Computational Statistics and Data Analysis 56 (2012) 2501–2512

Contents lists available at SciVerse ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Benchmark testing of algorithms for very robust regression: FS, LMS and LTS

Francesca Torti [a], Domenico Perrotta [b], Anthony C. Atkinson [c,*], Marco Riani [d]

[a] Dipartimento di Economia, Università di Parma, Italy
[b] European Commission, Joint Research Centre, Ispra, Italy
[c] The London School of Economics, WC2A 2AE London, United Kingdom
[d] Dipartimento di Economia, Università di Parma, Italy

ARTICLE INFO

Article history:
Received 10 August 2011
Received in revised form 2 February 2012
Accepted 3 February 2012
Available online 15 February 2012

Keywords:
Combinatorial search
Concentration step
Forward search
Least median of squares
Least trimmed squares
Logistic plots of power
Masking
Outlier detection

ABSTRACT

The methods of very robust regression resist up to 50% of outliers. The algorithms for very robust regression rely on selecting numerous subsamples of the data. New algorithms for LMS and LTS estimators that have increased computational efficiency due to improved combinatorial sampling are proposed. These and other publicly available algorithms are compared for outlier detection. Timings and estimator quality are also considered. An algorithm using the forward search (FS) has the best properties for both size and power of the outlier tests.

1. Introduction

Multiple regression is one of the main tools of applied statistics. It has however long been appreciated that ordinary least squares as a method of fitting regression models is exceptionally susceptible to the presence of outliers. Instead, very robust methods, that asymptotically resist 50% of outliers, are to be preferred.

Several algorithms have been proposed for very robust regression. All algorithms estimate the parameters by least squares applied to subsets of observations; they differ in the number and sizes of the subsets and how those subsets are found. We introduce new versions of standard algorithms with improved combinatorial searches for these subsets. The purposes of our paper are to describe these improvements and to compare publicly available versions of seven algorithms; the comparisons are of computational efficiency and of outlier detection.

Very robust regression was introduced by Rousseeuw (1984) who developed suggestions of Hampel (1975) that led to the Least Median of Squares (LMS) and Least Trimmed Squares (LTS) algorithms. The very robust regression estimators compared here have an asymptotic breakdown point of 50%; they share the property that the parameter estimates are not spoilt by up to half of the data being outliers, provided the two groups are sufficiently well separated. Maronna et al. (2006, Section 3.2) provide a discussion of the breakdown point, with some history in Yohai and Zamar (1988).

[*] Corresponding author.
E-mail addresses: francesca.torti@nemo.unipr.it (F. Torti), domenico.perrotta@ec.europa.eu (D. Perrotta), a.c.atkinson@lse.ac.uk (A.C. Atkinson), mriani@unipr.it (M. Riani).

0167-9473/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2012.02.003

The algorithms for finding these estimates search over many subsets of p observations, p being the number of parameters in the linear model. For these subsets the least squares fit is, of course, an exact fit to the observations. In LTS these subsets are used to construct subsets of predetermined size h, with $[(n+p+1)/2] \leq h \leq n$, where n is the number of observations, to which least squares is applied, so that subsets of two sizes are used. In the reweighted versions of these algorithms described in Section 2.2 all k observations that are identified as outliers are rejected and least squares estimation is used on the remaining $n-k$ observations. These reweighted methods therefore involve fits to either two or three subsets of the data, the size of the last being random.

More recently developed methods fit to subsets of many sizes. In the Forward Search (FS: Atkinson and Riani, 2000 with a recent discussion in Atkinson et al., 2010) subsets of increasing size m are used, starting from $m_0 = p$ and increasing until all observations not in the subset are identified as outliers. Least squares is used for parameter estimation for each subset. In this way the subset size flexibly responds to the properties of the data. The FS for regression is described by Riani and Atkinson (2007).

In order to speed up the LTS algorithm for large data sets, Rousseeuw and Van Driessen (2006) introduced a "concentration" step into their "Fast" LTS algorithm in which the most promising subsets of size $h = (n+p+1)/2$ are used to find a local optimum. In the Appendix we discuss our numerical experience of related algorithms, including our implementation of LTS with concentration steps for all subsets. A crucial component is the speed of sampling subsets, outlined at the beginning of the Appendix. Although we consider both "raw" and reweighted versions of LMS and LTS, we consider only reweighted versions of Fast LTS.

To compare these seven methods of robust regression we look at their ability to detect outliers, considering both the size and the power of the tests. The motivation comes from the study by Riani et al. (2009), who reported excellent size and power for the FS for multivariate data. We likewise find that the FS dominates the other six regression methods.

Our paper is structured as follows. The algorithms for very robust regression are described in Section 2: the six for comparison in Section 2.2 and the FS in Section 2.3. Comparison of the size of the tests is in Section 3, with power comparisons in Section 4. Brief conclusions are in Section 5. Details of the improved combinatorial searches over subsets are in the Appendix. We conclude with indicative comparisons of timings and of the mean squared errors of parameter estimates which highlight the importance of our improved search.

2. Algorithms for very robust regression

We compare and contrast the properties of seven methods for very robust regression. The algorithms that we use are the Forward Search Regression routine (FSR), the LMS and LTS routines and their reweighted versions contained in the MATLAB toolbox called FSDA (Forward Search Data Analysis) at http://www.riani.it/MATLAB, two versions of Fast LTS, the first based on the implementation contained in the LIBRA toolbox at http://wis.kuleuven.be/stat/robust (Verboven and Hubert, 2005, 2010) as originally proposed by Rousseeuw and Van Driessen (2006) and the second based on our implementation in FSDA. Although all our algorithms depend on searching over subsets of observations, other algorithmic approaches to robust regression continue to be explored, including those described by Li (2004), Mastronardia and O'Leary (2007), Flores (2010) and Nunkesser and Morell (2010). García-Escudero et al. (2010) describe methods for the clustering of robustly estimated regression models.

2.1. Least squares for subsets of observations

In the regression model $y = X\beta + \epsilon$, y is the $n \times 1$ vector of responses, X is an $n \times p$ full-rank matrix of known constants, with ith row x_i^T, and β is a vector of p unknown parameters. The normal theory assumptions are that the errors ϵ_i are i.i.d. $N(0, \sigma^2)$.

Let $S(m)$ be a subset of m observations for which the matrix of regressors is $X(m)$. Least squares on this subset of observations yields parameter estimates $\hat{\beta}(m)$ and $s^2(m)$, the mean square estimate of σ^2 on $m-p$ degrees of freedom. Residuals can be calculated for all observations including those not in $S(m)$. The n resulting least squares residuals are

$$e_i(m) = y_i - x_i^T \hat{\beta}(m), \quad i = 1, 2, \ldots, n. \tag{1}$$

The algorithms use different functions of the $e_i(m)$ to declare observations as outlying. The forward search algorithm is designed to have size α of declaring an outlier free sample to contain at least one outlier. Therefore, when evaluating the properties of the other algorithms, we use Bonferroni corrections for simultaneity, with level $\alpha^* = \alpha/n$, so taking the $1-\alpha^*$ cutoff value of the reference distribution. In our calculations $\alpha = 0.01$.

2.2. Some very robust procedures

- *LMS—Least Median of Squares.*

The LMS estimator minimizes the hth ordered squared residual $e^2_{[h]}(\beta)$ with respect to β, where $h = \lfloor (n+p+1)/2 \rfloor$ and $\lfloor . \rfloor$ denotes integer part. Algorithms for LMS find an approximation to the estimator, selecting the best fit, that is the one giving the smallest hth ordered squared residual out of all n residuals, to randomly chosen subsets of p observations (usually called elemental subsets). Since there are p parameters and p observations in the subset, the fit to the observations in the subset is exact. In the PROGRESS algorithm of Rousseeuw and Leroy (1987, Section 4.4) sampling of observations is at random. Samples containing duplicate rows are rejected, although the same subset may be selected more than once.

In our versions of LMS and LTS we instead sample without replacement from the lexicographic list of all possible subsets. If ${}^nC_p < 5 \times 10^7$ we store a list of all subsets in eight bit integer (int8) format. This very large number was chosen as being, at the time of writing, reasonably below the current limit of storage of a typical 32-bit PC. Subsets are then sampled without replacement from this list. For larger values of ${}^nC_p$ we use an algorithm which avoids explicit storage of the subsets. The strategy requires binomial coefficients which are either computed with an algorithm which can be dated back to Lehmer (1964) or retrieved from a look-up table of Pascal's triangle values, previously built using the property that each cell is given by the sum of the number above it and that above to the left. The strategy is built to trade off time and space requirements on the basis of the machine resources available at the time of execution. Details of this procedure can be found in the Appendix. In our comparisons we sampled 10 000 subsets.

Let $\tilde{\beta}_{\text{LMS}}$ be the LMS estimate of β. Rousseeuw (1984) bases the estimate of σ on the value of the median squared residual $e^2_{\text{med}}(\tilde{\beta}_{\text{LMS}})$. As in Rousseeuw and Leroy (1987, p. 202) we define

$$\tilde{\sigma}_{\text{LMS}} = 1.4826\{1 + 5/(n-p)\}\left\{e^2_{\text{med}}(\tilde{\beta}_{\text{LMS}})\right\}^{0.5}. \tag{2}$$

We declare as an outlier any observation i for which the absolute scaled residual

$$|e^{\text{S}}_{\text{LMS},i}| = |e_i(\tilde{\beta}_{\text{LMS}})|/\tilde{\sigma}_{\text{LMS}} > \Phi^{-1}(1-\alpha^*) \quad (i = 1, \ldots, n). \tag{3}$$

- LTS—Least Trimmed Squares.

The convergence rate of $\tilde{\beta}_{\text{LMS}}$ is $n^{-1/3}$. Rousseeuw (1984, p. 876) also suggested Least Trimmed Squares (LTS) which has a convergence rate of $n^{-1/2}$ and so better properties than LMS for large samples. As opposed to minimizing the median squared residual, we now find $\tilde{\beta}_{\text{LTS}}$ to

$$\text{minimize } SS_{\text{T}}\{\hat{\beta}(h)\} = \sum_{i=1}^{h} e_i^2\{\hat{\beta}(h)\}, \tag{4}$$

where, for any subset $\mathcal{H}$ of size h the parameter estimates $\hat{\beta}(h)$ are straightforwardly estimated by least squares.

Let the minimum value of (4) be $SS_{\text{T}}(\tilde{\beta}_{\text{LTS}})$. We base the estimator of σ^2 on this residual sum of squares. However, since the sum of squares contains only the central h observations from a normal sample, the estimate needs scaling. The variance of the truncated normal distribution containing the central h/n portion of the full distribution is

$$\sigma^2_T(h) = 1 - \frac{2n}{h}\Phi^{-1}\left(\frac{n+h}{2n}\right)\phi\left\{\Phi^{-1}\left(\frac{n+h}{2n}\right)\right\},$$

where $\phi(.)$ and $\Phi(.)$ are respectively the standard normal density and c.d.f. (see Croux and Rousseeuw (1992), Eq. (6.5), or the results of Tallis (1963) on elliptical truncation). To estimate σ^2 we accordingly take

$$\tilde{\sigma}^2_{\text{LTS}} = SS_{\text{T}}(\tilde{\beta}_{\text{LTS}})/\{h \times \sigma^2_T(h)\}. \tag{5}$$

Outliers are found as in (3) but with LTS parameter estimates.

- *LTSP—Small-Sample Corrected Least Trimmed Squares.*

The preceding consistency factor is not sufficient to make the LTS estimate of scale unbiased for small samples. Pison et al. (2002) use simulation and curve fitting to obtain correction factors for LTS (and for the MCD estimate in multivariate analysis). We include LTSP only in our comparisons with small sample sizes.

- *LMSR and LTSR—Reweighted Least Median of Squares and Reweighted Least Trimmed Squares.*

To increase efficiency, reweighted versions of the LMS and LTS estimators can be computed. These reweighted estimators are computed by giving weight 0 to observations which condition (3), or its LTS analogue, suggests are outliers. We then obtain a sample of reduced size $n-k$, possibly outlier free, to which OLS is applied. Rousseeuw and Leroy (1987, pp. 44–45), suggest to estimate σ^2 by dividing the residual sum of squares from OLS by $n-k-p$. On the other hand, for comparability with the implementation in the LIBRA library, we divided by $n-k-1$.

Let the parameter estimates be $\tilde{\beta}_{\text{LXSR}}$ and $\tilde{\sigma}_{\text{LXSR}}$, where by LXSR we mean either LMSR or LTSR. The outliers are the k^* observations rejected at this second stage, that is those for which

$$|e_i(\tilde{\beta}_{\text{RLXS}})|/\tilde{\sigma}_{\text{RLXS}} > \Phi^{-1}(1-\alpha^*) \quad (i = 1, \ldots, n). \tag{6}$$

We may find that k^* is greater equal or less than k. Both in (3) and (6) we perform a test of Bonferronised size α^*.

- *LTSRL—"Fast" Least Trimmed Squares.*

The simple LTS algorithm outlined above can be slow as the number of observations increases. Rousseeuw and Van Driessen (2006) introduced an improved algorithm with increased speed and several refinements. The most important of these is the concentration step, which is similar to the step used in the Forward Search in moving from a subset of size m to one of $m+1$.

In this algorithm the random subsets of size p are used to find subsets of size h (say H_1) by selecting the smallest h squared residuals from a fit based on p observations. Given the set of n residuals from the first parameter estimate $\tilde{\beta}(H_1)$ based on the subset of h observations, the absolute values of the residuals are ordered and a new subset H_2 formed as those observations giving the h smallest values of $|e_i\{\tilde{\beta}(H_1)\}|$. This process can be repeated a specified number of times or iterated to convergence. A discussion of the properties of such options in the more general context of S-estimation is in Maronna et al. (2006, Section 5.7).

Rousseeuw and Van Driessen (2006, p. 38) give the pseudo-code for their algorithm. For small values of n, perhaps less than 600, they suggest taking 500 random subsets of size p, leading to least squares estimates based on subsets of size h. The ten subsets so obtained with the smallest value of $SS_T\{\hat{\beta}(H_3)\}$ in (5) then have the concentration steps iterated to convergence, with the minimum value of $SS_T\{\hat{\beta}(H_\infty)\}$ providing the LTS estimate. For larger sample sizes more complicated, but related strategies are suggested. In practice, a divide and conquer strategy based on blocks of 300 observations is used to reduce the number of sorting operations and concentration steps. Finally, there is an adjustment for the intercept to improve precision (Rousseeuw and Van Driessen, 2006).

Once the estimate $\tilde{\beta}_{\text{LTS}}$ has been obtained, the estimation of σ and the identification of outliers are as in the LTS algorithm above.

- *LTSRF—Fully Iterated Reweighted Least Trimmed Squares.*

Our experience is that the LTSRL algorithm was slower, for problems of the size considered here; that is up to $n = 1000$ and $p = 11$, than our implementation of LTS that uses a very efficient random sample generation method and a number of subsets which is approximately four times greater than that of LTSRL (see the Appendix). We think this was, at least in part, also due to the housekeeping (storage and sorting) involved in the concentration step strategy in the algorithm of Rousseeuw and Van Driessen (2006). Therefore, in our implementation of the Fast LTS we made some simplifications in the use of the concentration steps. Instead of our usual 10 000 subsamples, we also drew only 500. However, all of these were subject to concentration steps to convergence. We observed that the number of iterations was essentially independent of the number of variables and was increasing, from about 5 iterations for $n = 50$ to 20 iterations for $n = 1000$, with a rate which decreased as the sample size increased.

2.3. *The Forward Search*

2.3.1. *Background*

The forward search fits subsets of observations of size m to the data, with $m_0 \leq m \leq n$. To start we take $m_0 = p$ and search over subsets of p observations to find the subset that yields one of the very robust estimates of β described in Section 2.2, for example LMS. It is pertinent to remark, however, that our numerical experience, over many examples, is that the final part of the search, which is important for outlier detection, is unaffected by the starting point we use (see for example Atkinson and Riani, 2007). We then order the residuals and take as the subset of size $p+1$ those observations giving rise to the $p+1$ smallest squared residuals. The parameters are re-estimated using least squares on the subset of size $p+1$ and the observations giving rise to the smallest $p+2$ squared residuals form the next subset. In general, let $S^*(m)$ be the subset of size m found by the forward search. Least squares on this subset of observations yields parameter estimates $\hat{\beta}(m^*)$ and $s^2(m^*)$. From (1) the n resulting least squares residuals can be written $e_i(m^*)$. The search moves forward with the augmented subset $S^*(m+1)$ consisting of the observations with the $m+1$ smallest absolute values of $e_i(m^*)$. Although the subset grows in size at each stage, observations can leave the subset as well as entering it.

2.3.2. *Testing for outliers*

In the search we want m to increase until all $n-m$ observations not in $S^*(m)$ are outliers. To test for outliers the deletion residuals are calculated for these $n-m$ observations not in $S^*(m)$. These residuals, which form the maximum likelihood tests for the outlyingness of individual observations, are

$$r_i(m^*) = e_i(m^*)/\sqrt{s^2(m^*)\{1+h_i(m^*)\}}, \tag{7}$$

where the leverage $h_i(.) = x_i^T\{X(.)^TX(.)\}^{-1}x_i$. The observation nearest to those forming $S^*(m)$ is that with the minimum value of $|r_i(m^*)|$, $i \notin S^*(m)$. Call this $i_{\min}$. To test whether observation $i_{\min}$ is an outlier we use the absolute value of the minimum deletion residual $r_{\min}(m^*)$. If the absolute value is too large, the observation $i_{\min}$ is considered to be an outlier, as well as all other observations not in $S^*(m)$. See Riani and Atkinson (2007) for further details.

2.3.3. *The FSR algorithm*

The automatic procedure in the FSR algorithm is based on that of Riani et al. (2009) who used scaled Mahalanobis distances to detect outliers in multivariate normal data. For regression these distances are replaced by the absolute value of the minimum deletion residual. Some further details of the regression algorithm are given by Torti (2011).

Table 1
Convention for lines and labelling used in all plots.

- Traditional LTS and LTSP, the version with a small-sample correction, are represented with a dashed line.
- LTS with the final reweighting step is represented with a dashed and dotted line.
  - LTSR stands for the standard re-weighted LTS.
  - LTSRL stands for the fast re-weighted LTS as in Libra.
  - LTSRF stands for the fast re-weighted LTS as in our FSDA.
- LMS and its re-weighted version are represented with a hatched line
- FS (Forward Search) is represented with a solid line.

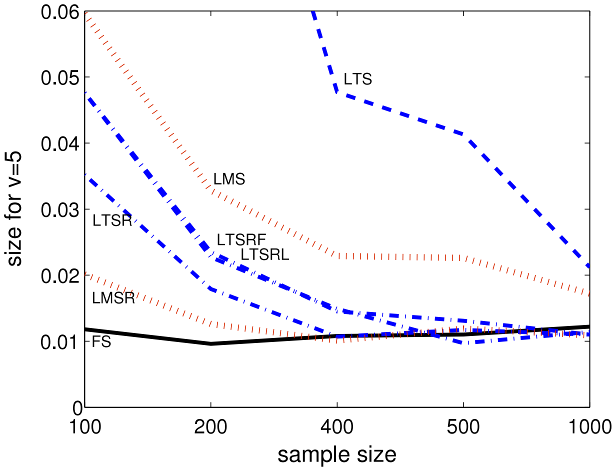

Fig. 1. Size for $v = 5$. The FS is stable around 1% even for smaller sample sizes, where the other methods perform badly. The size of LTS is particularly high. Note the uneven spacing of values of n.

The implementation in FSDA allows appreciable diagnostic and graphical output. Riani et al. (submitted for publication) present an illustrative examples of a FS analysis of the kind of trade data motivating Riani et al. (2008).

3. The size of the tests

In our simulation studies we considered regression models with an intercept and v explanatory variables over a range of values of v from 1 to 10 (so p ranged from 2 to 11). The values of the x_{ij} $(i = 1, \ldots, n; j = 1, \ldots, v)$ were sampled once for each pair of values of v and n from independent standard normal distributions $\mathcal{N}(0, 1)$. Since the tests for outliers are functions of residuals, which do not depend on the values of the parameters β, these values were set to zero. We added standard normal random variables to these null models, estimating the parameters and repeating the process $n_{\text{sim}} = 10,000$ times. We count the proportion of samples declared to contain at least one outlier.

We start with results for $v = 5$. Fig. 1 gives the size of the seven tests for sample sizes n in the range 100–1000. The notational conventions are in Table 1. For the largest sample sizes ($n = 500$ and 1000) all tests, except the traditional LTS and LMS without re-weighting, have sizes near 1%. Unlike the other tests, the size of the FS is stable around 1% even for the smallest sample sizes, whereas the other methods perform comparatively poorly. In particular note that the fast re-weighted LTS (both the LTSRF and the LTSRL versions) has larger sizes than the standard re-weighted LTS (LTSR). We believe this is because the concentration step of the fast LTS is based on a number of subsets, 500, which is not sufficient to visit all the local minima that are explored with the standard LTS estimate using 10 000 subsets. Further simulations with increased numbers of subsets did indeed show that the average squared norm of the parameter estimates decreased as the number of subsets increased. We illustrate this point in the Appendix.

The results for $v = 10$ in Fig. 2 are similar to, but more extreme than, those for $v = 5$. For the larger sample sizes all sizes, except LTS without re-weighting, are between 1% and 2%. For smaller sample sizes all reweighted versions of LTS perform with the two fast versions (LTSRF and LTSR) having larger sizes than LTSR.

As a last exploration of size we look at results for small sample sizes when $v = 1$ in which we also include LTSP, that is LTS with a small sample correction for the estimation of variance (Pison et al., 2002). As Fig. 3 shows, the sizes for LMSR and FS are closest to 1% for all values of n. The size for LMS increases with sample size up to a final 2% at $n = 95$. Standard LTS decreases from around 5% to 4%, whereas LTSP is more nearly constant around 3%. In our opinion this arises because the small sample factors have been calibrated for an individual testing procedure with a nominal individual size in the range 0.05–0.01, as opposed to the simultaneous size considered in this paper. Finally, the three re-weighted LTS methods (LTSR, LTSRF and LTSRL) seem virtually equivalent, the size decreasing with the sample size, from an initial 4% to a final 1.5%.

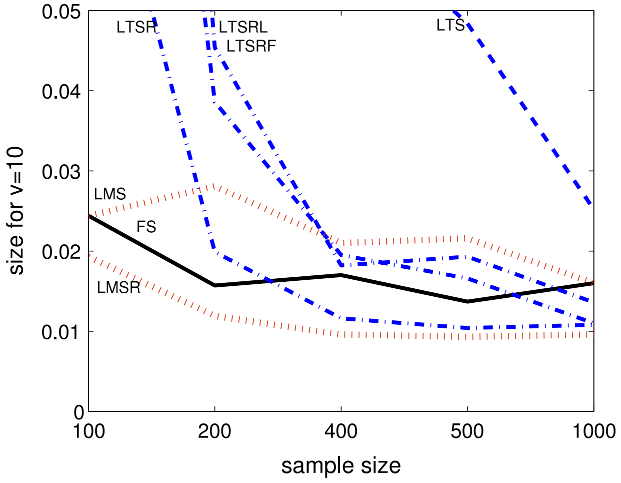

Fig. 2. Size for $v = 10$. For large sample sizes, all the curves except the LTS without re-weighting are between 1% and 2%. For smaller sample sizes LTS performs badly in all cases.

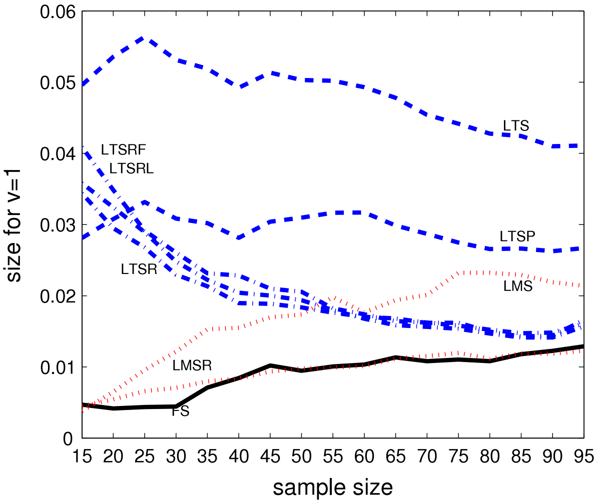

Fig. 3. Smoothed values of size for small n and $v = 1$. The FS and the LMS re-weighted (LMSR) are near 1% for all sample sizes. The standard LMS tends to increase with the sample size up to a final 2%. The standard LTS is constantly between 5% and 4% with LTSP close to 3% throughout. The three LTS re-weighted methods (LTSR, LTSRF and LTSRL) are almost equivalent and their size decreases with the sample size, from an initial 4% to a final 1.5%. In this case the fact of using only 500 subsets to estimate the fast LTS re-weighted (LTSRF and LTSRL) does not produce a big difference from the standard LTS re-weighted (LTSR), which uses 10 000 subsets.

4. The power of the tests

The comparisons of size suggest that, overall, FS and LMSR have the best performance. However, power comparisons reveal that LMSR has the lowest power of the seven procedures.

In our power calculations we shifted the mean of 5% or 30% of the observations by up to 7 units (the errors in our observations were standard normal) and calculated the average power, that is the average proportion of contaminated observations correctly identified. We start in Fig. 4 with 5% contamination when $v = 5$ and $n = 500$. FS clearly has the best power. The other methods are almost equal to each other. For the majority of shift values, the ordering of the rest from best to worst is standard LTS and LMS, the two fast re-weighted LTS (LTSRF and LTSRL), which are indistinguishable, and finally the LTS and LMS with re-weighting (LTSR and LMSR). The labels in the figure are positioned from the top to the bottom to reflect this ranking.

The conclusions from Fig. 4 appear clear. However, in general there are two problems of interpretation for such straightforward plots of power. One is that they are bounded below and above by zero and one; thus the eye focuses on comparisons in the centre of the plot, that is on powers around 50%. The other is that it is impossible to adjust by eye for the size of the different procedures. Accordingly we accompany our plots of power by logit plots.

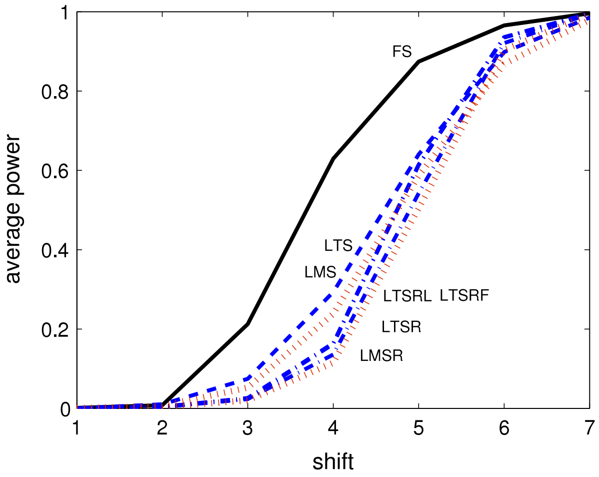

Fig. 4. Average power for $n = 500$, $v = 5$ and 5% contamination. The FS clearly has the best power. The other methods are almost equivalent. For most of the shift values, from the best to the worst we find the standard LTS and LMS, the two fast LTS re-weighted (LTSRF and LTSRL), indistinguishable, and finally the LTS and LMS with re-weighting step (LTSR and LMSR). The labels in the figure are positioned from the top to the bottom to reflect this ranking.

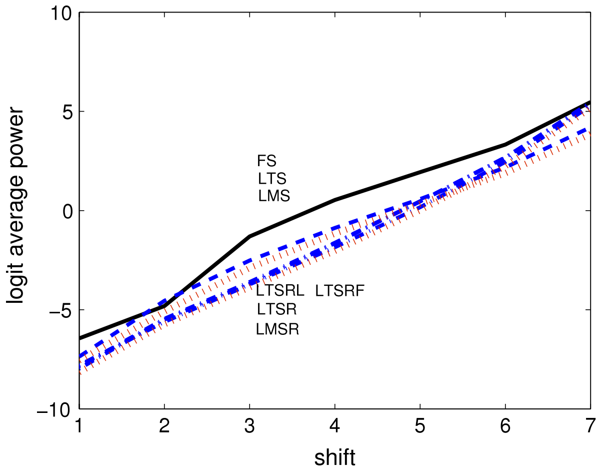

Fig. 5. Logit average power for $n = 500$, $v = 5$ and 5% contamination. For intermediate shift values, the FS has the best performance. The other methods have indistinguishable performance until the shift becomes large.

Fig. 5 repeats Fig. 4 with the power r replaced by logit$\{(r - 3/8)/(n + 1/4)\}$ (Cox and Hinkley, 1974, p. 470). Under this transformation procedures with the same power plot as parallel curves regardless of size. For an example see Atkinson (1985, Fig. 8.12). In our case, Fig. 5 again shows the superior performance of FS. However it is now apparent that the seemingly superior performance of LTS over LMS in Fig. 4 is a reflection of its larger size in Fig. 1. In Fig. 5 the plots for LTS and LMS are parallel. However, we know from the results in Fig. 1 for $n = 500$ that FS has the correct size, that is power when the shift is zero.

The difference between the procedures becomes more marked as the level of contamination increases. In our last two examples we have 30% contamination. In Figs. 6 and 7 we consider small samples with $n = 50$ when $v = 1$. The plots show that the re-weighted methods (LTSR, LTSRF and LTSRL) are indistinguishable over most of the range. LTSP has slightly lower power than LTS, which agrees with the smaller size of LTSP that was evident in Fig. 3. The highest power is for the FS, although its size was the lowest in Fig. 3.

An interesting feature of the logit transformed power in Fig. 7 is apparent around a shift of three where the three reweighted LTS methods are distinguishable. They and LMSR have slightly reduced, rather than increased, power compared to smaller shifts. This phenomenon comes from the masking of outliers which leads to an overestimate of the error variance and a reduction in the number of outliers detected. As the shift increases further the outliers become identifiable. An example of such inflation of variance due to unidentified outliers in multivariate data is in Section 7.3 of Atkinson et al. (2004).

The good performance of the FS becomes even more apparent when v increases from one to five. The results are in Figs. 8 and 9. Both figures show the outstanding performance of the FS. The logit transformation of Fig. 9 not only confirms the performance of the FS, but shows the strong effect of masking on the four reweighted rules.

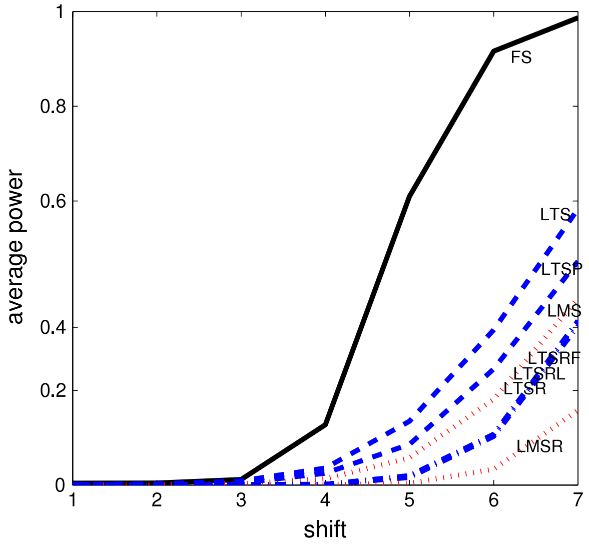

Fig. 6. Average power for $n = 50$, $v = 1$ and 30% contamination. The LTS re-weighted methods (LTSR, LTSRF and LTSRL) are indistinguishable. Power is highest for the FS, although the size was the lowest in Fig. 3. LTSP has lower power than its uncorrected version, LTS.

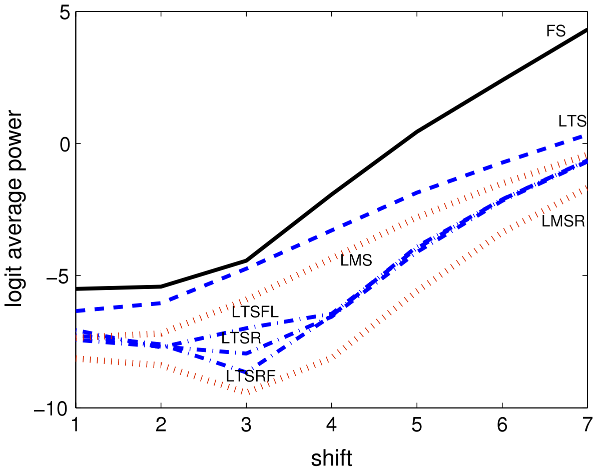

Fig. 7. Logit of average power for $n = 50$, $v = 1$ and 30% contamination. The re-weighted fast LTS (LTSRF and LTSRL) and the standard LTS re-weighted (LTSR) almost coincide for all shift values except three. These three methods suffer some masking for this intermediate contamination level, which disappears for larger contaminations.

5. Conclusions

The idea of reweighing very robust estimates to obtain information from all apparently non-outlying observations is intuitively appealing. However our power comparisons show that all four reweighted rules have a behaviour which is not as good as expected (at least for the sample sizes and the number of variables we considered). The best behaviour for both size and power is provided by the FS. Of the other two rules LMS behaves better for size than LTS. LTSP falls between the two for both size and power. The logit plots show that, once the tests have been adjusted for size, there is little to choose between LTS and LMS for power. This is perhaps surprising in view of the superior asymptotic properties of LTS; the convergence rate of the LMS estimates is $n^{-1/3}$ rather than $n^{-1/2}$ for LTS. However, the simulation results of Rousseeuw (1984, p. 183) fail to reveal any differences in the behaviour of the two estimates for values of n up to 2000.

Of course, the primary interest in fitting regression models in applied statistics is to use the fitted model rather than solely to detect outliers. Our paper concentrates exclusively on a thorough investigation of outlier detection. Riani et al. (2011) evaluate the properties of the parameter estimates, and the relationship with outlier detection, for several rules including S and MM estimates. Their results show a strong relationship between the variance and bias of parameter estimates in very robust regression and the ability of the fitted model to detect outliers. Although the comparisons of size and power for the tests are not as extensive as those given here, the indication is that S and MM estimates have properties close to those of LTS, with larger size, and lower power, than FS.

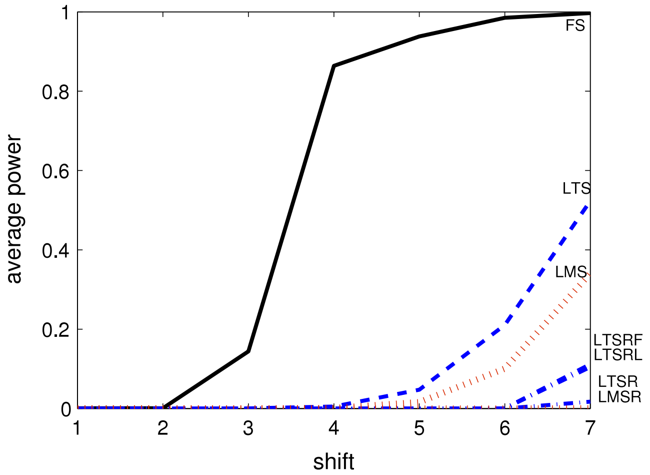

Fig. 8. Average power for $n = 500$, $v = 5$ and 30% contamination. The FS has the best power. The labels are positioned from the top to the bottom to reflect their ranking.

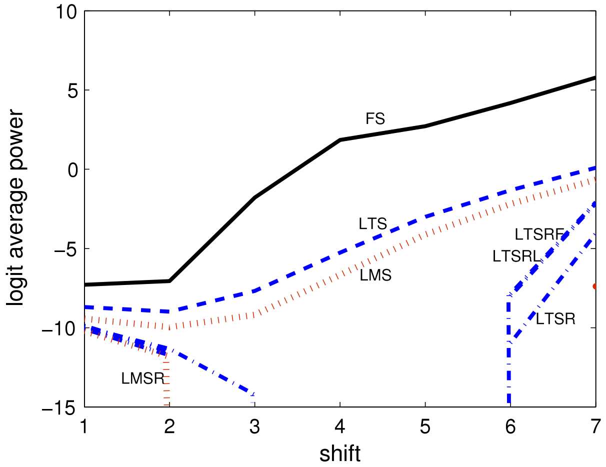

Fig. 9. Logit average power for $n = 500$, $v = 5$ and 30% contamination. For shift values between 2 and 6 the average power for LMS and LTS with reweighting (LMSR, LTSR, LTSRF and LTSRL) is zero and the logit is not defined. These methods suffer from masking for these intermediate contamination levels. This phenomenon was seen less strongly in Fig. 7. FS has by far the highest power.

Acknowledgements

This paper was completed at the Isaac Newton Institute for Mathematical Sciences in Cambridge, England, during the 2011 programme on the Design and Analysis of Experiments. The first author acknowledges support from the University of Milan-Bicocca. We are very grateful to the referees for informative and helpful comments.

Appendix. Efficient random sample generation

LMS and LTS estimation (and, in general, all algorithms of robust statistics) spend a large part of the computational time in sampling subsets of observations and then computing parameter estimates from the subsets. In addition, each new subset has to be checked as to whether it is in general position (that is, it has a positive determinant). For these reasons, when the total number of possible subsets ${}^{n}C_{p}$ is much larger than the number of distinct subsets used for estimation (e.g. $k = 500$ or $k = 10\,000$ as in this paper), we need an efficient method to generate a new random p-element subset without checking explicitly if it contains repeated elements. We also need to ensure that the current subset has not been previously extracted. The lexicographic approach that we present in this Appendix fulfils these requirements. In combinatorial terms, the problem can be reformulated as follows:

> Given a totally ordered set $S = \{1, 2, \ldots, n\}$, generate a set of k different p-combinations $\{s_1, \ldots, s_p\}$ of ordered elements of S.

Following Lehmer (1964), the p-combinations of elements in S can be generated in lexicographic order without repetitions. The lexicographic ordering is a biunivocal correspondence between the p-combinations and the set of integers $\{N \mid 0 \le N <^n C_p\}$. This correspondence, called by Knuth (2005, pp. 5–6) a "combinatorial number system", has an explicit formulation.

In one direction, the generation order N has the following unique p binomial coefficient terms representation (called RANK in the computer science literature)

$$N = \sum_{i=1}^{p} \binom{n - s_i}{p - i + 1}, \tag{A.1}$$

with $0 \le x_p < \cdots < x_2 < x_1 < n$, where $x_i = n - s_i$. For example, when $n = 7$, the generation order of the 3-combination $\{2, 5, 6\}$ is: $N = \binom{5}{3} + \binom{2}{2} + \binom{1}{1} = 12$. In the inverse direction (UNRANK) the function that, given the generation order N, produces the p-combination at position N in the lexicographic ordering is defined by the following greedy algorithm, again due to Lehmer:

$$\begin{aligned}
&x_1 \text{ is the greatest integer such that } \binom{x_1}{p} \le N \\
&x_2 \text{ is the greatest integer such that } \binom{x_2}{p-1} \le N - \binom{x_1}{p} \\
&x_3 \text{ is the greatest integer such that } \binom{x_3}{p-2} \le N - \binom{x_1}{p} - \binom{x_2}{p-1} \\
&\vdots \\
&x_p \text{ is the greatest integer such that } x_p \le N - \sum_{i=1}^{p-1} \binom{x_i}{p-i+1}.
\end{aligned} \tag{A.2}$$

Now, if we want to extract k different subsamples, we simply need to extract k random integers $N_1, \ldots, N_k$ between 0 and ${}^nC_p - 1$ and find the corresponding p-combinations. We employ the following strategy:

A. For small values of ${}^nC_p$, use a look-up table with all p-combinations built beforehand in lexicographic order, and extract rows $N_1, \ldots, N_k$.
B. For small to moderate values of ${}^nC_p$, use UNRANK (A.2) to build the set of k different p-combinations associated with the random integers $N_1, \ldots, N_k$.
C. For big values of ${}^nC_p$, abandon the lexicographic approach and:
   1. If $n \ge 4p$, repeatedly sample from $S = \{1, 2, \ldots, n\}$ until there are p unique values. Repeat this k times.
   2. If $n < 4p$, randomize S (i.e. make a number of switches between two elements chosen at random positions in S) and take the first p as a p-combination. Repeat this k times.

Strategy B (using UNRANK) requires a number of binomial coefficients. Depending on this number and the memory available, we adopt one of these two options:

a. Explicitly compute the binomial coefficients; each requires $\min(p - 1, n - p - 1)$ multiplications.
b. Build beforehand a Pascal triangle for given n and p, which requires $O(np)$ addition operations (each cell is given by the sum of the cell above it and that above to the left). Then, use the Pascal triangle as a look-up table.

If the random number generator is unbiased, as in the case of the Mersenne twister algorithm (Matsumoto and Nishimura, 1998) or the general linear congruential method (Knuth, 1997, pp. 10–26), our strategies ensure that every combination is chosen with equal probability $1/\binom{n}{p}$. To avoid duplicates in the random integers generated by strategies A and B we adopt a very simple and efficient systematic sampling technique, consisting in selecting every $\binom{n}{p}/k$th integer from an ordered list. See Cochran (1977) for systematic sampling.

We have observed that with this strategy the execution time of the traditional LTS and LMS algorithms are drastically reduced and become even better than that of the Fast LTS, at least for the combinations of n and p discussed in the present paper. The timings, of course, cannot be divorced from the quality of the estimates produced. In Fig. 10 we summarize these properties for four scenarios when $v = 10$:

1. LTSRF with 500 subsets, all subsets concentrated;
2. LTSRL with 500 subsets, 10 subsets concentrated;
3. LTSR with 3000 subsets and
4. LMSR with 3000 subsets.

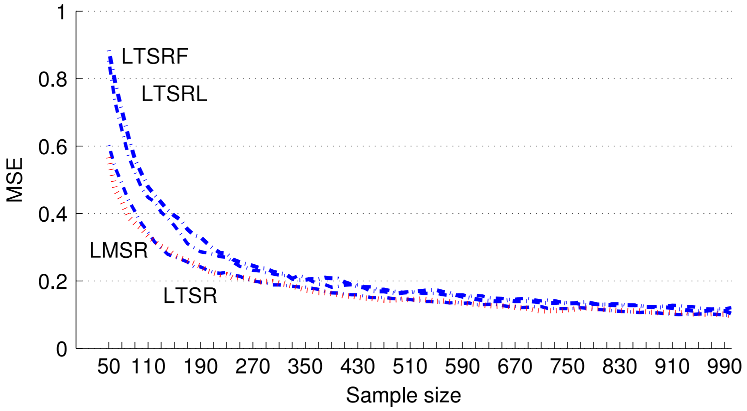

Fig. 10. Mean squared errors (MSEs) of parameter estimates, including intercept, for $v = 10$ and n from 50 to 1000. 500 subsets were taken for LTSRF and LTSRL; 3000 for LMSR and LTSR without concentration steps. For the larger values of n the properties of the estimates are similar.

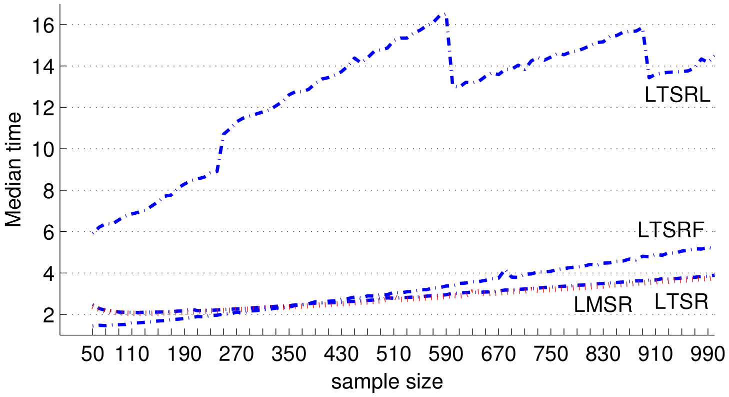

Fig. 11. Timings for the algorithms producing the parameter estimates of Fig. 10. The effect of the "divide and conquer" strategy of LTSRL is evident for $n = 600$ and 900.

The simulations were run in the absence of outliers. The response is the mean squared error (MSE) of the parameter estimates, including intercept. Since the x_i were generated as independent variables and all regression parameters were set to zero, this becomes the median value of $\sum \hat{\beta}_j^2$. The figure shows that for large values of n, perhaps above 500, the estimates have similar properties. However, for smaller n the algorithms with lexicographic sampling produce superior estimates.

We now turn to the timings shown in Fig. 11. Our fully iterated LTSRF in the FSDA toolbox turns out to be at least four times as fast as the Fast LTSRL in LIBRA, which uses just two concentration steps for all combinations of p and $n < 600$ and goes to convergence just for the 10 best subsets. The figure shows that, by comparison with LTSRL, there is little to choose between the timings for LTSRF, LTSR and LMSR, when the latter two take 3000 subsets. The figure also nicely illustrates the effect of the divide and conquer strategy of Rousseeuw and Van Driessen (2006) which is applied to reduce the application of the concentration step, first at $n = 600$ and then at $n = 900$. These times were measured on a computer with a 1.6 GHz Intel T-5450.

References

Atkinson, A.C., 1985. Plots, Transformations, and Regression. Oxford University Press, Oxford.

Atkinson, A.C., Riani, M., 2000. Robust Diagnostic Regression Analysis. Springer-Verlag, New York.

Atkinson, A.C., Riani, M., 2007. Exploratory tools for clustering multivariate data. Computational Statistics and Data Analysis 52, 272–285. doi:10.1016/j.csda.2006.12.034.

Atkinson, A.C., Riani, M., Cerioli, A., 2004. Exploring Multivariate Data with the Forward Search. Springer-Verlag, New York.

Atkinson, A. C., Riani, M., Cerioli, A., 2010. The forward search: theory and data analysis (with discussion). Journal of the Korean Statistical Society 39, 117–134. doi:10.1016/j.jkss.2010.02.007.

Cochran, W. G., 1977. Sampling Techniques, third ed.. Wiley, New York.

Cox, D.R., Hinkley, D.V., 1974. Theoretical Statistics. Chapman and Hall, London.

Croux, C., Rousseeuw, P.J., 1992. A class of high-breakdown scale estimators based on subranges. Communications in Statistics Theory and Methods 21, 1935–1951.

Flores, S., 2010. On the efficient computation of robust regression estimators. Computational Statistics and Data Analysis 54, 3044–3056. doi:10.1016/j.csda.2010.03.020.

García-Escudero, L.A., Gordaliza, A., Mayo-Iscar, A., San Martin, R., 2010. Robust clusterwise linear regression through trimming. Computational Statistics and Data Analysis 54, 3057–3069. doi:10.1016/j.csda.2009.07.002.

Hampel, F. R., 1975. Beyond location parameters: robust concepts and methods. Bulletin of the International Statistical Institute 46, 375–382.

Knuth, D. E., 1997. The Art of Computer Programmig, third ed.. In: Seminumerical Algorithms, vol. 2. Addison-Wesley, Reading, Mass.

Knuth, D., 2005. Generating all combinations and partitions. In: The Art of Computer Programming. In: Fascicle 3, vol. 4. Addison-Wesley, Reading, Mass.

Lehmer, D.H., 1964. The machine tools of combinatorics. In: Beckenbach, E.F. (Ed.), Applied Combinatorial Mathematics. Wiley, New York, pp. 5–31.

Li, L.M., 2004. An algorithm for computing exact least-trimmed squares estimate of simple linear regression with constraints. Computational Statistics and Data Analysis 48, 717–734. doi:10.1016/j.csda.2004.04.003.

Maronna, R.A., Martin, D.R., Yohai, V.J., 2006. Robust Statistics: Theory and Methods. Wiley, New York.

Mastronardia, N., O'Leary, D.P., 2007. Fast robust regression algorithms for problems with Toeplitz structure. Computational Statistics and Data Analysis 52, 1119–1131. doi:10.1016/j.csda.2007.05.008.

Matsumoto, M., Nishimura, T., 1998. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. ACM Transactions on Modeling and Computer Simulation 8, 3–30.

Nunkesser, R., Morell, O., 2010. An evolutionary algorithm for robust regression. Computational Statistics and Data Analysis 54, 3242–3248. doi:10.1016/j.csda.2010.04.017.

Pison, G., Van Aelst, S., Willems, G., 2002. Small sample corrections for LTS and MCD. Metrika 55, 111–123. doi:10.1007/s001840200191.

Riani, M., Atkinson, A.C., 2007. Fast calibrations of the forward search for testing multiple outliers in regression. Advances in Data Analysis and Classification 1, 123–141. doi:10.1007/s11634-007-0007-y.

Riani, M., Atkinson, A.C., Cerioli, A., 2009. Finding an unknown number of multivariate outliers. Journal of the Royal Statistical Society Series: B 71, 447–466.

Riani, M., Atkinson, A. C., Perrotta, D., 2011. Calibrated very robust regression. Technical Report NI11033-DAE, Isaac Newton Institute, Cambridge, UK.

Riani, M., Atkinson, A. C., Perrotta, D., 2012. Calibrated very robust regression for mixtures of regression models (submitted for publication).

Riani, M., Cerioli, A., Atkinson, A., Perrotta, D., Torti, F., 2008. Fitting mixtures of regression lines with the forward search. In: Fogelman-Soulié, F., Perrotta, D., Piskorski, J., Steinberger, R. (Eds.), Mining Massive Data Sets for Security. IOS Press, Amsterdam, pp. 271–286.

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

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

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

Tallis, G.M., 1963. Elliptical and radial truncation in normal samples. Annals of Mathematical Statistics 34, 940–944.

Torti, F., 2011. Advances in the Forward Search: Methodological and Applied Contributions. Italian Statistical Society, Padova.

Verboven, S., Hubert, M., 2005. LIBRA: a MATLAB library for robust analysis. Chemometrics and Intelligent Laboratory Systems 75, 127–136. doi:10.1016/j.chemolab.2004.06.003.

Verboven, S., Hubert, M., 2010. Matlab library LIBRA. WIREs Computational Statistics 2, 509–515.

Yohai, V.J., Zamar, R.H., 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.