A Fast Algorithm for S-Regression Estimates

Matías SALIBIAN-BARRERA and Víctor J. YOHAI

Equivariant high-breakdown point regression estimates are computationally expensive, and the corresponding algorithms become unfeasible for moderately large number of regressors. One important advance to improve the computational speed of one such estimator is the fast-LTS algorithm. This article proposes an analogous algorithm for computing S-estimates. The new algorithm, that we call "fast-S", is also based on a "local improvement" step of the resampling initial candidates. This allows for a substantial reduction of the number of candidates required to obtain a good approximation to the optimal solution. We performed a simulation study which shows that S-estimators computed with the fast-S algorithm compare favorably to the LTS-estimators computed with the fast-LTS algorithm.

Key Words: AU: Please give 3–5 key words that do not appear in the title.

1. INTRODUCTION

Consider the linear model

$$y_i = \boldsymbol{\beta}_0' \, \mathbf{x}_i + u_i \,, \quad i = 1, \ldots, n \,, \tag{1.1}$$

where $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})' \in R^p$ and $\boldsymbol{\beta}_0 \in R^p$ are the vectors of explanatory variables and regression coefficients, respectively, and the u_i's are the errors. When the model includes an intercept we set $x_{i1} = 1$ for $1 \leq i \leq n$.

The most commonly used estimator for $\boldsymbol{\beta}_0$ in these models is the least squares (LS) estimator. It is well known that this estimator is very sensitive to the presence of atypical points in the sample. An observation $(y_i, \mathbf{x}_i)$ is an atypical point (or outlier) if it does not follow the above regression model. It is well known that a single outlier can have an unbounded effect on the LS-estimator. Estimators that are not highly sensitive to outliers are called robust.

One measure of robustness of an estimate is its breakdown point. Heuristically, the breakdown point is the minimum fraction of arbitrary outliers that can take the estimate

Matías Salibian-Barrera is Assistant Professor, 333-6356 Agricultural Road, University of British Columbia, Vancouver, BC, V6T 1Z2 (E-mail: *matias@stat.ubc.ca*). Víctor J. Yohai is Professor, Departamento de Matematica, Universidad de Buenos Aires, Ciudad Universitaria, Pabellon 1, 1428 Buenos Aires, Argentina.

Journal of Computational and Graphical Statistics, Volume 15, Number 2, Pages 1–14
DOI: 10.1198/106186006X113629

beyond any limit. Hampel (1971) introduced the breakdown point as an asymptotic concept, and Donoho and Huber (1983) gave the corresponding finite sample notion. The maximum possible asymptotic breakdown point of an equivariant regression estimate is 0.5, and the maximum possible finite sample breakdown point is $[n-p+1]/2n$ (Rousseeuw and Leroy 1987), where $[k]$ denotes the integer part of k.

A desirable property for regression estimates is that the estimate be equivariant with respect to affine, regression, and scale transformations. This means that when we apply one of these types of transformations to the data, the estimates will transform in the "natural" way. Several equivariant estimators for linear regression with asymptotic breakdown point 0.5 were proposed in the literature. We can cite the least median of squares (LMS) and the least trimmed mean squares (LTS) estimators introduced by Rousseeuw (1984), and the S-estimators introduced by Rousseeuw and Yohai (1984). Proof of the breakdown point of S-estimators can be found in Müller and Neykov (2003). The LTS-estimator and the S-estimator are asymptotically normal with rate of convergence $n^{1/2}$ and their asymptotic efficiencies under normal errors are 0.0713 and 0.287, respectively. Instead, the LMS-estimator converges to a nonnormal distribution with rate of convergence $n^{1/3}$ (see Kim and Pollard 1990). More efficient estimates with asymptotic breakdown point 0.5 are the MM-estimates (Yohai 1987), the τ-estimates (Yohai and Zamar 1988) and the constrained S-estimates proposed by Mendes and Tyler (1996).

Exact algorithms to compute the LMS-estimator have been proposed by Steele and Steiger (1986), Stromberg (1993), and Agulló (1997a,b). Hawkins (1994) proposed an approximate algorithm for the LTS-estimator, while Agulló (2001) described an exact algorithm for the same estimator. Unfortunately, these algorithms are feasible only when n and p are small. Rousseeuw (1984) proposed an approximate algorithm for the LMS- and the LTS-estimators based on minimizing the corresponding objective functions over a large set of N candidates. Each candidate is obtained by taking a subsample of size p from the data and finding the hyperplane that contains these points. Let $\epsilon_0(n,p)$ be the breakdown point of the estimate being computed. By taking a sufficiently large number of subsamples N, this approximating algorithm can guarantee that the resulting breakdown point is arbitrarily close to $\epsilon_0(n,p)$ with high probability. More specifically, let $\epsilon < \epsilon_0(n,p)$, $0 < \alpha < 1$ and take N satisfying

$$N \geq \frac{\log(\alpha)}{\log\left(1-(1-\epsilon)^p\right)} \approx \frac{-\log(\alpha)}{(1-\epsilon)^p}\,.$$

Then with probability $1-\alpha$ the breakdown point of the resulting algorithm is at least ϵ (see Rousseeuw and Leroy 1987). Note that this lower bound for N grows exponentially with the number of covariates p. For example, when $\epsilon = 0.20$ (20% of outliers) and $\alpha = 0.01$ (probability 99% of finding a clean subsample) the required number of subsamples is 398 for $p = 20$, over 322,000 for $p = 50$ and more than 22 billion for $p = 100$. Therefore, even if this algorithm has a larger range of application than the exact algorithms mentioned above, it also becomes unfeasible when p is large.

Rousseeuw and Van Driessen (2002) proposed a modification of the subsampling algorithm for the LTS-estimator, called fast-LTS, that considerably improves its performance.

The main idea used in this algorithm was first proposed to compute the minimum covariance determinant estimator (fast-MCD algorithm) by Rousseeuw and Van Driessen (1999). Given any initial value, they defined the so-called concentration step (C-step) that improves the objective function. This step is applied to the N candidates obtained by subsampling, and it brings each candidate closer to the solution of the optimization problem. If the C-step is applied a sufficient (finite) number of times, a local minimum of the objective function is obtained. The fast-LTS algorithm improves each resampling candidate using a fixed number k of C-steps. Rousseeuw and Van Driessen (2002) recommended to take $k = 2$. They also showed that the fast-LTS is much faster than the approximating algorithms for the LTS-estimator that do not use the C-step.

This article describes an algorithm for S-estimates similar to the fast-LTS of Rousseeuw and Van Driessen (2002) which borrows from Ruppert's SURREAL algorithm (Ruppert 1992). This algorithm, that we call fast-S, is based on modifying each candidate with a step that generally improves the S-optimality criterion, and thus allows us to reduce the number of subsamples required to obtain a desired breakdown point with high probability. It is important to note that, unlike Ruppert's algorithm that only performs local improvements to resampling candidates that decrease the objective function, we improve every resampling candidate. In our view, this is a major difference. The advantage of improving every candidate is the following. Candidates that are obtained from ill-conditioned subsamples that do not contain outliers may initially have a large value of the objective function. However, after applying the local improvement step they can become much closer to the optimum. Another difference between the fast-S and Ruppert's algorithm is that the latter checks whether the local improvements actually decrease the objective function, whereas we prove that this is always the case (see Lemma 1 in Section 2).

We can mention two reasons why we expect S-estimators to behave more robustly than the LTS-estimator: S-estimators have smaller asymptotic bias and smaller asymptotic variance in contamination neighborhoods. Comparison of the asymptotic biases and variances of the S-estimators and the LTS-estimator can be found in Section 2. Our Monte Carlo simulation (see Section 3) gives empirical evidence of the better robustness behavior of the S-estimator computed with the fast-S algorithm.

The rest of the article is organized as follows. Section 2 gives the basic definitions and defines the local improvement step for S-estimates. Section 3 presents the results of a Monte Carlo study comparing the S- and LTS-estimators using the fast algorithms and the classical algorithms only based on subsampling. Finally, Section 4 contains some concluding remarks.

2. FAST-LTS AND FAST-S

Consider the linear regression model (1.1). In general, a noncentered scale of the vector of residuals $\mathbf{r} = (r_1, \ldots, r_n)'$ is a function $s : R^n \to R_+$ that satisfies: (a) $s(\mathbf{r}) \geq 0$; (b) $s(a\,\mathbf{r}) = a\,s(\mathbf{r})$ for all $a \geq 0$; (c) $s(|r_1|, \ldots, |r_n|) = s(r_1, \ldots, r_n)$; and (d) $s(r_{\pi_1}, \ldots, r_{\pi_n}) = s(r_1, \ldots, r_n)$ where $(\pi_1, \ldots, \pi_n)$ is any permutation of $\{1, \ldots, n\}$.

We can interpret $s(\mathbf{r})$ as a measure of the size of the absolute value of the residuals.

For each $\boldsymbol{\beta} \in R^p$ let $\mathbf{r}(\boldsymbol{\beta}) = (r_1(\boldsymbol{\beta}), \ldots, r_n(\boldsymbol{\beta}))'$ be the vector of residuals where $r_i(\boldsymbol{\beta}) = y_i - \mathbf{x}_i'\boldsymbol{\beta}$ for $1 \leq i \leq n$. Many regression estimates can be thought of as minimizing $s(\mathbf{r}(\boldsymbol{\beta}))$ for different scale estimates. For example, for the LS-estimator the scale s is given by

$$s(r_1, \ldots, r_n) = \sqrt{\sum_{i=1}^{n} r_i^2 \Big/ n}\,.$$

We now consider two robust scale estimators and their associated regression estimates. The scale estimate that corresponds to the LTS-estimator (Rousseeuw 1984) is the α-trimmed scale defined as follows. If $\mathbf{r} = (r_1, \ldots, r_n)'$ are n real numbers, their α-trimmed scale for $0 < \alpha < 1/2$ is defined as

$$s_\alpha^2(\mathbf{r}) = \frac{1}{n - [n\,\alpha]} \sum_{i=1}^{n-[n\,\alpha]} r_{(i)}^2\,,$$

where $[k]$ denotes the integer part of k and $r_{(1)}^2 \leq \cdots \leq r_{(n)}^2$ are the ordered squares of the observations.

Huber (1964) defined M-scale estimates for $\mathbf{r} = (r_1, \ldots, r_n)'$ as

$$s_M(\mathbf{r}) = \inf\left\{ s > 0 \,:\, \frac{1}{n}\sum_{i=1}^{n} \rho\left(\frac{r_i}{s}\right) \leq b \right\}\,,$$

where $0 \leq b \leq 1$ and $\rho : R \to R_+$ satisfies

A1: ρ is even;

A2: $\rho(r)$ is nondecreasing in $|r|$;

A3: $\rho(0) = 0$.

It is easy to see that if ρ is continuous and $\#\{r_i = 0\} < n(1-b)$, then the M-scale $s_M(\mathbf{r})$ satisfies:

$$\frac{1}{n}\sum_{i=1}^{n} \rho\left(\frac{r_i}{s_M(\mathbf{r})}\right) = b\,,$$

and if $\#\{r_i = 0\} > n(1-b)$, then $s_M(\mathbf{r}) = 0$. To guarantee consistency when the data are normally distributed, the constant b is usually chosen to be $E_\Phi(\rho(u))$, where Φ denotes the standard normal distribution.

The regression estimates associated with M-scales are the S-estimators proposed by Rousseeuw and Yohai (1984). In particular, they satisfy

$$\hat{\boldsymbol{\beta}}_n = \arg\min_{\boldsymbol{\beta} \in R^p} \sum_{i=1}^{n} \rho\left(\frac{r_i(\boldsymbol{\beta})}{\hat{s}}\right)\,, \tag{2.1}$$

with $\hat{s} = s_M(\mathbf{r}(\hat{\boldsymbol{\beta}}_n))$. Differentiating (2.1) we obtain the estimating equations for S-estimators:

$$\sum_{i=1}^{n} \psi\left(\frac{r_i(\hat{\boldsymbol{\beta}}_n)}{\hat{s}}\right)\mathbf{x}_i = 0\,, \tag{2.2}$$

Table 1. Maximum Bias of Robust Estimates With 50% Breakdown Point for Different Amounts ϵ of Contamination

	ϵ			
Estimator	*0.05*	*0.10*	*0.15*	*0.20*
LTS	0.63	1.02	1.46	2.02
LMS	0.53	0.83	1.13	1.52
S	0.56	0.88	1.23	1.65

where $\psi = \rho'$ and

$$\frac{1}{n}\sum_{i=1}^{n}\rho\left(\frac{r_i(\hat{\boldsymbol{\beta}}_n)}{\hat{s}}\right) = b\,.$$

Both the LTS- and the S-estimators are asymptotically normal [for results on S-estimators see Rousseeuw and Yohai (1984), and Davies (1990); for the LTS-estimator, heuristic arguments can be found in Rousseeuw and Leroy (1987), Tableman (1994), and Hössjer (1994), and a rigorous proof for the location model in Yohai and Maronna (1976)]. When the errors in model (1.1) are normally distributed the efficiencies of the 50% breakdown point LTS- and S-estimators based on the bi-squared function (2.3) are 0.0713 and 0.287, respectively. The efficiency of the LTS-estimator was obtained applying Theorem 4 of Rousseeuw and Leroy (1987, p. 180) and the one of the S-estimator was taken from Table 19 of Rousseeuw and Leroy (1987, p. 142). The asymptotic maximum biases for ϵ-contamination neighborhoods of the normal multivariate model can be found in Table 1. The maximum biases of the 50% breakdown point S-estimator were obtained applying Theorem 3.1 of Martin, Yohai, and Zamar (1989); the maximum biases of the LTS-estimator were obtained using Theorems 1 and 2 and the Remark in Section 4 of Berrendero and Zamar (2001). We observe that S-estimators are more efficient at the normal model and have smaller bias than the LTS-estimator for all values of ϵ.

We now turn our attention to a procedure that increases the speed of the resampling algorithm for computing S-estimators. Given an estimate $\boldsymbol{\beta}^{(0)}$ of the regression coefficients, the local improvement step (that we call I-step) for S-estimators is defined as follows:

I-step(a) Compute $s_0 = s_M(\mathbf{r}(\boldsymbol{\beta}^{(0)}))$ and the weights $w_i = w(r_i((\boldsymbol{\beta}^{(0)})/s_0))$ where $w(u) = \psi(u)/u$.

I-step(b) Define $\beta^{(1)}$ as the weighted LS-estimator where the ith observation receives weight w_i, for $i = 1, \ldots, n$.

Note that this I-step is one step of the iterated re-weighted least squares algorithm to solve Equation (2.2) starting from $\boldsymbol{\beta}^{(0)}$.

The following Lemma shows that the I-step decreases the scale s of the residuals and hence it improves the candidate $\boldsymbol{\beta}^{(0)}$. We need these additional regularity conditions on the loss function ρ:

A4: ρ is differentiable with continuous derivative $\psi = \rho'$; and

A5: $\psi(u)/u$ is decreasing in $u > 0$.

Lemma 1. *Suppose that ρ satisfies A1–A5 and that $\boldsymbol{\beta}^{(1)}$ was obtained by applying the I-step described above to the vector $\boldsymbol{\beta}^{(0)}$. Then $s_M(\mathbf{r}(\boldsymbol{\beta}^{(1)})) \leq s_M(\mathbf{r}(\boldsymbol{\beta}^{(0)}))$.*

Proof: By Remark 1 to Lemma 8.3 in Huber (1981, p. 186) (this lemma was originally proved by Dutter 1975) we have that

$$\frac{1}{n}\sum_{i=1}^{n}\rho\left(\frac{r_i(\boldsymbol{\beta}^{(1)})}{s_0}\right) \leq \frac{1}{n}\sum_{i=1}^{n}\rho\left(\frac{r_i(\boldsymbol{\beta}^{(0)})}{s_0}\right) = b\,.$$

By A1 and A2 we have that $s_M(\mathbf{r}(\widehat{\boldsymbol{\beta}}_1)) \leq s_0 = s_M(\mathbf{r}(\widehat{\boldsymbol{\beta}}_0))$. □

According to this lemma it is natural to define a fast-S algorithm similarly to the fast-LTS but using the I-step described above.

The following Lemma shows that if $\left\{\boldsymbol{\beta}^{(n)},\, n \geq 1\right\}$ is the sequence obtained by applying the I-step iteratively starting from an arbitrary $\boldsymbol{\beta}^{(0)}$, then any accumulation point is a local minimum of $s(\mathbf{r}(\boldsymbol{\beta}))$.

Lemma 2. *Assume that ρ satisfies A1–A5. Then, for any starting point $\boldsymbol{\beta}^{(0)}$, any accumulation point of the sequence $\boldsymbol{\beta}^{(n)}$ obtained by applying the I-step iteratively is a local minimum of $s(\mathbf{r}(\boldsymbol{\beta}))$.*

Proof: According to Lemma 1, the sequence $s^{(k)} = s(\mathbf{r}(\beta^{(k)}))$ is decreasing and nonnegative, therefore $s^{(k)} \to s_0$ for some limit s_0. Consider $\boldsymbol{\beta}^*$ any accumulation point of $\left\{\boldsymbol{\beta}^{(k)}\right\}_{k\geq 1}$. We are going to prove that $s(\mathbf{r}(\beta^*)) = s_0$ and that if $\boldsymbol{\beta}^{**}$ is the result of applying one I-step to $\boldsymbol{\beta}^*$, then $s(\mathbf{r}(\beta^{**})) = s_0$.

Take any subsequence $\left\{\boldsymbol{\beta}^{(k_i)}\right\}_{i\geq 1}$ converging to $\boldsymbol{\beta}^*$. Call $C : R^p \to R^p$ such that $C(\boldsymbol{\beta})$ is the result of applying one I-step to $\boldsymbol{\beta}$. By assumption, this transformation is continuous. Note that $\boldsymbol{\beta} \to s(\mathbf{r}(\boldsymbol{\beta}))$ is also a continuous function. Then, $\boldsymbol{\beta}^{(k_i+1)} = C(\boldsymbol{\beta}^{(k_i)})$ and $\lim_{i\to\infty}\boldsymbol{\beta}^{(k_i+1)} = C(\boldsymbol{\beta}^*)$. Therefore, $\lim_{i\to\infty}\boldsymbol{\beta}^{(k_i+1)} = \boldsymbol{\beta}^{**}$. Then, $s(\mathbf{r}(\boldsymbol{\beta}^*)) = \lim_{i\to\infty} s(\mathbf{r}(\boldsymbol{\beta}^{(k_i)})) = s_0$ and $s(\mathbf{r}(\boldsymbol{\beta}^{**})) = \lim_{i\to\infty} s(\mathbf{r}(\boldsymbol{\beta}^{(k_i+1)})) = s_0$. Thus

$$\frac{1}{n}\sum_{i=1}^{n}\rho\left(\frac{r_i(\boldsymbol{\beta}^*)}{s_0}\right) = b = \frac{1}{n}\sum_{i=1}^{n}\rho\left(\frac{r_i(\boldsymbol{\beta}^{**})}{s_0}\right)\,.$$

Therefore,

$$\sum_{i=1}^{n}\rho\left(\frac{r_i(\boldsymbol{\beta}^*)}{s_0}\right) = \sum_{i=1}^{n}\rho\left(\frac{r_i(\boldsymbol{\beta}^{**})}{s_0}\right)\,.$$

By Remark 1 to Lemma 8.3 of Huber (1981), we conclude that $\boldsymbol{\beta}^*$ is a local minimum of

$$f(\boldsymbol{\beta}) = \sum_{i=1}^{n}\rho\left(\frac{r_i(\boldsymbol{\beta})}{s_0}\right)\,.$$

We will now prove that it is also a local minimum of $s(\mathbf{r}(\boldsymbol{\beta}))$. Take a neighborhood $N(\boldsymbol{\beta}^*)$ around $\boldsymbol{\beta}^*$ such that $f(\boldsymbol{\beta}) \geq f(\boldsymbol{\beta}^*)$ for all $\boldsymbol{\beta} \in N(\boldsymbol{\beta}^*)$. Then given $\boldsymbol{\beta} \in N(\boldsymbol{\beta}^*)$ we will have

$$\frac{1}{n}\sum_{i=1}^{n}\rho\left(\frac{r_i(\boldsymbol{\beta})}{s_0}\right) \geq b\,,$$

and this implies that $s(\mathbf{r}(\boldsymbol{\beta})) \geq s_0 = s(\mathbf{r}(\boldsymbol{\beta}^*))$ proving the lemma. □

A simple first version of the fast-S algorithm, which is similar to the fast-LTS proposed by Rousseeuw and Van Driessen (2002), is as follows:

Step 1. Draw N random subsamples of size p and let $\boldsymbol{\beta}_j$, $j = 1, \ldots, N$, be the coefficients of each of the hyperplanes determined by them.

Step 2. Improve each of these candidates applying k I-steps as described above. Let $\boldsymbol{\beta}_j^C$, $j = 1, \ldots, N$, be the resulting improved candidates.

Step 3. For each $\boldsymbol{\beta}_j^C$ compute the M-scale $s_j = s(\mathbf{r}(\boldsymbol{\beta}_j^C))$ and keep the improved candidates with the best t scales ($1 \leq t \leq N$). Call ($\boldsymbol{\beta}_j^B$,s_j^B), $j = 1, \ldots, t$, these best improved candidates and their corresponding scales.

Step 4. Apply the I-step to each $\boldsymbol{\beta}_j^B$, $j = 1, \ldots, t$, until convergence, obtaining $(\boldsymbol{\beta}_j^F, s_j^F)$, $j = 1, \ldots, t$, where $s_j^F = s(\mathbf{r}(\boldsymbol{\beta}_j^F))$.

Step 5. The final estimate is the $\boldsymbol{\beta}_j^F$ associated with the smallest s_j^F.

Since computing the scales $s(\mathbf{r}(\boldsymbol{\beta}))$ is rather computationally costly, we will slightly modify Steps 2 and 3 of the above algorithm. In Step 2 we modify the I-step as follows: given a candidate $\boldsymbol{\beta}_j$ the improved vector $\boldsymbol{\beta}_j^C$ is computed as before but we replace $s(\mathbf{r}(\boldsymbol{\beta}_j))$ by an approximated value obtained at the rth step of any iterative algorithm used in the computation of $s(\mathbf{r}(\boldsymbol{\beta}_j))$, starting from $s = \text{MAD}(\mathbf{r}(\boldsymbol{\beta}_j))$. Even though we cannot prove a result analogous to Lemma 1 for the modified I-step, it worked extremely well in our simulations. In all the cases we considered in our Monte Carlo study of Section 3, the approximated Steps 2 and 3 with $r = 1$ resulted in improved scales. The average improvement ranged from 40% for $p = 2$ to 80% for $p = 30$.

In Step 3 we do not need to compute all the scales $s_j = s(\mathbf{r}(\boldsymbol{\beta}_j^C))$, $1 \leq j \leq N$. We can be more efficient by adapting an idea first proposed by Yohai and Zamar (1991). We proceed as follows:

(a) Compute $s_m = s(\mathbf{r}(\boldsymbol{\beta}_m^C))$, $m = 1, \ldots, t$. Let $A_t = \max_{1 \leq m \leq t} s_m$ and $I_t = \{1, \ldots, t\}$.

(b) Suppose that we have already examined r candidates, where $r \geq t$. Denote by I_r the set of indices of the t best scale estimates found after examining these r candidates, and let A_r be the maximum of these scales. Now we check if $\boldsymbol{\beta}_{r+1}^C$ will be included in I_{r+1}. This will only happen if $s(\mathbf{r}(\boldsymbol{\beta}_{r+1}^C)) < A_r$ and this is equivalent to

$$\sum_{i=1}^{n} \rho\left(\frac{r_i(\boldsymbol{\beta}_{r+1}^C)}{A_r}\right) < b\,. \tag{2.3}$$

Then, we compute $s(\mathbf{r}(\boldsymbol{\beta}_{r+1}^C))$ only if (2.3) holds, and we correspondingly update I_r and A_r to obtain I_{r+1} and A_{r+1}. If (2.3) does not hold, let $I_{r+1} = I_r$ and $A_{r+1} = A_r$.

This change allows the computation of the best t candidates with a significantly smaller number of scale calculations.

Computer code for R and S-Plus implementing this algorithm is available online at *http://hajek.stat.ubc.ca/~matias/soft.html*

2.1 THE CASE OF LARGE *n*

When the number of observations n is large the performance of the fast-S algorithm may still be too slow. One way to overcome this problem is to follow the suggestion by Rousseeuw and Van Driessen (2002) for the LTS-estimator, which consists of drawing a small random subsample (of 2,000 observations, say) which is randomly divided into smaller disjoint subsets. This strategy of dividing a large sample into smaller subsamples was first proposed by Woodruff and Rocke (1994) and Rocke and Woodruff (1996). Their proposal is slightly different because they propose to divide the whole dataset in smaller subsamples.

More specifically, the algorithm of Rousseeuw and Van Driessen (2002) proposed that the random subsample be subsequently partitioned into smaller disjoint subsets. We randomly divide our subsample of size 2,000 into four disjoint subsets of 500 observations each. The idea is that each of these four smaller sets will be representative of the whole dataset. Random subsampling candidates are found in each of these four segments and improved using the I-step as before. The best t candidates of each partition are then improved (via I-steps) over the larger subsample of size $n = 2{,}000$. Finally, the best t improved candidates are iterated until convergence using the whole dataset and the solution with the smallest scale estimator is reported as the S-estimator.

A detailed description of the algorithm is as follows:

(a) If $n > 2{,}000$, take a random sample E, of $n = 2{,}000$ observations and split it into four disjoint subsets $E_1, \ldots, E_4$ of $n = 500$ points each.
(b) For each subset E_i, $1 \leq i \leq 4$, apply Steps 1 to 3 of the fast-S algorithm described above obtaining the best t improved candidates. At the end of this step we will have a total of four t improved candidates.
(c) Apply the I-step k times to each of the four t candidates over the whole $n = 2{,}000$ subsample E and record the best t improved candidates.
(d) Apply the I-step until convergence using the whole dataset to each of the t improved candidates found in (c) and compute the associated scale estimates.
(e) Among the t final solutions found in (d), choose the one with the smallest associated scale as your final estimate.

3. MONTE CARLO STUDY

We compared the performance of the fast-S and fast-LTS algorithms using a Monte Carlo study. The S-estimate was based on a ρ function in the bi-square family with score functions

$$\rho'_c(u) = \begin{cases} u\left(1 - \left(\frac{u}{c}\right)^2\right)^2 & \text{if} \quad |u| \leq c \\ 0 & \text{if} \quad |u| > c\,. \end{cases} \tag{3.1}$$

To obtain an S-estimate with breakdown point 0.5 we chose $c = 1.547$ and $b = 0.5$ (see Rousseeuw and Leroy 1987, tab. 19, p. 142).

Because we considered a model with intercept, we have $\mathbf{x} = (1, \mathbf{z}')'$ in model (1.1). In our samples a proportion $(1 - \varepsilon)$ of the data $(\mathbf{z}', y)'$ followed a multivariate normal distribution. Because of the equivariance of the estimators, without loss of generality we can take the mean vector equal to 0 and the covariance matrix the identity. Note that these

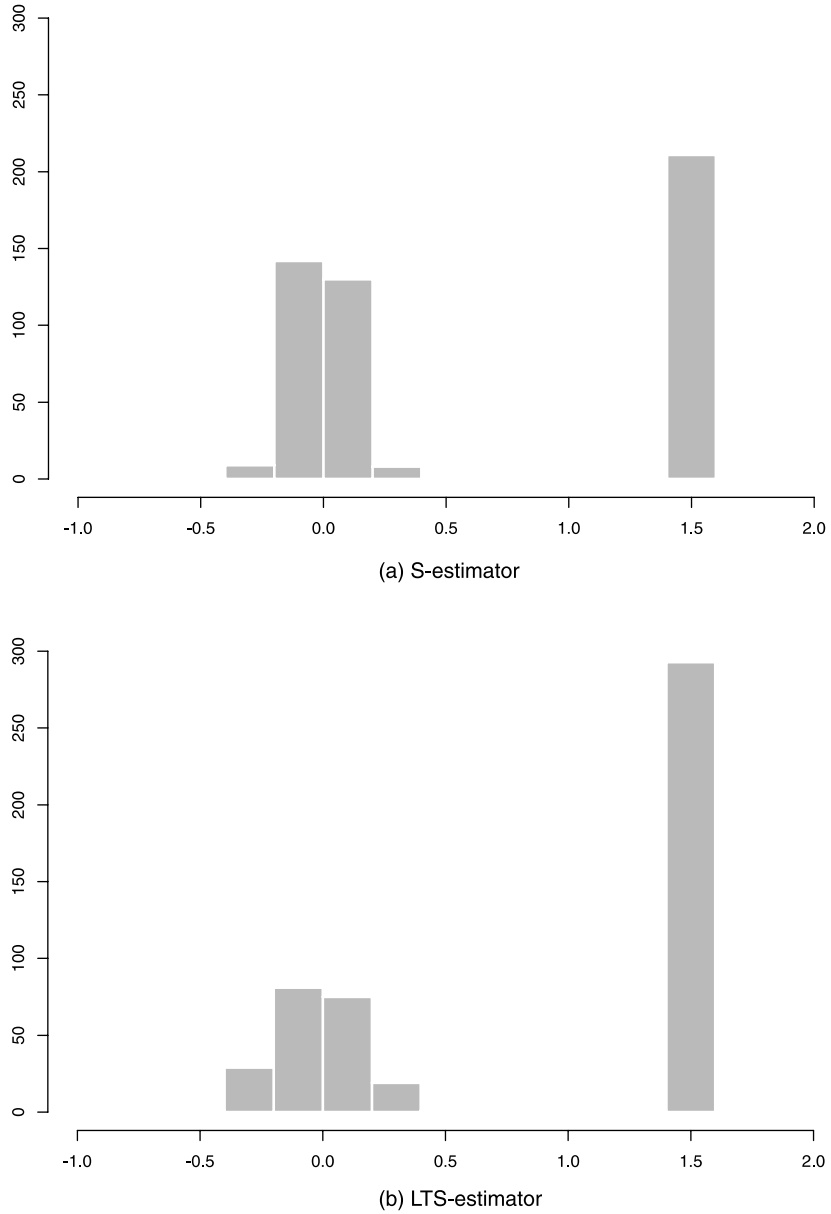

Figure 1. Histogram of 500 estimators $\hat{\boldsymbol{\beta}}_2$ for a sample with n = 400 observations with p = 36 and 10% of outliers located at $(\mathbf{x}_0', y_0)' = (1, 100, 0, \ldots, 0, 150)'$. The true slope is $\boldsymbol{\beta}_2 = 0$.

Table 2. Percentage of Samples Where Convergence Occurred to the Wrong Local Minimum. *N* is the number of subsamples. $n = 400$, $p = 36$, and 10% of outliers located at $(\mathbf{x}_0', y_0)' = (1, 100, 0, \ldots, 0, \text{slope} \times 100)'$

		Slope											
EST	*N*	*0.9*	*1.0*	*1.1*	*1.2*	*1.3*	*1.4*	*1.5*	*1.6*	*1.7*	*1.8*	*1.9*	*2.0*
Ruppert	1100	83	75	69	66	65	63	62	61	59	57	56	55
S0	2400	100	99	94	84	69	55	42	30	22	15	10	6
S1	540	89	58	27	7	1	0	0	0	0	0	0	0
LTS0	5200	100	99	97	91	82	71	59	46	35	23	14	7
LTS1	1400	100	97	85	64	39	21	7	3	1	0	0	0
FAST-LTS	900	100	100	96	86	65	42	22	9	4	1	0	0

observations follow model (1.1) with $\boldsymbol{\beta} = 0$ and iid errors u_i with a standard normal distribution. The remainder observations were high-leverage outliers with $y = M$ and $\mathbf{z} = (100, 0, \ldots, 0)'$. By the equivariance property of both estimators the effect of these outliers is the same as that of any fixed vector $(z_1, z_2, \ldots, z_{p-1}, M)$ with $(\sum_{i=1}^{p-1} z_j^2)^{1/2} = 100$.

Under high leverage contaminations the objective functions of the S- and the LTS-estimators typically have two distinct types of local minima: one close to the true value of the regression parameter, and another one close to the slope of the outliers. This is illustrated in Figure 1 where we plot the histogram of $\hat{\boldsymbol{\beta}}_2$ for the S- and LTS-estimators (without any improvement step) for 500 random samples of size $n = 400$ and $p = 36$. The samples were contaminated with 10% of high-leverage outliers at $(\mathbf{x}_0', y_0)' = (1, 100, \ldots, 0, 150)'$. Note that $\hat{\boldsymbol{\beta}}_2$ concentrates around zero (the true value of $\boldsymbol{\beta}_2$) and 1.5 (the slope of the contamination).

In our Monte Carlo experiment we used two measures of performance to compare the fast-S and fast-LTS algorithms: (a) the percentage of samples for which each algorithm converged to a wrong local minimum (a local minimum close to the slope of the outliers); and (b) the mean square errors.

The total computing time depends of the number N of initial candidates used in these algorithms. To make a fair comparison we set the value of N for each estimator in such a way that the computing times were approximately equal for each combination of sample size n, number of covariates p and number of improvement steps.

To illustrate the behavior of these algorithms for a range of different values of M we first considered samples of size $n = 400$ and $p = 36$ with $M = m100$, where the contamination slope m ranges between 0.9 and 2. The proportion of outliers ε was set to 0.10 and the number of replications was 500. We simulated the fast-S algorithm with 1 I-step (denoted S1) and the fast-LTS with 1 and 2 C-steps (denoted LTS1 and FAST-LTS, respectively) We also included the re-sampling algorithm for both estimates without improvement steps (denoted S0 and LTS0 for the S- and LTS-estimators respectively). Tables 2 and 3 contain the percentage of convergence to a wrong local minimum and the MSEs, respectively. From Table 2 we see that the fast-S with $k = 1$ has the lowest proportion of convergences to the wrong local minimum. From Table 3 we also see that the fast-S estimator with $k = 1$ has the smallest MSE for contaminations with slope between 0.9 and 1.6. Outside this range the MSE of the fast-S with $k = 1$ was comparable with that of the fast-LTS with $k = 1$ or $k = 2$.

Table 3. Mean Squared Errors. N is the number of subsamples. $n = 400$, $p = 36$, and 10% of outliers located at $(\mathbf{x}_0', y_0)' = (1, 100, 0, \ldots, 0, \text{slope} \times 100)'$

.

		Slope											
EST	*N*	*0.9*	*1.0*	*1.1*	*1.2*	*1.3*	*1.4*	*1.5*	*1.6*	*1.7*	*1.8*	*1.9*	*2.0*
Ruppert	1100	1.22	1.24	1.27	1.32	1.37	1.42	1.49	1.53	1.58	1.62	1.66	1.70
S0	2400	1.36	1.46	1.52	1.51	1.44	1.34	1.24	1.10	1.01	0.92	0.85	0.78
S1	540	1.26	1.10	0.89	0.73	0.68	0.67	0.67	0.67	0.67	0.67	0.67	0.67
LTS0	5200	1.69	1.79	1.89	1.93	1.94	1.87	1.78	1.65	1.51	1.33	1.17	1.04
LTS1	1400	1.24	1.33	1.32	1.22	1.02	0.87	0.73	0.69	0.66	0.66	0.65	0.65
FAST-LTS	900	1.25	1.35	1.42	1.43	1.31	1.11	0.92	0.77	0.71	0.67	0.66	0.66

The main conclusion from these tables is that for contaminations with slopes between 1.0 and 1.6 the fast-S estimator performs noticeably better than the fast-LTS in both measures. When the slopes are outside this range, their MSEs are comparable, but the fast-S converges more often to a local minimum closer to the true value of the parameters. There does not seem to be much difference in performance for the fast-LTS with 1 or 2 C-steps. However, for both the fast-S and fast-LTS there is an important improvement in both performance measures when at least one I-step or one C-step is used, respectively.

It is interesting to note that using one I-step for the S-estimator noticeably decreased the percentage of samples where convergence occurs to a wrong local minimum compared to the case when no I-step is used. In particular, when the slope of the contamination is between 0.9 and 1.6 the improvement in the fast-S is much higher than that in the fast-LTS with one C-step. Moreover, in some cases the LTS-estimator with two C-steps appears to perform worse than that with one C-step. This can be attributed to the fact that, to keep the

Table 4. Samples With 10% of Outliers Located at $(\mathbf{x}_0', y_0)' = (1, 100, 0, \ldots, 0, 100)'$. "CPU time" is in seconds of a P4 3.0GHz chip, "N" is the number of subsamples, "%wrong" is the percentage of samples where the algorithm converged to a wrong local minimum and "MSE" is the mean squared error.

		Average	*FAST S*			*FAST LTS*		
n	*p*	*CPU time*	*N*	*%wrong*	*MSE*	*N*	*%wrong*	*MSE*
100	2	0.06	500	33	0.46	500	52	0.76
	3	0.06	500	35	0.54	500	57	0.95
	5	0.08	500	48	0.87	500	66	1.29
500	5	0.34	500	14	0.18	550	63	0.79
	10	0.58	500	16	0.25	550	67	0.92
	20	1.52	500	26	0.50	600	82	1.26
1000	5	0.66	500	5	0.07	1500	60	0.70
	10	1.13	500	6	0.10	1500	70	0.87
	20	2.86	500	10	0.18	1600	84	1.17
5000	5	1.43	500	2	0.02	1900	71	0.76
	10	2.26	500	2	0.03	1050	78	0.88
	20	5.71	500	7	0.11	2900	84	1.07
10000	5	2.06	500	2	0.02	4300	78	0.82
	10	3.59	500	1	0.02	4800	87	0.94
	20	10.2	500	8	0.10	6500	93	1.11

Table 5. Samples With 20% of Outliers Located at $(\mathbf{x}_0', y_0)' = (1, 100, 0, \ldots, 0, 220)'$. "*N*" is the number of subsamples, "%wrong" is the percentage of samples where the algorithm converged to a wrong local minimum and "MSE" is the mean squared error.

		FAST S			*FAST LTS*		
n	*p*	*N*	*%wrong*	*MSE*	*N*	*%wrong*	*MSE*
100	2	500	11	0.679	500	42	2.547
	3	500	16	1.007	500	53	3.523
	5	500	27	2.008	500	69	5.150
500	5	500	0	0.038	550	25	1.450
	10	500	0	0.065	550	33	2.007
	20	500	9	0.721	600	74	4.683
1000	5	500	0	0.015	1500	12	0.686
	10	500	0	0.029	1500	20	1.184
	20	500	1	0.138	1600	52	3.174
5000	5	500	0	0.003	1900	2	0.089
	10	500	0	0.006	1050	1	0.073
	20	500	1	0.072	2900	1	0.105
10000	5	500	0	0.001	4300	9	0.439
	10	500	0	0.003	4800	10	0.522
	20	500	0	0.005	6500	31	1.622

computing time comparable, the fast-LTS with two C-steps was computed with a smaller number of initial candidates N.

We also ran the case $m = 1$ with $N = 540$ candidates for both estimators. In this case, the performance of the fast-LTS is comparable to those in Tables 2 and 3, and the required computing time is 2.7 times smaller than the fast-S. Nevertheless, the results of Table 2, where the values of N were chosen to match computing times, show that the advantage in performance of the fast-S cannot be overcome increasing the number of subsamples for the fast-LTS.

To make a more exhaustive comparison of these algorithms for different values of n, p, and ε we ran a Monte Carlo study similar to the previous one, but considering only one value of the contamination slope m, with one I-step for the fast-S and two C-steps for the fast-LTS. We used $\varepsilon = 0.10$ and $\varepsilon = 0.20$. For $\varepsilon = 0.10$ we set $m = 1$. This value was chosen based on Tables 2 and 3. Note that larger values of m correspond to outliers that are easier to identify by the S-estimator, whereas outliers with $m = 0.90$ are practically impossible to detect by both the S- and LTS-estimators. For similar reasons, based on an exploratory study, for $\varepsilon = 0.20$ we chose $m = 2.2$. Tables 4 and 5 show the results of these simulations. As in Tables 2 and 3, we observe that the fast-S has the best overall performance.

The fast-S and Ruppert's algorithms were implemented in R while the fast-LTS was computed using the function `ltsReg` in library `rrcov` for `R`. However, to be able to modify the number of C-steps, in Tables 2 and 3 we used the authors' implementation of the fast-LTS algorithm. Numerical experiments showed that both implementations of the fast-LTS algorithm produce very similar results.

4. CONCLUSION

We have proposed an algorithm to compute S-estimators of regression which can be considered an improvement over both Rousseeuw's approximated algorithm based on random subsamples and Ruppert's SURREAL algorithm. The basic idea of this procedure is, as in the fast-LTS of Rousseeuw and Van Driessen (2002), to apply a local improvement step to each resampling candidate. We compare our fast-S procedure with the fast-LTS algorithm and find that the fast-S gives better results. The reason for this difference in performance can be attributed to the better asymptotic efficiency and bias of S-estimators.

Even though the fast-S algorithm is slower than the fast-LTS, it can still be applied to very large datasets with reasonable computing times. For example, for $n = 10{,}000$ observations with $p = 45$ variables and 10% of outliers, the fast-S can be computed in approximately 44 seconds on a PC with a P4 3GHz CPU.

[Received October 2004. Revised September 2005.]

REFERENCES

Agulló, J. (1997a), "Computación de Estimadores Con Alto Punto de Ruptura," unpublished PhD Thesis, Universidad de Alicante.

——— (1997b), "Exact Algorithms to Compute the Least Median of Squares Estimate in Multiple Linear Regression," in *L_1-Statistical Procedures and Related Topics*, ed. Dodge, Y., vol. 31 of *IMS Lecture Notes, Monograph Series*, Hayward, CA: IMS, pp. 133–146.

——— (2001), "New Algorithms for Computing the Least Trimmed Squares Regression Estimator," *Computational Statistics and Data Analysis*, 36, 425–439.

Berrendero, J. R., and Zamar, R. (2001), "Maximum Bias Curves for Robust Regression with Non-elliptical Regressors," *The Annals of Statistics*, 29, 224–251

Davies, L. (1990), "The Asymptotics of S-estimators in the Linear Regression Model," *The Annals of Statistics*, 18, 1651–1675.

Donoho, D. L., and Huber, P. J. (1983), "The Notion of Breakdown-Point," in *A Festschrift for Erich L. Lehmann*, eds. P. J. Bickel, K. A. Doksum, and J. L Hodges, Jr., Belmont, CA: Wadsworth, pp. 157–184.

Dutter, R. (1975), "Robust Regression: Different Approaches to Numerical Solutions and Aalgorithms," Res. Rep. no. 6, Fachgruppe für Statistik, Eidgen, Technische Hochschule, Zurich.

Hampel, F. R. (1971), "A General Qualitative Definition of Robustness," *The Annals of Mathematical Statistics*, 42, 1887–1896.

Hawkins, D. M. (1994), "The Feasible Solution Algorithm for Least Trimmed Squares Regression," *Computational Statistics and Data Analysis*, 17, 185–196.

Hössjer, O. (1994), "Rank-Based Estimates in the Linear Model with High-Breakdown Point," *Journal of the American Statistical Association*, 89, 149–158.

Huber, P. J. (1964), "Robust Estimation of a Location Parameter," *Annals of Mathematical Statistics*, 35, 73–101.

——— (1981), *Robust Statistics*, New York: Wiley.

Kim, J., and Pollard, D. (1990), "Cube Root Asymptotics," *The Annals of Statistics*, 18, 191–219.

Martin, R. D., Yohai, V. J., and Zamar, R. (1989), "Min-max Bias Robust Regression," *The Annals of Statistics*, 17, 1608–1630.

Mendes, B., and Tyler, D. E. (1996), "Constrained M-estimation for Regression," in *Robust statistics, Data analysis, and Computer Intensive Methods*, ed. H. Rieder, Lecture Notes in Statistics, 109, New York: Springer, pp. 299–320.

Müller, C., and Neykov, N. M. (2003), "Breakdown Points of the Trimmed Likelihood and Related Estimators in Generalized Linear Models," *Journal of Statistical Planning and Inference*, 116, 503–519.

Rocke, D. M., and Woodruff, D. L. (1996), "Identification of Outliers in Multivariate Data," *Journal of the American Statistical Association*, 91, 1047–1061.

Rousseeuw, P. J. (1984), "Least Median of Squares Regression," *Journal of the American Statistical Association*, 79, 871–880.

Rousseeuw, P. J., and Leroy, A. M. (1987), *Robust Regression and Outlier Detection*, New York: Wiley.

Rousseeuw, P. J., and Van Driessen, K. (1999), "A Fast Algorithm for the Minimum Covariance Determinant Estimator," *Technometrics*, 41, 212–223.

——— (2002), "Computing LTS Regression for Large Datasets," *Estadistica*, 54, 163–190.

Rousseeuw, P. J., and Yohai, V. J. (1984), "Robust Regression by Means of S-Estimators," in *Robust and Nonlinear Time Series*, eds. J. Franke, W. Hardle, and D. Martin, *Lecture Notes in Statistics*, 26, Berlin: Springer-Verlag, pp. 256–272.

Ruppert, D. (1992), "Computing S-estimators for Regression and Multivariate Location/Dispersion," *Journal of Computational and Graphical Statistics*, 1, 253–270.

Steele, J. M., and Steiger, W. L. (1986), "Algorithms and Complexity for Least Median of Squares Regression," *Discrete Applied Mathematics. Combinatorial Algorithms, Optimization and Computer Science*, 14, 93–100.

Stromberg, A. J. (1993), "Computing the Exact Least Median of Squares Estimate and Stability Diagnostics in Multiple Linear Regression," *SIAM Journal of Scientific Computing*, 14, 1289–1299.

Tableman, M. (1994), "The Influence Functions for the Least Trimmed Squares and the Least Trimmed Absolute Deviations Estimators," *Statistics & Probability Letters*, 19, 329–337.

Woodruff, D. L., and Rocke, D. M. (1994), "Computable Robust Estimation of Multivariate Location and Shape in High Dimension Using Compound Estimators," *Journal of the American Statistical Association*, 89, 888–896.

Yohai, V. J. (1987), "High Breakdown Point and High Efficiency Robust Estimates for Regression," *The Annals of Statistics*, 15, 642–656.

Yohai, V. J., and Maronna, R. (1976), "Location Estimators Based on Linear Combinations of Modified Order Statistics," *Communications in Statistics, Part A—Theory and Methods*, 5, 481–486.

Yohai, V. J., and Zamar, R. (1988), "High Breakdown Point Estimates of Regression by Means of the Minimization of an Efficient Scale," *Journal of the American Statistical Association*, 83, 406–413.

——— (1991), Discussion of "Least Median of Squares Estimation in Power Systems," by Mili, L., Phaniraj, V., and Rousseeuw, P. J., *IEEE Transactions on Power Systems*, 6, 520.