Fast and robust bootstrap for LTS

Gert Willems [a,*], Stefan Van Aelst [b]

[a] *Department of Mathematics and Computer Science, University of Antwerp, Middelheimlaan 1, B-2020 Antwerp, Belgium*

[b] *Department of Applied Mathematics and Computer Science, Ghent University, Krijgslaan 281 S9, B-9000 Ghent, Belgium*

Abstract

The Least Trimmed Squares (LTS) estimator is a frequently used robust estimator of regression. When it comes to inference for the parameters of the regression model, the asymptotic normality of the LTS estimator can be used. However, this is usually not appropriate in situations where the use of robust estimators is recommended. The bootstrap method constitutes an alternative, but has two major drawbacks. First, since the LTS in itself is a computer-intensive estimator, the classical bootstrap can be extremely time-consuming. And second, the breakdown point of the procedure is lower than that of the estimator itself. To overcome these problems, an alternative bootstrap method is proposed which is both computationally simple and robust. In each bootstrap sample, instead of recalculating the LTS estimates, an approximation is computed using information from the LTS solution in the original sample. A simulation study shows that this method performs well, particularly regarding confidence intervals for the regression parameters. An example is given to illustrate the benefits of the method.

Key words: Least trimmed squares, Bootstrap, Robust inference

1 Introduction

It is well known that the classical least squares estimator for the linear regression model is extremely sensitive to outliers in the data. Therefore, several

* Corresponding author: Department of Mathematics and Computer Science, University of Antwerp, Middelheimlaan 1, B-2020 Antwerp, Belgium, Tel: +32-3-2653896, Fax:+32-3-2653777.

Email addresses: gert.willems@ua.ac.be (Gert Willems), stefan.vanaelst@ugent.be (Stefan Van Aelst).

robust alternatives have been investigated in the literature. Among those, Rousseeuw's Least Trimmed Squares (LTS) estimator [5] is a popular choice mainly because of its computability and its intuitively appealing definition. The LTS estimator minimizes a trimmed sum of squared residuals, thereby allowing some potentially influential observations to have large residuals. In this way, outliers do not necessarily affect the estimates of the model parameters, as they do in case of the least squares estimator.

When it comes to inference concerning the parameters of the regression model there are two standard possibilities. The first is to approximate the standard error of the LTS estimates by using their asymptotic variances, see [6]. However, this asymptotic result only holds for some specified underlying model distributions, such as the central normal model. Empirical versions of the asymptotic variances can be used in practice but they are not likely to yield accurate approximations in situations where robust methods are recommended.

Alternatively, the sampling distribution of LTS estimates can be estimated using the bootstrap method [3]. However, two important drawbacks arise when using classical bootstrap on a robust estimator like LTS. First, although nowadays there exists a reasonably fast algorithm to compute the LTS [7], the estimator still is computer-intensive. Especially for high dimensional data, computing e.g. 1000 recalculated bootstrap estimates might not be feasible due to the computational cost.

The second problem concerns the robustness of the method. Even if the estimator is resistant to the proportion of outliers appearing in the original data, when taking a bootstrap sample this proportion can become high enough to break down the estimator for that particular sample. As a consequence, variance estimates or confidence intervals based on the resulting bootstrap distribution can break down even if the original LTS estimate did not [9,10]. In other words, the classical bootstrap estimates are not as robust as the estimator that is bootstrapped.

Recently, a robust and fast bootstrap method was developed [8,11] for the class of robust regression estimators that can be represented as a solution of a smooth fixed-point equation. This class includes MM-, GM- and S-estimators, but not the LTS estimator. In this paper we propose a simple approximating bootstrap method for LTS which is both fast and robust. The idea is to draw bootstrap resamples (just as in classical bootstrap), but instead of applying the actual LTS algorithm to each resample, we compute an approximation by using information gathered from the LTS solution of the original data set. Simulations show that this fast method performs well, both in case of regular (outlier-free) data as in case of contaminated data. Hence this inference method is a preferable choice in all cases.

The rest of the paper is organized as follows. In Section 2 the LTS estimator and its properties are described. Section 3 introduces the fast and robust bootstrap method and Section 4 shows some simulation results. In Section 5 the method is illustrated on an example, while Section 6 concludes.

2 Least Trimmed Squares Estimator

Consider the univariate regression model given by

$$y_i = \boldsymbol{x}_i'\boldsymbol{\beta} + \epsilon_i, \qquad i = 1, \ldots, n.$$

Here, $\boldsymbol{x}_i \in I\!R^p$ are the regressors, $y_i \in I\!R$ is the response and $\epsilon_i \in I\!R$ is the error term. It is assumed that the errors are independent and homoscedastic with zero center and unknown scale σ. For every $\boldsymbol{\beta} \in I\!R^p$ we denote the corresponding residuals by $r_i(\boldsymbol{\beta}) = r_i := y_i - \boldsymbol{x}_i'\boldsymbol{\beta}$ and $r^2_{1:n} \leq \ldots \leq r^2_{n:n}$ denote the ordered squared residuals.

The LTS estimator minimizes the objective function $\sum_{i=1}^{h} r^2_{i:n}$, where h is to be chosen between $\frac{n}{2}$ and n. This is equivalent to finding the subset of size h with the smallest least squares objective function. The LTS estimate of $\boldsymbol{\beta}$ is then the least squares estimate of that subset. The estimate of σ is given by the corresponding least squares scale estimate, multiplied by a consistency factor depending on the ratio h/n, and a finite-sample correction factor depending on h, n and p to obtain unbiasedness at the normal model (see [4]).

If the data come from a continuous distribution, the breakdown value of the LTS equals $\min(n-h+1, h-p+1)/n$. We have that $h = [(n+p+1)/2]$ yields the maximum breakdown value, which asymptotically equals 50%, whereas $h = n$ gives the ordinary least squares estimator with breakdown value $1/n$. As is the case with most robust estimators, there is a trade-off between robustness and efficiency. We prefer to use $h \approx 0.75n$ which is considered to be a good compromise, yielding an asymptotic breakdown value of 25%.

The LTS is an intuitively appealing regression estimator that also has some desirable formal properties such as affine equivariance and asymptotic normality. Moreover, its influence function is bounded for both vertical outliers and bad leverage points. As already stated, the LTS breakdown value can be set to any value between 0% and 50%.

The computation of LTS estimates is not a straightforward task. For large datasets in high dimensions it is practically not feasible to find the exact solution. Usually one turns to approximating algorithms and in this paper we will use the recently developed FAST-LTS algorithm [7]. It should be noted that this particular algorithm then has to be regarded as the actual estimator.

The FAST-LTS algorithm aims to find the h-subset which yields the smallest objective function. To find the exact minimum, it would have to consider every possible subset of size h, which is not practical for large datasets. Therefore, the algorithm will typically find a local minimum which is close to the global minimum, but not necessarily equal to that global minimum. A key element of the algorithm is the fact that starting from any h-subset, it is possible to construct another h-subset yielding a lower value of the objective function. Rousseeuw and Van Driessen [7] call this a *C-step*, and it works as follows:

- suppose we have an h-subset H_{old} with corresponding LS-estimate $\boldsymbol{\beta}_{\text{old}}$
- compute the residuals $r_i = y_i - \boldsymbol{x}_i'\boldsymbol{\beta}_{\text{old}}$, $i = 1, \ldots, n$
- set $H_{\text{new}} := \{$ h observations with smallest $r_i^2\}$.

The least squares estimate $\boldsymbol{\beta}_{\text{new}}$, based on H_{new}, and its corresponding residuals then yield a value of the objective function that is smaller or equal to that of H_{old}. The basic idea of the FAST-LTS algorithm is to construct many initial h-subsets, apply C-steps to each of them until convergence, and keep the solution with the lowest value of the objective function.

3 Bootstrapping the LTS estimator

We are now interested in obtaining inference, such as confidence intervals, for the parameters $\boldsymbol{\beta}$ and σ. For this we can use the asymptotic normality of LTS but, as pointed out in the introduction, the asymptotic variance is not available for situations in which the use of LTS is recommended, such as the situation of suspected severe non-normality of the errors. Therefore we turn to bootstrap methods.

3.1 Classical bootstrap

The use of the bootstrap method is gradually increasing nowadays, due to increasing computer power. The basic idea is to generate a large number of samples by randomly drawing observations with replacement from the original dataset, and to recalculate the estimates for each of these bootstrap samples. The empirical distributions of the bootstrap estimates $\widehat{\boldsymbol{\beta}}^*$ and $\widehat{\sigma}^*$ are an approximation to the true sample distributions of $\widehat{\boldsymbol{\beta}}$ and $\widehat{\sigma}$. However, recalculating the LTS estimator (using FAST-LTS) for each bootstrap sample is extremely time-consuming, and might not be feasible for large datasets, despite considerable computer power. Table 1 lists some computation times for the classical bootstrap procedure for LTS on simulated datasets. The number of bootstrap samples was set to $B = 1000$, which is generally regarded as

Table 1
Computation time for classical bootstrap on LTS with $B = 1000$ (in CPU minutes, on Pentium IV 1.9 Ghz)

	$n = 20$	$n = 50$	$n = 200$	$n = 1000$
$p = 2$	5.5	17.0	32.9	168.9
$p = 5$	18.6	20.0	36.1	188.6

the minimum number needed to get accurate bootstrap confidence limits. We used a Matlab implementation of the FAST-LTS algorithm. As can be seen from Table 1, in practice the computation time is a serious drawback of the bootstrap method.

The second problem is the lack of robustness, in particular the fact that the bootstrap estimates are less robust than the LTS estimates itself. An important notion in robustness is the breakdown value ϵ_n^* of an estimator. It is defined as the minimum fraction of the observations in the dataset that need to be shifted for the estimator to take on arbitrary values (see e.g. [6]). In other words, ϵ_n^* indicates the minimum proportion of outliers in the data that the estimator is not able to resist anymore. Now, for any estimator, denote by V^* the regular bootstrap variance estimate, being the empirical variance of the bootstrap recalculations of the estimator. Then, as pointed out in [10],

$$\epsilon_n^*(V^*) \underset{B\to\infty}{\longrightarrow} \frac{1}{n},$$

regardless of the breakdown point of the estimator that is bootstrapped. This result is intuitively clear since the presence of a single bad observation in the dataset makes it theoretically possible that a bootstrap sample emerges with more bad observations than the estimator can resist. Furthermore, one contaminated bootstrap sample suffices to break down the empirical variance V^*. In practice, the convergence is rather slow and the bootstrap method is unlikely to fail if the proportion of bad observations is small, but break-down is still possible. Now denote by Q_t^* the t-th quantile of some marginal bootstrap distribution of the estimator. These quantiles are used to estimate confidence limits for the estimated parameters. Similar reasoning can now be applied to show that the breakdown point of Q_t^* is lower than that of the estimator itself, although not as low as that of V^* (see [9,8]). A number of straightforward robustifications of the bootstrap have been proposed in the literature (see [10]). For example, one could replace the empirical variance V^* by a more robust measure of variability, such as the interquartile range. Another possibility is to exclude bootstrap samples that select the same index more than m^* times, but the appropriate choice for m^* is unclear. Still another possibility is to delete the outliers identified by the initial estimator and bootstrap only the remaining observations. Clearly, it is not difficult to adapt the bootstrap for LTS such that the robustness problem is more or less solved. However, the problem of

computational feasibility remains. Next we will propose a bootstrap method for LTS, that solves both problems.

3.2 A fast and robust bootstrap method

The main reason for the high computation time of the FAST-LTS algorithm is that it needs to start from a large number of initial subsets. This large number is necessary to obtain a sufficient probability that at least one of the initial subsets is not contaminated with outliers. When bootstrap samples are drawn from the original dataset, they consist of observations also present in the original dataset. Often it will be the case that observations identified as outliers by the original LTS estimate, are also outliers when they appear in bootstrap samples. This consideration leads to a short-cut when searching for the LTS solution of a bootstrap sample. We propose the following bootstrap procedure which intends to mimic the classical bootstrap. First, in the original dataset, label those observations $(\boldsymbol{x}_i, y_i)$ for which $|r_i(\widehat{\boldsymbol{\beta}}_{LTS})/\widehat{\sigma}_{LTS}| > \Phi^{-1}(0.9875)$ as outliers. Then as in the classical procedure, draw B bootstrap samples from the complete original data, and for each bootstrap sample compute an approximate LTS solution in the following way:

- Draw 1 random initial h-subset out of the observations that were not labeled as outliers in the original dataset.
- Apply C-steps on this initial h-subset until convergence to obtain the approximate LTS solution. While excluded from the initial subset, observations labeled as outliers are allowed to be included in this process.
- In case the number of non-outlying observations h' in the resample is less than h, then replace h with h' in the previous steps for this particular resample. In this way we can deal with highly contaminated bootstrap samples.

So we assume here that indeed the outliers in a bootstrap sample are mostly the same observations as those identified in the original sample. In this way we avoid the need for many initial h-subsets. Furthermore we use the effectiveness of the C-step to obtain a local minimum of the LTS objective function starting from the initial subset.

We now argue that the 'short-cut' procedure satisfies the following properties:

(1) more robust than the classical bootstrap
(2) faster than the classical bootstrap
(3) the variability of the (FAST-)LTS estimator is accurately mimicked.

With regard to the first property, it is clear that due to the choice of the initial subset for the bootstrap samples the short-cut procedure generally will not break down as long as the original LTS estimate did not break down. Theoret-

Table 2
Average number of observations shared by the final h-subsets, with quartiles Q_1 and Q_3 (based on 500 samples with normal errors and $p = 5$ regressors).

# C-steps:		0	1	2	3	∞	
	Q_1	29	34	35	35	35	
$n = 50$	Avg	30.3	35.7	36.2	36.3	36.4	$(h = 39)$
	Q_3	31	37	38	38	39	
	Q_1	112	139	141	142	143	
$n = 200$	Avg	113.8	142.0	144.0	145.0	145.8	$(h = 151)$
	Q_3	116	145	148	149	150	

ically it could happen though that a bootstrap sample emerges with less than p non-outlying observations, in which case breakdown is possible. However, in practice these situations hardly occur. We thus assert that the proposed short-cut procedure is much more robust than the classical bootstrap. See also the example in Section 5.

Naturally, the short-cut procedure is much faster than running FAST-LTS for each bootstrap sample. Implementations of the FAST-LTS algorithm typically start with 500 random initial subsets, of which a number is discarded after 2 C-steps. Instead the short-cut procedure starts with just one (well chosen) initial subset. Note that we could also consider two or more well chosen initial subsets instead of one, which then still would be relatively fast and might perform even better. In this paper we restrict our method to one initial subset and show that this yields good results.

The short-cut in effect attempts to achieve an accurate reflection of the variability by aiming for a good (and fast) approximation of the FAST-LTS solution in each resample when there are not too many outliers, while on the other hand applying a robustification in case of severe contamination. A good approximation of the estimate in each resample would mean that the final h-subsets selected by both estimators, FAST-LTS and the short-cut, are sufficiently similar. We performed some simulations on datasets without outliers. Table 2 shows the average number of observations shared by the final h-subset of the FAST-LTS algorithm and that of the short-cut procedure after various numbers of performed C-steps. The ∞-entries correspond to the full short-cut (after convergence of the C-steps). The average is taken over 500 resamples, each one from a bootstrap procedure on a different simulated dataset. The number of regressors was set to $p = 5$, the errors were generated from a normal distribution. Results for $n = 50$ and $n = 200$ are given and we set $h \approx 0.75n$. The first and third quartile of the 500 samples are also shown. We see how the short-cut subset becomes more similar to the FAST-LTS subset with each additional C-step. The average number of C-steps that it takes to

converge is about 2.9 for $n = 50$ and 4.5 for $n = 200$. In the next section an extended simulation study will show how accurate the variance of the FAST-LTS is being approximated by the short-cut.

4 Simulations

In this section we will show that in spite of its simple and approximating nature, the short-cut procedure generally performs well. Through simulations we investigate the performance of two inference results provided by the procedure. The first is the estimated variance of the LTS estimates, while the second is the coverage and length of univariate confidence intervals for the regression parameters.

Simulations were performed for sample sizes $n = 50, 200$ and 500 and dimensions $p = 5$ and 10. An intercept term was included by setting $x_{ip} = 1, i = 1, \ldots, n$. The remaining regressors were generated from the $(p-1)$-variate normal distribution $N(\mathbf{0}, \mathbf{I})$. The true value of the parameter $\boldsymbol{\beta}$ was set equal to $(0, \ldots, 0)'$. However, this choice does not affect the performance results since the LTS is regression, scale and affine equivariant. We now consider the following cases:

(1) *normal* errors, generated from $N(0, 1)$
(2) *long-tailed* errors, generated from t_3 (Student-t, 3 d.f.)
(3) *far outliers*, proportion 80% of the errors generated from $N(0, 1)$ and proportion 20% generated from $N(10, 0.1)$.

For each case 2000 data sets were generated and we computed LTS estimates with $h \approx 0.75n$. On each data set we applied the short-cut bootstrap procedure as described in the previous section, with $B = 1000$. In computing the bootstrap scale estimates we used the consistency and small sample correction factors as proposed in [4]. These factors were also incorporated in the Matlab implementation of FAST-LTS that was used throughout this paper.

Let us first focus on the variance estimates of the estimators. The short-cut bootstrap variance estimate is the empirical variance of the B recalculated estimates. We also computed for each dataset an estimate of the asymptotic variance of LTS as given in [6] for the regression coefficients, and in [1] for the error scale σ. The estimate consists of an empirical version of the analytical result for symmetric distributions. For the coefficients we used:

$$\widehat{\mathrm{ASV}}((\widehat{\boldsymbol{\beta}}_{LTS})_j) = \left[\left(\sum_{i=1}^{n} \boldsymbol{x}_i \boldsymbol{x}_i^t\right)^{-1}\right]_{jj} \frac{\frac{1}{n}\sum_{i=1}^{h} r_{i:n}^2(\widehat{\boldsymbol{\beta}}_{LTS})}{(h/n - 2\widehat{z}_h\, \phi(\widehat{z}_h))^2},$$

Table 3
LTS variance estimates ($\times 100$), compared with Monte Carlo; normal errors

		$n = 50$				$n = 200$			
p		MC	0 C	∞ C	$\widehat{\text{ASV}}$	MC	0 C	∞ C	$\widehat{\text{ASV}}$
	slope	6.49	2.50	5.39	7.13	1.76	0.58	1.42	1.77
5	int	6.01	2.34	5.09	6.85	1.67	0.57	1.42	1.76
	scale	2.02	2.58	2.67	2.99	0.48	0.52	0.51	0.57
	slope	7.43	2.93	5.99	6.28	1.70	0.59	1.43	1.69
10	int	6.85	2.70	5.58	6.04	1.77	0.58	1.42	1.68
	scale	2.24	3.25	3.87	4.45	0.48	0.59	0.58	0.67

where $\widehat{z}_h = r_{h:n}(\widehat{\boldsymbol{\beta}}_{LTS})/\widehat{\sigma}_{LTS}$. For the scale estimate an analogous expression was used. Both methods are compared to a Monte Carlo estimate (MC) of the variances, which are considered to be good approximations of the variances of the LTS estimator. Note that we did not include the classical bootstrap in this simulation study because of its high computation time.

Table 3 lists the variance estimates for the case of normal errors. Besides the actual short-cut estimate (∞ C) we also show the results for the short-cut without C-steps (0 C), to show the effect of the C-steps (see also Table 2). The values shown are the averages over the 2000 samples. The entries corresponding to the slope are in turn averages over the $p-1$ coefficients. The results for $n = 500$ are not shown here due to lack of space, but were found to be comparable to those for $n = 200$. Table 4 and 5 give the results for the long-tailed errors and the far outliers respectively. Here we left out the entries for the short-cut without C-steps.

We see from Table 3 that the short-cut estimates are not too far from the MC

Table 4
LTS variance estimates ($\times 100$), compared with Monte Carlo; t_3 errors

		$n = 50$			$n = 200$			$n = 500$		
p		MC	∞ C	$\widehat{\text{ASV}}$	MC	∞ C	$\widehat{\text{ASV}}$	MC	∞ C	$\widehat{\text{ASV}}$
	slope	6.56	6.66	10.63	1.40	1.37	2.48	0.55	0.53	0.99
5	int	6.06	6.07	10.26	1.37	1.33	2.46	0.53	0.52	0.98
	scale	4.01	4.81	4.57	0.83	0.91	0.83	0.34	0.34	0.30
	slope	8.37	8.43	10.10	1.48	1.48	2.43	0.54	0.54	0.98
10	int	7.97	7.74	9.70	1.46	1.43	2.40	0.53	0.54	0.97
	scale	4.77	6.84	7.32	0.95	1.01	0.99	0.35	0.35	0.33

Table 5
LTS variance estimates(×100), compared with Monte Carlo; 20% far outliers

		$n = 50$			$n = 200$			$n = 500$		
p		MC	∞ C	$\widehat{\text{ASV}}$	MC	∞ C	$\widehat{\text{ASV}}$	MC	∞ C	$\widehat{\text{ASV}}$
	slope	3.47	4.41	17.21	0.97	0.98	3.62	0.36	0.38	1.43
5	int	3.37	4.01	16.47	0.89	0.95	3.58	0.36	0.38	1.42
	scale	3.99	6.81	7.89	0.87	2.01	1.28	0.31	0.76	0.45
	slope	3.45	5.83	20.24	0.93	1.02	3.67	0.37	0.39	1.44
10	int	3.29	5.35	19.42	0.91	1.00	3.63	0.37	0.38	1.43
	scale	5.86	8.95	15.85	0.94	2.33	1.59	0.32	0.83	0.51

variance of the LTS coefficients, but there seems to be a consistent underestimation. The latter is not present for the long-tailed errors, where the method apparently performs very well, as can be seen from Table 4. For the case of far outliers Table 5 indicates an overestimation, which becomes smaller as the sample size grows. These results can be explained as follows. For normal errors very few observations will be labeled as outliers. Therefore the initial h-subset is approximately a random h-subsample of the bootstrap sample. The estimate corresponding to this initial subset is then the least squares (LS) estimate of a random h-subsample. Accordingly, the variance estimates of the short-cut without C-steps (0 C) are close to the variance of the least squares estimator for sample size h. The C-steps are subsequently applied in an attempt to move the estimate roughly from the LS solution towards the LTS solution. This aim will not always be achieved and often the estimate ends up in between the LS and the LTS estimates. A result is that the variance of the bootstrap estimates is higher than the LS variance corresponding to sample size h, but still somewhat lower than the LTS variance. Presumably there are two reasons for the disappearance of the underestimation in the case of long-tailed errors. First, while the initial short-cut estimate in each bootstrap sample still somewhat resembles a least squares estimate, the latter is less efficient for this kind of errors, yielding a higher variance in itself. Second, the eventual short-cut solutions are now in fact more similar to the actual LTS estimates. The result for the case of 20% outliers can be explained along the same lines. It should be noted however that the bootstrap cannot be expected here to approximate the MC result in a truly accurate way. Indeed, the MC estimate is obtained from samples that contain exactly 20% outliers, while in the bootstrap samples this proportion varies.

For the scale σ the short-cut in all cases overestimates the variance. However, the bias is not dramatic except in case of outliers. Intuitively this overestimation stems from the fact that the scale corresponds to the objective value of the LTS estimator. Therefore short-cut bootstrap estimates for σ are greater

Table 6
Coverage and length of 95% confidence intervals (B=1000); normal errors

$p = 5$	$n = 50$			$n = 200$		
	0 C	∞ C	$\widehat{\text{ASV}}$	0 C	∞ C	$\widehat{\text{ASV}}$
slope	89.3	96.1	95.6	92.9	96.8	95.0
	(0.614)	(0.886)	(1.031)	(0.298)	(0.459)	(0.520)
int	89.8	96.3	96.2	93.3	97.5	95.8
	(0.596)	(0.869)	(1.016)	(0.297)	(0.464)	(0.518)

than or equal to the FAST-LTS estimates. This explains the higher variance for the short-cut estimates of σ, since we found the bootstrap distributions for $\widehat{\sigma}$ to be generally skewed to the right.

The empirical asymptotic variance performs better than the short-cut bootstrap for normal errors. However in the other two situations, which are cases where the use of LTS is recommended, this asymptotic variance estimate is not accurate at all and much worse than the short-cut bootstrap.

Concerning confidence intervals for the parameter $\boldsymbol{\beta}$, we compared 95% percentile bootstrap intervals with intervals based on the asymptotic normality of LTS, using the empirical asymptotic variance. Tables 6, 7 and 8 list the percentage of the intervals in the simulation that contained the value 0 (the true value of the parameter), as well as the average length of the intervals. Here we only produce the results for $p = 5$, those for $p = 10$ being similar. We see that the coverage of the short-cut bootstrap intervals in all cases is higher than the nominal value, while their length is quite short relative to the variance of the LTS. These results follow from the fact that the intervals are usually not centered around the original LTS estimate, but rather are somewhat shifted towards what would be the LS estimate of the non-outlying observations. Note that one could argue that the intervals could be even more precise if they were based on the least squares estimates of the non-outlying observations in each bootstrap sample. However, the coverage for those intervals turns out to be

Table 7
Coverage and length of 95% confidence intervals (B=1000); t_3 errors

$p = 5$	$n = 50$		$n = 200$		$n = 500$	
	∞ C	$\widehat{\text{ASV}}$	∞ C	$\widehat{\text{ASV}}$	∞ C	$\widehat{\text{ASV}}$
slope	96.7	97.9	96.4	98.8	96.0	99.1
	(0.984)	(1.256)	(0.453)	(0.615)	(0.215)	(0.389)
int	96.9	98.3	97.0	99.1	97.1	99.5
	(0.949)	(1.240)	(0.450)	(0.613)	(0.283)	(0.388)

Table 8
Coverage and length of 95% confidence intervals (B=1000); 20% far outliers

$p = 5$	$n = 50$		$n = 200$		$n = 500$	
	∞ C	$\widehat{\text{ASV}}$	∞ C	$\widehat{\text{ASV}}$	∞ C	$\widehat{\text{ASV}}$
slope	96.6	100	96.2	99.9	96.6	99.9
	(0.809)	(1.605)	(0.385)	(0.743)	(0.241)	(0.468)
int	96.1	100	97.0	100	96.9	100.0
	(0.781)	(1.578)	(0.383)	(0.740)	(0.242)	(0.467)

somewhat too low. Moreover, with such a bootstrap method we would not be able to obtain a fair estimate of the variance of the LTS estimator.

We can conclude that the short-cut bootstrap method for LTS yields reasonable variance estimates, which are relatively more efficient when the data has longer tails. The confidence intervals based on the bootstrap are short and conservative. Note that we only considered one type of outliers in this simulation, but it is expected that the performance is similar for different kinds of outliers, as long as the LTS itself is capable of resisting this contamination. Also note that it is not our intent to give a theoretical justification for the short-cut bootstrap method. In fact, it is clear from the simulations that the method will probably give an asymptotic bias. Rather we intended to show that by using fast approximations in the resamples, the bootstrap method can become a practical tool for inference on the LTS.

5 Example

For an illustration of the use as well as the benefits of the short-cut bootstrap procedure for LTS, we consider the so-called *Pilot-Plant* data [2], consisting of 20 observations on 2 variables. The response variable is the acid content determined by titration, and the explanatory variable is the organic acid content determined by extraction and weighing. A scatter plot of the data is given in Figure 1a. The regression lines estimated by least squares and LTS respectively are superimposed on the plot. These lines practically coincide here. Obviously this dataset contains no outlying observations so actually there is no need for a robust estimator. In fact, even if there would be an outlier present, since we only have one explanatory variable we would be able to easily spot the outlier and remove it if desired. It should be clear that robust regression in general is more beneficial in case of multiple regression. However, we chose to illustrate our method on an example of simple regression for reasons of clarity.

Now suppose that two of the observations have been wrongly recorded, yielding

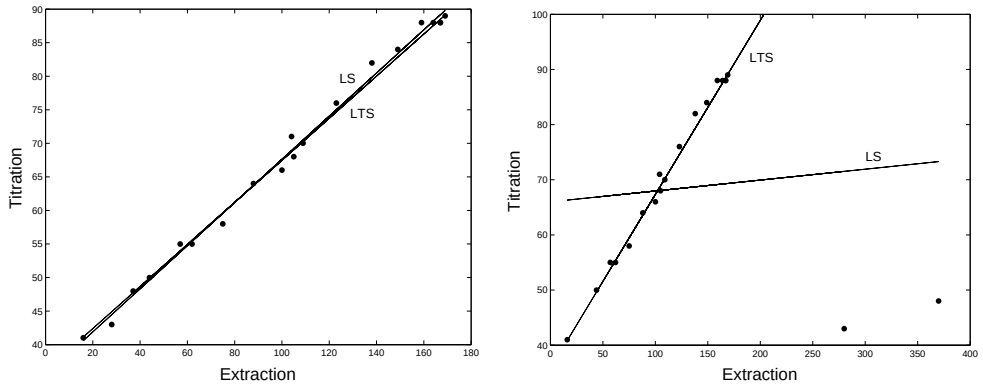

Fig. 1. Pilot-Plant data, regression fit (LS and LTS); (a) original data; (b) data with 2 observations shifted

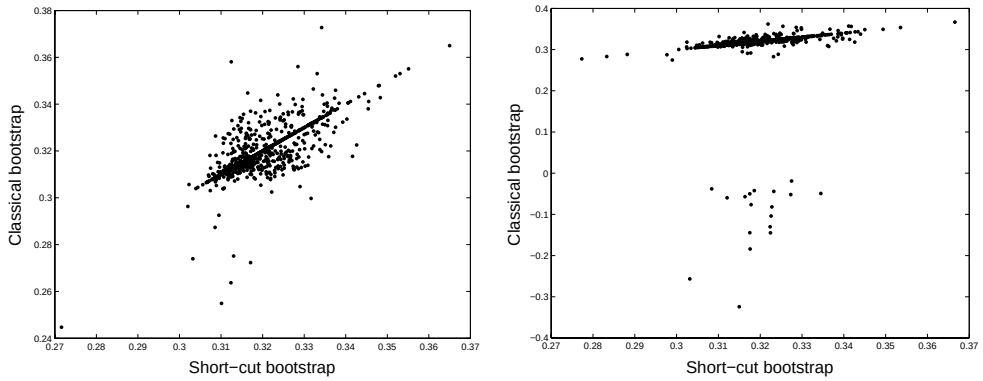

Fig. 2. Pilot-Plant data, LTS bootstrap estimates for the slope: classical versus short-cut estimates; (a) original data; (b) data with 2 observations shifted

the situation as depicted in Figure 1b. We now see that the least squares estimate is attracted to the leverage points, while LTS, with $h = 15$, was able to resist them. We applied the classical bootstrap as well as the short-cut bootstrap for LTS on both the original dataset and the contaminated dataset. For both bootstrap methods the same $B = 1000$ resamples were used. Figure 2 depicts the classical bootstrap estimates for the slope parameter versus the short-cut bootstrap estimates. The left panel corresponds to the original Pilot-Plant data, the right to the contaminated data. It can be seen that for the contaminated dataset the LTS estimate broke down in several bootstrap resamples, yielding e.g. a negative slope estimate. Note that, since $h = 15$, the LTS can cope with at most 5 recurrences of either one of the leverage points. This robustness problem is avoided when using the short-cut procedure.

Table 9 summarizes some inference results on the slope parameter for both datasets. The high contamination of several classical bootstrap estimates reveals itself in the fact that the lower 99% percentile confidence limit for the slope parameter has a negative value. The classical bootstrap estimate for the standard deviation is also affected. This clearly illustrates the non-robustness

Table 9
Pilot-Plant data: Bootstrap results for the slope

		Classical			Short-cut		
	$(\widehat{\boldsymbol{\beta}}_{LTS})_1$	$\widehat{\mathrm{SD}}$	99% Conf.Int.		$\widehat{\mathrm{SD}}$	99% Conf.Int.	
Original data	0.313	0.01017	[0.274	0.355]	0.00828	[0.304	0.348]
Contaminated	0.316	0.05776	[-0.144	0.356]	0.00767	[0.299	0.344]

of the classical bootstrap procedure, as opposed to the robustness of the LTS estimate itself, and of course as opposed to the robustness of the short-cut bootstrap procedure. Probably more important however than the robustness benefit is the fastness of the short-cut procedure. The computation time for the classical bootstrap (where 500 initial subsets were used for FAST-LTS) on the Pilot-Plant dataset was approximately 7 minutes, while the short-cut bootstrap took only about 5 seconds.

6 Conclusion

When it comes to inference methods for regression parameters using robust estimators, the bootstrap is a natural choice. However, the classical bootstrap often demands excessive computation time and has a problem of robustness. In this paper we proposed an alternative bootstrap method for the LTS estimator, which performs a 'short-cut' in each resampled dataset instead of the whole FAST-LTS algorithm. Through a simulation study, it is shown that this fast method generally yields accurate results and is more robust than the classical bootstrap. The method was also illustrated on an example.

Acknowledgements

We are grateful to Elke Maesen for contributing to the simulation study. We also thank both referees for their helpful comments and remarks.

References

[1] Croux, C. and Haesbroeck, G., 1999. Influence function and efficiency of the minimum covariance determinant scatter matrix estimator. J. Multivariate Anal., 71 161-190

[2] Daniel, C. and Wood, F.S., 1971. Fitting Equations to Data. Wiley, New York.

[3] Efron, B., 1979. Bootstrap methods: another look at the jackknife. Ann. Statist., 7 1-26.

[4] Pison, G., Van Aelst, S. and Willems, G., 2002. Small sample corrections for LTS and MCD. Metrika, 55 111-123.

[5] Rousseeuw, P.J., 1984. Least median of squares regression. J. Amer. Statist. Assoc., 79 871-880.

[6] Rousseeuw, P.J. and Leroy, A.M., 1987. Robust regression and outlier detection. Wiley, New York.

[7] Rousseeuw, P.J. and Van Driessen, K., Computing LTS regression for large data sets. Mimeo. (Dept. Mathematics, University of Antwerp, 1999).

[8] Salibian-Barrera, M. and Zamar, R., 2002. Bootstrapping robust estimates of regression. Ann. Statist., 30 556-582.

[9] Singh, K., 1998. Breakdown theory for bootstrap quantiles. Ann. Statist., 26 1719-1732.

[10] Stromberg, A.J., 1997. Robust covariance estimates based on resampling. J. Statist. Plann. Inference, 57 321-334.

[11] Van Aelst, S. and Willems, G., 2002. Robust bootstrap for S-estimators of multivariate regression. Statistics in Industry and Technology: Statistical Data Analysis, 201-212.