The Annals of Applied Statistics
2013, Vol. 7, No. 1, 226–248
DOI: 10.1214/12-AOAS575

SPARSE LEAST TRIMMED SQUARES REGRESSION FOR ANALYZING HIGH-DIMENSIONAL LARGE DATA SETS

BY ANDREAS ALFONS, CHRISTOPHE CROUX AND SARAH GELPER

KU Leuven, KU Leuven and Erasmus University Rotterdam

Sparse model estimation is a topic of high importance in modern data analysis due to the increasing availability of data sets with a large number of variables. Another common problem in applied statistics is the presence of outliers in the data. This paper combines robust regression and sparse model estimation. A robust and sparse estimator is introduced by adding an L_1 penalty on the coefficient estimates to the well-known least trimmed squares (LTS) estimator. The breakdown point of this sparse LTS estimator is derived, and a fast algorithm for its computation is proposed. In addition, the sparse LTS is applied to protein and gene expression data of the NCI-60 cancer cell panel. Both a simulation study and the real data application show that the sparse LTS has better prediction performance than its competitors in the presence of leverage points.

1. Introduction.

In applied data analysis, there is an increasing availability of data sets containing a large number of variables. Linear models that include the full set of explanatory variables often have poor prediction performance as they tend to have large variance. Furthermore, large models are in general difficult to interpret. In many cases, the number of variables is even larger than the number of observations. Traditional methods such as least squares can then no longer be applied due to the rank deficiency of the design matrix. For instance, gene expression or fMRI studies typically contain tens of thousands of variables for only a small number of observations. In this paper, we present an application to the cancer cell panel of the National Cancer Institute, in which the data consists of 59 observations and 22,283 predictors.

To improve prediction accuracy and as a remedy to computational problems with high-dimensional data, a penalty term on the regression coefficients can be added to the objective function. This approach shrinks the coefficients and reduces variance at the price of increased bias. Tibshirani (1996) introduced the least absolute shrinkage and selection operator (lasso), in which the penalty function is the L_1 norm. Let $\mathbf{y} = (y_1, \dots, y_n)'$ be the response and $\mathbf{X} = (x_{ij})_{1 \le i \le n, 1 \le j \le p}$ the matrix of predictor variables, where n denotes the number of observations and p the number of variables. In addition, let $\mathbf{x}_1, \dots, \mathbf{x}_n$ be the p-dimensional observations,

Received July 2011; revised May 2012.

Key words and phrases. Breakdown point, outliers, penalized regression, robust regression, trimming.

that is, the rows of $\mathbf{X}$. We assume a standard regression model

$$y_i = \mathbf{x}_i' \boldsymbol{\beta} + \varepsilon_i, \tag{1.1}$$

where the regression parameter is $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)'$, and the error terms ε_i have zero expected value. With a penalty parameter λ, the lasso estimate of $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}}_{\text{lasso}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \mathbf{x}_i' \boldsymbol{\beta})^2 + n\lambda \sum_{j=1}^{p} |\beta_j|. \tag{1.2}$$

The lasso is frequently used in practice since the L_1 penalty allows to shrink some coefficients to exactly zero, that is, to produce sparse model estimates that are highly interpretable. In addition, a fast algorithm for computing the lasso is available through the framework of least angle regression [LARS; Efron et al. (2004)]. Other algorithms are available as well [e.g., Wu and Lange (2008)]. Due to the popularity of the lasso, its theoretical properties are well studied in the literature [e.g., Knight and Fu (2000), Zhao and Yu (2006), Zou, Hastie and Tibshirani (2007)] and several modifications have been proposed [e.g., Zou (2006), Yuan and Lin (2006), Gertheiss and Tutz (2010), Radchenko and James (2011), Wang et al. (2011)]. However, the lasso is not robust to outliers. In this paper we formally show that the breakdown point of the lasso is $1/n$, that is, only one single outlier can make the lasso estimate completely unreliable. Therefore, robust alternatives are needed.

Outliers are observations that deviate from the model assumptions and are a common problem in the practice of data analysis. For example, for many of the 22,283 predictors in the NCI data set used in Section 7, (log-transformed) responses on the 59 cell lines showed outliers. Robust alternatives to the least squares regression estimator are well known and studied; see Maronna, Martin and Yohai (2006) for an overview. In this paper, we focus on the least trimmed squares (LTS) estimator introduced by Rousseeuw (1984). This estimator has a simple definition, is quite fast to compute, and is probably the most popular robust regression estimator. Denote the vector of squared residuals by $\mathbf{r}^2(\boldsymbol{\beta}) = (r_1^2, \dots, r_n^2)'$ with $r_i^2 = (y_i - \mathbf{x}_i' \boldsymbol{\beta})^2$, $i = 1, \dots, n$. Then the LTS estimator is defined as

$$\hat{\boldsymbol{\beta}}_{\text{LTS}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{h} (\mathbf{r}^2(\boldsymbol{\beta}))_{i:n}, \tag{1.3}$$

where $(\mathbf{r}^2(\boldsymbol{\beta}))_{1:n} \leq \cdots \leq (\mathbf{r}^2(\boldsymbol{\beta}))_{n:n}$ are the order statistics of the squared residuals and $h \leq n$. Thus, LTS regression corresponds to finding the subset of h observations whose least squares fit produces the smallest sum of squared residuals. The subset size h can be seen as an initial guess of the number of good observations in the data. While the LTS is highly robust, it clearly does not produce sparse model estimates. Furthermore, if $h < p$, the LTS estimator cannot be computed.

A sparse and regularized version of the LTS is obtained by adding an L_1 penalty with penalty parameter λ to (1.3), leading to the *sparse LTS* estimator

$$\hat{\boldsymbol{\beta}}_{\text{sparseLTS}} = \operatorname*{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{h} (\mathbf{r}^2(\boldsymbol{\beta}))_{i:n} + h\lambda \sum_{j=1}^{p} |\beta_j|. \tag{1.4}$$

We prove in this paper that sparse LTS has a high breakdown point. It is resistant to multiple regression outliers, including leverage points. Besides being highly robust, and similar to the lasso estimate, sparse LTS (i) improves the prediction performance through variance reduction if the sample size is small relative to the dimension, (ii) ensures higher interpretability due to simultaneous model selection, and (iii) avoids computational problems of traditional robust regression methods in the case of high-dimensional data. For the NCI data, sparse LTS was less influenced by the outliers than competitor methods and showed better prediction performance, while the resulting model is small enough to be easily interpreted (see Section 7).

The sparse LTS (1.4) can also be interpreted as a trimmed version of the lasso, since the limit case $h = n$ yields the lasso solution. Other robust versions of the lasso have been considered in the literature. Most of them are penalized M-estimators, as in van de Geer (2008) and Li, Peng and Zhu (2011). Rosset and Zhu (2004) proposed a Huber-type loss function, which requires knowledge of the residual scale. A least absolute deviations (LAD) type of estimator called LAD-lasso is proposed by Wang, Li and Jiang (2007),

$$\hat{\boldsymbol{\beta}}_{\text{LAD-lasso}} = \operatorname*{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{n} |y_i - \mathbf{x}_i'\boldsymbol{\beta}| + n\lambda \sum_{j=1}^{p} |\beta_j|. \tag{1.5}$$

However, none of these methods is robust with respect to leverage points, that is, outliers in the predictor space, and can handle outliers only in the response variable. The main competitor of the sparse LTS is robust least angle regression, called RLARS, and proposed in Khan, Van Aelst and Zamar (2007). They develop a robust version of the LARS algorithm, essentially replacing correlations by a robust type of correlation, to sequence and select the most important predictor variables. Then a nonsparse robust regression estimator is applied to the selected predictor variables. RLARS, as will be confirmed by our simulation study, is robust with respect to leverage points. A main drawback of the RLARS algorithm of Khan, Van Aelst and Zamar (2007) is the lack of a natural definition, since it is not optimizing a clearly defined objective function.

An entirely different approach is taken by She and Owen (2011), who propose an iterative procedure for outlier detection. Their method is based on imposing a sparsity criterion on the estimator of the mean-shift parameter $\boldsymbol{\gamma}$ in the extended regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\gamma} + \boldsymbol{\varepsilon}. \tag{1.6}$$

They stress that this method requires a nonconvex sparsity criterion. An extension of the method to high-dimensional data is obtained by also assuming sparsity of the coefficients $\boldsymbol{\beta}$. Nevertheless, their paper mainly focuses on outlier detection and much less on sparse robust estimation. Note that another procedure for simultaneous outlier identification and variable selection based on the mean-shift model is proposed by Menjoge and Welsch (2010).

The rest of the paper is organized as follows. In Section 2 the breakdown point of the sparse LTS estimator is obtained. Further, we also show that the lasso and the LAD-lasso have a breakdown point of only $1/n$. A detailed description of the proposed algorithm to compute the sparse LTS regression estimator is provided in Section 3. Section 4 introduces a reweighted version of the estimator in order to increase statistical efficiency. The choice of the penalty parameter λ is discussed in Section 5. Simulation studies are performed in Section 6. In addition, Section 7 presents an application to protein and gene expression data of the well-known cancer cell panel of the National Cancer Institute. The results indicate that these data contain outliers such that robust methods are necessary for analysis. Moreover, sparse LTS yields a model that is easy to interpret and has excellent prediction performance. Finally, Section 8 presents some computation times and Section 9 concludes.

2. Breakdown point.

The most popular measure for the robustness of an estimator is the *replacement finite-sample breakdown point* (FBP) [e.g., Maronna, Martin and Yohai (2006)]. Let $\mathbf{Z} = (\mathbf{X}, \mathbf{y})$ denote the sample. For a regression estimator $\hat{\boldsymbol{\beta}}$, the breakdown point is defined as

$$\varepsilon^*(\hat{\boldsymbol{\beta}}; \mathbf{Z}) = \min\left\{\frac{m}{n} : \sup_{\tilde{\mathbf{Z}}} \|\hat{\boldsymbol{\beta}}(\tilde{\mathbf{Z}})\|_2 = \infty\right\}, \tag{2.1}$$

where $\tilde{\mathbf{Z}}$ are corrupted data obtained from $\mathbf{Z}$ by replacing m of the original n data points by arbitrary values. We obtained the following result, from which the breakdown point of the sparse LTS estimator immediately follows. The proof is in the Appendix.

THEOREM 1. *Let $\rho(x)$ be a convex and symmetric loss function with $\rho(0) = 0$ and $\rho(x) > 0$ for $x \neq 0$, and define $\boldsymbol{\rho}(\mathbf{x}) := (\rho(x_1), \dots, \rho(x_n))'$. With subset size $h \leq n$, consider the regression estimator*

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{h} (\boldsymbol{\rho}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}))_{i:n} + h\lambda \sum_{j=1}^{p} |\beta_j|, \tag{2.2}$$

where $(\boldsymbol{\rho}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})))_{1:n} \leq \cdots \leq (\boldsymbol{\rho}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}))_{n:n}$ are the order statistics of the regression loss. Then the breakdown point of the estimator $\hat{\boldsymbol{\beta}}$ is given by

$$\varepsilon^*(\hat{\boldsymbol{\beta}}; \mathbf{Z}) = \frac{n - h + 1}{n}.$$

The breakdown point is the same for any loss function ρ fulfilling the assumptions. In particular, the breakdown point for the sparse LTS estimator $\hat{\beta}_{\text{sparseLTS}}$ with subset size $h \leq n$, in which $\rho(x) = x^2$, is still $(n - h + 1)/n$. The smaller the value of h, the higher the breakdown point. By taking h small enough, it is even possible to have a breakdown point larger than 50%. However, while this is mathematically possible, we are not advising to use $h < n/2$ since robust statistics aim for models that fit the majority of the data. Thus, we do not envisage to have such large breakdown points. Instead, we suggest to take a value of h equal to a fraction α of the sample size, with $\alpha = 0.75$, such that the final estimate is based on a sufficiently large number of observations. This guarantees a sufficiently high statistical efficiency, as will be shown in the simulations in Section 6. The resulting breakdown point is then about $1 - \alpha = 25\%$. Notice that the breakdown point does not depend on the dimension p. Even if the number of predictor variables is larger than the sample size, a high breakdown point is guaranteed. For the nonsparse LTS, the breakdown point does depend on p [see Rousseeuw and Leroy (2003)].

Applying Theorem 1 to the lasso [corresponding to $\rho(x) = x^2$ and $h = n$] yields a finite-sample breakdown point of

$$\varepsilon^*(\hat{\boldsymbol{\beta}}_{\text{lasso}}; \mathbf{Z}) = \frac{1}{n}.$$

Hence, only one outlier can already send the lasso solution to infinity, despite the fact that large values of the regression estimate are penalized in the objective function of the lasso. The nonrobustness of the Lasso comes from the use of the squared residuals in the objective function (1.2). Using other convex loss functions, as done in the LAD-lasso or penalized M-estimators, does not solve the problem and results in a breakdown point of $1/n$ as well. The theoretical results on robustness are also reflected in the application to the NCI data in Section 7, where the lasso is much more influenced by the outliers than the sparse LTS.

3. Algorithm.

We first present an equivalent formulation of the sparse LTS estimator (1.4). For a fixed penalty parameter λ, define the objective function

$$Q(H, \boldsymbol{\beta}) = \sum_{i \in H} (y_i - \mathbf{x}_i' \beta)^2 + h\lambda \sum_{j=1}^{p} |\beta_j|, \tag{3.1}$$

which is the L_1 penalized residual sum of squares based on a subsample $H \subseteq \{1, \ldots, n\}$ with $|H| = h$. With

$$\hat{\boldsymbol{\beta}}_H = \operatorname*{argmin}_{\boldsymbol{\beta}} Q(H, \boldsymbol{\beta}), \tag{3.2}$$

the sparse LTS estimator is given by $\hat{\boldsymbol{\beta}}_{H_{\text{opt}}}$, where

$$H_{\text{opt}} = \operatorname*{argmin}_{H \subseteq \{1,\ldots,n\}: |H| = h} Q(H, \hat{\boldsymbol{\beta}}_H). \tag{3.3}$$

Hence, the sparse LTS corresponds to finding the subset of $h \leq n$ observations whose lasso fit produces the smallest penalized residual sum of squares. To find this optimal subset, we use an analogue of the FAST-LTS algorithm developed by Rousseeuw and Van Driessen (2006).

The algorithm is based on *concentration steps* or C-steps. The C-step at iteration k consists of computing the lasso solution based on the current subset H_k, with $|H_k| = h$, and constructing the next subset H_{k+1} from the observations corresponding to the h smallest squared residuals. Let H_k denote a certain subsample derived at iteration k and let $\hat{\boldsymbol{\beta}}_{H_k}$ be the coefficients of the corresponding lasso fit. After computing the squared residuals $\mathbf{r}_k^2 = (r_{k,1}^2, \ldots, r_{k,n}^2)'$ with $r_{k,i}^2 = (y_i - \mathbf{x}_i' \hat{\boldsymbol{\beta}}_{H_k})^2$, the subsample H_{k+1} for iteration $k+1$ is defined as the set of indices corresponding to the h smallest squared residuals. In mathematical terms, this can be written as

$$H_{k+1} = \{i \in \{1, \ldots, n\} : r_{k,i}^2 \in \{(\mathbf{r}_k^2)_{j:n} : j = 1, \ldots, h\}\},$$

where $(\mathbf{r}_k^2)_{1:n} \leq \cdots \leq (\mathbf{r}_k^2)_{n:n}$ denote the order statistics of the squared residuals. Let $\hat{\boldsymbol{\beta}}_{H_{k+1}}$ denote coefficients of the lasso fit based on H_{k+1}. Then

$$Q(H_{k+1}, \hat{\boldsymbol{\beta}}_{H_{k+1}}) \leq Q(H_{k+1}, \hat{\boldsymbol{\beta}}_{H_k}) \leq Q(H_k, \hat{\boldsymbol{\beta}}_{H_k}), \tag{3.4}$$

where the first inequality follows from the definition of $\hat{\boldsymbol{\beta}}_{H_{k+1}}$, and the second inequality from the definition of H_k. From (3.4) it follows that a C-step results in a decrease of the sparse LTS objective function, and that a sequence of C-steps yields convergence to a local minimum in a finite number of steps.

To increase the chances of arriving at the global minimum, a sufficiently large number s of initial subsamples H_0 should be used, each of them being used as starting point for a sequence of C-steps. Rather than randomly selecting h data points, any initial subset H_0 of size h is constructed from an *elemental subset* of size 3 as follows. Draw three observations from the data at random, say, $\mathbf{x}_{i_1}$, $\mathbf{x}_{i_2}$ and $\mathbf{x}_{i_3}$. The lasso fit for this elemental subset is then

$$\hat{\boldsymbol{\beta}}_{\{i_1, i_2, i_3\}} = \operatorname*{argmin}_{\boldsymbol{\beta}} Q(\{i_1, i_2, i_3\}, \boldsymbol{\beta}), \tag{3.5}$$

and the initial subset H_0 is then given by the indices of the h observations with the smallest squared residuals with respect to the fit in (3.5). The nonsparse FAST-LTS algorithm uses elemental subsets of size p, since any OLS regression requires at least as many observations as the dimension p. This would make the algorithm not applicable if $p > n$. Fortunately the lasso is already properly defined for samples of size 3, even for large values of p. Moreover, from a robustness point of view, using only three observations is optimal, as it ensures the highest probability of not including outliers in the elemental set. It is important to note that the elemental subsets of size 3 are only used to construct the initial subsets of size h for the C-step algorithms. All C-steps are performed on subsets of size h.

In this paper, we used $s = 500$ initial subsets. Using a larger number of subsets did not lead to better prediction performance in the case of the NCI data. Following the strategy advised in Rousseeuw and Van Driessen (2006), we perform only two C-steps for all s subsets and retain the $s_1 = 10$ subsamples with the lowest values of the objective function (3.1). For the reduced number of subsets s_1, further C-steps are performed until convergence. This is a standard strategy for C-step algorithms to decrease computation time.

Estimation of an intercept: the regression model in (1.1) does not contain an intercept. It is indeed common to assume that the variables are mean-centered and the predictor variables are standardized before applying the lasso. However, computing the means and standard deviations over all observations does not result in a robust method, so we take a different approach. Each time the sparse LTS algorithm computes a lasso fit on a subsample of size h, the variables are first centered and the predictors are standardized using the means and standard deviations computed from the respective subsample. The resulting procedure then minimizes (1.4) with squared residuals $r_i^2 = (y_i - \beta_0 - \mathbf{x}_i'\boldsymbol{\beta})^2$, where β_0 stands for the intercept. We verified that adding an intercept to the model has no impact on the breakdown point of the sparse LTS estimator of $\boldsymbol{\beta}$.

4. Reweighted sparse LTS estimator.

Let α denote the proportion of observations from the full sample to be retained in each subsample, that is, $h = \lfloor (n+1)\alpha \rfloor$. In this paper we take $\alpha = 0.75$. Then $(1 - \alpha)$ may be interpreted as an initial guess of the proportion of outliers in the data. This initial guess is typically rather conservative to ensure that outliers do not impact the results, and may therefore result in a loss of statistical efficiency. To increase efficiency, a reweighting step that downweights outliers detected by the sparse LTS estimator can be performed.

Under the normal error model, observations with standardized residuals larger than a certain quantile of the standard normal distribution may be declared as outliers. Since the sparse LTS estimator—like the lasso—is biased, we need to center the residuals. A natural estimate for the center of the residuals is

$$\hat{\mu}_{\text{raw}} = \frac{1}{h} \sum_{i \in H_{\text{opt}}} r_i, \tag{4.1}$$

where $r_i = y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}_{\text{sparseLTS}}$ and H_{opt} is the optimal subset from (3.3). Then the residual scale estimate associated to the raw sparse LTS estimator is given by

$$\hat{\sigma}_{\text{raw}} = k_\alpha \sqrt{\frac{1}{h} \sum_{i=1}^{h} (\mathbf{r}_{\text{c}}^2)_{i:n}}, \tag{4.2}$$

with squared centered residuals $\mathbf{r}_{\text{c}}^2 = ((r_1 - \hat{\mu}_{\text{raw}})^2, \ldots, (r_n - \hat{\mu}_{\text{raw}})^2)'$, and

$$k_\alpha = \left(\frac{1}{\alpha} \int_{-\Phi^{-1}((\alpha+1)/2)}^{\Phi^{-1}((\alpha+1)/2)} u^2 \, d\Phi(u) \right)^{-1/2}, \tag{4.3}$$

a factor to ensure that $\hat{\sigma}_{\text{raw}}$ is a consistent estimate of the standard deviation at the normal model. This formulation allows to define binary weights

$$w_i = \begin{cases} 1, & \text{if } \left|(r_i - \hat{\mu}_{\text{raw}})/\hat{\sigma}_{\text{raw}}\right| \leq \Phi^{-1}(1-\delta), \\ 0, & \text{if } \left|(r_i - \hat{\mu}_{\text{raw}})/\hat{\sigma}_{\text{raw}}\right| > \Phi^{-1}(1-\delta), \end{cases} \qquad i = 1, \ldots, n. \tag{4.4}$$

In this paper $\delta = 0.0125$ is used such that 2.5% of the observations are expected to be flagged as outliers in the normal model, which is a typical choice.

The *reweighted sparse LTS* estimator is given by the weighted lasso fit

$$\hat{\boldsymbol{\beta}}_{\text{reweighted}} = \operatorname*{argmin}_{\boldsymbol{\beta}} \sum_{i=1}^{n} w_i (y_i - \mathbf{x}_i' \boldsymbol{\beta})^2 + \lambda n_w \sum_{j=1}^{p} |\beta_j|, \tag{4.5}$$

with $n_w = \sum_{i=1}^{n} w_i$ the sum of weights. With the choice of weights given in (4.4), the reweighted sparse LTS is the lasso fit based on the observations not flagged as outliers. Of course, other weighting schemes could be considered. Using the residual center estimate

$$\hat{\mu}_{\text{reweighted}} = \frac{1}{n_w} \sum_{i=1}^{n} w_i (y_i - \mathbf{x}_i' \hat{\boldsymbol{\beta}}_{\text{reweighted}}), \tag{4.6}$$

the residual scale estimate of the reweighted sparse LTS estimator is given by

$$\hat{\sigma}_{\text{reweighted}} = k_{\alpha_w} \sqrt{\frac{1}{n_w} \sum_{i=1}^{n} w_i (y_i - \mathbf{x}_i' \hat{\boldsymbol{\beta}}_{\text{reweighted}} - \hat{\mu}_{\text{reweighted}})^2}, \tag{4.7}$$

where k_{α_w} is the consistency factor from (4.3) with $\alpha_w = n_w/n$.

Note that this reweighting step is conceptually different from the adaptive lasso by Zou (2006). While the adaptive lasso derives individual penalties on the predictors from initial coefficient estimates, the reweighted sparse LTS aims to include all nonoutlying observations into fitting the model.

5. Choice of the penalty parameter.

In practical data analysis, a suitable value of the penalty parameter λ is not known in advance. We propose to select λ by optimizing the Bayes Information Criterion (BIC), or the estimated prediction performance via cross-validation. In this paper we use the BIC since it requires less computational effort. The BIC of a given model estimated with shrinkage parameter λ is given by

$$\text{BIC}(\lambda) = \log(\hat{\sigma}) + df(\lambda) \frac{\log(n)}{n}, \tag{5.1}$$

where $\hat{\sigma}$ denotes the corresponding residual scale estimate, (4.2) or (4.7), and $df(\lambda)$ are the degrees of freedom of the model. The degrees of freedom are given by the number of nonzero estimated parameters in $\hat{\boldsymbol{\beta}}$ [see Zou, Hastie and Tibshirani (2007)].

As an alternative to the BIC, cross-validation can be used. To prevent outliers from affecting the choice of λ, a robust prediction loss function should be used. A natural choice is the root trimmed mean squared prediction error (RTMSPE) with the same trimming proportion as for computing the sparse LTS. In k-fold cross-validation, the data are split randomly in k blocks of approximately equal size. Each block is left out once to fit the model, and the left-out block is used as test data. In this manner, and for a given value of λ, a prediction is obtained for each observation in the sample. Denote the vector of squared prediction errors $\mathbf{e}^2 = (e_1^2, \ldots, e_n^2)'$. Then

$$\text{RTMSPE}(\lambda) = \sqrt{\frac{1}{h}\sum_{i=1}^{h}(\mathbf{e}^2)_{i:n}}. \tag{5.2}$$

To reduce variability, the RTMSPE may be averaged over a number of different random splits of the data.

The selected λ then minimizes BIC(λ) or RTMSPE(λ) over a grid of values in the interval $[0, \hat{\lambda}_0]$. We take a grid with steps of size $0.025\hat{\lambda}_0$, where $\hat{\lambda}_0$ is an estimate of the shrinkage parameter λ_0 that would shrink all parameters to zero. If $p > n$, 0 is of course excluded from the grid. For the lasso solution we take

$$\hat{\lambda}_0 = \frac{2}{n} \max_{j \in \{1,\ldots,p\}} \text{Cor}(\mathbf{y}, \mathbf{x}_j), \tag{5.3}$$

exactly the same as given and motivated in Efron et al. (2004). In (5.3), Cor($\mathbf{y}, \mathbf{x}_j$) stands for the Pearson correlation between $\mathbf{y}$ and the jth column of the design matrix $\mathbf{X}$. For sparse LTS, we need a robust estimate $\hat{\lambda}_0$. We propose to replace the Pearson correlation in (5.3) by the robust correlation based on bivariate winsorization of the data [see Khan, Van Aelst and Zamar (2007)].

6. Simulation study. This section presents a simulation study for comparing the performance of various sparse estimators. The simulations are performed in R [R Development Core Team (2011)] with package *simFrame* [Alfons, Templ and Filzmoser (2010), Alfons (2012a)], which is a general framework for simulation studies in statistics. Sparse LTS is evaluated for the subset size $h = \lfloor (n+1)0.75 \rfloor$. Both the raw and the reweighted version (see Section 4) are considered. We prefer to take a relatively large trimming proportion to guarantee a breakdown point of 25%. Adding the reweighting step will then increase the statistical efficiency of sparse LTS. We make a comparison with the lasso, the LAD-lasso and robust least angle regression (RLARS), discussed in the introduction. We selected the LAD-lasso estimator as a representative of the class of penalized M-estimators, since it does not need an initial residual scale estimator.

For every generated sample, an optimal value of the shrinkage parameter λ is selected. The penalty parameters for sparse LTS and the lasso are chosen using the BIC, as described in Section 5. For the LAD-lasso, we estimate the shrinkage parameter in the same way as in Wang, Li and Jiang (2007). However, if $p > n$, we cannot use their approach and use the BIC as in (5.1), with the mean absolute value of residuals (multiplied by a consistency factor) as scale estimate. For RLARS, we add the sequenced variables to the model in a stepwise fashion and fit robust MM-regressions [Yohai (1987)], as advocated in Khan, Van Aelst and Zamar (2007). The optimal model when using RLARS is then again selected via BIC, now using the robust scale estimate resulting from the MM-regression.

6.1. *Sampling schemes.* The first configuration is a latent factor model taken from Khan, Van Aelst and Zamar (2007) and covers the case of $n > p$. From $k = 6$ latent independent standard normal variables $\mathbf{l}_1, \ldots, \mathbf{l}_k$ and an independent normal error variable $\mathbf{e}$ with standard deviation σ, the response variable $\mathbf{y}$ is constructed as

$$\mathbf{y} := \mathbf{l}_1 + \cdots + \mathbf{l}_k + \mathbf{e},$$

where σ is chosen so that the signal-to-noise ratio is 3, that is, $\sigma = \sqrt{k}/3$. With independent standard normal variables $\mathbf{e}_1, \ldots, \mathbf{e}_p$, a set of $p = 50$ candidate predictors is then constructed as

$$\begin{aligned}
\mathbf{x}_j &:= \mathbf{l}_j + \tau \mathbf{e}_j, \qquad j = 1, \ldots, k,\\
\mathbf{x}_{k+1} &:= \mathbf{l}_1 + \delta \mathbf{e}_{k+1},\\
\mathbf{x}_{k+2} &:= \mathbf{l}_1 + \delta \mathbf{e}_{k+2},\\
&\vdots\\
\mathbf{x}_{3k-1} &:= \mathbf{l}_k + \delta \mathbf{e}_{3k-1},\\
\mathbf{x}_{3k} &:= \mathbf{l}_k + \delta \mathbf{e}_{3k},\\
\mathbf{x}_j &:= \mathbf{e}_j, \qquad j = 3k + 1, \ldots, p,
\end{aligned}$$

where $\tau = 0.3$ and $\delta = 5$ so that $\mathbf{x}_1, \ldots, \mathbf{x}_k$ are low-noise perturbations of the latent variables, $\mathbf{x}_{k+1}, \ldots, \mathbf{x}_{3k}$ are noise covariates that are correlated with the latent variables, and $\mathbf{x}_{3k+1}, \ldots, \mathbf{x}_p$ are independent noise covariates. The number of observations is set to $n = 150$.

The second configuration covers the case of moderate high-dimensional data. We generate $n = 100$ observations from a p-dimensional normal distribution $N(0, \Sigma)$, with $p = 1000$. The covariance matrix $\boldsymbol{\Sigma} = (\Sigma_{ij})_{1 \le i,j \le p}$ is given by $\Sigma_{ij} = 0.5^{|i-j|}$, creating correlated predictor variables. Using the coefficient vector $\boldsymbol{\beta} = (\beta_j)_{1 \le j \le p}$ 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 (1.1), where the error terms follow a normal distribution with $\sigma = 0.5$.

Finally, the third configuration represents a more extreme case of high-dimensional data with $n = 100$ observations and $p = 20{,}000$ variables. The first

1000 predictor variables are generated from a multivariate normal distribution $N(0, \Sigma)$ with $\Sigma_{ij} = 0.6^{|i-j|}$. Furthermore, the remaining 19,000 covariates are standard normal variables. Then the response variable is generated according to (1.1), where the coefficient vector $\boldsymbol{\beta} = (\beta_j)_{1 \le j \le p}$ 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.

For each of the three simulation settings, we apply contamination schemes taken from Khan, Van Aelst and Zamar (2007). To be more precise, we consider the following:

(1) *No contamination.*

(2) *Vertical outliers*: 10% of the error terms in the regression model follow a normal $N(20, \sigma)$ instead of a $N(0, \sigma)$.

(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.

In addition, we investigate a fourth and more stressful outlier scenario. Keeping the contamination level at 10%, outliers in the predictor variables are drawn from independent $N(10, 0.01)$ distributions. Note the small standard deviation such that the outliers form a dense cluster. Let $\tilde{\mathbf{x}}_i$ denote such a leverage point. Then the values of the response variable of the contaminated observations are generated by $\tilde{y}_i = \eta \tilde{\mathbf{x}}_i' \boldsymbol{\gamma}$ with $\boldsymbol{\gamma} = (-1/p)_{1 \le j \le p}$. The direction of $\boldsymbol{\gamma}$ is very different from the one of the true regression parameter $\boldsymbol{\beta}$ in the following ways. First, $\boldsymbol{\gamma}$ is not sparse. Second, all predictors have a negative effect on the response in the contaminated observations, whereas the variables with nonzero coefficients have a positive effect on the response in the good data points. Furthermore, the parameter η controls the magnitude of the leverage effect and is varied from 1 to 25 in five equidistant steps.

This results in a total of 12 different simulations schemes, which we think to be representative for the many different simulation designs we tried out. The first scheme has $n > p$, the second setting has $p > n$, and the third setting has $p \gg n$. The choices for the contamination schemes are standard, inducing both vertical outliers and leverage points in the samples.

6.2. *Performance measures.* Since one of the aims of sparse model estimation is to improve prediction performance, the different estimators are evaluated by the *root mean squared prediction error* (RMSPE). For this purpose, n additional observations from the respective sampling schemes (without outliers) are generated as test data, and this in each simulation run. Then the RMSPE is given by

$$\mathrm{RMSPE}(\hat{\boldsymbol{\beta}}) = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i^* - \mathbf{x}_i^{*\prime} \hat{\boldsymbol{\beta}})^2},$$

where y_i^* and $\mathbf{x}_i^*$, $i = 1, \ldots, n$, denote the observations of the response and predictor variables in the test data, respectively. The RMSPE of the oracle estimator, which uses the true coefficient values $\boldsymbol{\beta}$, is computed as a benchmark for the evaluated methods. We report average RMSPE over all simulation runs.

Concerning sparsity, the estimated models are evaluated by the *false positive rate* (FPR) and the *false negative rate* (FNR). A false positive is a coefficient that is zero in the true model, but is estimated as nonzero. Analogously, a false negative is a coefficient that is nonzero in the true model, but is estimated as zero. In mathematical terms, the FPR and FNR are defined as

$$\mathrm{FPR}(\hat{\boldsymbol{\beta}}) = \frac{|\{j \in \{1, \dots, p\} : \hat{\beta}_j \neq 0 \wedge \beta_j = 0\}|}{|\{j \in \{1, \dots, p\} : \beta_j = 0\}|},$$

$$\mathrm{FNR}(\hat{\boldsymbol{\beta}}) = \frac{|\{j \in \{1, \dots, p\} : \hat{\beta}_j = 0 \wedge \beta_j \neq 0\}|}{|\{j \in \{1, \dots, p\} : \beta_j \neq 0\}|}.$$

Both FPR and FNR should be as small as possible for a sparse estimator and are averaged over all simulation runs. Note that false negatives in general have a stronger effect on the RMSPE than false positives. A false negative means that important information is not used for prediction, whereas a false positive merely adds a bit of variance.

6.3. *Simulation results.* In this subsection the simulation results for the different data configurations are presented and discussed.

6.3.1. *Results for the first sampling scheme.* The simulation results for the first data configuration are displayed in Table 1. Keep in mind that this configuration is exactly the same as in Khan, Van Aelst and Zamar (2007), and that the contamination settings are a subset of the ones applied in their paper. In the scenario without contamination, LAD-lasso, RLARS and lasso show excellent performance with low RMSPE and FPR. The prediction performance of sparse LTS is good, but it

TABLE 1

Results for the first simulation scheme, with $n = 150$ and $p = 50$. Root mean squared prediction error (RMSPE), the false positive rate (FPR) and the false negative rate (FNR), averaged over 500 simulation runs, are reported for every method

	No contamination			**Vertical outliers**			**Leverage points**		
Method	**RMSPE**	**FPR**	**FNR**	**RMSPE**	**FPR**	**FNR**	**RMSPE**	**FPR**	**FNR**
Lasso	1.18	0.10	0.00	2.44	0.54	0.09	2.20	0.00	0.16
LAD-lasso	1.13	0.05	0.00	1.15	0.07	0.00	1.27	0.18	0.00
RLARS	1.14	0.07	0.00	1.12	0.03	0.00	1.22	0.09	0.00
Raw sparse LTS	1.29	0.34	0.00	1.26	0.32	0.00	1.26	0.26	0.00
Sparse LTS	1.24	0.22	0.00	1.22	0.25	0.00	1.22	0.18	0.00
Oracle	0.82			0.82			0.82		

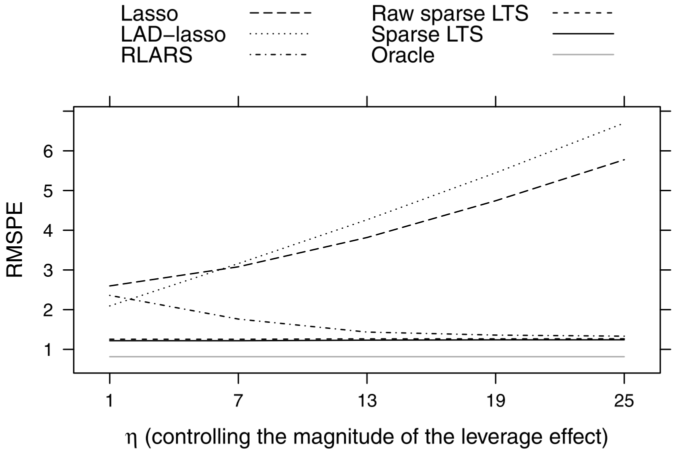

FIG. 1. *Root mean squared prediction error* (*RMSPE*) *for the first simulation scheme*, *with* $n = 150$ *and* $p = 50$, *and for the fourth contamination setting*, *averaged over* 500 *simulation runs. Lines for raw and reweighted sparse LTS almost coincide.*

has a larger FPR than the other three methods. The reweighting step clearly improves the estimates, which is reflected in the lower values for RMSPE and FPR. Furthermore, none of the methods suffer from false negatives.

In the case of vertical outliers, the nonrobust lasso is clearly influenced by the outliers, reflected in the much higher RMSPE and FPR. RLARS, LAD-lasso and sparse LTS, on the other hand, keep their excellent behavior. Sparse LTS still has a considerable tendency toward false positives, but the reweighting step is a significant improvement over the raw estimator.

When leverage points are introduced in addition to the vertical outliers, the performance of RLARS, sparse LTS and LAD-lasso is comparable. The FPR of RLARS and LAD-lasso slightly increased, whereas the FPR of sparse LTS slightly decreased. The LAD-lasso still performs well, and even the lasso performs better than in the case of only vertical outliers. This suggests that the leverage points in this example do not have a bad leverage effect.

In Figure 1 the results for the fourth contamination setting are shown. The RMSPE is thereby plotted as a function of the parameter η. With increasing η, the RMSPE of the lasso and the LAD-lasso increases. RLARS has a considerably higher RMSPE than sparse LTS for lower values of η, but the RMSPE gradually decreases with increasing η. However, the RMSPE of sparse LTS remains the lowest, thus, it has the best overall performance.

6.3.2. *Results for the second sampling scheme.* Table 2 contains the simulation results for the moderate high-dimensional data configuration. In the scenario without contamination, RLARS and the lasso perform best with very low RMSPE and almost perfect FPR and FNR. Also, the LAD-lasso has excellent prediction

TABLE 2

Results for the second simulation scheme, with $n = 100$ and $p = 1000$. Root mean squared prediction error (RMSPE), the false positive rate (FPR) and the false negative rate (FNR), averaged over 500 simulation runs, are reported for every method

	No contamination			**Vertical outliers**			**Leverage points**		
Method	**RMSPE**	**FPR**	**FNR**	**RMSPE**	**FPR**	**FNR**	**RMSPE**	**FPR**	**FNR**
Lasso	0.62	0.00	0.00	2.56	0.08	0.16	2.53	0.00	0.71
LAD-lasso	0.66	0.08	0.00	0.82	0.00	0.01	1.17	0.08	0.01
RLARS	0.60	0.01	0.00	0.73	0.00	0.10	0.92	0.02	0.09
Raw sparse LTS	0.81	0.02	0.00	0.73	0.02	0.00	0.73	0.02	0.00
Sparse LTS	0.74	0.01	0.00	0.69	0.01	0.00	0.71	0.02	0.00
Oracle	0.50			0.50			0.50		

performance, followed by sparse LTS. The LAD-lasso leads to a slightly higher FPR than the other methods, though. When vertical outliers are added, RLARS still has excellent prediction performance despite some false negatives. We see that the sparse LTS performs best here. In addition, the prediction performance of the nonrobust lasso already suffers greatly from the vertical outliers. In the scenario with additional leverage points, sparse LTS remains stable and is still the best. For RLARS, sparsity behavior according to FPR and FNR does not change significantly either, but there is a small increase in the RMSPE. On the other hand, LAD-lasso already has a considerably larger RMSPE than sparse LTS, 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.

The results for the fourth contamination setting are presented in Figure 2. As for the previous simulation scheme, the RMSPE for the lasso and the LAD-lasso is increasing with increasing parameter η. The RMSPE for RLARS, however, is gradually decreasing. Sparse LTS shows particularly interesting behavior: the RMSPE is close to the oracle at first, then there is a kink in the curve (with the value of the RMSPE being in between those for the LAD-lasso and the lasso), after which the RMSPE returns to low values close to the oracle. In any case, for most of the investigated values of η, sparse LTS has the best performance.

6.3.3. *Results for the third sampling scheme.* Table 3 contains the simulation results for the more extreme high-dimensional data configuration. Note that the LAD-lasso was no longer computationally feasible with such a large number of variables. In addition, the number of simulation runs was reduced from 500 to 100 to lower the computational effort.

In the case without contamination, the sparse LTS suffers from an efficiency problem, which is reflected in larger values for RMSPE and FNR than for the

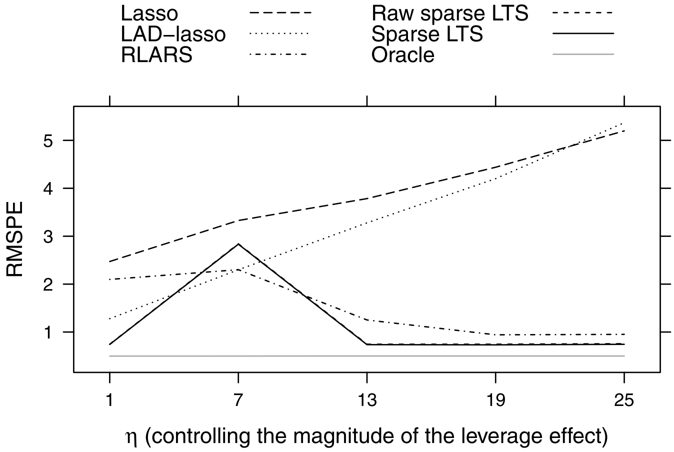

FIG. 2. *Root mean squared prediction error* (*RMSPE*) *for the second simulation scheme*, *with* $n = 100$ *and* $p = 1000$, *and for the fourth contamination setting*, *averaged over* 500 *simulation runs*. *Lines for raw and reweighted sparse LTS almost coincide*.

other methods. The lasso and RLARS have considerably better performance in this case. With vertical outliers, the RMSPE for the lasso increases greatly due to many false negatives. Also, RLARS has a larger FNR than sparse LTS, resulting in a slightly lower RMSPE for the reweighted version of the latter. When leverage points are introduced, sparse LTS clearly exhibits the lowest RMSPE and FNR. Furthermore, the lasso results in a very large FNR.

Figure 3 shows the results for the fourth contamination setting. Most interestingly, the RMSPE of RLARS in this case keeps increasing in the beginning and even goes above the one of the lasso, before dropping dropping continuously in the remaining steps. Sparse LTS again shows a kink in the curve for the RMSPE, but clearly performs best.

TABLE 3

Results for the third simulation scheme, *with* $n = 100$ *and* $p = 20{,}000$. *Root mean squared prediction error* (*RMSPE*), *the false positive rate* (*FPR*) *and the false negative rate* (*FNR*), *averaged over* 100 *simulation runs*, *are reported for every method*

	No contamination			**Vertical outliers**			**Leverage points**		
Method	**RMSPE**	**FPR**	**FNR**	**RMSPE**	**FPR**	**FNR**	**RMSPE**	**FPR**	**FNR**
Lasso	1.43	0.000	0.00	5.19	0.004	0.49	5.57	0.000	0.83
RLARS	1.54	0.001	0.00	2.53	0.000	0.38	3.34	0.001	0.45
Raw sparse LTS	3.00	0.001	0.19	2.59	0.002	0.11	2.59	0.002	0.10
Sparse LTS	2.88	0.001	0.16	2.49	0.002	0.10	2.57	0.002	0.09
Oracle	1.00			1.00			1.00		

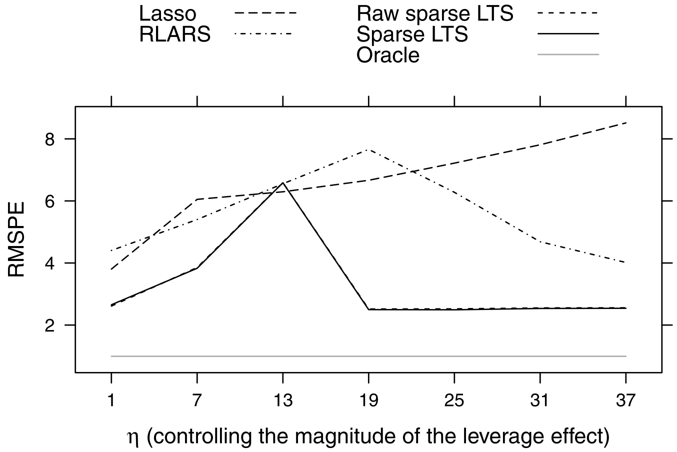

FIG. 3. *Root mean squared prediction error* (*RMSPE*) *for the third simulation scheme*, *with* $n = 100$ *and* $p = 20{,}000$, *and for the fourth contamination setting*, *averaged over* 100 *simulation runs*. *Lines for raw and reweighted sparse LTS almost coincide*.

6.3.4. *Summary of the simulation results*. Sparse LTS shows the best overall performance in this simulation study, if the reweighted version is taken. Concerning the other investigated methods, RLARS also performs well, but suffers sometimes from an increased percentage of false negatives under contamination. It is also confirmed that the lasso is not robust to outliers. The LAD-lasso still sustains vertical outliers, but is not robust against bad leverage points.

7. NCI-60 cancer cell panel. In this section the sparse LTS estimator is compared to the competing methods in an application to the cancer cell panel of the National Cancer Institute. It consists of data on 60 human cancer cell lines and can be downloaded via the web application CellMiner (http://discover.nci.nih.gov/cellminer/). We regress protein expression on gene expression data. The gene expression data were obtained with an Affymetrix HG-U133A chip and normalized with the GCRMA method, resulting in a set of $p = 22{,}283$ predictors. The protein expressions based on 162 antibodies were acquired via reverse-phase protein lysate arrays and $\log_2$ transformed. One observation had to be removed since all values were missing in the gene expression data, reducing the number of observations to $n = 59$. More details on how the data were obtained can be found in Shankavaram et al. (2007). Furthermore, Lee et al. (2011) also use this data for regression analysis, but consider only nonrobust methods. They obtain models that still consist of several hundred to several thousand predictors and are thus difficult to interpret.

Similar to Lee et al. (2011), we first order the protein expression variables according to their scale, but use the MAD (median absolute deviation from the median, multiplied with the consistency factor 1.4826) as a scale estimator instead of the standard deviation. We show the results for the protein expressions based on the KRT18 antibody, which constitutes the variable with the largest MAD, serving as one dependent variable. Hence, our response variable measures the expression levels of the protein *keratin 18*, which is known to be persistently expressed in carcinomas [Oshima, Baribault and Caulín (1996)]. We compare raw and reweighted sparse LTS with 25% trimming, lasso and RLARS. As in the simulation study, the LAD-lasso could not be computed for such a large p. The optimal models are selected via BIC as discussed in Section 5. The raw sparse LTS estimator thereby results in a model with 32 genes. In the reweighting step, one more observation is added to the best subset found by the raw estimator, yielding a model with 33 genes for reweighted sparse LTS (thus also one more gene is selected compared to the raw estimator). The lasso model is somewhat larger with 52 genes, whereas the RLARS model is somewhat smaller with 18 genes.

Sparse LTS and the lasso have three selected genes in common, one of which is KRT8. The product of this gene, the protein *keratin 8*, typically forms an intermediate filament with keratin 18 such that their expression levels are closely linked [e.g., Owens and Lane (2003)]. However, the larger model of the lasso is much more difficult to interpret. Two of the genes selected by the lasso are not even recorded in the Gene database [Maglott et al. (2005)] of the National Center for Biotechnology Information (NCBI). The sparse LTS model is considerably smaller and easier to interpret. For instance, the gene expression level of MSLN, whose product *mesothelin* is overexpressed in various forms of cancer [Hassan, Bera and Pastan (2004)], has a positive effect on the protein expression level of keratin 18.

Concerning prediction performance, the root trimmed mean squared prediction error (RTMSPE) is computed as in (5.2) via leave-one-out cross-validation (so $k = n$). Table 4 reports the RTMSPE for the considered methods. Sparse LTS clearly shows the smallest RTMSPE, followed by RLARS and the lasso. In addition, sparse LTS detects 13 observations as outliers, showing the need for a robust procedure. Further analysis revealed that including those 13 observations changes the correlation structure of the predictor variables with the response. Consequently,

TABLE 4

*Root trimmed mean squared prediction error (RTMSPE) for protein expressions based on the KRT*18 *antibody (NCI*-60 *cancer cell panel data), computed from leave-one-out cross-validation*

Method	**RTMSPE**
Lasso	1.058
RLARS	0.936
Raw sparse LTS	0.727
Sparse LTS	0.721

the order in which the genes are added to the model by the lasso algorithm on the full sample is completely different from the order on the best subset found by sparse LTS. Leaving out those 13 observations therefore yields more reliable results for the majority of the cancer cell lines.

It is also worth noting that the models still contain a rather large number of variables given the small number of observations. For the lasso, it is well known that it tends to select many noise variables in high dimensions since the same penalty is applied on all variables. Meinshausen (2007) therefore proposed a relaxation of the penalty for the selected variables of an initial lasso fit. Adding such a relaxation step to the sparse LTS procedure may thus be beneficial for large p and is considered for future work.

8. Computational details and CPU times. All computations are carried out in R version 2.14.0 [R Development Core Team (2011)] using the packages *robustHD* [Alfons (2012b)] for sparse LTS and RLARS, *quantreg* [Koenker (2011)] for the LAD-lasso and *lars* [Hastie and Efron (2011)] for the lasso. Most of sparse LTS is thereby implemented in C++, while RLARS is an optimized version of the R code by Khan, Van Aelst and Zamar (2007). Optimization of the RLARS code was necessary since the original code builds a $p \times p$ matrix of robust correlations, which is not computationally feasible for very large p. The optimized version only stores an $q \times p$ matrix, where q is the number of sequenced variables. Furthermore, the robust correlations are computed with C++ rather than R.

Since computation time is an important practical consideration, Figure 4 displays computation times of lasso, LAD-lasso, RLARS and sparse LTS in seconds. Note that those are average times over 10 runs based on simulated data with $n = 100$ and varying dimension p, obtained on an Intel Xeon X5670 machine. For sparse LTS and the LAD-lasso, the reported CPU times are averages over a grid

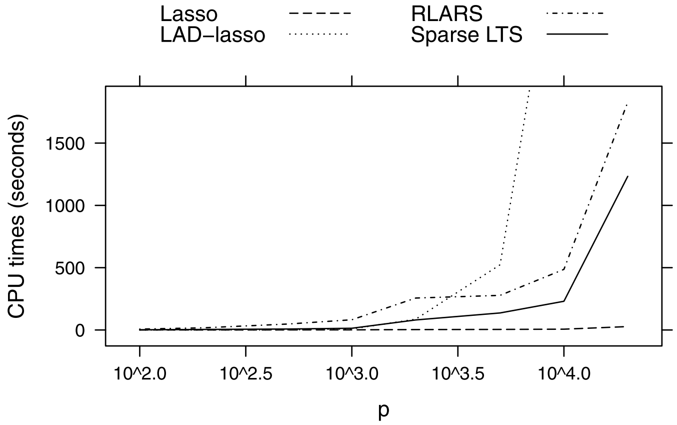

FIG. 4. *CPU times* (*in seconds*) *for* $n = 100$ *and varying* p, *averaged over* 10 *runs*.

of five values for λ. RLARS is a hybrid procedure, thus, we only report the CPU times for obtaining the sequence of predictors, but not for fitting the models along the sequence.

As expected, the computation time of the nonrobust lasso remains very low for increasing p. Sparse LTS is still reasonably fast up to $p \approx 10{,}000$, but computation time is a considerable factor if p is much larger than that. However, sparse LTS remains faster than obtaining the RLARS sequence. A further advantage of the subsampling algorithm of sparse LTS is that it can easily be parallelized to reduce computation time on modern multicore computers, which is future work.

9. Conclusions and discussion.

Least trimmed squares (LTS) is a robust regression method frequently used in practice. Nevertheless, it does not allow for sparse model estimates and cannot be applied to high-dimensional data with $p > n$. This paper introduced the sparse LTS estimator, which overcomes these two issues simultaneously by adding an L_1 penalty to the LTS objective function. Simulation results and a real data application to protein and gene expression data of the NCI-60 cancer cell panel illustrated the excellent performance of sparse LTS and showed that it performs as well or better than robust variable selection methods such as RLARS. In addition, an advantage of sparse LTS over algorithmic procedures such as RLARS is that the objective function allows for theoretical investigation of its statistical properties. As such, we could derive the breakdown point of the sparse LTS estimator. However, it should be noted that efficiency is an issue with sparse LTS. A reweighting step can thereby lead to a substantial improvement in efficiency, as shown in the simulation study.

In the paper, an L_1 penalization was imposed on the regression parameter, as for the lasso. Other choices for the penalty are possible. For example, an L_2 penalty leads to ridge regression. A robust version of ridge regression was recently proposed by Maronna (2011), 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. Another possible choice for the penalty function is the smoothly clipped absolute deviation penalty (SCAD) proposed by Fan and Li (2001). It satisfies the mathematical conditions for sparsity but results in a more difficult optimization problem than the lasso. Still, a robust version of SCAD can be obtained by optimizing the associated objective function over trimmed samples instead of over the full sample.

There are several other open questions that we leave for future research. For instance, we did not provide any asymptotics for sparse LTS, as was, for example, done for penalized M-estimators in Germain and Roueff (2010). Potentially, sparse LTS could be used as an initial estimator for computing penalized M-estimators.

All in all, the results presented in this paper suggest that sparse LTS is a valuable addition to the statistics researcher's toolbox. The sparse LTS estimator has an intuitively appealing definition and is related to the popular least trimmed squares estimator of robust regression. It performs model selection, outlier detection and robust estimation simultaneously, and is applicable if the dimension is larger than the sample size.

APPENDIX: PROOF OF BREAKDOWN POINT

PROOF OF THEOREM 1. In this proof the L_1 norm of a vector $\boldsymbol{\beta}$ is denoted as $\|\boldsymbol{\beta}\|_1$ and the Euclidean norm as $\|\boldsymbol{\beta}\|_2$. Since these norms are topologically equivalent, there exists a constant $c_1 > 0$ such that $\|\boldsymbol{\beta}\|_1 \geq c_1\|\boldsymbol{\beta}\|_2$ for all vectors $\boldsymbol{\beta}$. The proof is split into two parts.

First, we prove that $\varepsilon^*(\hat{\boldsymbol{\beta}}; \mathbf{Z}) \geq \frac{n-h+1}{n}$. Replace the last $m \leq n-h$ observations, resulting in the contaminated sample $\tilde{\mathbf{Z}}$. Then there are still $n-m \geq h$ good observations in $\tilde{\mathbf{Z}}$. Let $M_y = \max_{1\leq i\leq n}|y_i|$ and $M_{x_1} = \max_{1\leq i\leq n}|x_{i1}|$. For the case $\beta_j = 0$, $j = 1, \dots, p$, the value of the objective function is given by

$$Q(\mathbf{0}) = \sum_{i=1}^{h}(\boldsymbol{\rho}(\tilde{\mathbf{y}}))_{i:n} \leq \sum_{i=1}^{h}(\boldsymbol{\rho}(\mathbf{y}))_{i:n} \leq h\rho(M_y).$$

Now consider any $\boldsymbol{\beta}$ with $\|\boldsymbol{\beta}\|_2 \geq M := (h\rho(M_y) + 1)/(\lambda c_1)$. For the value of the objective function, it holds that

$$Q(\boldsymbol{\beta}) \geq \lambda\|\boldsymbol{\beta}\|_1 \geq \lambda c_1\|\boldsymbol{\beta}\|_2 \geq h\rho(M_y) + 1 > Q(\mathbf{0}).$$

Since $Q(\hat{\boldsymbol{\beta}}) \leq Q(0)$, we conclude that $\|\hat{\boldsymbol{\beta}}(\tilde{\mathbf{Z}})\|_2 \leq M$, where M does not depend on the outliers. This concludes the first part of the proof.

Second, we prove that $\varepsilon^*(\hat{\boldsymbol{\beta}}; \mathbf{Z}) \leq \frac{n-h+1}{n}$. Move the last $m = n-h+1$ observations of $\mathbf{Z}$ to the position $\mathbf{z}(\gamma,\tau) = (\mathbf{x}(\tau)', y(\gamma,\tau))' = ((\tau, 0, \dots, 0), \gamma\tau)'$ with $\gamma, \tau > 0$, and denote $\mathbf{Z}_{\gamma,\tau}$ the resulting contaminated sample. Assume that there exists a constant M such that

$$\sup_{\tau,\gamma}\|\hat{\boldsymbol{\beta}}(\mathbf{Z}_{\gamma,\tau})\|_2 \leq M, \tag{A.1}$$

that is, there is no breakdown. We will show that this leads to a contradiction.

Let $\boldsymbol{\beta}_\gamma = (\gamma, 0, \dots, 0)' \in \mathbb{R}^p$ with $\gamma = M+2$ and define $\tau > 0$ such that $\rho(\tau) \geq \max(h-m, 0)\rho(M_y + \gamma M_{x_1}) + h\lambda\gamma + 1$. Note that τ is always well defined due to the assumptions on ρ, in particular, since $\rho(\infty) = \infty$. Then the objective function is given by

$$Q(\boldsymbol{\beta}_\gamma) = \begin{cases} \displaystyle\sum_{i=1}^{h-m}(\boldsymbol{\rho}(y - \mathbf{X}\boldsymbol{\beta}_\gamma))_{i:(n-m)} + h\lambda|\gamma|, & \text{if } h > m, \\ h\lambda|\gamma|, & \text{else,} \end{cases}$$

since the residuals with respect to the outliers are all zero. Hence,

$$Q(\boldsymbol{\beta}_\gamma) \leq \max(h-m, 0)\rho(M_y + \gamma M_{x_1}) + h\lambda\gamma \leq \rho(\tau) - 1. \tag{A.2}$$

Furthermore, for $\boldsymbol{\beta} = (\beta_1, \dots, \beta_p)'$ with $\|\boldsymbol{\beta}\|_2 \le \gamma - 1$ we have

$$Q(\boldsymbol{\beta}) \ge \rho(\gamma\tau - \tau\beta_1),$$

since at least one outlier will be in the set of the smallest h residuals. Now $\beta_1 \le \|\boldsymbol{\beta}\|_2 \le \gamma - 1$, so that

$$Q(\boldsymbol{\beta}) \ge \rho\bigl(\tau(\gamma - \beta_1)\bigr) \ge \rho(\tau), \tag{A.3}$$

since ρ is nondecreasing.

Combining (A.2) and (A.3) leads to

$$\|\hat{\boldsymbol{\beta}}(\mathbf{Z}_{\gamma,\tau})\|_2 \ge \gamma - 1 = M + 1,$$

which contradicts the assumption (A.1). Hence, there is breakdown. □

Acknowledgments. We would like to thank the Editor and two anonymous referees for their constructive remarks that led to an improvement of the paper.

REFERENCES

ALFONS, A. (2012a). *simFrame*: Simulation framework. R package version 0.5.0.

ALFONS, A. (2012b). *robustHD*: Robust methods for high-dimensional data. R package version 0.1.0.

ALFONS, A., TEMPL, M. and FILZMOSER, P. (2010). An object-oriented framework for statistical simulation: The R package *simFrame*. *Journal of Statistical Software* **37** 1–36.

EFRON, B., HASTIE, T., JOHNSTONE, I. and TIBSHIRANI, R. (2004). Least angle regression. *Ann. Statist.* **32** 407–499. MR2060166

FAN, J. and LI, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *J. Amer. Statist. Assoc.* **96** 1348–1360. MR1946581

GERMAIN, J.-F. and ROUEFF, F. (2010). Weak convergence of the regularization path in penalized M-estimation. *Scand. J. Stat.* **37** 477–495. MR2724509

GERTHEISS, J. and TUTZ, G. (2010). Sparse modeling of categorial explanatory variables. *Ann. Appl. Stat.* **4** 2150–2180. MR2829951

HASSAN, R., BERA, T. and PASTAN, I. (2004). Mesothelin: A new target for immunotherapy. *Clin. Cancer Res.* **10** 3937–3942.

HASTIE, T. and EFRON, B. (2011). *lars*: Least angle regression, lasso and forward stagewise. R package version 0.9-8.

KHAN, J. A., VAN AELST, S. and ZAMAR, R. H. (2007). Robust linear model selection based on least angle regression. *J. Amer. Statist. Assoc.* **102** 1289–1299. MR2412550

KNIGHT, K. and FU, W. (2000). Asymptotics for lasso-type estimators. *Ann. Statist.* **28** 1356–1378. MR1805787

KOENKER, R. (2011). *quantreg*: Quantile regression. R package version 4.67.

LEE, D., LEE, W., LEE, Y. and PAWITAN, Y. (2011). Sparse partial least-squares regression and its applications to high-throughput data analysis. *Chemometrics and Intelligent Laboratory Systems* **109** 1–8.

LI, G., PENG, H. and ZHU, L. (2011). Nonconcave penalized M-estimation with a diverging number of parameters. *Statist. Sinica* **21** 391–419. MR2796868

MAGLOTT, D., OSTELL, J., PRUITT, K. D. and TATUSOVA, T. (2005). Entrez gene: Gene-centered information at NCBI. *Nucleic Acids Res.* **33** D54–D58.

MARONNA, R. A. (2011). Robust ridge regression for high-dimensional data. *Technometrics* **53** 44–53. MR2791951

MARONNA, R. A., MARTIN, R. D. and YOHAI, V. J. (2006). *Robust Statistics*: *Theory and Methods*. Wiley, Chichester. MR2238141

MEINSHAUSEN, N. (2007). Relaxed lasso. *Comput*. *Statist*. *Data Anal*. **52** 374–393. MR2409990

MENJOGE, R. S. and WELSCH, R. E. (2010). A diagnostic method for simultaneous feature selection and outlier identification in linear regression. *Comput*. *Statist*. *Data Anal*. **54** 3181–3193. MR2727745

OSHIMA, R. G., BARIBAULT, H. and CAULÍN, C. (1996). Oncogenic regulation and function of keratins 8 and 18. *Cancer and Metastasis Rewiews* **15** 445–471.

OWENS, D. W. and LANE, E. B. (2003). The quest for the function of simple epithelial keratins. *Bioessays* **25** 748–758.

R DEVELOPMENT CORE TEAM (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

RADCHENKO, P. and JAMES, G. M. (2011). Improved variable selection with forward-lasso adaptive shrinkage. *Ann*. *Appl*. *Stat*. **5** 427–448. MR2810404

ROSSET, S. and ZHU, J. (2004). Discussion of "Least angle regression," by B. Efron, T. Hastie, I. Johnstone and R. Tibshirani. *Ann*. *Statist*. **32** 469–475.

ROUSSEEUW, P. J. (1984). Least median of squares regression. *J*. *Amer*. *Statist*. *Assoc*. **79** 871–880. MR0770281

ROUSSEEUW, P. J. and LEROY, A. M. (2003). *Robust Regression and Outlier Detection*, 2nd ed. Wiley, Hoboken.

ROUSSEEUW, P. J. and VAN DRIESSEN, K. (2006). Computing LTS regression for large data sets. *Data Min*. *Knowl*. *Discov*. **12** 29–45. MR2225526

SHANKAVARAM, U. T., REINHOLD, W. C., NISHIZUKA, S., MAJOR, S., MORITA, D., CHARY, K. K., REIMERS, M. A., SCHERF, U., KAHN, A., DOLGINOW, D., COSSMAN, J., KALDJIAN, E. P., SCUDIERO, D. A., PETRICOIN, E., LIOTTA, L., LEE, J. K. and WEINSTEIN, J. N. (2007). Transcript and protein expression profiles of the NCI-60 cancer cell panel: An integromic microarray study. *Molecular Cancer Therapeutics* **6** 820–832.

SHE, Y. and OWEN, A. B. (2011). Outlier detection using nonconvex penalized regression. *J*. *Amer*. *Statist*. *Assoc*. **106** 626–639. MR2847975

TIBSHIRANI, R. (1996). Regression shrinkage and selection via the lasso. *J*. *Roy*. *Statist*. *Soc*. *Ser*. *B* **58** 267–288. MR1379242

VAN DE GEER, S. A. (2008). High-dimensional generalized linear models and the lasso. *Ann*. *Statist*. **36** 614–645. MR2396809

WANG, H., LI, G. and JIANG, G. (2007). Robust regression shrinkage and consistent variable selection through the LAD-lasso. *J*. *Bus*. *Econom*. *Statist*. **25** 347–355. MR2380753

WANG, S., NAN, B., ROSSET, S. and ZHU, J. (2011). Random lasso. *Ann*. *Appl*. *Stat*. **5** 468–485. MR2810406

WU, T. T. and LANGE, K. (2008). Coordinate descent algorithms for lasso penalized regression. *Ann*. *Appl*. *Stat*. **2** 224–244. MR2415601

YOHAI, V. J. (1987). High breakdown-point and high efficiency robust estimates for regression. *Ann*. *Statist*. **15** 642–656. MR0888431

YUAN, M. and LIN, Y. (2006). Model selection and estimation in regression with grouped variables. *J*. *R*. *Stat*. *Soc*. *Ser*. *B Stat*. *Methodol*. **68** 49–67. MR2212574

ZHAO, P. and YU, B. (2006). On model selection consistency of lasso. *J*. *Mach*. *Learn*. *Res*. **7** 2541–2563. MR2274449

ZOU, H. (2006). The adaptive lasso and its oracle properties. *J*. *Amer*. *Statist*. *Assoc*. **101** 1418–1429. MR2279469

ZOU, H., HASTIE, T. and TIBSHIRANI, R. (2007). On the “degrees of freedom” of the lasso. *Ann. Statist*. **35** 2173–2192. MR2363967

A. ALFONS
C. CROUX
ORSTAT RESEARCH CENTER
FACULTY OF BUSINESS AND ECONOMICS
KU LEUVEN
NAAMSESTRAAT 69
3000 LEUVEN
BELGIUM
E-MAIL: andreas.alfons@econ.kuleuven.be
christophe.croux@econ.kuleuven.be

S. GELPER
ROTTERDAM SCHOOL OF MANAGEMENT
ERASMUS UNIVERSITY ROTTERDAM
BURGEMEESTER OUDLAAN 50
3000 ROTTERDAM
THE NETHERLANDS
E-MAIL: sgelper@rsm.nl