Robust penalized least squares of depth trimmed residuals regression for high-dimensional data

Yijun Zuo

Department of Statistics and Probability
Michigan State University, East Lansing, MI 48824, USA

Abstract: Challenges with data in the big-data era include (i) the dimension $p$ is often larger than the sample size $n$ (ii) outliers or contaminated points are frequently hidden and more difficult to detect. Challenge (i) renders most conventional methods inapplicable. Thus, it attracts tremendous attention from statistics, computer science, and bio-medical communities. Numerous penalized regression methods have been introduced as modern methods for analyzing high-dimensional data. Disproportionate attention has been paid to the challenge (ii) though. Penalized regression methods can do their job very well and are expected to handle the challenge (ii) simultaneously. The fact is most of them can break down by a single outlier (or single adversary contaminated point) as revealed in this article. The latter systematically examines leading penalized regression methods in the literature in terms of their robustness and provides quantitative assessment and reveals that most of them can break down by a single outlier. Consequently, a novel robust penalized regression method based on the least sum of squares of depth trimmed residuals is proposed and studied carefully. Ex-
periments with simulated and real data reveal that the newly proposed method can outperform some leading competitors in terms of estimation and prediction accuracy in the cases considered.

*Key words and phrases:* robust regularized regression estimators, least squares of depth trimmed residual regression, finite sample prediction error bound.

1. Introduction

**Least squares regression, the classical setting** Consider the classic linear regression model,

\[ y_i = (1, x_i') \beta_0 + e_i := w_i' \beta_0 + e_i, \]  

(1.1)

where random variables \( y_i \) and \( e_i \in \mathbb{R} \), random vector \( x_i \in \mathbb{R}^{p-1} \), and \( \beta_0 \in \mathbb{R}^p \) is an unknown parameter of interest. ' stands for the transpose.

One wants to estimate the \( \beta_0 \) based on a given sample \( Z^{(n)} := \{(x_i', y_i)', i \in \{1, \cdots, n\}\} \) from a parent model \( y = w' \beta_0 + e \).

Call the difference between \( y_i \) (observed value) and \( w_i' \beta \) (predicted value), \( r_i \), for a candidate coefficient vector \( \beta \) (which is often suppressed).

\[ r_i := r_i(\beta) = y_i - (1, x_i') \beta := y_i - w_i' \beta. \]  

(1.2)

To estimate \( \beta_0 \), the classic least squares (LS) estimator is the minimizer of the sum of the squared residuals (SSR): \( \hat{\beta}_{ls} = \arg \min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} r_i^2 \). Alternatively, one can replace the square by absolute value to obtain the least
absolute deviations (lad) estimator (aka, $L_1$ estimator, in contrast to the $L_2$ (LS) estimator). A straightforward calculus derivation leads to

$$\hat{\beta}_{ls} = (X_n^\prime X_n)^{-1} X_n^\prime Y_n.$$  

(1.3)

where $Y_n = (y_1, \cdots, y_n)^\prime$, $X_n = (w_1, \cdots, w_n)^\prime$ and the columns of $X_n$ are assumed to be linearly independent (i.e. $X_n$ has a full rank $p$ ($n \geq p$)).

The LS estimator is popular in practice across a broader spectrum of disciplines due to its (i) great computability (with the computation formula); and (ii) optimal properties (the best linear unbiased estimator (BLUE) and the uniformly minimum variance unbiased estimator (UMVUE), page 186 of Shao (2003) when the i.i.d. error $e_i$ follows a normal $\mathcal{N}(0, \sigma^2)$.

It, however, can behave badly when the error distribution is slightly departed from the normal distribution, particularly when the errors are heavy-tailed or contain outliers.

**Penalized regression, the state of the art**  In modern applied data analysis, the number of variables often is even larger than the number of observations. Traditional methods such as LS can then no longer be applied due to the design matrix $X$ being less than $p$ rank ($n < p$), hence LS estimator is no longer unique and its variance is large if $X$ is close to collinear. Furthermore, models that include the full set of explanatory
variables often have poor prediction performance as they tend to have large variance while large models are in general difficult to interpret.

**Ridge regression**, minimizing SSR, subject to a constraint \( \sum_{i=1}^{p} |\beta_i|^2 < t \)

\[
\hat{\beta}_{ridge}(\lambda) := \arg \min_{\beta \in \mathbb{R}^p} \left\{ \sum_{i=1}^{n} r_i^2 + \lambda \sum_{i=1}^{p} |\beta_i|^2 \right\}, \tag{1.4}
\]

first proposed by Hoerl and Kennard (1970a,b), is a useful tool for improving prediction in regression situations with highly correlated predictors and tackling the non-inverse issue,

\[
\hat{\beta}_{ridge}(\lambda) = (X_n'X_n + \lambda I_{d \times d})^{-1} X_n'Y_n, \tag{1.5}
\]

its variance is smaller than that of the LS estimator. Therefore, better estimation can be achieved on the average in terms of mean squared error (MSE) with a little sacrifice of bias, and predictions can be improved overall.

The ridge regression was generalized in Frank and Friedman (1993) that introduced **bridge regression**, which minimizes SSR subject to a constraint \( \sum_{i=1}^{p} |\beta_i|^\gamma \leq t \) with \( \gamma \geq 0 \),

\[
\hat{\beta}_{bridge}(\lambda, \gamma) := \arg \min_{\beta \in \mathbb{R}^p} \left\{ \sum_{i=1}^{n} r_i^2 + \lambda \sum_{i=1}^{p} |\beta_j|^\gamma \right\}. \tag{1.6}
\]

Ridge regression (\( \gamma = 2 \)) and subset selection (\( \gamma = 0 \)) are special cases.

**Least absolute shrinkage and selection operator** (lasso) was introduced in Tibshirani (1996), minimizing SSR subject to a constraint \( \sum_{j=1}^{p} |\beta_j| \leq t \),
is a special case of the bridge with $\gamma = 1$. As pointed out by Tibshirani (1996), the lasso shrinks the LS estimator $\hat{\beta}_{ls}$ towards 0 and potentially sets $\hat{\beta}_j = 0$ for some $j$. That is, it performs as a variable selection operator.

Other approaches of regularized regression include, among others, (i) She and Owen (2011), who proposed an iterative procedure for outlier detection and consider the model $y_i = \sum_{j=1}^p x_{ij} \beta_j + \gamma_i + \epsilon_i$, in which the parameter $\gamma_i$ is nonzero when observation $i$ is an outlier. An earlier mean-shift model was proposed by Menjoge and Welsch (2010). (ii) elastic nets, introduced in Zou and Hastie (2005), a generalization of the ridge and lasso models, which combines the two penalties and yields

$$\hat{\beta}_{enet}(\lambda_1, \lambda_2) := \arg\min_{\beta \in \mathbb{R}^p} \left\{ \sum_{i=1}^n r_i^2 + \lambda_1 \sum_{i=1}^p |\beta_i| + \lambda_2 \sum_{i=1}^p \beta_i^2 \right\}. \quad (1.7)$$

(iii) To avoid the pre-estimation of standard deviation $\sigma$ of the error term in lasso and achieve a better performance, square-root lasso, introduced in Belloni, Chernozhukov and Wang (2011), is defined as

$$\hat{\beta}_{sqrt-lasso} = \arg\min_{\beta \in \mathbb{R}^p} \left\{ \left( \sum_{i=1}^n r_i^2 \right)^{1/2} + \lambda \sum_{i=1}^p |\beta_i| \right\}. \quad (1.8)$$

(iv) Aim to control the false discover rate (FDR), slope (Sorted L-One Penalized Estimation) introduced in Bogdan et al. (2015),

$$\hat{\beta}_{slope} = \arg\min_{\beta \in \mathbb{R}^p} \left\{ \sum_{i=1}^n r_i^2 + \sum_{i=1}^p \lambda_i |\beta_{(i)}| \right\}. \quad (1.9)$$
where \( \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0 \) and \( |\beta_{(1)}| \geq |\beta_{(2)}| \geq \cdots \geq |\beta_{(p)}| \).

Strong connections between some modern methods and a method called least angle regression (lar) was revealed in Efron et al. (2004) where they developed an algorithmic framework that includes all of these methods (lasso, boosting, forward stagewise regression) and provided a fast implementation, for which they used the term ‘lars’. lars is a promising technique/algorithm for variable selection applications, offering a nice alternative to stepwise regression. For a fine review on lar and lasso, see Hesterberg et al. (2008).

Other outstanding penalized regression estimators include, among others, SCAD Fan (1997), Fan and Li (2001) and MCP Zhang (2010). It is not our goal to review all existing penalized/regularized regression estimators in the literature. For a detailed account about lasso and its variants, refer to Table 6 of Freijeiro-González, Febrero-Bande and González-Manteiga (2022) or Fig. 1 of Wang, Chen and Yang (2022), and Zhang and Zhang (2012) and references therein.

The penalized regression estimators above improve prediction accuracy meanwhile enhance the interpretability of the model. They, however, pay the price of inducing a little bit of bias in addition to the lack of robustness. There are numerous published articles related to lasso and regularized regression in the literature. However, there are disproportionately few ad-
dressing the robustness of the estimators. Are they robust as supposed (or expected)? Or rather can they resist the influence of just a single contaminated point (or outlier) that is typically buried in high-dimensional data?

Robust versions of the lasso(or ridge) estimators have been sporadically considered in the literature. The LS in lasso (or in the ridge), is replaced by M-estimators, as in Van De Geer (2008) and Li, Peng and Zhu (2011); or replaced by a Huber-type loss function, as in Rosset and Zhu (2004) and Sun Zhou and Fan (2020); or by lads, as in Wang, Li and Jiang (2007);

\[
\hat{\beta}_{lad-lasso} = \arg\min_{\beta \in \mathbb{R}^p} \left\{ \sum_{i=1}^{n} |r_i| + \lambda \sum_{i=1}^{p} |\beta_i| \right\}, \tag{1.10}
\]
or replacing correlations in lars by a robust type of correlation, as in Khan, Van Aelst and Zamar (2007) (Rlars); or by S- and MM- (Yohai (1987)) estimators, as in Maronna (2011) for ridge regression (Rrr); or by the least trimmed squares (LTS) (Rousseeuw (1984)), as in Alfons, Croux and Gelper (2013). The LTS is defined as

\[
\hat{\beta}_{ltss}^n := \arg\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{h} (r^2)_{i:n}, \tag{1.11}
\]
where \((r^2)_{1:n} \leq (r^2)_{2:n} \leq \cdots, (r^2)_{n:n}\) are the ordered squared residuals, \([n/2] \leq h \leq n\), and \([\cdot]\) is the ceiling function. Alfons, Croux and Gelper (2013) replaced the SSR by the objective function of LTS and defined

\[
\hat{\beta}_{ltss-lasso} = \arg\min_{\beta \in \mathbb{R}^p} \left\{ \sum_{i=1}^{h} (r^2)_{i:n} + h\lambda \sum_{i=1}^{p} |\beta_i| \right\}, \tag{1.12}
\]
The idea of Alfons, Croux and Gelper (2013) has extended to logistic regression with elastic net penalty in Kurnaz, Hoffmann and Filzmoser (2018), and penalized weighted M-type estimators for the logistic regression have also been studied in Bianco, Boente and Chebi (2022).

Most estimators above (except Rlars, Rrr, and \( \hat{\beta}_{lts-lasso} \)), like both \( L_1 \) and \( L_2 \) (LS) estimators, unfortunately, have a pathetic 0% asymptotic breakdown point (i.e., one bad point can ruin (break down) the estimator), in sharp contrast to the 50% of the least sum of squares of trimmed (LST) residuals estimator (see Section 3.1 of Zuo and Zuo (2023) or Section 3 here). Kurnaz, Hoffmann and Filzmoser (2018) and Bianco, Boente and Chebi (2022) both assert their estimators are robust, but no qualitative robustness assessment of their estimators has been established yet. The same situation with the estimator in She, Wang and Shen (2022).

Now let us take a close look at the three exceptions above. The main drawback of the Rlars is the lack of a natural definition or a clear objective function, as commented in Alfons, Croux and Gelper (2013). The main focus of Maronna (2011) is robustifying ridge regression (Rrr).

Only \( \hat{\beta}_{lts-lasso} \) in Alfons, Croux and Gelper (2013) has an established high finite sample breakdown point (see Section 3 for definition). Their result, though covers the lasso-type estimators, does not cover the elastic nets
and other estimators; the authors failed to (i) explain why their estimator can have a breakdown point higher than 50% and (ii) study the properties (such as equivariance and consistency) of their estimator. Furthermore, the LTS is notorious for its inefficiency (i.e., usually has a large variance). On the other hand, the LST introduced in Zuo and Zuo (2023) can outperform the LTS (especially in efficiency) as demonstrated in Zuo and Zuo (2023).

Based on observations above, questions we want to address are: (i) Can one replace the LS with a robust LST in the penalized regression? How does the resulting estimator perform? (ii) Is it more robust, compared with existing ones? Can one provide a more general breakdown robustness assessment that covers more regularized regression estimators and provide an explanation of a breakdown point higher than 50%? (iii) Besides robustness, what are other desirable properties for a regression estimator?

The main contributions of this article include (i) it proves that most leading penalized regression estimators can break down by a single adversary contaminating point; (ii) hence, a novel and robust penalized least squares of depth trimmed regression estimator ($\hat{\beta}_{l_{1}et}$) is introduced that outperforms leading competitors in the cases considered; and (iii) efficient computational algorithm for the estimator is proposed and tested for simulated and real high-dimensional data.
The rest of article is organized as follows. Section 2 establishes a robust result for general regularized regression estimators and reveals that most of leading estimators (including lasso, lars, and enet) has the worst breakdown point robustness. Section 3 introduces the least sum of squares of (depth) trimmed residuals (LST) regression and studies its robust property. Section 4 introduces a class of penalized regression estimators based on LST and studies their properties including, existence and uniqueness, robustness, and equivariance. Section 5 is devoted to the establishment of the finite sample prediction error bound and estimator consistency. Section 6 addresses the computation issue of \( \hat{\beta}_{lst-enet} \). Section 7 consists of simulation/comparison study and real data application of five competing methods. Section 8 ends the article with some concluding discussions. Proofs of main results and remarks are deferred to an Appendix in the supplementary material.

2. Robustness of the penalized regression estimators

Are existing numerous penalized regression methods mentioned above robust as they are expected or believed? Or rather can they resist the influence of just a single outlier (or adversary single-point contamination)? We now formally address this question.

A robustness measure
In the finite sample practice, the most prevailing quantitative measure of the robustness of any regression or location estimators is the finite sample breakdown point, introduced by Donoho and Huber (1983).

**Definition** Donoho and Huber (1983) The finite sample replacement breakdown point (RBP) of a regression estimator $T$ at a given sample $Z^{(n)} = \{Z_1, Z_2, \ldots, Z_n\}$, where $Z_i := (x_i', y_i')'$, is defined as

$$\text{RBP}(T, Z^{(n)}) = \min_{1 \leq m \leq n} \left\{ \frac{m}{n} : \sup_{Z_m^{(n)}} \|T(Z_m^{(n)}) - T(Z^{(n)})\|_2 = \infty \right\}, \quad (2.13)$$

where $Z_m^{(n)}$ stands for an arbitrary contaminated sample by replacing $m$ original sample points in $Z^{(n)}$ with arbitrary points in $\mathbb{R}^p$ and $\|x\|_q = (\sum_{i=1}^n x_i^q)^{1/q}$ is the $\ell_q$-norm for vector $x \in \mathbb{R}^n$.

We now present a general RBP result on the penalized regression estimators.

**A general result on penalized regression estimators**

**Theorem 2.1** For any given data set $Z^{(n)} = \{(x_i', y_i')', i \in \{1, \cdots, n\}\}$ in $\mathbb{R}^p$ ($p > 1$), let $\hat{\beta}^*(\lambda_1, \lambda_2, \gamma, Z^{(n)})$ be the penalized regression estimator, which minimizes the objective

$$O(\beta, \lambda_1, \lambda_2, \gamma, Z^{(n)}):= \frac{1}{n} \sum_{i=1}^n \mathcal{L}(r_i) + g(\beta, \lambda_1, \lambda_2, \gamma), \quad (2.14)$$

where $\lambda_1, \gamma \geq 0$, the combined penalty function $g(\beta, \lambda_1, \lambda_2, \gamma) \geq 0$ and the loss function $\mathcal{L}(x)$ is non-negative, non-decreasing over $(0, \infty)$, $\mathcal{L}(0) = 0$.
and $\mathcal{L}(x) \to \infty$ when $x \to \infty$. Then

$$
\text{RBP}(\hat{\beta}^*(\lambda_1, \lambda_2, \gamma, Z^{(n)}), Z^{(n)}) = \frac{1}{n}.
$$

**Proof:** see the appendix in the supplementary material.

**Remarks 2.1** see the appendix in the supplementary material.

Now that most of the existing penalized regression estimators can be broken down by a single outlier (or single-point contamination). Existing robust penalized regression estimators are most ad hoc, e.g., Rlars of Khan, Van Aelst and Zamar (2007) is for robustifying lars, and Rrr of Maronna (2011) is for robustifying ridge regression, and Bianco, Boente and Chebi (2022) is mainly for robustifying the penalized logistic regression estimators.

Only $\hat{\beta}_{\text{Lts-lasso}}$ of Alfons, Croux and Gelper (2013) and $\hat{\beta}_{\text{enetLTS}}$ of Kurnaz, Hoffmann and Filzmoser (2018) that employed LTS to replace the SSR in lasso have really high breakdown robustness meanwhile do the variable selection job. But the major drawback of the LTS is its inefficiency (it has a larger variance) as demonstrated in Zuo and Zuo (2023) and Figures 2 and 3 of Kurnaz, Hoffmann and Filzmoser (2018).

A natural question is: can one construct a penalized regression estimator that is robust against the outliers or contamination and more efficient (i.e., with a smaller variance than the LTS)? In the following, we achieve
this goal by introducing a robust alternative to the LS estimator, called an LST (least squares of depth trimmed residuals estimator), and applying it to the penalized regression setting.

3. The least sum of squares of trimmed residuals regression

Definition of LST
To robustify the LS estimator, Rousseeuw (1984) introduced least trimmed squares (LTS) estimator. The procedure orders the squared residuals and then trims the larger ones and keeps at least $h \geq \lceil n/2 \rceil$ squared residuals, the minimizer of the sum of those trimmed squared residuals is called an LTS estimator as defined in (1.11). $\hat{\beta}_{lts}^n$ is highly robust but is not very efficient, as reported in Maronna, Martin and Yohai (2006) (page 132) having just 7% or 8% asymptotic efficiency. A more efficient competitor, least sum of squares of trimmed (LST) residuals estimator, is introduced in Zuo and Zuo (2023), overcoming LTS drawback while sharing its high robustness and fast computation advantages.

For a given sample $Z^{(n)} = \{(x_i', y_i)' , i \in \{1, \cdots, n\}\}$ in $\mathbb{R}^p$ and a $\beta \in \mathbb{R}^p$, define $\mu(Z^{(n)}, \beta) = \text{Med}_i \{ r_i \}$, $\sigma(Z^{(n)}, \beta) = \text{MAD}_i \{ r_i \}$, where $r_i$ is defined in (1.2), $\text{Med}_i \{ r_i \} = \text{median} \{ r_i , i \in \{1, \cdots, n\}\}$ is the median of $r_i$s, and $\text{MAD}_i \{ r_i \} = \text{Med}(\{|r_i - \text{Med}(r_i)| , i \in \{1, \cdots, n\}\})$ is the median
of absolute deviations to the center (median) of $r_i$s. Operators Med and MAD are used for discrete data sets (and distributions as well).

The outlyingness (or equivalently, depth) of a point $x$ in Zuo (2003) is defined to be (strictly speaking, depth=$1/(1+\text{outlyingness})$)

$$D(x, X^{(n)}) = |x - \text{Med}(X^{(n)})|/\text{MAD}(X^{(n)}),$$

(3.15)

where $X^{(n)} = \{x_1, \ldots, x_n\}$ is a data set in $\mathbb{R}^1$. It is readily seen that $D(x, X^{(n)})$ is a generalized standard deviation, or equivalent to the one-dimensional projection depth/outlyingness (see Zuo and Serfling (2000), Zuo (2003, 2006) for a high dimensional version). For notion of outlyingness, cf Stahel (1981), and Donoho and Gasko (1992). For a given $\alpha$ (throughout constant $\alpha \geq 1$, default value is one) in the depth trimming scheme, consider the quantity

$$Q^n(\beta) := Q(Z^{(n)}, \beta, \alpha) = \frac{1}{n} \sum_{i=1}^{n} r_i^2 \mathbb{1}(D(r_i, R^{(n)}) \leq \alpha),$$

(3.16)

where $\mathbb{1}(A)$ is the indicator of $A$ (i.e., it is one if $A$ holds and zero otherwise) and $R^{(n)} := \{r_i, i \in \{1, 2, \ldots, n\}\}$. Namely, residuals with their outlyingness (or depth) greater than $\alpha$ (or less than $1/(1 + \alpha)$) will be trimmed. When there is a majority ($\geq \lfloor (n + 1)/2 \rfloor$) identical $r_i$s, we define $\sigma(Z^{(n)}, \beta) = 1$. Minimizing $Q(Z^{(n)}, \beta, \alpha)$, one gets the least sum of squares.
of trimmed (LST) residuals estimator,

\[ \hat{\beta}^{\text{lst}}_n := \hat{\beta}^{\text{lst}}(Z^{(n)}, \alpha) = \arg \min_{\beta \in \mathbb{R}^p} Q(Z^{(n)}, \beta, \alpha). \]  

(3.17)

Compared with the LTS definition (1.11), it is readily seen that both estimators trim residuals. However, there are two essential differences: (i) the trimming schemes are different. The LTS employs a rank-based trimming scheme that focuses only on the relative position of points (squared residuals) with respect to others and ignores the magnitude of the point and the relative distance between points whereas the LST exactly catches these two important attributes. It orders data from a center (the median) outward and trims the points that are far away from the center. This is known as depth-based trimming. (ii) Besides the trimming scheme difference, there is another difference between the LTS and the LST, that is, the order of trimming and squaring. In the LTS, squaring is first, followed by trimming whereas, in the LST, the order is reversed.

All the difference leads to an unexpected performance difference in the LTS and the LST as demonstrated in the small illustration example (Ex 1.1) and in Figure 1 of Zuo and Zuo (2023).

Existence and uniqueness of \( \hat{\beta}^{\text{lst}}_n \) have been addressed in Zuo and Zuo (2023), it is also equivariant (see Zuo and Zuo (2023)).

A regression estimator \( T \) is called \textit{regression, scale, and affine equiv-}
arian if, respectively (see page 116 of Rousseeuw and Leroy (1987)) with
\[ N = \{1, 2, \ldots, n\} \]
\[
T(\{(w_i', y_i + w_i'b)\}') = T(\{(w_i', y_i)\}') + b, \ i \in N, \ \forall \ b \in \mathbb{R}^p
\]
\[
T(\{(w_i', s y_i)\}') = s T(\{(w_i', y_i)\}'), \ i \in N, \ \forall \ s \in \mathbb{R}^1
\]
\[
T(\{(A'w_i)', y_i\}') = A^{-1} T(\{(w_i', y_i)\}'), \ i \in N, \ \forall \ \text{nonsingular} \ A_{p \times p}
\]
Now with the measure of robustness (in section 2), naturally one wants to ask: is \( \hat{\beta}_{\text{lst}}^n \) theoretically more robust than the LS estimator \( \hat{\beta}_{\text{ls}}^n \)?

Robustness of LST

We shall say \( Z^{(n)} \) is in general position when any \( p \) of observations in \( Z^{(n)} \) gives a unique determination of \( \beta \). That is, any (p-1) dimensional subspace of the space \( (x', y)' \) contains at most p observations of \( Z^{(n)} \).

Theorem 3.2 Zuo and Zuo (2023) For \( \hat{\beta}_{\text{lst}}^n \) defined in (3.17) and \( Z^{(n)} \) in general position, we have
\[
\text{RBP}(\hat{\beta}_{\text{lst}}^n, Z^{(n)}) = \begin{cases} 
\lfloor (n + 1)/2 \rfloor /n, & \text{if } p = 1, \\
(\lfloor n/2 \rfloor - p + 2)/n, & \text{if } p > 1.
\end{cases}
\] (3.18)

The LST not only shares the best 50% asymptotic breakdown value of the LTS, it is much more efficient than the LTS as demonstrated in the Table 2 of Zuo and Zuo (2023).
4. A class of penalized regression estimators based on the LST

**Definition** Now that we have a much more robust regression estimator than the LS, which turns out to be more efficient than the LTS. It is quite natural to replace the SSR in (1.7) by the $Q_n$ defined by (3.16), and minimize it, subject to two constraints: $\ell_\gamma$-constraint $\sum_{i=1}^{p} |\beta_i|^{\gamma} \leq t_1$, $t_1 \geq 0$, $\gamma \geq 1$; and $\ell_2$-constraint $\sum_{i=1}^{p} \beta_i^2 \leq t_2$, $t_2 \geq 0$, the minimizer is

$$\hat{\beta}_{lst-net}^n(\alpha, \lambda_1, \lambda_2, \gamma) := \arg \min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} r_i^2 w_i + \lambda_1 \sum_{i=1}^{p} |\beta_j|^{\gamma} + \lambda_2 \sum_{i=1}^{p} \beta_i^2 \right\},$$

(4.19)

where $\lambda_i := \lambda(t_i) \geq 0$, $\alpha$, $\gamma \geq 1$, and $w_i := w_i(\beta) := w_i(\beta, r_i, Z^{(n)}) = 1 \left( D(r_i, R^{(n)}) \leq \alpha \right)$. Before studying its robustness, let us first address existence and uniqueness of $\hat{\beta}_{lst-net}^n(\alpha, \lambda_1, \lambda_2, \gamma)$.

**Existence and uniqueness**

Existence and uniqueness are implicitly assumed for many other penalized regression estimators in the literature. We formally address them below for $\hat{\beta}_{lst-net}^n(\alpha, \lambda_1, \lambda_2, \gamma)$.

**Theorem 4.1**

(i) $\hat{\beta}_{lst-net}^n(\alpha, \lambda_1, \lambda_2, \gamma)$ in (4.19) always exists for any $\lambda_i \geq 0$, $\alpha$, $\gamma \geq 1$;

(ii) $\hat{\beta}_{lst-net}^n(\alpha, \lambda_1, \lambda_2, \gamma)$ in (4.19) is unique provided that (a) $\lambda_1 > 0$ and $\gamma > 1$ or (b) $\lambda_2 > 0$. 
Proof: see the appendix in the supplementary material.

Remarks 4.1 (see the appendix in the supplementary material).

The most relevant question now is: Is $\hat{\beta}_{\text{lat-enet}}(\alpha, \lambda_1, \lambda_2, \gamma)$ much more robust than the existing ones? Or rather, what is its RBP? The next result covers both the LST and the LTS based regularized estimators and provides an affirmative answer to the question.

Theorem 4.2 Let $\hat{\beta}(\lambda_1, \lambda_2, \gamma, \mathbf{Z}^{(n)})$ be the penalized regression estimator which minimizes the objective function

$$Q(\beta, \lambda_1, \lambda_2, \gamma, \mathbf{Z}^{(n)}) := \frac{1}{n} \sum_{i=1}^{n} r_i^2 w_i + \lambda_1 \sum_{i=1}^{p} |\beta_i|^\gamma + \lambda_2 \sum_{i=1}^{p} \beta_i^2, \quad (4.20)$$

where $w_i \in \{0, 1\}$ is an indicator function: $1(r_i^2 \leq r_{h}^2)$ or $1(D(r_i, R^{(n)}) \leq \alpha)$ and $\sum_{i=1}^{n} w_i = k ([n/2] \leq k \leq n)$, $\lambda_i \geq 0$, and $\lambda_1 + \lambda_2 > 0$, $1 \leq \gamma \leq 2$.

Then

$$\text{RBP}(\hat{\beta}(\lambda_1, \lambda_2, \gamma, \mathbf{Z}^{(n)}), \mathbf{Z}^{(n)}) = (n - k + 1)/n.$$ 

Proof: see the appendix in the supplementary material.

Remarks 4.2 (see the appendix in the supplementary material).

Equivariance

Among regression, scale, and affine equivariance, the three desired properties (discussed in Section 3), the regression equivariance is the most fundamental, it demands that if one shifts response variable $y$ up and down,
then the regression line (or hyperplane) should shift accordingly up and down. The LS estimator and all its robust alternatives mentioned so far satisfy the three properties. But this is not the case for most of regularized regression estimators. In fact,

**Theorem 4.3** Among three equivariant properties, only scale equivariance is processed by \( \hat{\beta}_{\text{ridge}} \) in (1.4), the \( \hat{\beta}_{\text{sqrt-lasso}} \) in (1.8), the \( \hat{\beta}_{\text{lad-lasso}} \) in (1.10), and \( \hat{\beta}_{\text{lst-enet}}^n(\alpha, \lambda_2, \gamma) \) in (4.19) among all penalized regression estimators discussed previously.

**Proof:** see the appendix in the supplementary material. ■

**Remarks 4.3** (see the supplementary material) ■

5. **Finite sample prediction error bounds—consistency**

In this section we assume that the true model is \( Y = X\beta_0 + e \) where \( Y = (y_1, \ldots, y_n)' \), \( X = (w_1, \ldots, w_n)' \), and \( e = (e_1, \ldots, e_n)' \) with \( y_i, e_i, \) and \( w_i \) defined in (1.1) and (1.2). We investigate the difference between \( X\hat{\beta}_{\text{lst-enet}}^n \) and \( X\beta_0 \) (prediction error). Write \( \hat{\beta}^n \) for \( \hat{\beta}_{\text{lst-enet}}^n \) for simplicity.

Define an index set \( I(\beta) := \{ i : w_i = 1 \} \), the scalar \( w_i \in \{0, 1\} \) in (4.19) is different from the vector \( w_i \) above. Write \( D(\beta) = \text{diag}(w_1, \cdots, w_n) \) with \( w_i \) defined in (4.19). Let \( A \) be a \( n \) by \( n \) symmetric positive semidefinite matrix, a norm (or seminorm) induced by \( A \) is \( \|x\|_A^2 = x'Ax \) for any \( x \in \mathbb{R}^n \).
Although $\hat{\beta}^n$ provides predictions for all $i$, but we just employed residuals $r_i$ with $i \in I(\hat{\beta}^n)$ in (4.19), so instead of looking at $\|X(\hat{\beta}^n - \beta_0)\|^2$, we will focus on the squared perdition error $\|X(\hat{\beta}^n - \beta_0)\|^2_{D(\hat{\beta}^n)}$.

**Lemma 5.1** Assume that $\beta_0$ is the true parameter of the model in (1.1), $\hat{\beta}^n := \hat{\beta}^n_{lst-enet}$ is defined in (4.19). We have

$$\|X(\hat{\beta}^n - \beta_0)\|^2_{D(\hat{\beta}^n)} \leq \frac{2}{n} e' D(\hat{\beta}^n) X(\hat{\beta}^n - \beta_0) + \frac{1}{n} (\|e\|^2_{D(\beta_0)} - \|e\|^2_{D(\hat{\beta}^n)}) + \lambda_1 \|\beta_0\|_{\gamma} + \lambda_2 \|\beta_0\|^2 - \lambda_1 \|\hat{\beta}^n\|_{\gamma} - \lambda_2 \|\hat{\beta}^n\|^2. \quad (5.21)$$

**Proof**: see the appendix in the supplementary material.

Write $(e_1^*, \ldots, e_n^*) := (e^*)'$ with $e_i^* = e_i * 1 (i \in I(\hat{\beta}^n))$. Define two sets

$$\mathcal{S}_1 := \left\{ \max_{1 \leq j \leq p} 2\| (e^*)' x^{(j)} \| / n \leq q_1 \right\}, \quad \mathcal{S}_2 := \left\{ \| e \|^2_{D(\cdot)} / \sigma^2 - N_d \leq q_2 \right\},$$

where $x^{(j)}$ is the $j$th column of the fixed design matrix $X_{n \times p}$, $D^* = D(\beta_0) - D(\hat{\beta}^n)$, a diagonal matrix with $D^*(i,i) = 1 (D(\beta_0)(i,i) = 1$ and $D(\hat{\beta}^n)(i,i) = 0)$. Let $N_d = |I(\beta_0)| - |I(\beta_0) \cap I(\hat{\beta}^n)|$, it is readily seen that $0 \leq N_d \leq (n - 1)$. Note that $e' D(\hat{\beta}^n) = (e_1^*, \ldots, e_n^*) := (e^*)'$. **Assume hereafter** that $\max_{1 \leq j \leq p} \|x^{(j)}\|_2 \leq c_x$ for a constant $c_x$.

In the classical setting $e_i$ in (1.1) is assumed $N(0, \sigma^2)$, it is needed for the second result below, but for the first, it can be relaxed to be a sub-Gaussian variable. For the definition of the latter, we refer to Definition 1.2 of [Rigollet and Hütter (2017)] and/or Theorem 2.1.1 of [Pauwels (2020)].
Lemma 5.2 (i) Let $e_i$ in (1.1) be independent sub-Gaussian variables that have variance proxy $\sigma^2$, then $(e^i)'x^{(j)}/c_x$ is a sub-Gaussian variable with variance proxy $\sigma^2$. (ii) Let $e_i$ in (1.1) be i.i.d. $N(0, \sigma^2)$, then $\|e\|_D^2/\sigma^2$ follows a $\chi^2$ distribution with $N_d$ degrees of freedom.

Proof: see the appendix in the supplementary material.

Lemma 5.3 Assume that $e_i$ in (1.1) be i.i.d. $N(0, \sigma^2)$ and other assumptions in Lemmas 5.1-5.2 hold, for any $\delta \in (0, 1)$ let

$$q_1 = \frac{4c_x\sigma}{n} \left(2\sqrt{p} + \sqrt{2\log(2/\delta)}\right); \quad q_2 = 2\sqrt{\log(2/\delta)} \left(\sqrt{|I(\beta_0)|} + \sqrt{\log(2/\delta)}\right),$$

then

$$P(\mathcal{S}_1) \geq 1 - \delta/2; \quad P(\mathcal{S}_2) \geq 1 - \delta/2. \quad (5.22)$$

Proof: see the appendix in the supplementary material.

In light of all Lemmas we are in the position to present the main result.

Theorem 5.1 Set $\gamma$ in (4.19) to be one and assume that the assumptions in Lemma 5.3 hold. For any $\delta \in (0, 1)$, selecting $\lambda_1 \geq q_1$. Then with probability at least $1 - \delta$, one has

$$\|X(\hat{\beta}^n - \beta_0)\|_{D(\hat{\beta}^n)}^2 \leq 2\lambda_1\sqrt{p}\|\beta_0\|_2 + \lambda_2\|\beta_0\|_2^2 + \frac{\sigma}{n}(q_2 + N_d). \quad (5.23)$$

Remarks 5.1 (see the appendix in the supplementary material).
6. Computation algorithm

Re-parametrizations If we set \( \lambda^* = \lambda_1 + \lambda_2 \) and \( \alpha^* = \lambda_2 / (\lambda_1 + \lambda_2) \) (note that \( \lambda_1 + \lambda_2 > 0 \), otherwise we have a non-penalized problem addressed in Zuo and Zuo (2023), then we have
\[
O_n(\beta, \lambda_1, \lambda_2, \gamma) = O_n(\beta, \alpha^*, \lambda^*, \gamma) := \frac{1}{n} \| Y - X \beta \|_{D(\beta)}^2 + \lambda^* \left( (1-\alpha^*) \sum_{j=1}^p \| \beta_j \| \gamma + \alpha^* \| \beta \|_2^2 \right).
\]
Note that \( \alpha^* \in [0, 1) \) (a pure ridge regression case is excluded) and \( \lambda^* \in (0, \lambda_0] \) for some \( \lambda_0 \) (which is set to be \( \max_{1 \leq j \leq p} |2Y'x^{(j)}|/n \) as in the literature, see e.g., Alfons, Croux and Gelper (2013). Boundedness of parameters is the advantage of this formulation. For a given data set \( Z^{(n)} = \{(x_i', y_i)', i \in \{1, \cdots, n\}\} \), we now present the outline our approximate algorithm (AA) for \( \hat{\beta}_{lst-enet}^n \).

For Pseudocode for computing \( \hat{\beta}_{lst-enet}^n \) (lst-enet), see the appendix.

7. Illustration examples and comparison

7.1 Simulation

Five regularized regression procedures We like to compare the performance of our procedure lst-enet with leading regularized regression procedures including lasso, lars, enet, and enetLTS. lasso will be computed via \textbf{R} package “lars”, it can be obtained via “elasticnet”. The latter package is mainly for the enet whereas the former mainly focuses on lars. Though lasso could be obtained via “glmnet” but due to the contamination scenario,
the glmnet often does not work. Unfortunately, enetLTS employing glmnet in its CV calculation, it can not handle the model \( y = w'\beta_0 + e \) appeared in (1.1) (an error message “glmnet fails at standardization step”). We use an alternative model given below.

All R code for simulation and examples as well as figures in this article (downloadable via https://github.com/left-github-4-codes/lst-enet) were run on a desktop Intel(R)Core(TM) 21 i7-2600 CPU @ 3.40 GHz.

**Simulation designs** To copy with the situation above, we simulate data from the true model: \( y = X\beta_0 + \sigma e \), \( e \sim N(0,1) \), where the true unknown parameter \( \beta_0 \) is assumed to be a \( p \)-dimensional vector with the first \( p_1 := \lceil 6\% \times p \rceil \) components are ones and the rest \( p_2 := p - p_1 \) components are zeros. \( \sigma \) is set to be 0.5 but could be changed to other values (leading to different signal-to-noise ratio).

**Design I**: take sample of \( X \) from \( N(0,\sigma I_{p \times p}) \) and \( e \) from \( N(0,1) \).

**Design II**: take sample from \( X \sim N(0,\Sigma) \) with \( \Sigma(i,j) = \rho_1^{\lvert i-j \rvert}, 1 \leq i,j \leq p_1 \), \( \Sigma(i,j) = \rho_2^{\lvert i-j \rvert}, p_1 < i,j \leq p \), \( \rho_1 = 0.95 \), \( \rho_2 = 0.05 \), all other entries of \( \Sigma \) are zeros and \( e \sim N(0,1) \). We take \( n \in \{50,100\} \) samples from the \( X \) and \( e \) above and calculate the response \( y_i = X_i\beta_0 + \sigma e_i, i \in \{1,2,\cdots,n\} \).

**Contamination levels and schemes** Let \( \varepsilon \) be the contamination
level, when \( \varepsilon = 0 \) there is no contamination, an ideal situation (and not realistic). Consider the scenario \( \varepsilon \in \{0, 0.05, 0.1, 0.2\} \) (i.e., 0%, 5%, 10%, 20% contamination). Let \( m = |\varepsilon \times n| \), sample \( m \) indices from \( \{1, \ldots, n\} \).

Contamination **Scheme I**: add 20 to the corresponding \( m \) components of \((e_1, \ldots, e_n)\), compute \( y_i = X_i\beta_0 + \sigma e_i, i \in \{1, 2, \ldots, n\} \), and add 20 (component-wise) to the corresponding \( m \) rows of \((X_1, \ldots, X_n)'\). **Scheme II**: add 20 to the corresponding \( m \) components of \((e_1, \ldots, e_n)\), compute \( y_i = X_i\beta_0 + \sigma e_i, i \in \{1, 2, \ldots, n\} \). Replace the corresponding \( m \) rows of \((X_1, \ldots, X_n)'\) by a \( p \)-vector with its first component being \( 10^4 \) and the rest are zeros, do the same for the corresponding \( m \) components of \((y_1, \ldots, y_n)\) but with a scalar \( 10^{10} \).

**Four performance criteria** The first measure is the estimation error, or L2-error/L2-loss between the true parameter \( \beta_0 \) and the estimator \( \hat{\beta}_P \) via procedure \( P \) and is defined as:

\[
\text{L2-error}(\beta_0, \hat{\beta}_P) := \|\beta_0 - \hat{\beta}_P\|_2^2, \tag{7.24}
\]

where \( \|a - b\|_2 \) is the \( \ell_2 \)-norm between the two \( p \)-dimensional vectors.

On the other hand, one has to take the performance measure into the context of the sparsity model consideration. In the following we introduce the **true sparsity discovery rate** (TSDR) and the **false sparsity discovery**
rate (FSDR). For notation simplicity, we denote the unknown parameter by $\beta^0$ (assume it has at least one zero coordinate), an estimator by $\hat{\beta}_P$.

$$TSDR(\beta^0, \hat{\beta}_P) := \frac{\sum_{i=1}^p 1(\beta^0_i = 0, \hat{\beta}_P_i = 0)}{\sum_{i=1}^p 1(\beta^0_i = 0)}, \quad (7.25)$$

namely, the fraction of correctly detecting/discovering the zero coordinates of the true parameter $\beta^0$. The higher the TSDR, the better the $\hat{\beta}_P$.

$$FSDR(\beta^0, \hat{\beta}_P) := \frac{\sum_{i=1}^p 1(\beta^0_i \neq 0, \hat{\beta}_P_i = 0)}{\sum_{i=1}^p 1(\beta^0_i \neq 0)}, \quad (7.26)$$

namely, the fraction of falsely detecting/discovering as zero coordinate for the true parameter $\beta^0$. The lower the FSDR, the better the $\hat{\beta}_P$.

The fourth performance measure is a popular one, it is (square-)root of mean squared (prediction) error (RMSE) on testing data. That is, for a given data set, one first partitions data into training and testing two parts (we take the ratio 7:3 for partition). Then fit the model and get estimator based on the training data and using the testing data to get the RMSE. Testing data sets are often assumed to be clean (have no contamination or outliers) in the literature. This, however, is not realistic in practice.

Let $X_{test}$, $y_{test}$ be the testing data and $\hat{\beta}_P$ be the estimator obtained from the training data. Then

$$RMSE(\hat{\beta}_P) := \left( \text{mean}((y_{test} - X_{test}\hat{\beta}_P)^2) \right)^{1/2}. \quad (7.27)$$
7.1 Simulation

Figure 1: Boxplots for five procedures (P1 stands for lst-enet, P2 for lasso, P3 for lars, P4 for enetLTS, P5 for enet) and 50 samples each with \( n = 100 \) and \( p = 50 \) that are generated from design I with 0% contamination rate.

The four performance measures above were discussed in the literature before, all are hoped to be small except the TSDR which is hoped to be as high as possible. All (but RMSE) depend on the unknown parameter \( \beta^0 \).

Example 7.1 We first consider \( \varepsilon = 0 \). For simplicity, data are generated according to design I and set \( n = 100, p = 50 \) (low dimension case) or \( n = 50, p = 300 \) (high dimension and sparsity case).

We generated 50 samples for \( X \) and \( e \) and obtained corresponding re-
sponses $y$. The simulation results are displayed in Figure 1. For description simplicity, we use hereafter P1 for lst-enet, P2 for lasso, P3 for Lars, P4 for enetLTS, P5 for enet in the Figures.

Inspecting Figure 1 reveals that (i) with respect to (w.r.t.) FSDR, all four perform equally well with 0% mis-discovery rate; (ii) w.r.t. TSDR, lst-enet, Lars, and lasso perform stably and at a highest rate while enet with a less stable lower rate but enetLTS performs most unstable with the lowest median rate; (iii) w.r.t. RMSE, lasso and lars are the best followed by lst-enet, enetLST has the median RMSE that is also among the best but with the widest spread of RMSE while enet has the largest (and wider spread of) RMSE; (iv) w.r.t. L2-error, lst-enet, lars, and lasso are among the best while enetLTS has the worst performance followed by enet. Overall, lst-enet, lars, and lasso are among the best whereas enetLTS performs worst overall followed by enet.

**Example 7.2** Perfect normal data are not realistic in practice. We now consider $\varepsilon = 0.05$ (i.e. 5% contamination), all others are the same as Example 7.1 except the contamination scheme II will be adopted (in 7.1 contamination scheme does not matter). We first consider $n = 100, p = 50$ (low dimension case) and for simplicity generate data according design I. Performance of five procedures in 50 samples is displayed in Figure 2.
7.1 Simulation

Figure 2: Boxplots for five procedures and 50 samples each with \( n = 100 \) and \( p = 50 \) that are generated from design I with 5% contamination rate.

Inspecting the Figure reveals that (i) w.r.t. L2-error, lst-enet, enetLTS and enet are the best performers while lasso and lars are equally dissatisfactory; (ii) w.r.t. RMSE, the situation is the same as in the L2-error case; (iii) w.r.t. TSDR, enet is the best performer (this perhaps is false best since it might assign zero to all components of the estimator \( \hat{\beta} \) that could lead to 100% of its FSDR) while enetLTS is the worst; (iv) w.r.t. FSDR, lst-enet, enetLTS are the best performers followed by lasso and lars while enet is the loser. Overall, lst-enet is the only winner.
7.1 Simulation

The simulation study above with 5% contamination is repeated but \( n = 50 \) and \( p = 300 \) (high dimensional case) and simulation design II is adopted. Results are displayed in Figure 3.

Reviewing the Figure discovered that (i) w.r.t. L2-error, lst-enet, enetLTS, and enet are the best performers while lasso and lars are disappointing; (ii) w.r.t. RMSE; the situation is almost the same as in L2-error case; (iii) w.r.t. TSDR, lst-enet, lasso, lars and enet are the best performers while
enetLTS is the loser; (iv) w.r.t. FSDR, enet is the worst performer (since its FSDR is almost 100%), enetLTS has the lowest median value while it has the widest spread. lst-enet is the second best performer, lasso and lars are disappointed. Overall, lst-enet is the only winner.

Figure 4: Boxplots for four procedures and 50 samples each with \( n = 50 \) and \( p = 300 \) that are generated from design I with 10% contamination rate.

**Example 7.3** In practice, 10% (or even 20%) contamination is not rare. Next we consider the case \( \varepsilon = 0.1 \) (i.e., 10% contamination), contamination scheme II will be adopted. Samples of 50 with \( n = 50, p = 300 \) are generated with simulation design I. Due to the higher level contamination and the
usage of R package glmnet in its background CV calculation, enetLTS fails to go through the computation we have to drop it in our comparison. Simulation results are displayed in Figure 4.

Inspecting the Figure reveals that (i) w.r.t. L2-error, lst-enet and enet are the best while lasso and lars are inferior; (ii) w.r.t. RMSE, the situation is the same as in L2-error case; (iii) w.r.t. TSDR, enet is the worst performer (it assigns zeros to almost all components of \( \hat{\beta} \) that will lead to 100% of its FSDR), lst-enet and lasso are the best performers followed by lars; (iv) w.r.t. FSDR, lst-enet is the best performer, enet is the worst one while lasso and lars perform dissatisfactory. Overall, lst-enet is the winner.

The advantage of lst-enet is even better demonstrated in Figure 5 when \( n = 100 \) and \( p = 50 \) and \( \varepsilon = 0.2 \) (i.e., 20% contamination).

7.2 A read data example

Example 7.4 To analyze a realistic dataset with very large number of variables, we consider the well-known cancer data from the National Cancer Institute (NCI60); see [Tan et al. (2009)] for more detail about this dataset. A total of 59 of the human cancer cell-lines (n = 59) were assayed for gene expression and protein expression. The data set, downloadable from the CellMiner program package, NCI (http://discover.nci.nih.gov/cellminer/) and available from the R package robustHD, has been repeatedly studied.
7.2 A read data example

Figure 5: Boxplots for four procedures and 50 samples each with $n = 100$ and $p = 50$ that are generated from design I with 20% contamination rate.

in the literature, see e.g., Lee et al. (2011).

We process the data set by following the approach in the literature and treat the gene expression microarray data as the predictors $X_{\text{raw}}$ (a 59 by 22283 matrix) and the protein expression data as responses variables $Y_{\text{raw}}$ (a 59 by 162 matrix). Similar to Lee et al. (2011) or Alfons, Croux and Gelper (2013), we order the protein expression variables according to their scale (employing MAD as a scale estimator instead of the standard deviation) and select the one with median MAD, serving as our dependent variable.
It is the column of the protein expression data matrix. Denote it by $Y$.

Next, we selected out genes using their correlations with $Y$. Here we adopt the robust correlation measure in Khan, Van Aelst and Zamar (2007). We obtain 22283 ordered (decreasing) correlations and select top $k_1 = 100$ corresponding columns of $X_{raw}$ and combined with the bottom $1000 - k_1$ columns as our final $X$, reducing the number of genes from 2,2283 to $p = 1,000$. The number $p$ could easily be changed by adjusting $k_1$.

We partition (by rows) $X_{50 \times 1000}$ and $Y$ into $x.train$, $y.train$ and $x.test$, $y.test$ according the rate 7 : 3. That is 41 rows of $X$ and $Y$ for the training data sets, the rest 18 rows as testing data sets. We do this step 50 times.
and each time we calculate the RMSE (the only measure that still valid without the given $\beta_0$) for the five procedures. The results are displayed in Figure 6 where the fifth performance measure is introduced, that is, the empirical mean squared error defined as:

$$EMSE(\hat{\beta}_P) := \frac{1}{R} \sum_{i=1}^{R} (\hat{\beta}_P - \overline{\beta}_P)^2,$$

(7.28)

where $R$ is the replication number, namely, $\frac{R}{R-1}EMSE$ is the sample variance of $\hat{\beta}_P$. $\overline{x}$ stands for the sample mean of $x_i$.

Inspecting the Figure reveals that (i) lasso, LSRS, enetLTS (enetLTS has the wider spread) have the smallest RMSE but their sample variances (EMSE) are among the largest; (ii) RMSE of the lst-enet is the second smallest but it is the most stable estimator with the distinguished smallest sample variance (EMSE) which means that with different training and testing data sets obtained by random partitioning, lst-enet produces very closed solutions; (iii) enetLTS has the lowest median RMSE but its sample variance is the remarkably large; (iv) enet has the categorical largest RMSR while its sample variance is also the largest. Overall, lst-enet is recommended with the rivals enetLTS, lasso and lars.
8. Concluding discussions

Most of leading penalized regression estimators for high-dimensional sparse data can breakdown by a single outlier (or contaminating point). The newly proposed lst-enet estimator not only processes a high breakdown robustness but also performs well in simulation studies and a read data example, serving as a robust alternative to regularized regression estimators.

Robust measure Finite sample breakdown point has been served as a prevailing quantitative robustness measure in finite sample practice, the main advantage/beauty is its non-randomness and probability-free nature that is exactly why it was enthusiastically welcomed and quick became adopted in a broad spectrum of disciplines after its introduction in 1983.

Critics (e.g., She, Wang and Shen (2022)), however, would like to have a more complicated version, a version that includes randomness and Orlicz norm. They argued that worst case performance might not be a good robustness measure. On the other hand, it is common practice to use the worst case performance as in the complexity of an algorithm or the safety of a passenger cars case.

Future possible work (a) Further performance measure could be pursued including (i) whether \( \hat{\beta} \) performs well on future samples (i.e., whether
$E(Y - w'\hat{\beta})^2$ is small); (ii) whether $\hat{\beta}$ closely approximates the “true” parameter $\beta_0$ (i.e., whether $\|\hat{\beta} - \beta_0\|$ is small with high probability); or (iii) whether $\hat{\beta}$ correctly identifies the relevant coordinates of the “true,” sparse parameter $\beta_0$ (i.e., whether $(\beta_{0j} = 0) \Leftrightarrow (\hat{\beta}_j = 0)$ with high probability).

(b) Extension of current regression work to a more general setting to cover discriminant analysis, logistic regression, and other topics.

**Supplementary Material** It includes the proofs of theorems, associated remarks and computation details and pseudocode.

**Acknowledgments**

The author thanks Prof.s Hao lei Weng, Yiyuan She and Wei Shao for insightful comments and stimulating discussions which improved the manuscript.

**References**


REFERENCES


REFERENCES


Department of Statistics and Probability
Michigan State University
East Lansing, MI 48823

E-mail: (zuo@msu.edu)