AUSTRIAN JOURNAL OF STATISTICS
Volume 41 (2012), Number 1, 45–58

Computing Robust Regression Estimators: Developments since Dutter (1977)

Moritz Gschwandtner and Peter Filzmoser

Vienna University of Technology, Austria

Abstract: The proposal of M estimators for regression (Huber, 1973) and the development of an algorithm for its computation (Dutter, 1977) has lead to an increased activity for further research in this area. New regression estimators were introduced that combine a high level of robustness with high efficiency. Also fast algorithms have been developed and implemented in several software packages. We provide a review of the most important methods, and compare the performance of the algorithms implemented in **R** .

Zusammenfassung: Die Einführung von M Schätzern für Regression (Huber, 1973) und die Entwicklung eines Algorithmus für die Berechnung (Dutter, 1977) hat zu einer regen Forschungstätigkeit auf diesem Gebiet geführt. Neue Schätzer für Regression wurden eingeführt, die einen hohen Grad von Robustheit mit hoher Effizienz kombinieren. Auch schnelle Algorithmen wurden entwickelt und in diversen Softwarepaketen implementiert. Wir geben einen Überblick über die wichtigsten Methoden und vergleichen die in **R** implementierten Algorithmen.

Keywords: MM Estimator, LTS Regression, Outlier Detection, Leverage Point.

1 Introduction

Linear regression belongs to the most important methods in statistics. There are numerous applications of linear regression in practice, and usually the least squares (LS) estimator is taken as the standard estimator. The LS estimator has excellent theoretical properties, and its computation is usually very simple. However, this estimator also relies on quite strict assumptions, and a violation of these assumptions may lead to useless results.

The idea of robust statistics is to account for certain deviations from idealized model assumptions. Typically, robust methods reduce the influence of outlying observations on the estimator. In the context of regression, there are two types of outliers:

- *vertical outliers:* these are outliers from the linear trend along the response variable;
- *leverage points:* these are outliers in the space of the explanatory variables.

Leverage points can even improve the estimation if the values of the response variable correspond to the linear trend. Here we assume that their response values are also deviating from the linear trend, which is often referred to as *bad leverage points* (e.g. Rousseeuw and Leroy, 2003). The LS estimator is sensitive with respect to both types of outliers. Leverage points may even "lever" the regression line or hyperplane. The goal of robust regression estimators is to protect against both types of outliers.

The history of robust regression dates back to the 19th century, where L$_1$ regression was treated in Edgeworth (1887); it was even earlier mentioned than LS regression (Boscovich, 1757). However, it turns out that L$_1$ regression is only robust with respect to vertical outliers, but not to leverage points. A first formal approach to robust regression was done with the *M estimator* in Huber (1973). This estimator was difficult to compute, and Dutter (1977) introduced an algorithm and compared with several other proposals. Extensions of this work are available in Dutter (1983). Since this time, a lot of research was devoted to developing robust regression estimators and algorithms for their computation. In this contribution we focus on the theoretical and numerical developments in robust regression since Dutter (1977). Several key proposals are compared in a simulation study.

2 Basic Concepts

For the following sections we consider the linear regression model

$$y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \epsilon_i \quad \text{for } i = 1, \ldots, n\,,$$

where $\boldsymbol{x}_i$ is a p-dimensional vector of explanatory variables, y_i is the response, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$ is the vector of unknown regression coefficients, and $\epsilon_i \sim N(0, \sigma^2)$ are independent normally distributed errors. By setting $x_{i1} = 1$, an intercept term is included in the model. The regression residuals are defined as $r_i(\boldsymbol{\beta}) = y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}$. It is clear that meaningful regression estimators attempt to minimize the regression residuals, or more precisely, a function of the residuals.

2.1 Least Squares Regression

The most common approach for estimating $\boldsymbol{\beta}$ is least squares (LS) regression, which minimizes the sum of squared residuals:

$$\hat{\boldsymbol{\beta}}_{LS} = \underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{n} r_i(\boldsymbol{\beta})^2\,. \tag{1}$$

The squared loss function is not bounded, which means that large (absolute) residuals contribute much to the overall sum. This is a particular problem if leverage points are present in the data. Figure 1 depicts the effect of a leverage point (triangle) in simple linear regression. While the six regular observations (circles) form a linear trend, the leverage point deviates from this trend and is outlying in the x space. The amount of outlyingness of the leverage point is smaller in the left picture, and more severe in the right plot. The LS regression line (solid line) is attracted by the leverage point in both situations, and the effect is much more severe in the right plot. Any diagnostics based on the size of the residuals would be misleading, since the residual of the leverage point is smaller than some of the residuals of the regular observations.

We conclude that already a single observation may cause the LS estimator to break down. In general, we define the proportion of observations that cause an estimator to give arbitrary results as *breakdown point*. For an exact definition of the breakdown point see, e.g., Hampel (1971). Thus, LS regression has a breakdown point of 0 %.

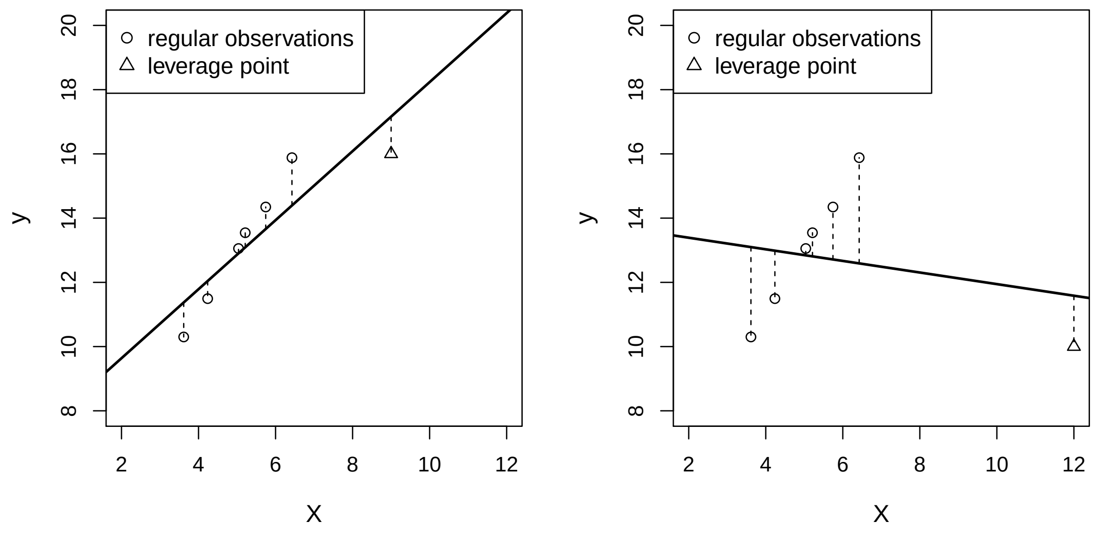

Figure 1: Effect of a single leverage point (triangle) on the LS regression line.

LS regression is straightforward and very fast to compute because of its explicit solution. In the software environment **R** it is available as function **lm**:

```
> lm(y ~ x)
```

2.2 L$_1$ Regression

One of the main reasons for the strong influence of outliers on LS regression are the properties of the squared loss function in (1), since large residuals have an even magnified effect on the overall sum. A possibility to reduce this effect is to replace the squared residuals with their absolute values:

$$\hat{\boldsymbol{\beta}}_{L1} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{n} |r_i(\boldsymbol{\beta})| \,. \tag{2}$$

This type of regression is called L$_1$ regression (Edgeworth, 1887). It was the first step in the direction of robustifying LS regression. However, it can be shown that L$_1$ regression does not achieve a higher breakdown point. Again, a single outlying observation in the x space may corrupt the results of the regression.

L$_1$ regression comes at the cost of a more complex solution. As there exists no analytical way of solving the minimization problem (2), an iterative algorithm is required. A popular method is the simplex based Barrodale-Roberts algorithm (Barrodale and Roberts, 1973), an implementation of which can be found in **R** in the **quantreg** package (function `rq`):

```
> require(quantreg)
> rq(y ~ x)
```

2.3 M Estimators

M estimators for regression were introduced by Huber (1973), and they generalize the idea of LS regression in the following way: The squared loss function is replaced by a function ρ with certain properties. This leads to the M regression criterion:

$$\hat{\boldsymbol{\beta}}_M = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{n} \rho(r_i(\boldsymbol{\beta}))\,. \tag{3}$$

We see that for $\rho(r) = r^2$, (3) is equivalent to (1) and for $\rho(r) = |r|$ we obtain the L_1 regression criterion in (2).

The criterion in (3) has the disadvantage of not being equivariant with respect to scaling of the response variable. In order to obtain the desired *regression equivariance*, the residuals r_i are standardized using a robust scale estimator $\hat{\sigma}$:

$$\hat{\boldsymbol{\beta}}_M = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \sum_{i=1}^{n} \rho\left(\frac{r_i(\boldsymbol{\beta})}{\hat{\sigma}}\right). \tag{4}$$

Differentiation of the objective function (4) with respect to the regression coefficients $\boldsymbol{\beta}$, and setting the result to zero leads to a system of p equations,

$$\sum_{i=1}^{n} \psi\left(\frac{r_i(\boldsymbol{\beta})}{\hat{\sigma}}\right) x_{ij} = 0 \quad \text{for} \quad j = 1, \ldots, p\,, \tag{5}$$

where $\psi = \rho'$. For LS regression we have $\psi(r) = r$, and in this case (5) are the usual *normal equations*.

If we define weights

$$w_i := \psi\left(\frac{r_i\,(\boldsymbol{\beta})}{\hat{\sigma}}\right) \Big/ \left(\frac{r_i\,(\boldsymbol{\beta})}{\hat{\sigma}}\right)\,, \tag{6}$$

depending on the scaled residuals, we obtain a system of weighted equations

$$\sum_{i=1}^{n} w_i \left(y_i - \boldsymbol{x}_i^\top \boldsymbol{\beta}\right) \boldsymbol{x}_i = \mathbf{0}\,. \tag{7}$$

For LS regression we get $w_i = 1$ for all i. In contrast, robust methods attempt to downweight outliers. Popular choices of the functions ρ, ψ, and the weights w_i are shown in Table 1. M estimates based on Huber's ψ function have computational advantages, but they are sensitive to leverage points. A bounded ρ function as given by Tukey's biweight is able to handle large x-values in a better way, but it results in a more difficult computation. For details, see Maronna et al. (2006). Unless the explanatory variables follow a fixed design (Mizera and Müller, 1999), the breakdown point of M estimators is $1/p$, and thus for larger p it can become very small.

In order to solve the system (7) with respect to $\boldsymbol{\beta}$, an iterative procedure called *iteratively reweighted least squares* (IRWLS) can be applied. The method repeatedly solves (7) for $\boldsymbol{\beta}$. The convergence is guaranteed for a decreasing weight function $w(r)$ (Maronna

Table 1: Different ρ functions, together with the corresponding derivatives ψ and the resulting weights w.

Type	LS	L_1	Huber	Tukey's biweight
$\rho(r)$	$\frac{r^2}{2}$	$\lvert r\rvert$	$\begin{cases} \frac{r^2}{2} & \lvert r\rvert \le c \\ c\lvert r\rvert - \frac{c^2}{2} & \lvert r\rvert > c \end{cases}$	$\begin{cases} 1-\left(1-(r/c)^2\right)^3 & \lvert r\rvert \le c \\ 1 & \lvert r\rvert > c \end{cases}$
$\psi(r)$	r	$\text{sign}(r)$	$\begin{cases} r & \lvert r\rvert \le c \\ c\cdot\text{sign}(r) & \lvert r\rvert > c \end{cases}$	$\begin{cases} 6r\left(1-(r/c)^2\right)^2 & \lvert r\rvert \le c \\ 0 & \lvert r\rvert > c \end{cases}$
$w(r)$	1	$\frac{1}{\lvert r\rvert}$	$\begin{cases} 1 & \lvert r\rvert \le c \\ \frac{c}{\lvert r\rvert} & \lvert r\rvert > c \end{cases}$	$\begin{cases} 6\left(1-(r/c)^2\right)^2 & \lvert r\rvert \le c \\ 0 & \lvert r\rvert > c \end{cases}$

et al., 2006). Since there may exist many local minima, it is important to initialize the IRWLS algorithm with a suitable starting value $\hat{\boldsymbol{\beta}}_{\mathbf{0}}$. Furthermore, the solution depends on the choice of $\hat{\sigma}$.

An alternative is the so-called H-algorithm, as presented by Huber and Dutter (1974) and Dutter (1977). The method iteratively performs LS regression using pseudo observations $\tilde{y}_i = \hat{y}_i + \psi(r_i)$, where $\hat{y}_i$ are the fitted values obtained from the previous step and r_i are the corresponding residuals. In each step, the estimates of $\boldsymbol{\beta}$ and σ are computed simultaneously. The procedure is repeated until convergence to obtain a final estimate. An implementation of the algorithm is given in the computer program LINWDR (see Dutter, 1976, 1983).

In **R** , the M estimator is available as function `rlm` in the package **MASS**, which uses the IRWLS algorithm:

```
> require(MASS)
> rlm(y ~ x, psi=psi.huber, method="M")
```

2.4 S Estimators

In the following we want to investigate regression estimators which are based on a robust scale measure $\hat{\sigma}$:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}}\, \hat{\sigma}(r_1(\boldsymbol{\beta}), \ldots, r_n(\boldsymbol{\beta}))\,. \tag{8}$$

An S estimator $\hat{\boldsymbol{\beta}}_{\boldsymbol{S}}$ is a solution of (8), with $\hat{\sigma}$ being a solution of

$$\frac{1}{n}\sum_{i=1}^{n} \rho\left(\frac{r_i(\boldsymbol{\beta})}{\sigma}\right) = \delta\,, \tag{9}$$

where δ is a constant. Equation (9) can be solved iteratively. The solution $\hat{\sigma}$ is called an *M estimator of scale*. After that, the S estimator can be computed using IRWLS. Again, a reliable starting value is needed in order to obtain a global minimum in the procedure.

The S estimator is implemented in the **R** package **robustbase**:

```
> require(robustbase)
> lmrob.S(x, y, lmrob.control())
```

Note that this function is not intended to be used on its own, because the S estimator has a low efficiency. The advantage of the S estimator is its high breakdown point of 50 %, and therefore it is used as an initial estimator for MM regression, see Section 2.7.

2.5 Least Trimmed Squares Regression

Another intuitive way of robustifying LS regression results from the following idea: If the sum of squared residuals in (1) is minimized including not all but only $h < n$ observations, possible outliers would not affect the parameter estimate $\hat{\boldsymbol{\beta}}$. This leads to the least trimmed squares (LTS) estimator which was introduced in Rousseeuw (1984). The criterion can be written as follows:

$$\left(\hat{\boldsymbol{\beta}}_{LTS}, H_{opt}\right) = \underset{(\boldsymbol{\beta},H)}{\operatorname{argmin}} \sum_{i \in H} r_i(\boldsymbol{\beta})^2 , \tag{10}$$

where $H \subseteq \{1, \ldots, n\}$ and $|H| = h < n$. Of course, the parameter h has to be chosen carefully, in order to not exclude too much of the information from the regression. Note that this criterion is equivalent to (8), if

$$\hat{\sigma} = \sqrt{\sum_{i=1}^{h} r_{(i)}^2(\boldsymbol{\beta})} ,$$

where $r_{(i)}$ are the ordered residuals. If $h = \lfloor n/2 \rfloor + 1$, a breakdown point of $(\lfloor n/2 \rfloor - p + 1)/n$ is obtained (see Rousseeuw, 1984).

A fast algorithm for the LTS regression estimator is available in the **R** package **robustbase**:

```
> require(robustbase)
> ltsReg(y ~ x)
```

2.6 Least Median of Squares Regression

The advantage of robustness of the median over the mean is well known and can be extended to the regression context: The LS criterion in (1) is equivalent to minimizing the mean of squared residuals. Rousseeuw (1984) introduced the least median of squares (LMS) estimator by replacing the mean with the median:

$$\hat{\boldsymbol{\beta}}_{LMS} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \operatorname{med}\{r_i(\boldsymbol{\beta})^2 : i = 1, \ldots, n\} . \tag{11}$$

This corresponds to (8) for $\hat{\sigma} = \operatorname{med}\{r_i(\boldsymbol{\beta})^2\}$. Like for LTS regression, an asymptotic breakdown point of 0.5 is obtained.

The LMS regression estimator is available in the **R** package **MASS**:

```
> require(MASS)
> lqs(y ~ x, method = "lms")
```

However, a reliable solution in higher dimension is computationally expensive, and thus this estimator will not be used in the simulations in Section 3.

2.7 MM Estimators

MM estimators (Yohai, 1987) reach a high level of robustness as well as high (tunable) efficiency, by combining the properties of M estimators and S estimators. Let $\hat{\boldsymbol{\beta}}_0$ be an S estimator, and let $\hat{\sigma}$ be the corresponding M estimator of scale, solving (9) for $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}}_0$. The MM estimator is then defined as local solution of

$$\hat{\boldsymbol{\beta}}_{MM} = \underset{\boldsymbol{\beta}}{\text{argmin}} \sum_{i=1}^{n} \rho\left(\frac{r_i(\boldsymbol{\beta})}{\hat{\sigma}}\right), \tag{12}$$

obtained with IRWLS and initial value $\hat{\boldsymbol{\beta}}_0$.

A implementation of MM estimators is available in the package **MASS** (function `rlm`). A fast alternative can be found in the **R** package **robustbase**:

```
> require(robustbase)
> lmrob(y ~ x)
```

3 Comparison

The robust regression methods listed in the previous sections have been intensively studied during the last years (decades). Accordingly, a lot is known about their theoretical properties, in particular about their robustness properties. These properties, however, can be partially weakened by the algorithms for the computation of the estimators. In **R** we can find fast algorithms, like a fast algorithm for LTS regression (Rousseeuw and Van Driessen, 2006), or a fast algorithm for regression S estimators (Salibian-Barrera and Yohai, 2006) (and therefore for MM estimators). Without these fast implementations it would be impossible to deal with larger data sets, but one has to be aware that the algorithms not necessarily find the global optimum. This is in particular problematic if the number of explanatory variables gets larger. In the following we compare the previously described regression methods in simulated data examples, where the true regression parameters are known.

3.1 Different Contamination Schemes

In this simulation experiment we generate the data as follows:

$\boldsymbol{x}_i \sim N(\mathbf{0}, \boldsymbol{I}_p)$ and $\epsilon_i \sim N(0, 0.5)$, for $i = 1, \ldots, n$,
$\beta_j = 1/\sqrt{p}$ for $j = 1, \ldots p$, and thus $\|\boldsymbol{\beta}\| = 1$,
$y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \epsilon_i$, for $i = 1, \ldots, n$.

This setting is referred to as *uncontaminated data*. In a second setting we create *vertical outliers* by changing the first n_{out} elements of the error term to

$$\epsilon_i \sim N(10, 0.1), \text{ and } y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \epsilon_i, \text{ for } i = 1, \dots, n_{out}.$$

The third setting refers to *leverage points*, where the first n_{out} observations of the explanatory variables and the response are replaced with

$$\boldsymbol{x}_i = (10, \dots, 10)^\top \text{ and } y_i = \boldsymbol{x}_i^\top \boldsymbol{a}, \text{ for } i = 1, \dots, n_{out},$$

where $\boldsymbol{a}$ is orthogonal to $\boldsymbol{\beta}$, and is constructed using a vector $\boldsymbol{v}$ with entries $(-1)^j$ by $\boldsymbol{a} = \boldsymbol{v} - (\boldsymbol{v}^\top \boldsymbol{\beta})\boldsymbol{\beta}$, normalized to unit norm. For $p = 1$ we take $a = -1$. The leverage points are thus placed on the least favorable position, see also Figure 5.

In this simulation we take $n = 200$ observations, and $n_{out} = 20$ are replaced by either vertical outliers or leverage points. The number p of explanatory variables is taken from the set $\{1, 5, 10, 25, 50\}$. For the three scenarios, we apply the described regression methods in $m = 100$ replications. As a measure of performance we use the mean squared error between the estimated parameters in the l-th repetition, $\hat{\boldsymbol{\beta}}^{(l)}$, and the true parameters $\boldsymbol{\beta}$:

$$\text{MSE} = \frac{1}{m} \sum_{l=1}^{m} \frac{1}{p} \|\hat{\boldsymbol{\beta}}^{(l)} - \boldsymbol{\beta}\|^2 . \tag{13}$$

The results are shown in Figure 2. The upper plot panels represent the uncontaminated case, the middle plot panels are for the vertical outliers, and the lower plot panels for the leverage points. The right panels zoom into the left figures to show some details. The legend is placed in the lower left panel. In the uncontaminated case we see that, independent of the dimension, LS regression gives the smallest MSE (as expected), closely followed by M and MM regression. L$_1$ and LTS regression are slightly worse, and S regression is the worst among these estimators due to its low efficiency. When including vertical outliers, LS regression is only reliable for $p = 1$. This situation is a special case, caused by the specific design of the simulated data. For the other methods we see that MM regression gives the best results, closely followed by LTS. For the scenario with leverage points, MM estimation gives again the best performance, followed by LTS and S estimation. Their resulting MSEs are even close to the uncontaminated case. Figure 3 (lower left panel) shows that LS, L$_1$ and M regression break down, however, one gets the impression that with increasing dimension these regression methods improve. This is not correct for the following reasons. Since the leverage points were included in the orthogonal direction to $\boldsymbol{\beta}$, the worst possible breakdown of a regression estimator is a vector of estimated regression coefficients $\hat{\boldsymbol{\beta}}$ orthogonal to $\boldsymbol{\beta}$, i.e. $\hat{\boldsymbol{\beta}}^\top \boldsymbol{\beta} = 0$. Since the simulation design is without intercept, for $p = 1$ we have $\beta = 1$ and the worst possible estimation $\hat{\beta} = -1$. The resulting expected MSE according to (13) is 4, which, due to Figure 3, is indeed attained for LS regression. For larger p, the expected MSE gives in the worst possible situation

$$\text{MSE} = \frac{1}{p}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) = \frac{1}{p}(\hat{\boldsymbol{\beta}}^\top \hat{\boldsymbol{\beta}} + \boldsymbol{\beta}^\top \boldsymbol{\beta} - 0) = \frac{2}{p} ,$$

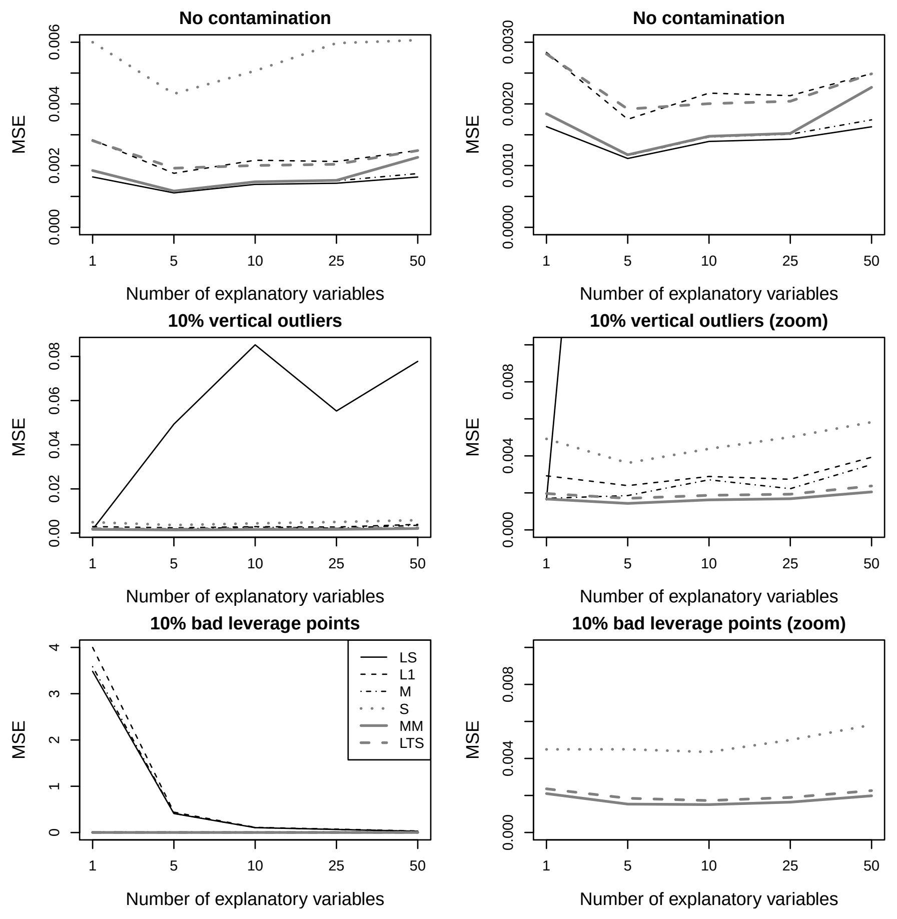

Figure 2: Mean squared errors between estimated and true regression parameters for different numbers of explanatory variables. The upper panels are for the uncontaminated case, the middle panels for vertical outliers, and the lower panes for leverage points. The right panels zoom into interesting parts of the left panels. The legend is placed in the lower left panel.

if we assume that $\hat{\beta}$ has norm 1. LS regression is indeed close to these values, which have to decrease with increasing dimension.

Figure 3 shows the average computation times in seconds of the regression methods (left: original scale, right: log-scale), using a standard personal computer. With an increasing number of explanatory variables, the time for computing the S estimator increases rapidly. Since the MM estimator is based on the S estimator, its computing time is almost the same. The computation time for LTS also increases exponentially, but it is much lower in absolute terms. The time increase for the other estimators is much smaller.

3.2 Breakdown

We use a very similar simulation design as above to evaluate the regression methods (algorithms) for their breakdown behavior. The idea is to include an increasing amount of

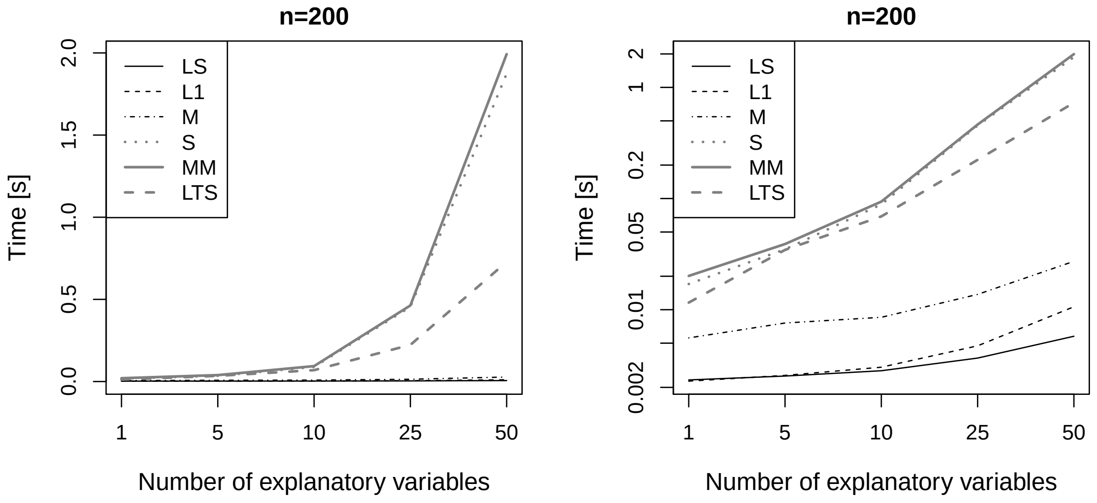

Figure 3: Average computation time (in seconds) depending on the number of explanatory variables; left: original scale, right: log-scale.

contamination. The most severe contamination is to include leverage points, and thus the simulation design is as follows:

$$\boldsymbol{x}_i \sim N(\boldsymbol{0}, \boldsymbol{I}_p) \text{ and } \epsilon_i \sim N(0, 0.1), \text{ for } i = 1, \ldots, n,$$
$$\beta_j = 1/\sqrt{p} \text{ for } j = 1, \ldots p, \text{ and thus } ||\boldsymbol{\beta}|| = 1,$$
$$y_i = \boldsymbol{x}_i^\top \boldsymbol{\beta} + \epsilon_i, \text{ for } i = 1, \ldots, n.$$

The first n_{out} values of the response and the explanatory variables are replaced with

$$\boldsymbol{x}_i = (10, \ldots, 10)^\top, \text{ and } y_i = \boldsymbol{x}_i^\top \boldsymbol{a}, \text{ for } i = 1, \ldots, n_{out},$$

where $\boldsymbol{a}$ is taken as before, and $n_{out} = f \cdot n$ with $f = 0, 0.05, \ldots, 0.5$ and $n = 200$. Then the MSE from (13) is computed over $m = 100$ repetitions of the simulation. Figure 4 shows the results (legend in the bottom panels). Theoretically, the breakdown point of the estimators S, MM and LTS, is 50 %, and also the default parameters in their **R** implementations are set to achieve this maximum possible breakdown point. Practically, the breakdown depends on the simulation setting, and mainly on the implemented algorithm. We can see that with increasing number of explanatory variables, the breakdown occurs much earlier. The S, MM and LTS estimator show a very comparable performance. On the other hand, LS regression breaks down already with 5 % contamination. It is interesting that for $p = 50$, L$_1$ and M regression are still robust against 5 % contamination (leverage points). This might be due to the special geometry in higher-dimensional spaces.

In a final simulation example we demonstrate that the breakdown of a regression line can indeed depend on the regression problem. We use the simulation design described in this subsection for 25 % contamination and $p = 1$. Here the leverage points are not placed on one point but spread along the orthogonal direction for visual purposes. Figure 5 shows a generated data set according to this setting. In the left plot we use $\epsilon_i \sim N(0, 0.5)$, as in the simulation of Section 3.1, in the right plot we increase the error variance using $\epsilon_i \sim N(0, 1)$. Thus, in the latter case, the relation between X and y becomes weaker, but

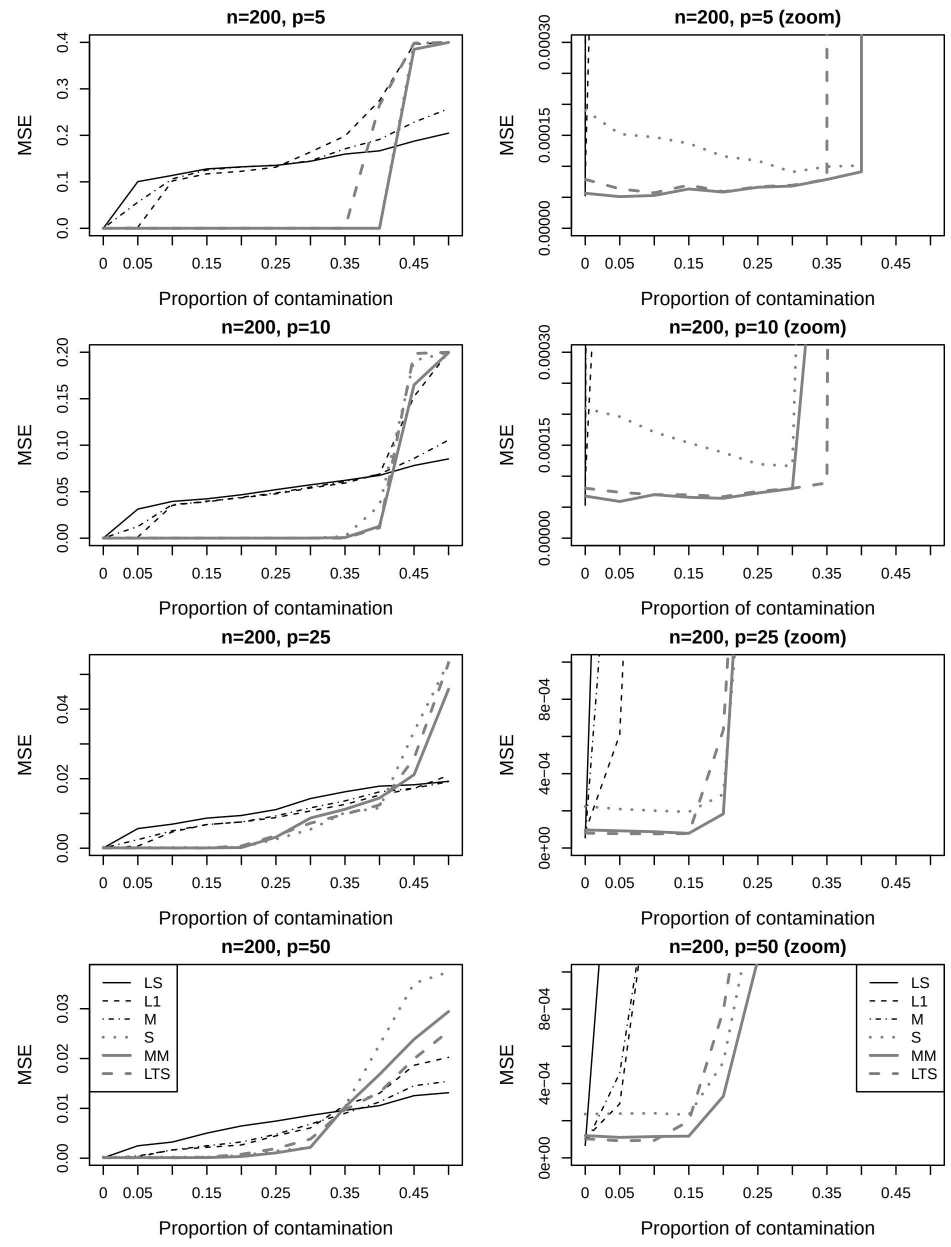

Figure 4: Mean squared errors between estimated and true regression parameters for different proportions of contamination and different numbers of explanatory variables. The right panels zoom into interesting parts of the left panels. The legend is placed in the lower panels.

LS regression on the clean data still results in an R^2 measure of 42 %. We applied MM and LTS regression, and in the first case both methods are robust, while in the second case they break down. This is not a consequence of the implemented algorithm, but of the simulated data. One has to keep this behavior in mind when applying robust regression to real data, where it can happen that the theoretical properties of the estimators can be quite different from the empirical results.

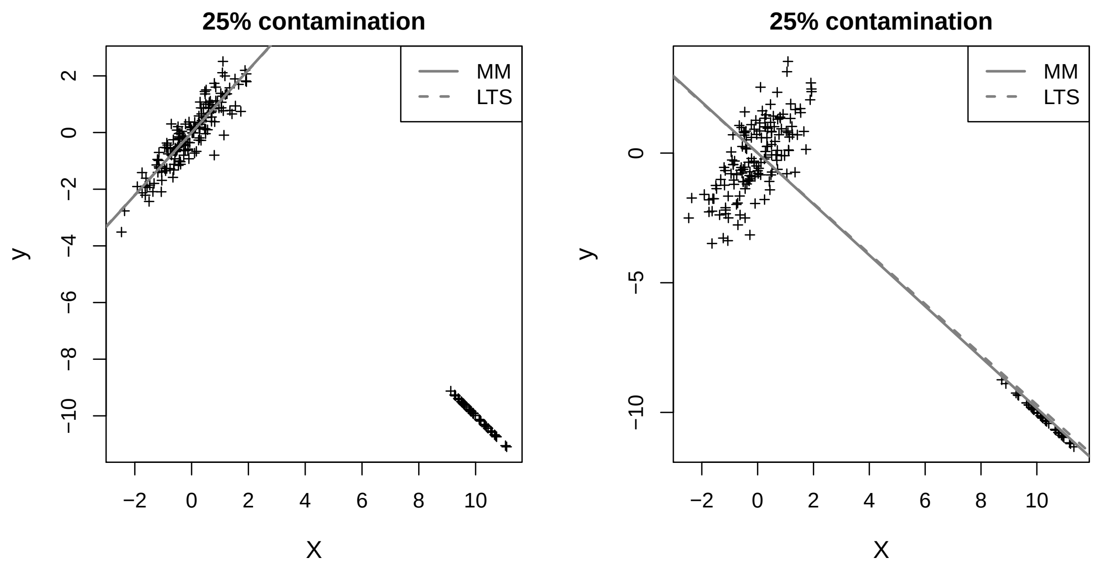

Figure 5: Simulated data example with 25 % leverage points. MM and LTS regression is robust in the left panel, but fails in the right panel where the residual variance has been increased.

4 Conclusions

The **R** package **robustbase** offers various possibilities for robust regression, in particular routines for computing the MM and LTS regression estimators (`lmrob` and `ltsReg`). These functions are implementations of fast algorithms (Salibian-Barrera and Yohai, 2006; Rousseeuw and Van Driessen, 2006), which can handle regression problems with a reasonably large number of explanatory variables. As we have shown by simulations, the theoretical robustness properties of these estimators can be far too optimistic in practice, which is due to the algorithm but also due to the problem at hand. Especially in higher dimension the estimators can break down already at moderate levels of contamination, much lower than 50 %. This could be avoided by changing the default parameters of the routines—a task that is not straightforward to the average user. This does not mean that robust regression becomes useless in higher dimension; in contrary: Thanks to the fast algorithms it is possible to obtain results in reasonable time, but these are a compromise between computation time and optimal robustness properties.

Acknowledgment

We are grateful to Rudi Dutter for all we could learn from him—and this is a lot more than just statistics.

References

Barrodale, I., and Roberts, F. D. K. (1973). An improved algorithm for discrete l1 linear approximation. *SIAM Journal on Numerical Analysis*, *10*(5), 839–848.

Boscovich, R. (1757). De literaria expeditione per pontificiam ditionem et synopsis amplioris operis. *Bononiensi Scientiarum et Artum Instituto atque Academia Commentarii*, *4*, 353-396.

Dutter, R. (1976). *Computer linear robust curve fitting program LINWDR* (Tech. Rep.). ETH Zürich, Switzerland: Fachgruppe für Statistik.

Dutter, R. (1977). Numerical solution of robust regression problems: Computational aspects, a comparison. *Journal of Statistical Computation and Simulation*, *5*, 207–238.

Dutter, R. (1983). *Computer program BLINWDR for robust and bounded influence regression* (Tech. Rep.). Technische Universität, Graz, Austria: Institut für Statistik.

Edgeworth, F. Y. (1887). On observations relating to several quantities. *Hermathena*, *6*, 279–285.

Hampel, F. R. (1971). A general qualitative definition of robustness. *Annals of Mathematical Statistics*, *42*, 1887–1896.

Huber, P. (1973). Robust regression: Asymptotics, conjectures, and monte carlo. *Annals of Statistics*, *1*(5), 799–821.

Huber, P., and Dutter, R. (1974). Numerical solution of robust regression problems. In *Proceedings in computational statistics* (pp. 165–172). Vienna: COMPSTAT 1974.

Maronna, R. A., Martin, R. D., and Yohai, V. J. (2006). *Robust statistics. theory and methods*. New York: John Wiley & Sons, Ltd.

Mizera, I., and Müller, C. (1999). Breakdown points and variation exponents of robust M-estimators in linear models. *Annals of Statistics*, *27*(4), 1164-1177.

Rousseeuw, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association*, *79*(388), 871–880.

Rousseeuw, P. J., and Leroy, A. (2003). *Robust regression and outlier detection*. New York: John Wiley & Sons, Ltd.

Rousseeuw, P. J., and Van Driessen, K. (2006). Computing LTS regression for large data sets. *Data Mining and Knowledge Discovery*, *12*(1), 29-45.

Salibian-Barrera, M., and Yohai, V. (2006). A fast algorithm for S-regression estimates. *Journal of Computational and Graphical Statistics*, *15*, 414-427.

Yohai, V. J. (1987). High breakdown-point and high efficiency robust estimates for regression. *The Annals of Statistics*, *15*(2), 642–656.

Authors' address:

Moritz Gschwandtner and Peter Filzmoser
Department of Statistics and Probability Theory
Vienna University of Technology
Wiedner Hauptstr. 8-10
1040 Vienna
Austria