.162**	**1.528**	0.947	0.756	**1.308**	**1.632**	0.998
RPE	CTFR	**2.758**	0.522	**2.149**	**2.399**	**8.809**	**3.585**	**1.405**	**3.76**	**6.616**	**11.561**
	MCP	0.581	0.310	0.385	**1.558**	**3.064**	0.167	0.316	0.209	**1.587**	**4.892**
	SCAD	0.249	0.287	**1.802**	0.783	**2.235**	0.105	0.220	**1.320**	0.821	**2.827**
	aLasso	**1.189**	**1.446**	**4.137**	**4.281**	0.963	0.706	**1.135**	**2.268**	**4.001**	0.759
		snr = 10					snr = 20				
RTP	CTFR	1.000	0.293	0.245	0.473	**1.411**	1.000	0.167	0.284	0.404	**2.381**
	MCP	0.613	**1.510**	0.415	0.570	0.566	0.952	1.000	0.547	0.563	0.406
	SCAD	0.388	0.939	**2.31**	0.846	0.421	0.257	0.962	**3.610**	0.896	0.393
	aLasso	0.166	0.624	**1.571**	**5.810**	**1.118**	0.138	0.270	**3.676**	**7.130**	**1.080**
RFP	CTFR	0.000	0.180	0.043	0.020	0.282	0.008	0.022	0.048	0.017	0.191
	MCP	0.021	0.018	0.029	0.027	0.055	0.036	0.020	0.014	0.028	0.022
	SCAD	0.064	0.076	0.246	0.199	0.149	0.064	0.123	0.180	0.174	0.123
	aLasso	0.361	**1.757**	**1.318**	**2.505**	**1.386**	0.866	0.908	**1.974**	**1.721**	**1.116**
RPE	CTFR	**12.338**	**1.191**	**2.573**	**16.630**	**10.936**	**8.255**	0.762	**6.301**	**23.682**	**34.628**
	MCP	0.161	0.265	0.228	**1.703**	**4.286**	0.360	0.291	0.150	**2.436**	**8.167**
	SCAD	0.112	0.219	**1.008**	**1.353**	**1.777**	0.132	0.368	**1.246**	**1.142**	**3.267**
	aLasso	0.509	**1.614**	**2.451**	**3.982**	0.807	0.461	**1.050**	**3.556**	**4.045**	0.512

Table 10 Simulation results for $p = 1000$ and $\rho = 0.8$

	α	0.1	0.3	0.5	0.7	0.9	0.1	0.3	0.5	0.7	0.9
		snr = 0.5					snr = 1				
RTP	CTFR	0.582	0.268	0.697	0.170	**1.033**	0.833	0.511	0.197	1.000	0.655
	MCP	**2.733**	0.143	0.427	0.091	0.107	0.483	**1.062**	0.223	0.202	0.164
	SCAD	0.759	0.633	0.505	0.293	0.576	0.745	0.879	0.875	0.691	0.197
	aLasso	**1.470**	0.286	**4.033**	**1.044**	**2.012**	0.202	**1.127**	**1.328**	**5.104**	**1.638**
RFP	CTFR	0.065	0.468	0.096	0.310	0.309	0.278	0.067	0.228	0.159	0.340
	MCP	0.036	0.037	0.040	0.056	0.048	0.046	0.027	0.060	0.018	0.118
	SCAD	0.064	0.403	0.073	0.402	0.206	0.377	0.054	0.264	0.097	0.299
	aLasso	0.991	**1.968**	**1.305**	**1.972**	**2.809**	**1.341**	**2.514**	**1.936**	**1.401**	**2.072**
RPE	CTFR	**1.502**	**1.170**	**1.705**	0.667	**2.097**	**1.834**	**2.234**	0.477	**6.729**	**1.310**
	MCP	**1.150**	0.610	**2.362**	0.911	0.628	0.840	0.647	**1.513**	**1.281**	**1.856**
	SCAD	0.398	0.988	**1.710**	**1.042**	0.734	**1.031**	0.347	**1.092**	**4.971**	0.274
	aLasso	**2.057**	**1.376**	**4.975**	**2.465**	**2.898**	0.814	**3.821**	**2.565**	**5.006**	**3.550**
		snr = 2					snr = 5				
RTP	CTFR	0.906	0.382	0.383	0.508	**1.103**	0.825	0.140	0.623	0.939	**2.294**
	MCP	0.552	0.938	0.491	0.336	0.230	0.735	0.825	0.229	0.353	0.521
	SCAD	0.272	0.828	**2.094**	0.382	0.510	0.307	0.794	**1.905**	0.355	0.66
	aLasso	0.409	0.487	**2.322**	**2.875**	**1.467**	0.288	0.485	**2.765**	**4.016**	**1.101**
RFP	CTFR	0.268	0.285	0.025	0.204	0.188	0.143	0.018	0.283	0.281	0.262
	MCP	0.025	0.029	0.033	0.043	0.037	0.009	0.015	0.036	0.035	0.036
	SCAD	0.097	0.147	0.083	0.227	0.126	0.051	0.046	0.248	0.159	0.146
	aLasso	**1.692**	**1.414**	**1.431**	**2.054**	**1.93**	**1.564**	**1.151**	**2.919**	**2.468**	**1.227**
RPE	CTFR	**2.688**	0.926	**3.577**	**1.259**	**4.694**	**3.573**	**1.127**	**2.03**	**2.677**	**6.923**
	MCP	0.560	**1.216**	0.900	**1.235**	**1.378**	0.307	0.618	0.599	**1.491**	**2.622**
	SCAD	0.254	**1.053**	**2.655**	0.725	**1.873**	0.197	0.534	**2.194**	0.813	**2.224**
	aLasso	**1.869**	**1.158**	**3.204**	**4.027**	**1.362**	**1.414**	**1.723**	**3.468**	**4.318**	**1.200**
		snr = 10					snr = 20				
RTP	CTFR	0.633	0.388	0.194	0.212	**1.217**	0.680	0.227	0.240	0.376	**1.617**
	MCP	0.416	**1.127**	0.829	0.419	0.207	0.599	**1.400**	0.813	0.525	0.163
	SCAD	0.207	0.807	**4.266**	0.898	0.231	0.243	0.898	**3.760**	0.712	0.395
	aLasso	0.127	0.418	**2.255**	**8.316**	**1.048**	0.170	0.417	**2.431**	**5.820**	**1.041**
RFP	CTFR	0.000	0.298	0.009	0.006	0.503	0.000	0.030	0.017	0.063	0.534
	MCP	0.020	0.122	0.017	0.025	0.046	0.011	0.020	0.015	0.029	0.071
	SCAD	0.046	0.226	0.084	0.074	0.167	0.012	0.044	0.039	0.054	0.283
	aLasso	0.300	**1.106**	**1.875**	**3.584**	**1.277**	0.488	0.755	**1.424**	**2.096**	**1.264**
RPE	CTFR	**6.281**	**1.475**	**6.634**	**15.067**	**6.288**	**16.466**	**3.31**	**12.632**	**13.979**	**5.139**
	MCP	0.688	**1.860**	0.435	**2.743**	**1.850**	0.132	0.796	0.443	**1.520**	**1.732**
	SCAD	0.265	**1.516**	**1.271**	**3.562**	**1.096**	0.065	0.450	**1.012**	**2.413**	**1.076**
	aLasso	0.544	**1.187**	**3.235**	**4.389**	**1.295**	0.580	**1.044**	**2.128**	**3.466**	**1.353**

3.3 Observations

In this section, we briefly summarize our simulation study in three aspects.

3.3.1 In terms of True Positives

- We see from simulation results that the performance of Lasso is superior to CTFR, MCP, and SCAD in terms of selecting true parameters when the true model is very sparse, the signal-to-noise ratio is low, and the predictors are uncorrelated.
- When the predictors are uncorrelated, as the model gets less sparse (for larger α) the TP performance of CTFR improves and in some cases outperforms Lasso.
- As the number of predictors increases, the true positive value of Lasso and MCP decreases. To put it differently, the variable selection performance of Lasso and MCP decreases under the higher dimensional settings.
- Overall, adaptive Lasso had better performance in variable selection compared to Lasso, but its performance was still inferior to that of CTFR and MCP.

3.3.2 In terms of False Positives

- CTFR and MCP outperformed Lasso in terms of FP under any sparsity, signal-to-noise ratio, and the correlation between predictors cases. This reflected the fact that a cross-validated Lasso tends to over select.
- SCAD also outperformed Lasso in terms of FP unless signal-to-noise ratio and the correlation between predictors are high. Therefore, for a two-stage variable selection procedure, applying CTFR or MCP in the second stage after applying Lasso in the first stage would help to eliminate the noise variables selected by Lasso.
- Under the presence of high correlation between predictors, FP of MCP was smaller than that of CTFR. In other words, MCP had better performance in screening out irrelevant variables than CTFR when predictors were highly correlated to each other.

3.3.3 In terms of Prediction Errors

- Lasso outperformed CTFR and aLasso under almost all simulation set-ups.
- Lasso's prediction performance is significantly better than CTFR as the model became less sparse, with higher signal-to-noise ratio, and greater correlation value in the design.

– In real data examples it is usually assumed that there are some signals in the data and predictors are not independent to each other. For this parameter range ($\rho = 0.2$ and snr = 2 or 5), MCP exhibits a better performance than Lasso in terms of false positive selection and prediction error performance.

3.4 Non-normal Error Distributions

In this section, we compare the variable selection and prediction performance of the Lasso and CTFR when the errors in the linear model, with $n = 100$ and $p = 500$, have a heavier tail. We only consider Cauchy and t-distribution with a degree of freedom 2.

In Table 11, we summarize the simulation results for some selected parameters when the errors come from Cauchy distribution. We observe that CTFR is no longer better than Lasso in terms of FP regardless of the sparsity level of model, signal-to-noise ratio and correlation level among predictors. On the other hand, it outperforms Lasso in terms of variable selection performance. CTFR has much lower prediction error than that of Lasso in all cases except the case that the model is very sparse and has high signal-to-noise ratio.

In Table 12, we summarize the simulation results for some selected parameters when the errors come from t-distribution with two degrees of freedom. When we compare the results in these two tables, we see that the prediction errors significantly decrease as the tail properties of the error distribution get closer to the normal distribution. In various combinations of sparsity level and the value of correlation among predictors, as long as signal-to-noise ratio increases, CTFR performs better than Lasso in terms of false positive selection; that is, it selects fewer noise variables.

Table 11 Errors in the linear models are Cauchy distributed

	Cauchy distribution					
	True Positive		False Positive		Prediction Error	
(α,ρ,snr)	Lasso	CTFR	Lasso	CTFR	Lasso	CTFR
(0.1,0.0,0.5)	0.08	0.34	3.10	56.22	2872	122
(0.1,0.0,20)	0.90	1.44	5.66	48.92	19, 632	70, 573
(0.1,0.8,0.5)	0.00	0.26	2.64	65.80	32, 902	488
(0.1,0.8,20)	0.46	0.90	4.76	55.54	1, 115, 701	21, 232
(0.5,0.0,0.5)	0.12	1.16	5.00	51.40	1662	108
(0.5,0.0,20)	1.26	2.52	7.12	40.08	1499	116
(0.5,0.8,0.5)	0.12	1.40	2.74	63.06	50, 765	986
(0.5,0.8,20)	0.42	1.66	6.42	54.50	6871	168

Table 12 Errors in the linear models are t-distributed with two degrees of freedom

	t distribution with two degrees of freedom					
	True Positive		False Positive		Prediction Error	
(α,ρ,snr)	Lasso	CTFR	Lasso	CTFR	Lasso	CTFR
(0.1,0.0,0.5)	0.56	0.84	10.34	18.26	6.95	3.89
(0.1,0.0,20)	1.98	2.00	14.00	4.68	6.51	8.54
(0.1,0.8,0.5)	0.18	0.34	5.38	31.44	10.87	3.81
(0.1,0.8,20)	1.16	1.02	13.28	11.24	10.8	6.32
(0.5,0.0,0.5)	0.72	1.26	7.66	21.40	6.78	4.08
(0.5,0.0,20)	6.02	4.28	25.24	5.26	16.31	14.38
(0.5,0.8,0.5)	0.40	0.64	10.40	27.38	8.79	3.62
(0.5,0.8,20)	2.60	2.22	16.60	12.30	6.75	6.03

4 Real Data Examples

In this section we apply the regression methods considered in this paper to two real data sets and summarize their results.

4.1 Prostate Data

In this section, we provide an application of each regression methods we studied in this paper on prostate cancer data. The data come from a study conducted by Stamey et al. [25]. The predictors in the data are log of cancer volume (lcavol), log of prostate weight (lweight), age, log of benign prostatic hyperplasia amount (lbph), seminal vesicle invasion (svi), log of capsular penetration (lcp), Gleason score (gleason), and percentage Gleason scores 4 or 5 (pgg45). The response of interest is log of prostate specific antigen (lpsa). The total number of observations in the dataset is 97. All the regression methods were applied after standardizing predictors.

In Table 13, we report estimated coefficients by different methods for the prostate data. Further, the average prediction errors (APEs) are shown therein. According to these results, the CTFR picks up four response variables; the Lasso, SCAD, and MCP do not eliminate any response variables but aLasso eliminates only one. We also observe that the aLasso has the minimum APE and then CTFR, MCP, Lasso, and SCAD follow, respectively. From Fig. 1, it looks like error distributions are symmetric, and inter-quartile ranges are almost the same for each method.

Table 13 Estimated coefficients and APEs for prostate data

	Lasso	CTFR	SCAD	MCP	aLasso
(Intercept)	2.478	2.478	2.478	2.478	2.478
lcavol	0.655	0.512	0.670	0.665	0.647
lweight	0.263	0.090	0.264	0.266	0.265
age	−0.148	0.000	−0.156	−0.158	−0.118
lbph	0.135	0.000	0.141	0.140	0.115
svi	0.305	0.115	0.312	0.315	0.276
lcp	−0.124	0.000	−0.147	−0.148	−0.044
gleason	0.032	0.000	0.005	0.036	0.067
pgg45	0.116	0.000	0.146	0.126	0.000
APE	0.589	0.527	0.593	0.574	0.511

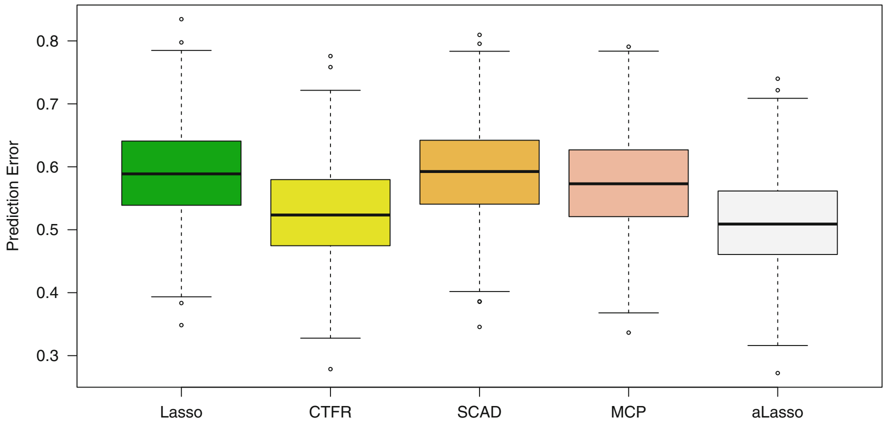

Fig. 1 Comparison of the estimators through prediction errors for prostate data

4.2 *Production of Riboflavin in Bacillus subtilis Data*

In [4], Bühlmann et al. made a data set on riboflavin production with *Bacillus subtilis* publicly available. In the data set, the logarithm of the riboflavin production rate is considered as the response variable corresponding to 4088 predictors which measure the logarithm of the expression level of 4088 genes. There are $n = 71$ samples. Therefore the design matrix for the dataset is $\mathbf{X} \in \mathbb{R}^{71\times 4088}$. We performed Lasso, SCAD, MCP, CTFR, aLasso to select a small subset of genes as the most important predictors for the model.

In Table 14, for each penalty estimator, we list the top 20 genes. Magnitude (absolute value) of the coefficient estimate was accompanied by each selected gene. For instance, in the case of CTFR, the absolute value of the coefficient estimate for the gene XHLA is 0.18. The table displays genes in a decreasing order with respect to their corresponding estimated values. As one can see, Lasso, SCAD, and aLasso selected more than 20 genes, while CTFR and MCP only selected 7 and 8 genes,

Table 14 Top 20 genes selected by each regression method from riboflavin data and their corresponding estimates

Lasso		CTFR		SCAD		MCP		aLasso	
gene	$\hat{\beta}$	gene	$\hat{\beta}$	gene	$\hat{\beta}$	gene	$\hat{\beta}$	alasso-gene	$\hat{\beta}$
YOAB	0.81	YOAB	0.23	YOAB	1.41	YOAB	1.45	YXLE	1.74
YEBC	0.53	XHLA	0.18	YHDZ	0.84	YHDZ	0.99	YXLF	1.25
LYSC	0.30	YXLD	0.13	SPOVAA	0.58	YXLD	0.46	YOAC	0.91
SPOVAA	0.26	YCKE	0.09	YEBC	0.51	SPOVAA	0.44	AADK	0.56
YQJU	0.23	LYSC	0.02	YXLD	0.44	CARA	0.39	PRKA	0.55
YXLD	0.22	YDAR	0.01	ARGF	0.36	XHLA	0.25	SPOVAB	0.54
YCLB	0.19	XTRA	0.01	XHLB	0.22	YSHB	0.06	YDDM	0.51
ARGF	0.19	YCGN	0.00	YTET	0.11	YEBC	0.04	YQJV	0.47
XHLB	0.16	AADK	0.00	YDDJ	0.08	AADK	0.00	YMFF	0.45
YFHE	0.15	AAPA	0.00	YQJU	0.08	AAPA	0.00	YYDB	0.44
YFIO	0.15	ABFA	0.00	YESJ	0.06	ABFA	0.00	PKSB	0.41
YHDS	0.14	ABH	0.00	YACN	0.06	ABH	0.00	YCLG	0.38
DNAJ	0.14	ABNA	0.00	YVDI	0.05	ABNA	0.00	ARGG	0.38
YBFI	0.14	ABRB	0.00	PTA	0.05	ABRB	0.00	YFIQ	0.36
YDDK	0.12	ACCA	0.00	YJCL	0.05	ACCA	0.00	YCLC	0.33
YKBA	0.11	ACCB	0.00	SPOIIAA	0.03	ACCB	0.00	YKVK	0.25
YYDA	0.11	ACCC	0.00	YHDS	0.02	ACCC	0.00	YCGP	0.25
PRIA	0.10	ACDA	0.00	YQIQ	0.02	ACDA	0.00	YLXX	0.24
YXLE	0.09	ACKA	0.00	YIST	0.02	ACKA	0.00	DNAK	0.22
YLXW	0.07	ACOA	0.00	KINA	0.02	ACOA	0.00	YQJU	0.19
APE	0.264		0.406		0.321		0.322		0.863

respectively. Lasso had the lowest average prediction errors among all five different penalty estimators.

From Fig. 2, we see that the error distributions are symmetric for Lasso, SCAD, and MCP, but it is left-skewed for CTFR and aLasso. CTFR and aLasso have a significantly larger inter-quartile range than those of Lasso, SCAD, and MCP.

Figure 3 depicts variable selection results for each penalty estimators. The blue line represents a general trend of the number of selected predictors by Lasso estimator for a given tuning parameter value. Each colored point represents the number of selected predictors with the corresponding tuning parameter value, which minimizes cross validation error. For instance, in the case of SCAD estimator, the estimator selected 26 predictors to minimize cross validation error. As CTFR's variable selection result is independent of the value of tuning parameter, the graph does not include it. However, from Table 14, one can observe that CTFR selected seven predictors, which is the least among all five penalty estimators of consideration.

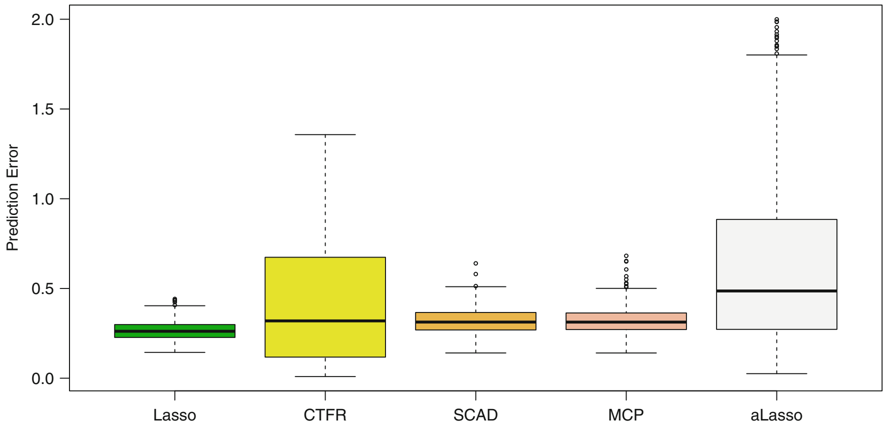

Fig. 2 Comparison of the estimators through prediction errors for Riboflavin data

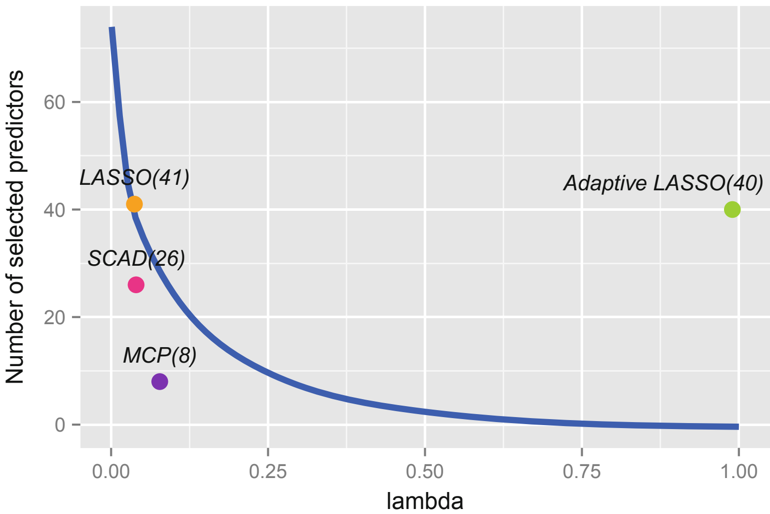

Fig. 3 The number of selected variables of each penalty estimators for the riboflavin data

5 Concluding Remarks and Future Research Directions

We have investigated the relative performance of high-dimensional regression strategies under the correlated design matrix and various signal-to-noise ratios. We have conducted an extensive simulation study to investigate the performance of the suggested strategies in terms of variable selection and prediction performance. The simulation results clearly demonstrate that none of the estimators considered here are better than their competitors under all possible correlation and signal-to-noise ratio scenarios.

As a future research project, one could investigate the distribution of the estimated parameters in CTFR and construction of confidence intervals for the estimated parameters. There are no theoretical results for CTFR in terms of variable selection performance so it is another research direction which needs to be explored.

Acknowledgements The research of S. Ejaz Ahmed is supported by the Natural Sciences and the Engineering Research Council of Canada (NSERC).

References

1. Ahmed, S. E. (2014). *Penalty, shrinkage and pretest strategies: Variable selection and estimation*. New York: Springer.
2. Belloni, A., Chernozhukov, V., & Wang, L. (2011). Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika, 98*(4), 791–806.
3. Bickel, P. J., Ritov, Y., & Tsybakov, A. B. (2009). Simultaneous analysis of lasso and Dantzig selector. *The Annals of Statistics, 37*, 1705–1732.
4. Bühlmann, P., Kalisch, M., & Meier, L. (2014). High-dimensional statistics with a view towards applications in biology. *Annual Review of Statistics and Its Applications, 1*, 255–278.
5. Bühlmann, P., & van de Geer, S. (2011). *Statistics for high-dimensional data. Methods,theory and applications. Springer series in statistics*. Heidelberg: Springer.
6. Chatterjee, S. (2015). High dimensional regression and matrix estimation without tuning parameters. https://arxiv.org/abs/1510.07294
7. Chatterjee, S., & Jafarov, J. (2016). Prediction error of cross-validated Lasso. https://arxiv.org/abs/1502.06291
8. Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of Statistics, 32*(2), 407–499.
9. Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association, 96*(456), 1348–1360.
10. Friedman, J., Hastie, T., & Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software, 33*(1), 1.
11. Frank, I., & Friedman, J. (1993). A statistical view of some chemometrics regression tools (with discussion). *Technometrics, 35*, 109–148 (1993).
12. Fu, W. J. (1998). Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics, 7*(3), 397–416.
13. Greenshtein, E., & Ritov, Y. A. (2004). Persistence in high-dimensional linear predictor selection and the virtue of overparametrization. *Bernoulli, 10*(6), 971–988.
14. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning. Data mining, inference, and prediction. Springer series in statistics* (2nd ed.). New York: Springer.
15. Hastie, T., Wainwright, M., & Tibshirani, R. (2016). *Statistical learning with sparsity: The lasso and generalizations*. Boca Raton, FL: Chapman and Hall/CRC.
16. Hebiri, M., & Lederer, J. (2013). How correlations influence lasso prediction. *IEEE Xplore: IEEE Transactions on Information Theory, 59*(3), 1846–1854.
17. Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for non-orthogonal problems. *Technometrics, 12*, 69–82.
18. Huang, J., Ma, S., & Zhang, C. H. (2008). Adaptive Lasso for sparse high-dimensional regression models. *Statistica Sinica, 18*, 1603–1618
19. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning with applications in R. Springer texts in statistics*. New York: Springer.
20. Knight, K., & Fu, W. (2000). Asymptotics for lasso-type estimators. *The Annals of Statistics, 28*, 1356–1378.

21. Lederer, J., & Muller, C. (2014). Don't fall for tuning parameters: Tuning-free variable selection in high dimensions with the TREX. Preprint. arXiv:1404.0541.
22. Leng, C., Lin, Y., & Wahba, G. (2006). A note on the lasso and related procedures in model selection. *Statistica Sinica, 16*, 1273–1284.
23. Meinshausen, N., & Bühlmann, P. (2006). High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics, 34*(3), 1436–1462.
24. Reid, S., Tibshirani, R., & Friedman, J. (2016). A study of error variance estimation in lasso regression. *Statistica Sinica, 26*(1), 35–67.
25. Stamey, T. A., Kabalin, J. N., McNeal, J. E., Johnstone, I. M., Freiha, F., Redwine, E. A., et al. (1989). Prostate specific antigen in the diagnosis and treatment of adenocarcinoma of the prostate. II. Radical prostatectomy treated patients. *The Journal of Urology, 141*(5), 1076–1083.
26. Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B Statistical Methodology, 58*, 267–288.
27. Tibshirani, R. (2013). The lasso problem and uniqueness. *Electronic Journal of Statistics, 7*, 1456–1490
28. Zhang, C. H. (2007). *Penalized linear unbiased selection*. Department of Statistics and Bioinformatics, Rutgers University, 3
29. Zhang, C. H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics, 38*, 894–942.
30. Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association, 101*(476), 1418–1429.