American Journal of Theoretical and Applied Statistics
2015; 4(3): 78-84
Published online March 30, 2015 (http://www.sciencepublishinggroup.com/j/ajtas)
doi: 10.11648/j.ajtas.20150403.12
ISSN: 2326-8999 (Print); ISSN: 2326-9006 (Online)

Robust Linear Regression Using L_1-Penalized MM-Estimation for High Dimensional Data

Kamal Darwish, Ali Hakan Buyuklu

Yildiz Technical University, Department of Statistics, Istanbul, Turkey

Email address:
kdarweesh.scom@gmail.com (K. Darwish), hbuyuklu@yildiz.edu.tr (A. H. Buyuklu)

To cite this article:
Kamal Darwish, Ali Hakan Buyuklu. Robust Linear Regression Using L_1-Penalized MM-Estimation for High Dimensional Data. *American Journal of Theoretical and Applied Statistics*. Vol. 4, No. 3, 2015, pp. 78-84. doi: 10.11648/j.ajtas.20150403.12

Abstract: Large datasets, where the number of predictors p is larger than the sample sizes n, have become very popular in recent years. These datasets pose great challenges for building a linear good prediction model. In addition, when dataset contains a fraction of outliers and other contaminations, linear regression becomes a difficult problem. Therefore, we need methods that are sparse and robust at the same time. In this paper, we implemented the approach of MM estimation and proposed L_1-Penalized MM-estimation (MM-Lasso). Our proposed estimator combining sparse LTS sparse estimator to penalized M-estimators to get sparse model estimation with high breakdown value and good prediction. We implemented MM-Lasso by using C programming language. Simulation study demonstrates the favorable prediction performance of MM-Lasso.

Keywords: MM Estimate, Sparse Model, LTS Estimate, Robust Regression

1. Introduction

In modern real-world applications, high-dimensional datasets, where the total number of variables p is much larger than sample size n, but the number of important variables is typically smaller than n, are not uncommon. Examples are gene expression microarray data and functional Magnetic Resonance Imaging (fMRI) data. A typical goal in sparse high-dimensional model fitting is to ensure high prediction accuracy and discovering relevant predictive variables. However, ill-conditioned design matrix in high-dimensional model makes statistical estimation is fundamentally different from the estimation problems in the classical settings.

Regularized or penalized estimations have been widely used to overcome the computational problems with high-dimensional data and to improve prediction accuracy. A penalty parameter can be added to the objective function on the regression coefficients to tradeoff between variance and bias as ridge estimation [1]. A popular approach is the least absolute shrinkage and selection operator (Lasso) [2] that uses the L_1 penalty.

Consider the linear regression problem in the matrix form

$$y_i = \mathrm{x}_i'\beta + \varepsilon_i, \tag{1}$$

where y is an n-vector of random responses, X an $n \times p$ design matrix, β a p-vector of parameters, and ε an n-vector of iid random errors have zero expected value. With a penalty parameter λ, the Lasso estimate of β is

$$\hat{\beta}_{lasso} = \arg\min\left\{L(\mathrm{x}, y, \beta) : \beta \in \mathbb{R}^p\right\} \tag{2}$$

where the loss function L is

$$L(\mathrm{x}, y, \beta) = \sum_{i=1}^{n}(y_i - \mathrm{x}_i'\beta)^2 + n\lambda\sum_{j=1}^{p}\left|\beta_j\right| \tag{3}$$

As with ridge regression (Lasso) shrinks the coefficients, however. L_1-penalty forces some of the coefficient estimates to be exactly equal to zero if the tuning parameter λ is sufficiently large, i.e., to produce sparse model estimates that are highly interpretable. Hence, much like best subset selection, the Lasso performs variable selection. It can effectively select important explanatory variables and estimate regression parameters simultaneously. In contrast to classical L_0 penalized variable selection methods, AIC, BIC, C_P and so on, the Lasso is computationally feasible for high-dimensional data. A fast algorithm for computing the Lasso is available through the framework of least angle regression (LARS) [3]. The satisfactory finite-sample performance of Lasso under normal errors has been demonstrated numerically in [2], and its statistical properties have been studied (e.g [4, 5]).

The main drawback of the Lasso is that it is not robust to outliers. The breakdown point of the Lasso is 1/n [6] i.e., even a single outlier can severely distort the Lasso estimate completely.

To produce more robust estimator than Lasso, the least absolute deviations (LAD) regression is combined with Lasso regression to produce an estimator called LAD-lasso [7],

$$\hat{\beta}_{LAD-lasso} = \arg\min \sum_{i=1}^{n} |y_i - \mathrm{x}_i'\beta| + n\lambda \sum_{j=1}^{p} |\beta_j| \qquad (4)$$

All estimators mentioned until now, are a special case of a more general estimator, the penalized M-estimator [8],

$$\hat{\beta}_M = \arg\min \sum_{i=1}^{n} \rho(y_i - \mathrm{x}_i'\beta) + n\lambda \sum_{j=1}^{p} J(\beta_j), \qquad (5)$$

with loss function $\rho : \mathrm{R} \rightarrow \mathrm{R}$ and penalty function $J : \mathrm{R} \rightarrow \mathrm{R}$. Using convex loss functions make these estimators non robust with respect to leverage points and result in a breakdown point of 1/*n* [6].

A robust version of ridge regression was proposed by [9], using L_2 penalized MM-estimators. Even though the resulting estimates are not sparse, prediction accuracy is improved by shrinking the coefficients, and the computational issues with high-dimensional robust estimators are overcome due to the regularization.

Ref [10] proposed a robust estimator with respect to leverage points, called RLARS. RLARS is a robust version of the stepwise algorithm LARS which is computationally very efficient but sensitive to outliers. A main drawback of the RLARS algorithm is the lack of a natural definition, since it is not optimizing a clearly defined objective function.

A popular robust estimator is the least trimmed squares (LTS) estimator [11]. Although its simple definition and fast computation make it interesting for practical application, it cannot be computed for high-dimensional data ($p > n$). Combining the Lasso estimator with the LTS estimator, developed the sparse LTS-estimator [6],

$$\hat{\beta}_{spLTS} = \arg\min \frac{1}{h} \sum_{i=1}^{h} r_{(i)}^2(\beta) + \lambda \sum_{j=1}^{p} |\beta_j|, \qquad (6)$$

where $r_i^2(\beta) = (y_i - x_i\beta)^2$ denotes the squared residuals and $r^2_{(1)}(\beta) \leq \ldots \leq r^2_{(\mathrm{n})}(\beta)$ their order statistics. Here $\lambda \geq 0$ is a penalty term and $h \leq n$ the size of the subsample that is deemed to consist of non-outlying observations. This estimator can be applied to high-dimensional data with good prediction performance and high robustness. It also has a high breakdown point [6].

However, sparse LTS can be applied to high-dimensional data, it should be noted that its efficiency is an issue. In this paper, we attempt to combine reweighting sparse LTS estimator with penalized M-estimators (Tukey's biweight functions with L_1-penalty). We employ the approach of MM estimation which was first proposed in [12]; using sparse LTS as an initial estimator for computing L_1-penalized M-estimator yields L_1- penalized MM-estimator.

The rest of the paper is organized as follows. In Section 2, we illustrate the proposed algorithm to combine high breakdown value estimation and efficient estimation. Section 3 presents the results of a simulation study that compares the performance of the estimated models by the *root mean squared prediction error* (RMSPE). In addition concerning sparsity, the estimated models are evaluated by the *false positive rate* (FPR) and the *false negative rate* (FNR). The results indicate that, L_1-Penalized MM-estimation yields a model that achieves excellent prediction accuracy with a sparse representation of the predictors in the model. Finally, Section 4 concludes.

2. L_1-Penalized MM-Estimation

To combine high breakdown value estimation with efficient estimation under the normal model, the approach of MM estimation is employed. The L_1-Penalized MM-estimation (henceforth MM-Lasso) can be constructed by a three-stage procedure. In the first stage, we compute an initial consistent estimator $\hat{\beta}_0$ with high breakdown point(BDP) but possibly low normal efficiency. In the second stage, we compute a robust M-scale estimator $\hat{\sigma}$ of the residuals based on the initial estimate. In the third stage, we compute an L_1-Penalized M estimator with fixed scale $\hat{\sigma}$; starting the iterations from $\hat{\beta}_0$; and using a loss function that ensures the desired efficiency. Here, efficiency will be loosely defined as similarity with the classical lasso estimator at the normal model.

Let $\rho_0(r) = \rho_{\mathrm{BI}}(r/k_0)$, $\rho(r) = \rho_{\mathrm{BI}}(r/k_1)$, and assume that each of the ρ-functions is bounded even in the sense of [13]. The scale M estimator (an M-scale for short) $\hat{\sigma}$ satisfies

$$\frac{1}{n-\hat{q}} \sum_{i=1}^{n} \rho_0\left(\frac{r(\beta)}{\hat{\sigma}}\right) = \delta. \qquad (7)$$

Where $\hat{q}$ is the number of non-zero estimated parameters in $\hat{\beta}$ which depends on λ.

To obtain consistency when the errors are normal, the constant δ satisfies $\delta = E_\Phi[\rho(Z)]$, with Φ the standard normal distribution. Note that if $\rho(t) = t^2$ and $\delta = 1$ then $\hat{\sigma} = s$ the residual standard deviation . The MM-Lasso is defined with

$$L(\mathrm{x}, y, \beta) = \hat{\sigma}^2 \sum_{i=1}^{n} \rho\left(\frac{r_i(\beta)}{\hat{\sigma}}\right) + n\lambda \sum_{j=1}^{p} |\beta_j| \qquad (8)$$

where the factor $\hat{\sigma}^2$ before the summation is employed to make the estimator coincide with the classical one when $\rho(t) = t^2$. Let ρ satisfy $\rho \leq \rho_0$, from [13] it is easy to show if $\hat{\beta}$ satisfies $L(\mathrm{x}, \mathrm{y}, \hat{\beta}) \leq L(\mathrm{x}, \mathrm{y}, \hat{\beta}_0)$, then $\hat{\beta}$'s BDP is not less than that of $\hat{\beta}_0$. The value of k_0 should be chosen in order to attain high breakdown point of the MM-Lasso. The choice of k_1 will to determine asymptotic efficiency of the estimate without affecting its breakdown point. In order to let $\rho \leq \rho_0$, we must have $k_1 \geq k_0$; the larger the k_1 is, the higher efficiency the MM-Lasso can attain at the normal distribution.

2.1. IRLS Algorithm

The penalty function in equation (2) is convex but it is a non-differentiable function, hence it is difficult to obtain analytic form solution of equation (2). Here we can obtain an approximate closed form solution as in [2]:

$$\hat{\beta}_{lasso} = \arg\min \sum_{i=1}^{n} (y_i - X_i\beta)^2 + n\lambda \sum_{j=1}^{p} \left(\beta_j^2 / |\beta_j| \right) \quad (9)$$

where the L_1 Penalization had a similar form to an L_2 norm with a weight $|\beta_j|$. Iteratively re-weighted least squares (IRLS) algorithm for the Lasso estimate in equation (9) can be obtained by computing the ridge regression iteratively as:

$$\hat{\beta}_{lasso}^{(i+1)} = (X'X + \lambda\Lambda^{(i)})^{-1}(X'y), \quad (10)$$

where $\Lambda^{(i)}$ is the generalized inverse (pseudo-inverse) of matrix= $\text{diag}\left\{ |\beta_{lasso,1}^{(i)}|, |\beta_{lasso,2}^{(i)}|, ..., |\beta_{lasso,p}^{(i)}| \right\}$ and i = 0,1... is the iteration number. Similarly, an iteratively re-weighted least squares (IRLS) algorithm for the MM-Lasso estimate. Define

$$\psi(t) = \rho'(t), \quad W(t) = \frac{\psi(t)}{t} \quad (11)$$

Let

$$t_i = \frac{r_i}{\hat{\sigma}}, \quad w_i = \frac{W(t)}{2},$$

$$\boldsymbol{w} = (w_1, w_2, ..., w_n)', \quad \boldsymbol{W} = diag(\boldsymbol{w}). \quad (12)$$

by differentiating of (5) with respect to β and setting the derivative to zero, one gets,

$$\boldsymbol{w}' = (y - X\hat{\beta}) = 0 \quad (13)$$

and

$$\hat{\beta}_{MM-lasso}^{(i+1)} = (X'\boldsymbol{W}^{(i)}X + \lambda\Lambda^{(i)})^{-1}(X'\boldsymbol{W}^{(i)}y). \quad (14)$$

Since for the chosen ρ, W(t) is a decreasing function of |t|, observations with larger residuals will receive lower weight w. The iteration will stop until a maximum number is reached or the difference between two successive iteration steps is small enough.

The following is the procedure to obtain the estimator $\hat{\beta}$

1) A high breakdown estimator is used to find an initial estimate $\hat{\beta}_0$ (in this paper we choose sparse LTS estimator in [6]). Using this estimate the residuals, $r_i(\hat{\beta}_0) = y_i - x_i'\hat{\beta}_0$, are computed, for $1 \le i \le n$.
2) Using these residuals from the robust fit, an M-estimate of scale $\hat{\sigma}$ with high BDP is computed from (6).
3) At each iteration with $\hat{\sigma}$ remains fixed throughout, calculate residuals $r_i^{(j-1)}$ and associated weight $\boldsymbol{w}\left(r_i^{(j-1)}\right)$ according to the weight function.
4) Solve the following for iteratively re-weighted least squares (IRLS) equation,

$$\hat{\beta}_{MM-lasso}^{(j)} = (X'\boldsymbol{W}^{(j-1)}X + \lambda\Lambda^{(j)})^{-1}(X'\boldsymbol{W}^{(j-1)}y),$$

Steps 3) and 4) are repeated until $\frac{\left|r_i^{(j)} - r_i^{(j-1)}\right|}{r_i^{(j-1)}}$ becomes less than tolerance.

2.2. Weight Functions and Choosing the Constants

Several types of weight functions are proposed for IRLS algorithm in literature. Each set of functions given includes tuning constants, which allow for the shape of the function to be slightly altered. Beaton and Tukey [14] proposed the IRLS algorithm with Tukey's bisquare function that enables to remove the influence of extreme outliers completely from the estimation.

$$\rho_{BI}(t) = \begin{cases} \frac{k_{BI}}{6} \cdot \left[1 - (1 - (t/k_{BI})^2)^3 \right] & if \ |t| \le k_{BI}, \\ \frac{k_{BI}}{6} & if \ |t| > k_{BI}. \end{cases} \quad (15)$$

$$\psi(t)_{BI} = \begin{cases} (1 - (t/k_{BI})^2)^2 & if \ |t| \le k_{BI}, \\ 0 & if \ |t| > k_{BI}. \end{cases} \quad (16)$$

where $\psi(t) = \rho'(t)$ is called a redescending function, and the value k_{BI} for bisquare function is a *tuning constant*. In particular, The value k_0 =2.937 such that the asymptotically consistent scale estimate $\hat{\sigma}$ has the breakdown value of 25%, while the value k_1 = 3.44 yields 0.85 asymptotic efficiency at the normal model when λ = 0 [11].

However, the scale estimate $\hat{\sigma}$ requires a correction for high dimensional data. According to [15], there are two problems appear when fitting a standard MM estimator to data with a high ratio p/n:

(1) The scale based on the residuals from the initial regression estimator underestimates the true error scale.
(2) Even if the scale is correctly estimated, the actual efficiency of the MM estimator can be much lower than the nominal one. For this reason, $\hat{\sigma}$ is corrected using (9) in [15] as

$$\tilde{\sigma} = \frac{\hat{\sigma}}{1 - (k_1 + k_2/n)\hat{q}/n} \quad \text{with } k_1 = 1.29, \ k_2 = -6.02.$$

2.3. Choosing the Penalty Parameter

We propose to select λ by the estimated prediction error of MM-Lasso for different values of λ via cross-validation. We can use the k-fold cross validation process, which requires recomputing the estimate k times. For k = n ("leave-one-out") we can use an approximation to avoid recomputing.

Call $\hat{y}_{-i}$ the fit of y_i computed without using the i-th observation; i.e., $y_{-i} = x_i'\hat{\beta}^{(-i)}$, where $\hat{\beta}^{(-i)}$ is the MM-Lasso estimate computed without observation i. Then a first-order

Taylor approximation of the estimator yields the approximate prediction errors

$$r_{-i} = y_i - \hat{y}_{-i} \approx \left(1 + \frac{W(t_i)h_i}{1-h_i\psi'(t_i)}\right) \quad (16)$$

with

$$h_i = x_i'\left(\sum_{i=1}^{n}\psi'(t_i)x_i x_i' + 2\lambda\Lambda^{(i)}\right)^{-1} x_i$$

where, ψ and t_i are defined in (11)-(12), x_i is the i-th row of X defined in (13) and $\Lambda^{(i)}$ is the generalized inverse (pseudo-inverse) defined in (10). Given the prediction errors $\boldsymbol{r}_{-} = (r_{-1},...,r_{-n})'$, we compute a robust mean squared error (MSE) as $\tau(r_{-})^2$, where τ is a "τ-scale" with tuning constant $c_\tau = 5$ [16], and choose the λ minimizing this MSE.

3. Simulation Study

To investigate the behaviour of our robust estimator, a simulation study for comparing the performance of various sparse estimators are performed in R (R Development Core Team, 2011) with package simFrame[17], which is a general framework for simulation studies in statistics. As in [6] we make a comparison with the lasso, the LAD-Lasso, robust least angle regression (RLARS) and Sparse LTS with reweighted step. Sparse LTS is evaluated for the subset size $h = \lfloor (n+1)0.75 \rfloor$ to guarantee a breakdown point of 25%. All computations are carried out in R version 3.1.2 (R Development Core Team, 2011) using the packages robustHD [18] for sparse LTS and RLARS, quantreg [19] for the LAD-Lasso and lars [20] for the Lasso. We implemented MM-Lasso by using C programming language.

For every generated sample, an optimal value of the shrinkage parameter λ is selected. The penalty parameters for MM-Lasso are chosen using k-fold cross validation process as described in subsection 2.4, and the other methods are optimized via BIC as described in [6].

3.1. Sampling Schemes

In this study we take the three configurations from [6] to represent low, moderate and high dimensional data. Firstly in the case of $n > p$, we create a linear model. From $k = 6$ latent independent standard normal variables, L_1 , L_2 , ... , L_k and an independent normal error variable e with standard deviation σ, the response variable y is constructed as

$$y = L_1 + L_2 + \cdots + L_k + \sigma\varepsilon, \quad (16)$$

The value of σ is chosen so that the signal to noise ratio is equal to 3. A set of $p = 50$ candidate predictors is then constructed as follows. Let e_1, ..., e_d be independent standard normal variables and let

$$X_i = L_i + \tau e_i,\ i = 1, ..., k$$

$$X_{k+1} = L_1 + \delta e_{k+1}$$

$$X_{k+2} = L_1 + \delta e_{k+2}$$

$$X_{k+3} = L_2 + \delta e_{k+3}$$

$$X_{k+4} = L_2 + \delta e_{k+4}$$

$$\vdots$$

$$X_{3k-1} = L_k + \delta e_{3k-1}$$

$$X_{3k} = L_k + \delta e_{3k}$$

and $X_i = e_i$, $i = 3k + 1$, ..., p

The constants $\delta = 5$ and $\tau = 0.3$ are chosen so that corr(X_1, X_{k+1}) = corr(X_1, X_{k+2}) = corr(X_2, X_{k+3}) = · · · = corr(X_k, X_{3k}) = 0.5. Note that covariates X_1, ..., X_k are "low noise" perturbations of the latent variables and constitute our "target covariates". Variables X_{3k+1}, ..., X_d are independent noise covariates and variables X_{k+1}, ..., X_3. The number of observations is set to $n = 150$.

The case of moderate high-dimensional data is represented by the second configuration. We generate $n = 100$ observations from a p-dimensional normal distribution $N(0, \Sigma)$, with $p = 250$. The covariance matrix $\Sigma = (\Sigma_{ij})$ $1 \le i,j \le p$ is given by $\Sigma_{ij} = 0.5^{|i-j|}$, creating correlated predictor variables. Using the coefficient vector $\beta = (\beta_j)$ $1 \le j \le p$ with $\beta_1 = \beta_7 = 1.5$, $\beta_2 = 0.5$, $\beta_4 = \beta_{11} = 1$, and $\beta_j = 0$ for $j \in \{1, \ldots, p\} \setminus \{1, 2, 4, 7, 11\}$, the response variable is generated according to the regression model (1), where the error terms follow a normal distribution with $\sigma = 0.5$.

Finally, the third configuration covers the case of high dimensional data with $n = 100$ observations and $p = 500$ variables. The first 250 predictor variables are generated from a multivariate normal distribution $N(0, \Sigma)$ with $\Sigma_{ij} = 0.6^{|i-j|}$. Furthermore, the remaining 250 covariates are standard normal variables. Then the response variable is generated according to (1), where the coefficient vector $\beta = (\beta_j)$ $1 \le j \le p$ is given by $\beta_j = 1$ for $1 \le j \le 10$ and $\beta_j = 0$ for $11 \le j \le p$, and the error terms follow a standard normal distribution.

To allow for a fraction of outliers we considered the following sampling distributions, listed in increasing order of difficulty

1. No contamination.

2. Vertical outliers: 10% of the error terms in the regression model follow a normal N(20, σ) instead of a N(0, σ). 3. Leverage points: Same as in 2, but the 10% contaminated observations contain high-leverage values by drawing the predictor variables from independent N(50,1) distributions.

4. The outliers form a dense cluster: Keeping the contamination level at 10%, outliers in the predictor variables are drawn from independent N(10, 0.01) distributions. Let $\tilde{x}_i$ denote such a leverage point. Then the values of the response variable of the contaminated observations are generated by $\tilde{y}_i = \eta\tilde{x}_i$ with $\gamma = (-1/p)_{1 \le j \le p}$. The parameter η controls the magnitude of the leverage effect and is varied from 1 to 25 in five equidistant steps.

3.2. Simulation Results

In this subsection, the results for the different data scheme are presented and discussed. The performance of the estimated models are compared by the *root mean squared prediction error* (RMSPE). For this purpose, we generate *n* additional observations from the respective sampling schemes (without outliers) as test data, and this in each simulation run. The RMSPE of the oracle estimator, which uses the true coefficient values β, is computed as a standard for the evaluated methods. In addition considering sparsity, the estimated models are evaluated by the *false positive rate* (FPR) and the *false negative rate* (FNR). Both FPR and FNR should be as small as possible for a sparse estimator. RMSPE, FPR and FNR, averaged over 100 simulation runs, are reported for every method.

3.2.1. The First Sampling Scheme

The simulation results for the first data are represented in table 1. It can be seen that when there is no contamination in the data LAD-Lasso, RLARS and Lasso have excellent performance in RMSPE and FPR, while sparse LTS and MM-Lasso have a good prediction, but they have larger FPR than other methods. In addition, MM-Lasso improves the estimates of sparse LTS, which is reflected in the lower values for RMSPE and FPR. On the other hand, there are no false negatives in all of these methods.

In the case of vertical outliers, the higher values of RMSPE and FPR show that Lasso is non-robust estimator.

All of methods are still have excellent performance in RMSPE but sparse LTS and MM-Lasso have considerable values of FPR. As showed in Table 1 RMSPE and FPR of MM-Lasso are 1.1765, 0.237 while sparse LTS have RMSPE and FPR equals 1.2378, 0.293 respectively. Ultimately, MM-Lasso has a significant improvement over Sparse LTS. In the third scenario, when we introduce leverage points in addition to vertical outliers, RLARS, MM-Lasso, and sparse LTS have a good performance. However, the RMSPE and FPR of RLARS increased (1.1210 to 1.2236, and 0.029 to 0.126, respectively) also the FPR of sparse LTS (0.293 to 0.319) and MM-Lasso (0.237 to 0.250) slightly increase. MM-Lasso still increases the performance of sparse LTS in RMSPE and FPR (1.1792 and 0.250 respectively). LAD-lasso has large RMSPE and suffers from false positives, while Lasso has large RMSPE and FNR. This suggests that the leverage points have a bad leverage effect.

Figure 1 refers to the results for the fourth contamination setting. The RMSPE for the more robust methods is plotted as a function of the parameter η. RLARS has a considerably higher RMSPE than MM-Lasso and sparse LTS for lower values of η, but the RMSPE gradually decreases with increasing η. The RMSPE of sparse LTS in beginning slightly increased then decreased in the next steps. However, MM-Lasso has the lowest RMSPE; thus, their overall performance is the best.

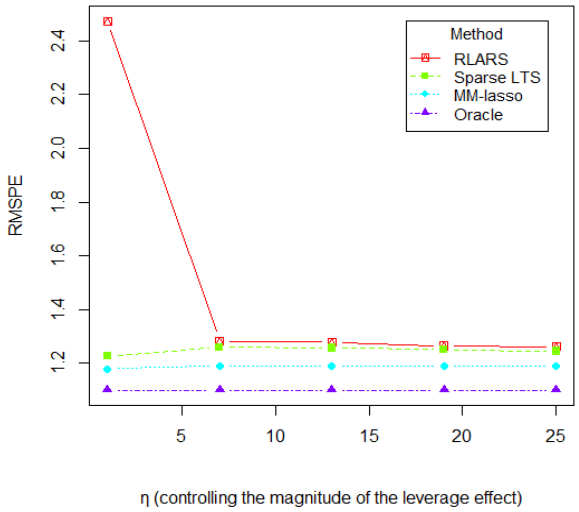

Figure 1. *Root mean squared prediction error (RMSPE) for the first simulation scheme, with n =150 and p = 50, and for the fourth contamination setting.*

Table 1. *Results for the first simulation scheme, with n = 150 and p = 50.*

Method	No contamination			Vertical outliers			Leverage points		
	RMSPE	FPR	FNR	RMSPE	FPR	FNR	RMSPE	FPR	FNR
Lasso	1.1778	0.080	0.000	1.7376	0.225	0.088	2.5205	0.066	0.766
LAD-Lasso	1.1316	0.092	0.000	1.1640	0.161	0.000	1.9939	0.316	0.002
RLARS	1.1450	0.066	0.000	1.1210	0.029	0.000	1.2236	0.126	0.030
Sparse LTS	1.2623	0.265	0.000	1.2378	0.293	0.000	1.2345	0.319	0.000
MM-Lasso[a]	1.1705	0.213	0.000	1.1765	0.237	0.000	1.1792	0.250	0.000
Oracle	1.1073			1.1073			1.1073		

a Proposed method.

3.2.2. The Second Sampling Scheme

Table 2 shows the simulation results for the second data configuration (the moderate high-dimensional data). In the case without contamination, MM-Lasso, and RLARS have the best performance. Also, the LAD-Lasso and Lasso have excellent prediction performance but a slightly higher FPR than the other methods, followed by sparse LTS. In the case of vertical outliers, RLARS still has excellent prediction performance despite some false negatives. We notice that RMSPE and FPR of MM-Lasso are 0.5766 and 0.034 respectively. While, for Sparse LTS are 0.6688, 0.039 respectively. Hence, MM-Lasso achieves good sparse prediction without false negative. Drastically, Lasso is non-robust against vertical outliers.

In the scenario with additional leverage points, it can be concluded that sparse LTS has RMSPE equal 0.6691 and FPR equal 0.039 also MM-Lasso has RMSPE equal 0.5738 and

FPR equal 0.035. It means that there is stability in these methods. For RLARS, there is small increase in the RMSPE, FPR and FNR. On the other hand, LAD-Lasso already has a considerably large RMSPE, and again a slightly higher FPR than the other methods. Furthermore, the Lasso is still highly influenced by the outliers, which is reflected in a very high FNR and poor prediction performance. Briefly, compared to other methods MM-Lasso is deemed a best performance.

Figure 2 clarifies the results for the fourth contamination setting. As in first scheme, we plotted the RMSPE for the more robust methods. The RMSPE of RLARS is gradually decreasing. The RMSPE of MM-Lasso and sparse LTS have constant low values. MM-Lasso clearly performs best for all values of η.

Table 2. *Results for the second simulation scheme, with n = 100 and p =250.*

Method	No contamination			Vertical outliers			Leverage points		
	RMSPE	FPR	FNR	RMSPE	FPR	FNR	RMSPE	FPR	FNR
Lasso	0.5848	0.105	0.000	2.3551	0.185	0.092	2.6857	0.013	0.632
LAD-Lasso	0.6020	0.067	0.000	0.7446	0.011	0.000	1.8398	0.096	0.112
RLARS	0.5506	0.016	0.000	0.6092	0.015	0.055	0.7901	0.072	0.098
Sparse LTS	0.7195	0.028	0.000	0.6688	0.039	0.000	0.6691	0.039	0.000
MM-Lasso[a]	0.5526	0.022	0.000	0.5766	0.034	0.000	0.5738	0.035	0.000
Oracle	0.4998			0.4998			0.4998		

a Proposed method.

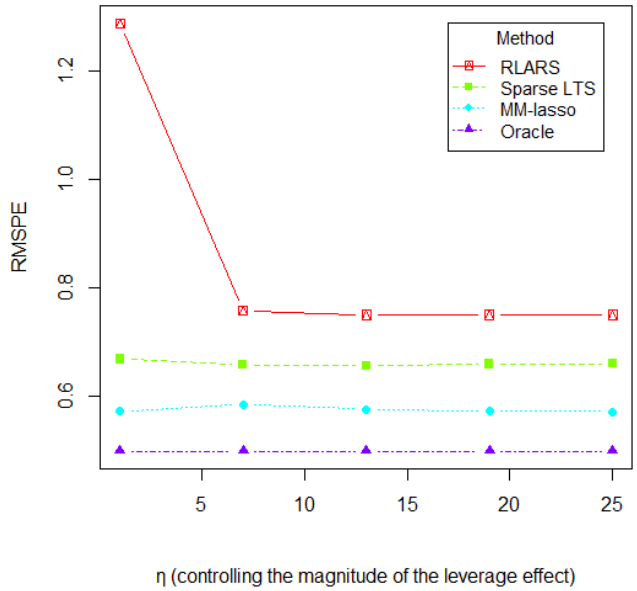

Figure 2. *Root mean squared prediction error (RMSPE) for the second simulation scheme, with n =100 and p = 250, and for the fourth contamination setting.*

3.2.3. The Third Sampling Scheme

Table 3 presents the simulation results for the high dimensional data configuration. When the data is free from contamination, the sparse LTS is characterized as the lowest efficiency due to have larger values of RMSPE than other methods. In the other hand, MM-Lasso and RLARS have considerably better performance in this case. Lasso and LAD-Lasso have a good behavior. With vertical outliers, the RMSPE for the Lasso increases extremely due to a high FNR, while LAD-Lasso still has good prediction performance. In addition, RLARS has a larger FNR, resulting in a slightly lower RMSPE. When leverage points are introduced, MM-Lasso exhibits the lowest RMSPE and sparse LTS keep its excellent behavior.

Figure 3 shows the results for the fourth contamination setting. It can be seen that RMSPE of RLARS is higher in the beginning, and then decreases continuously in the remaining steps. MM-Lasso and Sparse LTS have low and constant values for RMSPE but MM-Lasso is close to Oracle.

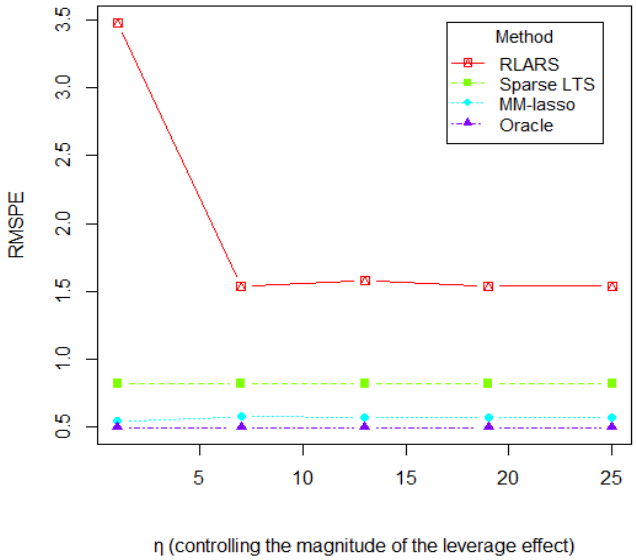

Figure 3. *Root mean squared prediction error (RMSPE) for the third simulation scheme, with n =100 and p =500, and for the fourth contamination setting.*

3.3. Summary of the Simulation Results

This study shows that MM-Lasso has RMSPE values close to Oracle, and does not suffer from any false positives at all. Hence, MM-Lasso is the best overall performance. Sparse LTS generally have good prediction accuracy; however, MM-Lasso can improve this prediction. Although RLARS has good achievement, contamination data makes FNR values increased. These simulation results also enhance that the Lasso is not robust to outliers and LAD-Lasso is not robust

against bad leverage points but is resistant to vertical outliers.

Table 3. *Results for the third simulation scheme, with n =100 and p = 500.*

Method	No contamination			Vertical outliers			Leverage points		
	RMSPE	FPR	FNR	RMSPE	FPR	FNR	RMSPE	FPR	FNR
Lasso	0.6114	0.073	0.000	3.3311	0.061	0.280	3.6073	0.054	0.350
LAD-Lasso	0.6493	0.023	0.000	0.8316	0.006	0.000	2.8667	0.074	0.132
RLARS	0.5767	0.009	0.000	0.8954	0.009	0.081	1.4805	0.052	0.112
Sparse LTS	0.9804	0.006	0.000	0.7912	0.005	0.000	0.7520	0.005	0.001
MM-Lasso[a]	0.5429	0.005	0.000	0.5626	0.004	0.000	0.5725	0.005	0.000
Oracle	0.4983			0.4983			0.4983		

a Proposed method.

4. Conclusion

Sparse Least trimmed squares (Sparse LTS) is a robust, shrinkage and selection regression estimation with high breakdown value and good prediction estimation, However, it should be noted that efficiency is an issue with sparse LTS. Our proposed estimator MM-Lasso, an approach of MM estimation, used sparse LTS estimator as initial estimator to penalized M-estimators (Tukey's biweight functions with L_1-penalty). Our model, MM-Lasso can improve prediction estimation of sparse LTS and its overall performance is the best.

Acknowledgements

We would like to express our sincerely thanks to the Editor, and referees for their valuable and constructive comments that led to considerable improvement of the paper.

References

[1] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," Technometrics, vol. 12, no. 1, pp. 55–67, 1970.

[2] R. Tibshirani, "Regression shrinkage and selection via the lasso," J. Royal. Statist. Soc B., vol. 58, no. 1, pp. 267–288, 1996.

[3] B. Efron, T. Hastie, and R.Tibshirani, "Least angle regression," The Annals of Statistics, vol. 32, pp, 407–499, 2004.

[4] K. Knight and W. Fu, "Asymptotics for Lasso-Type Estimators," The Annals of Statistics, vol. 28, pp. 1356–1378, 2000.

[5] J. Fan and R. Li, "Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties," Journal of the American Statistical Association, vol. 96, no. 456, pp. 1348–1360, 2001

[6] A. Alfons, C. Croux, and S. Gelper, "Sparse least trimmed squares regression for analyzing high dimensional large data sets," The Annals of Applied Statistics, vol. 7, no. 1, pp. 226–248, 2013.

[7] H.Wang, G. Li, and G. Jiang, "Robust regression shrinkage and consistent variable selection through the LAD-lasso," Journal of Business & Economic Statistics, vol. 25, pp. 347-355, 2007.

[8] G. Li, H. Peng, and L. Zhu,"Nonconcave penalized M-estimation with a diverging number of parameters," Statitica Sinica , vol. 21, no. 1, pp. 391–419, 2013.

[9] R. A. Maronna, "Robust ridge regression for high-dimensional data," Technometrics, vol. 53, pp. 44–53, 2011.

[10] J. A. Khan, Aelst, S. Van. and R. H. Zamar, "Robust linear model selection based on least angle regression," Journal of the Statistical Association, vol. 102, pp. 1289–1299, 2007.

[11] P. Rousseeuw and A. Leroy, Robust regression and outlier detection. John Wiley & Sons, 1987.

[12] V. J. Yohai, "High Breakdown-point and High Efficiency Estimates for Regression," The Annals of Statistics, vol. 15, pp. 642-65, 1987.

[13] R. Maronna, D. Martin, and V. Yohai, Robust Statistics. John Wiley & Sons, Chichester. ISBN 978-0-470-01092-1, 2006.

[14] A. E. Beaton, and J. W. Tukey, "The fitting of power series, meaning polynomials, illustrated on band-spectroscopic data," Technometrics, vol. 16, pp. 147-185, 1974.

[15] R. A. Maronna, and V. J. Yohai, "Correcting MM Estimates for Fat Data Sets," Computational Statistics & Data Analysis, vol. 54, pp. 3168-3173, 2010.

[16] V. J. Yohai and R.H. Zamar, "High breakdown-point estimates of regression by means of the minimization of an efficient scale," Journal of the American Statistical Association, vol. 83, pp. 406–413, 1988.

[17] A. Alfons, simFrame: Simulation framework. R package version 0.5, 2012b.

[18] A. Alfons, robustHD: Robust methods for high-dimensional R pakage version 0.1.0, 2012a.

[19] R. Koenker, quantreg: Quantile regression. R package version 4.67, 2011.

[20] T. Hasti and B. Efron, lars: Least angle regression, lasso and forward stagewise. R package version 0.9-8, 2011.